

29 PAGES OF EXPERT TUTORIALS

FREE OVER 5 HOURS OF JS VIDEOS

web

# designer™

HTML5 CSS3 jQuery JS JavaScript

@WebDesignerMag /WebDesignerUK

DESIGN WITH SKETCH 21 killer tips

## The NEW KINGS of JAVASCRIPT

UNLOCK LAYOUTS WITH CSS GRID Code flexible magazine designs

13 MUST-KNOW TECHNIQUES FOR THE MODERN DAY DEVELOPER



CREATE A CAROUSEL

ANIMATE SVG

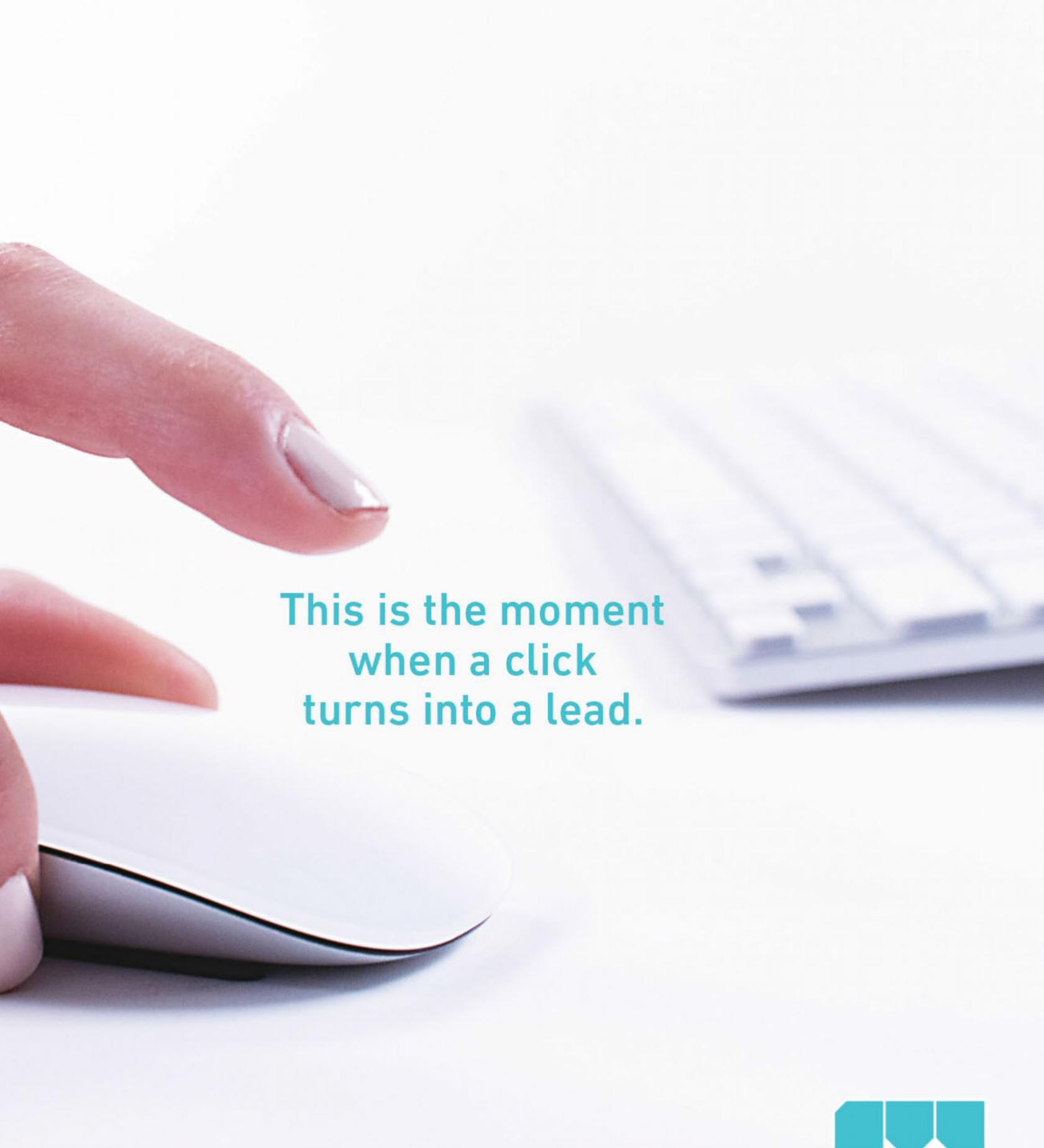
BUILD BROWSER EXTENSIONS

INTERACTIVE BACKGROUNDS

ALSO INSIDE THE LATEST UPDATES IN ANGULAR 5

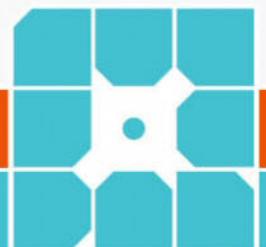


OPTIMISE SEO How to make your content count



This is the moment  
when a click  
turns into a lead.

**PRESS AHEAD**



WP Engine's digital experience platform drives your business forward faster. [wpengine.co.uk](https://www.wpengine.co.uk)

**WP**engine®

# Welcome to the issue



Steven Jenkins  
Editor

## THE WEB DESIGNER MISSION

To be the most accessible and inspiring voice for the industry, offering cutting-edge features and techniques vital to building future-proof online content

## Get the JavaScript that suits you



JavaScript has more frameworks than ever demanding a developer's attention. But which ones should you be using? This issue we take a closer look at the five biggest libraries and frameworks out there and give you the lowdown on why you should be using them, what to use them for and the latest techniques.

So, who makes the 'big five'? For our latest lead feature (page 42) it's the established giants – React and Angular – alongside new kid on the block Vue and emerging talents Polymer and Aurelia. All are making a big impression, and you can find a collection of must-know techniques that will help improve your projects and your workflow.

Previously it (React) had to be either a single component or 'null', the latest version allows strings, numbers, arrays and a new concept called 'portals'

Veering towards the other side of the web design spectrum we delve into the world of Sketch. The popular app is/has been adopted by the many and with good reasons. Designer and Sketch power user Daniel Schwartz unveils 21 of those reasons with his favourite collection of tips and shortcuts to take your workflow and designs to the next level.

Elsewhere we take a closer look at browser plugins and how they can now be built using classic web technologies. Finally, there is the option to create one option that works for all. About time we say.

Want to know what's new in Angular 5, how to optimise content for a better ranking and use CSS Grid for magazine-style layouts. It's all here in the latest issue. Enjoy.

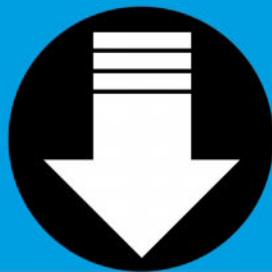
## Highlight



In our early days, when everyone was focused on Flash, we doubled down on HTML/CSS/JS.

Design is woven into the DNA of talented digital pioneers AREA 17. Web Designer decided to find out more. Page 34

Follow us on Twitter for all the news & conversation @WebDesignerMag  
Visit our blog for opinion, freebies & more [www.creativebloq.com](http://www.creativebloq.com)



## FREE - exclusive with this issue

# 94 Designer resources

- Video Tuition** - Over five hours of expert JavaScript video guides from Pluralsight ([pluralsight.com](http://pluralsight.com))
- Assets** - 90 Gradient triangulate backgrounds, and Real photo product box mockup actions from Sparklestock ([sparklestock.com](http://sparklestock.com))
- Tutorial files and assets



[www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

# This issue's panel of experts

Welcome to that bit of the mag where we learn more about the featured writers and contributors...



**Matt Crouch**

Matt is an experienced front-end developer who works in London. In this issue he takes a look at the biggest and most popular JavaScript frameworks and evaluates their strengths and values. Plus, he reveals a selection of the latest must-know techniques that the modern day developer needs to know today. **Page 42**

The state is a single object that holds all the necessary data for the entire application. The way this object gets structured depends on the project

## Daniel Schwarz



Daniel is a designer, creator writer and blogger. He runs his own studio called Airwalk Studios. He has also written a book called Jump Start Sketch. This issue he reveals 21 killer Sketch tips to get the most from the app. **Page 66**

## Mark Shufflebottom



This issue Mark shows how to make use of CSS Grid Layouts with three easy examples, before diving into a more complex responsive layout. When not creating web content, he can be found teaching at Sheridan College. **Page 52**

## Tam Hanna



Tam has experience in the Mozilla arena and this makes him a prime authority at pointing out the value of Mozilla-born technologies. This issue he explores building browser plugins with web technologies. **Page 74**

## Paul Betteridge



Paul has 15 years' experience, leading digital organisations and pioneering digital marketing across a plethora of competitive markets. This issue he tells you how to optimise your content for SEO success. **Page 56**

## Steven Roberts



Steven is a Frontend designer and developer from Middlesbrough. His recent passion has been SVG, specifically for animation. This issue he'll be teaching you how to use SVG and the GSAP animation library. **Page 60**

## Luke Harrison



Luke is a web developer from Sheffield, UK who is all about scalable and efficient front-end architecture. In this issue, he explores how to create a simple, flexible carousel component in Vue.js, the JavaScript framework. **Page 80**

## David Howell



David is a journalist with over 20 years' experience in publishing and runs his own business, Nexus Publishing. This issue he takes time out to talk to the talented crew at Area 17 and find out what makes them tick. **Page 34**

## Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he reveals a host of techniques, as seen on the top-class websites featured in our Lightbox. **Page 14**

## Follow us!

**Facebook:** [www.facebook.com/WebDesignerUK](http://www.facebook.com/WebDesignerUK)  
**Twitter:** <https://twitter.com/webdesignermag>

Future Publishing Limited  
Richmond House, 33 Richmond Hill  
Bournemouth, Dorset, BH2 6EZ

## Editorial

Editor **Steven Jenkins**  
[steve.jenkins@futurenet.com](mailto:steve.jenkins@futurenet.com)  
01202 586233

Designer **Harriet Knight**  
Senior Art Editor **Will Shum**  
Editor in Chief **Amy Hennessey**

## Contributors

Leonard Souza, Matt Crouch, Mark Shufflebottom, Paul Betteridge, Steven Roberts, Leon Brown, Daniel Schwarz, Luke Harrison, Tam Hanna, Mark Billen, David Howell, Rob Mead-Green

## Photography

**James Sheppard**

All copyrights and trademarks are recognised and respected

## Advertising

Media packs are available on request  
Commercial Director **Clare Dove**  
[clare.dove@futurenet.com](mailto:clare.dove@futurenet.com)  
Senior Advertising Manager **Mike Pyatt**  
[michael.pyatt@futurenet.com](mailto:michael.pyatt@futurenet.com)  
01225 687538

Account Director **George Lucas**  
[george.lucas@futurenet.com](mailto:george.lucas@futurenet.com)  
Account Manager **Chris Mitchell**  
[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

## International

Web Designer is available for licensing. Contact the International department to discuss partnership opportunities  
International Licensing Director **Matt Ellis**  
[matt.ellis@futurenet.com](mailto:matt.ellis@futurenet.com)  
+44 (0) 1225 442244

## Print subscriptions & back issues

Web [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
Email [contact@myfavouritemagazines.co.uk](mailto:contact@myfavouritemagazines.co.uk)  
Tel **0344 848 2852**  
International **+44 (0) 344 848 2852**

## Circulation

Head of Newstrade **Tim Mathers**  
01202 586200

## Production

Head of Production US & UK **Mark Constance**  
Production Project Manager **Clare Scott**  
Advertising Production Manager **Joanne Crosby**  
Digital Editions Controller **Jason Hudson**  
Production Manager **Nola Cokely**

## Management

Managing Director **Aaron Asadi**  
Editorial Director **Paul Newman**  
Art & Design Director **Ross Andrews**  
Head of Art & Design **Greg Whittaker**  
Commercial Finance Director **Dan Jotcham**

## Printed by

William Gibbons, 28 Planetary Road,  
Willenhall, WV13, 3XT

## Distributed by

Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU  
[www.marketforce.co.uk](http://www.marketforce.co.uk) Tel: 0203 787 9001

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing paper mill holds full FSC (Forest Stewardship Council) certification and accreditation

## Disclaimer

All contents © 2017 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.

ISSN 1745-3534

Future

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR).  
[www.futureplc.com](http://www.futureplc.com)

**Chief executive** Zillah Byng-Thorne  
**Non-executive chairman** Peter Allen  
**Chief financial officer** Penny Ladkin-Brand

Tel +44 (0)1225 442 244

LOSE YOURSELF IN A WORLD OF

# Vinyl

FIND YOURSELF IN  
OXFAM'S ONLINE SHOP

[oxfam.org.uk/shop](http://oxfam.org.uk/shop)



OXFAM

# Contents

Cutting-edge features, techniques and inspiration for web creatives

Chat with the team and other readers and discuss the latest tech, trends and techniques. Here's how to stay in touch...  
steve.jenkins@futurenet.com • @WebDesignerMag • www.creativebloq.com

**08 Welcome to Firefox 57**

The latest version of the browser has been virtually rebuilt. We take a closer look

**10 WebKit: The best must-try resources out there**

Discover the libraries and frameworks that will make your site a better place to visit

**11 Designers shouldn't code**

The future of web development is in designers' hands according to Webflow's Leonard Souza

**14 Lightbox**

A showcase of inspirational sites and the techniques used to create them

**26 Directing traffic**

Montreal studio FFunction reveals how it transformed poignant data into an engaging online story

**34 Engineered design**

Design is woven into the DNA of AREA 17. **Web Designer** finds out more

**42 The new kings of JavaScript**

We take a look at the five biggest frameworks and show you how to get the best out of them for your next project

**66 Sketch: 21 killer tips**

Discover an essential selection of techniques, tips, shortcuts and must-try plugins

**74 Browser plugins for the web**

The arrival of Firefox 57 brings with it the power to build plugins with web technologies

**92 Hosting listings**

An extensive list of web hosting companies. Pick the perfect host for your needs

**94 Course listings**

Want to start learning online? Check out what courses are out there with this list

**98 Next month**

What's in the next issue of **Web Designer**?

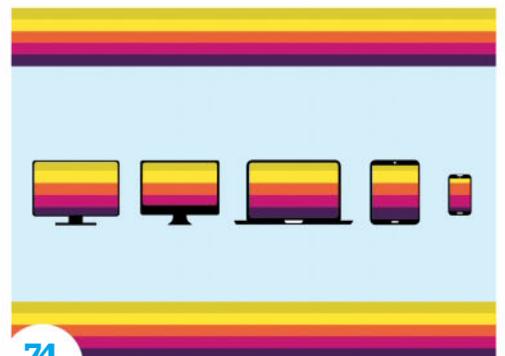
**Cover focus**



42



**66**  
**21 killer tips for Sketch**  
A must-know collection of tips and shortcuts



**74**  
**Build browser plugins**  
Create plugins with web technologies

# FileSilo

96 Get the latest must-have resources and videos

- Over five hours of expert JavaScript video guides from Pluralsight
- Real photo product box mockups
- 90 Gradient triangulate backgrounds



Never miss an issue  
**Subscribe**  
Turn to page 32 now

Turn to page 85 for an unbeatable Christmas offer



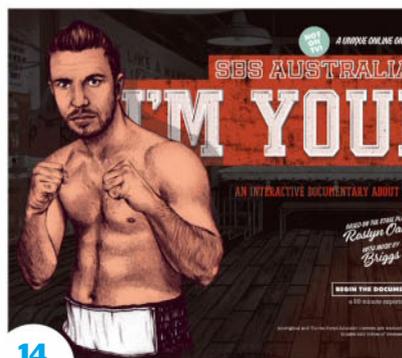
What if we could 'code' without coding at all? What if that 'code' could display instant visual feedback?



COMMENT – Leonard Souza – p11



34 **ProFile: AREA 17**  
Bringing understanding to digital



14 **Lightbox: I'm your man**  
Take on a virtual opponent

Visit the **WEB DESIGNER** online shop at

**Future** myfavouritemagazines  
myfavouritemagazines.co.uk

for the latest issue, back issues and specials

## Tutorials

Web gurus take you step-by-step through professional techniques



### 52 **Complex layouts with CSS Grid**

Learn how to create eye-catching magazine-style layouts with the help of CSS Grid

### 56 **Optimise content for better SEO**

Discover the ten steps that will help you understand what's required to influence Google rankings

### 60 **Animate SVG with JavaScript**

Create crisp, clean, animated graphics using the powerful GSAP library from GreenSock

## Web Workshop

### 50 **Multiple interactive backgrounds**

[serpentiform.bulgari.com/en/home](http://serpentiform.bulgari.com/en/home)

Add interest to home page design with parallax

### 64 **Splash image with rolling content**

[pomwonderful.com](http://pomwonderful.com)

Keep the focus on the product

## Web Developer

### 74 **Browser plugins for the web**

Discover the art of creating browser plugins using popular web technologies

### 80 **Create a carousel component**

Learn how to create a flexible carousel component using the Vue JavaScript framework

### 86 **Angular 5: What's new?**

The latest version has a host of new features. Get a closer look at what's on offer

# Header

The tools, trends and news to inspire your web projects

# Welcome to the new Firefox browser

The latest version of Firefox is faster, friendlier and better-looking than ever before. Have you tried it out yet?

Nowadays the arrival of a new web browser isn't typically the grand occasion it once used to be when they were only being updated about once a year. Regular incremental updates are now the way forward for the most popular browsers, and this means we no longer have to wait so long before getting our hands on a new feature or better support.

But the latest version of Firefox is more, much more than just another number added to the latest version. Firefox 57 by name may not mean much – in fact it sounds very much like every other version of the browser – but it really does offer something new. Firefox 57, also known as Quantum, is twice as fast as it used to be. Are you using it? If not, then why not download it from [mozilla.org/en-US/firefox/new](https://mozilla.org/en-US/firefox/new) and try it out for yourself.

We at Web Designer have been putting Quantum through its paces and have to say we like what we see. The design and

the speed increase are the two obvious bonus points. It took a few minutes to get to grips with placement of old features, but once we had overcome this small bump in the road we sat back and enjoyed the experience. So, what else is on offer?

It has a new CSS engine called Quantum CSS. Turn it on by going to `about:config` and setting `layout.css.servo.enabled` to true. Find out more at [mzl.la/2wkuFQP](https://mzl.la/2wkuFQP)

What don't we like? Very little, but we are a fussy lot and one small point was trying to move the browser around. We kept grabbing tabs by mistake, but we've got

used to it now. Others have mentioned legacy add-ons not working. We simply decided it was time to refresh our extensions/add-ons line-up and choose new ones where needed. And we are happy with everything chosen so far.

So, we know we like it, but we didn't know what the great web design public thought, so we asked. 36 per cent liked it, 33 per cent thought it was better than Chrome, while 31 per cent thought it could be better. A pretty even split. We like it, so go get it and let us know what you think [twitter.com/WebDesignerMag](https://twitter.com/WebDesignerMag)

## FIREFOX IN NUMBERS

**2**  
times faster  
than before

**700**  
the number of  
code contributors

**6,000**  
extensions and  
add-ons available

**1 YEAR**  
How long it  
took to make

**4,888,199**  
lines of code added

## CREATIVE BLOQ

[www.creativebloq.com](http://www.creativebloq.com)

In-depth tutorials, expert tips, cutting-edge features, industry interviews, inspiration and opinion. Make sure to get your daily dose of creativity, design and development.

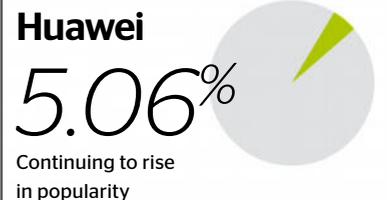
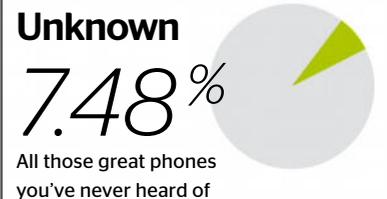
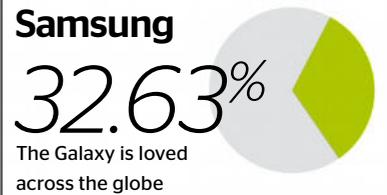
### WEB DESIGNER DIGITAL EDITION

Do you want to get your hands on a digital edition of your favourite web design magazine? Head to your preferred app store – Google Play ([bit.ly/2wetvGp](https://bit.ly/2wetvGp)) or iTunes ([apple.co/2igtBYq](https://apple.co/2igtBYq)) – then download, install and purchase the issue of choice from within the app.

# STAT ATTACK

## MOBILE VENDORS

Who is the most popular mobile phone maker: Apple, Samsung, or Xiaomi?



Source: [gs.statcounter.com](https://gs.statcounter.com)

# Sites of the month

01.

Menu NIHYAKU-ICHI® IIOI°

**Widen Your Perspective  
To Unlock Your  
Creativity**

02.

BOUTIQUE WEDDINGS

HOME ABOUT US SERVICES VENUES MEMORIES CONTACT

WEDDING PLANNERS

03.

Prored Partners

WHO WE ARE BUSINESS WORKS CLIENT COMPANY NEWS CAREER

thinkOut

最新にこだわらるから、考え出す

CONCEPT MOVIE

04. Duroc

ABOUT PRODUCTS VALUES

- 01. 201**  
201d.jp/en/  
Masking effects that add instant interest and engagement.
- 02. Boutique Weddings**  
boutique-weddings.com  
Gorgeous colour schemes and simple transitions bring the site to life.
- 03. Prored Partners**  
prored-p.com  
Look at this site just for the transitions alone. And then admire the rest.
- 04. Duroc**  
duroc.ma/en/  
Quirky and engaging video snippets, which keep site visitors happy.

**Graphics**  
Oscar Llorens  
behance.net/ollorens

A talented illustrator, Oscar Llorens has produced gorgeous work for a whole host of well-known names.

**Colour picker**  
Mint  
bit.ly/2zsnbcQ

#2B0F10  
#3B6B56  
#00987E  
#3FBEB1  
#BA874F

**Typesetter**  
Vodka  
bit.ly/2k10k5o

Vodka is a display combo mixing styles and fonts. Its core is two fonts, one chunky sans serif, one script, ideal for creating together.

*ABCabc*  
**ABCABC**

**WordPress**  
Trackstore  
trackstore.elated-themes.com

An eCommerce theme with sports as its core subject. A host of visually-led designs ensure the product is seen.

# webkit

Discover the must-try resources that will make your site a better place

## Reframe.js

Reframe.js is a javascript plugin that makes unresponsive elements responsive.

[Tweet](#) [Star](#)



## Reframe.js

[bit.ly/2cTwQ45](http://bit.ly/2cTwQ45)

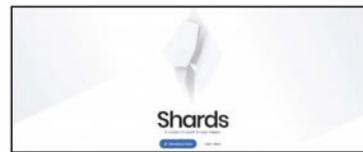
Got an unresponsive element and want to make it responsive quickly and easily? Then try out the Reframe.js plugin. It removes a specified element's height and width attributes and then wraps that element in a responsive div that is an intrinsic ratio of the original element. The plugin is ideal for embedded content such as video.



### Animated Flexbox Playground

[codepen.io/osublake/full/dMLQJr](http://codepen.io/osublake/full/dMLQJr)

A quick and visual introduction to flex components and how they affect layouts. Great as a reference.



### Shards

[designrevision.com/demo/shards](http://designrevision.com/demo/shards)

Shards is a modern design system based on Bootstrap 4 that has a host of custom components. It provides layouts, buttons, icons and much more.



### Loomio

[loomio.org](http://loomio.org)

Stop the emails, stop the meetings and start being more productive. Loomio looks to keep everything in one place and help groups make clear decisions.

# TOP 5 Web conferences – Jan 2018

Get yourself a seat at the biggest and best conferences coming your way soon



## FrontEndNorth

[frontendnorth.com](http://frontendnorth.com)

A Make Do event with a focus on front-end development. A host of speakers talk browsers, React, JavaScript and more.



## Script18

[scriptconf.org](http://scriptconf.org)

Fancy a trip to Linz in Austria, and learn about JavaScript at the same time? Then this is the conference for you.



## Next Money

[ff18.nextmoney.org](http://ff18.nextmoney.org)

The largest fintech conference in Hong Kong covers design, innovation, entrepreneurship, and a host of startups.



## Ecommerce Expo

[bit.ly/2hVHqZc](http://bit.ly/2hVHqZc)

eCommerce is huge and this eCommerce expo has over 100 exhibitors to talk to.



## Vibrant

[vibrantdigitalfuture.uk](http://vibrantdigitalfuture.uk)

One day and 30 speakers talking business strategy, government, governance, and cybersecurity. Listen and learn.

# Designers shouldn't (have to) code

The future of web development will be in designers' hands.



**Leonard Souza**

Engineering Manager at Webflow  
[webflow.com](http://webflow.com)

“What if we could ‘code’ without coding at all? What if that ‘code’ could display instant hammer-to-nail visual feedback?”

If you'd have asked me ten years ago, "Hey, is there an app that will let me code websites visually?", I'd have given you that awful stare that only an annoyed web developer can give. "No," I would have said. Then I might have shrugged my shoulders, pursed my lips, and pointed to a nearby bookshelf with a five-inch thick tome dedicated to JavaScript, HTML and CSS. "Good luck," and I'd have patted you on the back and been on my way.

Two things glare at me as I recount the above: 1) Wow, I hope my people skills have improved, and 2) My attitude toward visual coding has about-faced, and I couldn't be more excited. We've seen smart industries choose to empower creatives rather than belittle or neglect their unique and potentially cataclysmic (for good!) talents. The film and gaming industries have adopted tools that leverage a visual thinker's mental model. I mean, have you seen the latest 3D software packages or game engines? They are mind-bogglingly good. They give creatives a visual means to solve complex problems without employing a single line of maths. This is a staggering engineering feat. Just peruse Dribbble or Behance for the stunning effects technical ambition has wrought.

The web development industry has yet to enjoy such an epoch. For an industry delicately predicated on the co-operation of design and technical skills, it's a wonder we have not seen tools bring those disciplines closer. Until now. If you listen closely, the rattling breath of these medieval times is giving way to the sharp inhale of a creative renaissance. The uprising of user experience practitioners, who focus on empathy over analytics, have forced us to contend, design for, and fix problems wholly unrelated to our own. These stewards of empathy have lighted the path to empower a legion of new doers. Bret Victor ([worrydream.com](http://worrydream.com)) is one such steward. His talk, *Inventing On Principle*, and essay, *A Brief Rant On The Future Of Interaction Design*, delve into why technology must solve problems for human needs foremost by providing creators an immediate connection to their work.

Coding does not solicit immediate feedback. It can take seconds, or much more, to verify the result of a single character change. This latency cripples the creative process. Imagine daubing a canvas with a loaded paintbrush... and seconds later the colour appears. How infuriating would that be? Inflicting on humans obvious violations of the physical world is a recurring theme in technology, and I hope to see a big, black curtain fall on it in the near future. Bret gives us simple language through which to address and repair this persistent theme. We have "problems," such as pushing a sliver of metal into wood, and a need for "tools" that amplify human capacity, which in this case the hammer is an admirable response. The hammer fits a person's hand, snugly, naturally, as if it were designed for a human. Its feedback is instant: strike, and see an embedded nail. It's a human-first solution, and it keeps the human's innate capabilities top-of-mind.

Interaction design can learn a lot from a hammer. What if we could "code" without coding at all? What if that "code" could display instant hammer-to-nail visual feedback? What if we could solve many of the problems in web design through a user interface mapped to a creative's mental model? How many more people could join the fray and steer us toward a more beautiful and usable web?

As Bret Victor has proposed, and the game and film industries have proven: this reality of visual problem solving is all too possible. As an engineering manager at a company whose sole impetus is to empower designers, I haven't been this excited about the challenges that lay ahead, and I haven't had this much pride for the successes my team has already earned. But I'm most excited for the empowered and for all their patrons. Let the renaissance begin.

# webkit

Discover the must-try resources that will make your site a better place



Check JavaScript files ES version against a specified ES version 🏆



## ES Check ✓

**ES Check** checks JavaScript files against a specified version of ECMAScript (ES) with a shell command. If a specified file's ES version doesn't match the ES version argument passed in the ES Check command, ES Check will throw an error and log the files that didn't match the check.

## ES Check

[bit.ly/2BgA3Do](https://bit.ly/2BgA3Do)

In a nutshell, ES Check “Checks the version of ES in JavaScript files with simple shell commands”. In other words, it checks the chosen JavaScript files against a specified version of ECMAScript. If the selected files' ES version doesn't match the ES version argument passed through, ES Check then will display an error.



### Tailwind CSS

[tailwindcss.com](https://tailwindcss.com)

So what is Tailwind? It's a “utility-first CSS framework for rapidly building custom user interfaces”. It is not a UI kit, it provides components, not the design.



### Mockoon

[mockoon.com](https://mockoon.com)

Need a mock server? Then Mockoon is a quick and easy way to run mock APIs locally. It offers unlimited environments and customisable paths.



### SiteStacks

[sitestacks.com](https://sitestacks.com)

Want to discover the tech stack for a site you like the look of? Then this is exactly what SiteStacks does. Add a URL and wait for a list to appear with details.

## TOP 5 Codepens

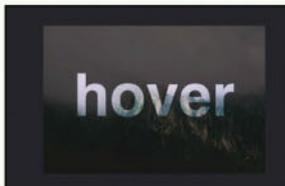
Admire a selection of the latest additions and dig into the code



### Breathing Halftone

[bit.ly/2k7zloU](https://bit.ly/2k7zloU)

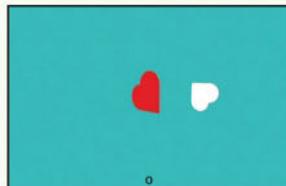
Take an image, transform to halftone, add some animation to get a pulsating head.



### z-index

[bit.ly/2ACygfK](https://bit.ly/2ACygfK)

A hover effect that sees the transparent background break up and disappear in the style of pixels on a TV screen.



### Codevember #24

[bit.ly/2Ac35qJ](https://bit.ly/2Ac35qJ)

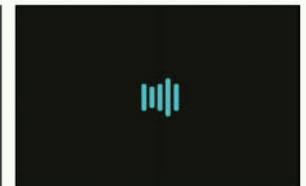
A simple game where you need to match the direction of the heart. Low on HTML, high on CSS and Javascript.



### Word animation

[bit.ly/2iWR1Qp](https://bit.ly/2iWR1Qp)

A blurry blob slowly stretches and animates into a selection of words. All done with a tiny bit of HTML and very little CSS.



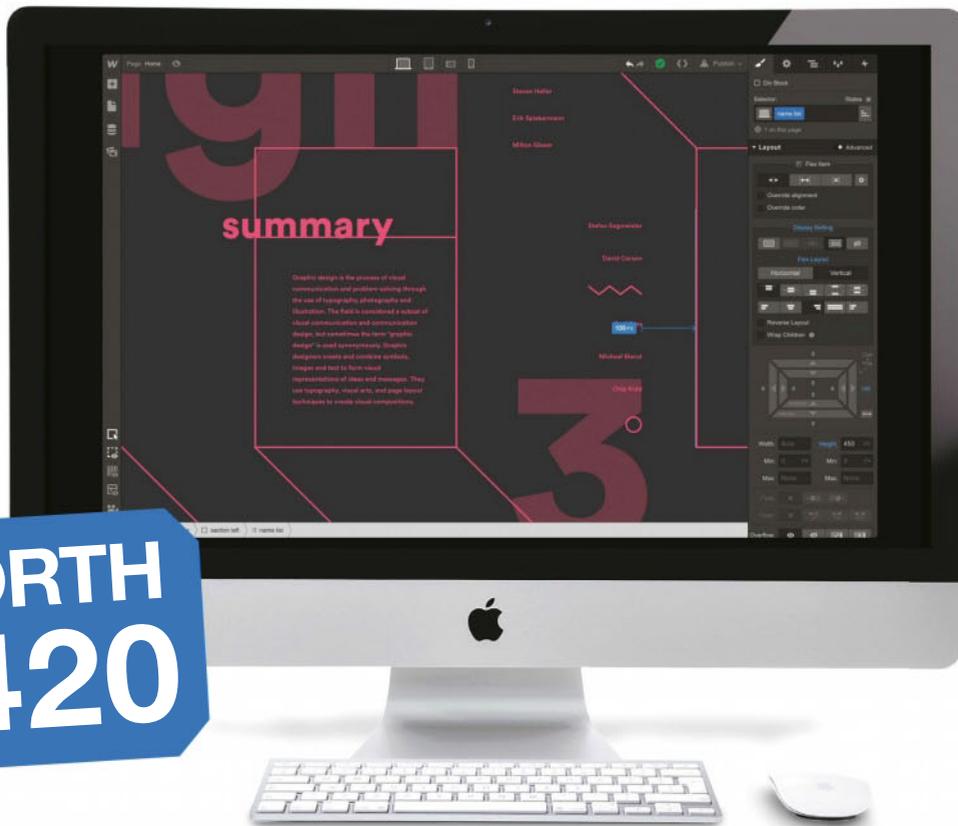
### Sound Wave

[bit.ly/2Ab6lxd](https://bit.ly/2Ab6lxd)

A simple animated preloader that mimics the movement of a graphic equaliser. Experiment with the code and colours.

# WIN

## A YEAR'S WORTH OF WEBFLOW PRO



**webflow**

Webflow gives you all of the power of HTML, CSS, and JavaScript, but in a 100 percent visual interface. This enables you to build production-ready, responsive websites powered by dynamic content — in your

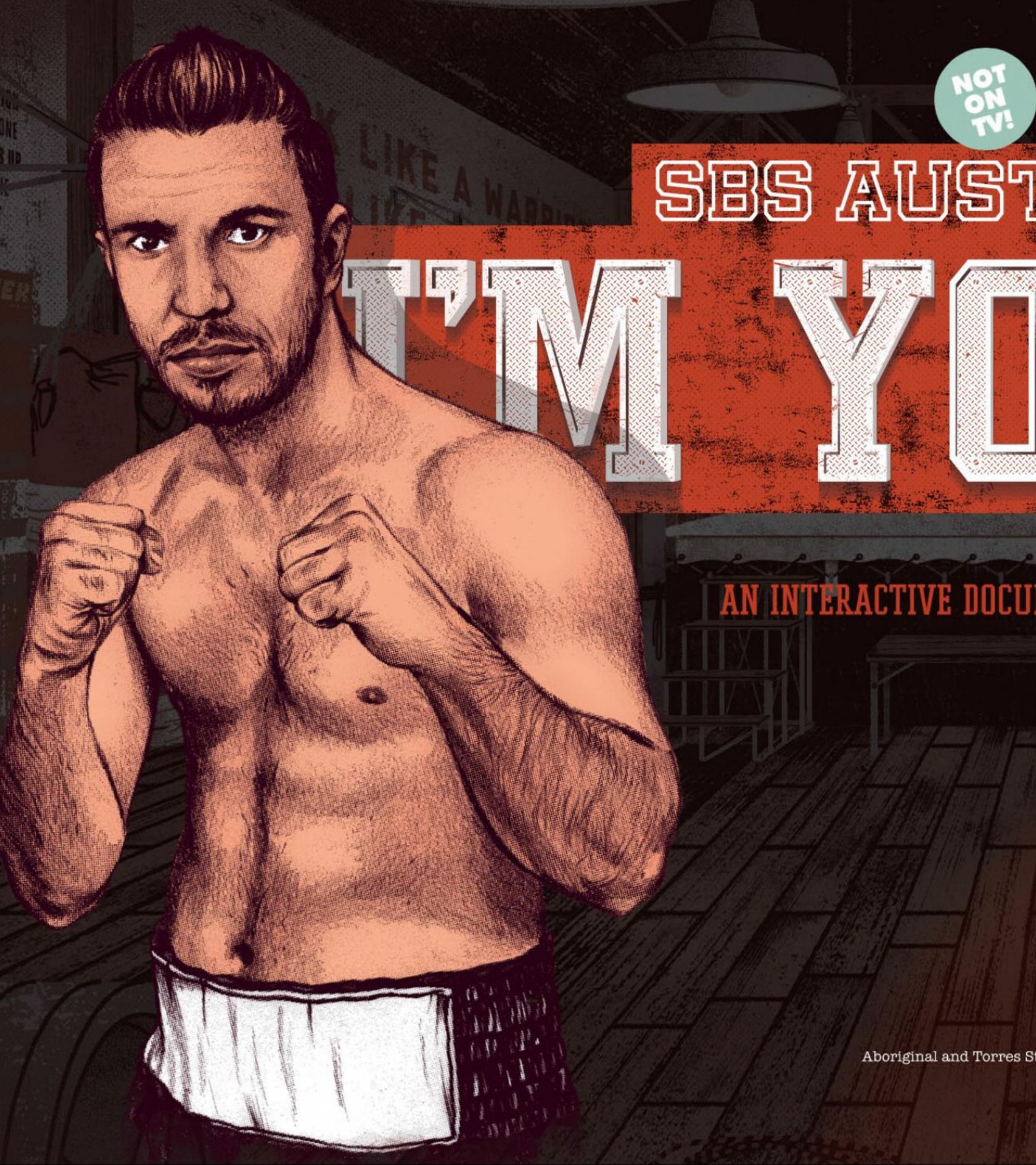
browser, and without writing a single line of code.

You can experience the speed and power for yourself, for free, by visiting [Webflow.com](http://Webflow.com). Once you tried it you will want to use, and then win it.

## HOW TO ENTER

All you need to do is answer one simple question. Head over to [bit.ly/2Agk0WZ](http://bit.ly/2Agk0WZ) and choose the correct answer. It really is that easy.

\*TERMS AND CONDITIONS: The competition is open to UK entrants only. Under 18s must obtain parental consent to enter this competition and be able to demonstrate this to Web Designer's reasonable satisfaction. Answers must be received by 11/01/2018. The winners will be selected at random from all correct entries received and will be sent the prize free of charge. For full terms and conditions, please go [futurepic.com/competition-rules/](http://futurepic.com/competition-rules/)



NOT ON TV!

SBS AUSTRALIA

# I'M YOUR MAN

AN INTERACTIVE DOCUMENTARY

Aboriginal and Torres Strait

## I'm Your Man

[sbs.com.au/imyourman](http://sbs.com.au/imyourman)

Designer: SBS Online – [sbs.com.au](http://sbs.com.au)

Development technologies GSAP, HTML5, CSS3, Apple Mobile, FontAwesome

“SBS Australia’s online team present a gesture-based interactive documentary experience that relives iconic boxing matches in the nation’s history”

A UNIQUE ONLINE ONLY EXPERIENCE!

LightBox  
I'm Your Man

# AUSTRALIA PRESENTS OUR MAN

DOCUMENTARY ABOUT AUSTRALIA'S BOXING LEGENDS.

BASED ON THE STAGE PLAY BY  
*Roslyn Oades*  
WITH MUSIC BY  
*Briggs*

**BEGIN THE DOCUMENTARY**

a 20 minute experience

Portrait Islander viewers are warned that the following program may contain images and voices of deceased persons.



#F4AF91



#BE412E



#9DC4B8



#2E221E

abcABC abcABC  
1234567 1234567

**Above**  
The ITC American Typewriter font by Joel Kaden and Tony Stan is used in Light and Medium typefaces.

*abcABC*  
*1234567890*

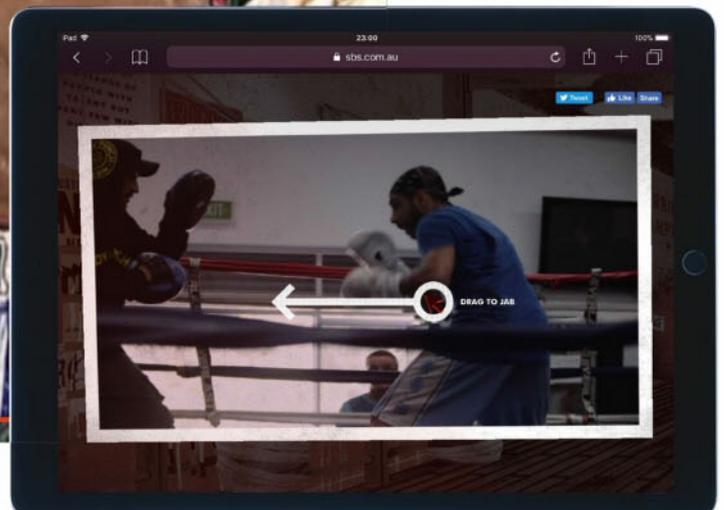
**Above**  
Ed's Market by Laura Worthington evokes a painted sign look, ideal for a retro boxing gym aesthetic.



**Above**  
Visitors can explore a stylised illustrated boxing gym environment, completing chapter-based encounters to unlock new fighters



**Below**  
Optimised for touchscreen devices, fight actions are directed by popup gestures, which echo the successful boxing manoeuvres



**Above**  
The user experience cleverly fuses real video footage with time-based events to gamify a compelling sports retrospective

# Create an animated opening text effect for any website

Use CSS and JavaScript to welcome visitors with an enticing title animation

## 1. HTML document initiation

Start the project by initiating the HTML document. This defines the HTML document container, along with its inner child and body sections. The head section's primary purpose is to store the references to the external CSS and JavaScript files. The body section will store the visible content created in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Flickering Text Transition</title>
<link rel="stylesheet" type="text/css"
media="screen" href="styles.css"/>
<script src="code.js"></script>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. HTML content

The opening words are contained in a text container that has the 'opening' class. An H2 element is used for this example, but any element could be used as long as the text is stored directly within the element; that is not within another container element. For this case, an H2 is used to present the effect as a title.

```
<h2 class="opening">
  All cows eat grass
</h2>
```

## 3. JavaScript definition

Create a new file called 'code.js'. JavaScript is used to give more control to CSS without adding complexity to the HTML. This code will search for elements with the 'opening' class and replace their text to be wrapped inside span tags. The aim of doing this is to allow CSS to manage the presentation of individual words.

```
window.addEventListener("load", function()
{
  var nodes = document.querySelectorAll(".
opening");
  for(var i=0; i<nodes.length; i++){
    var words = nodes[i].innerText.split("
");
    var html = "";
    for(var i2=0; i2<words.length; i2++){
      html += "<span>"+words[i2]+"</span>"
    }
    nodes[i].innerHTML = html;  }
});
```



## 4. Initiate CSS

Create a new file called 'styles.css' for defining the visual presentation rules. The first step in this file is to set the page document and body to appear with a black background. A potential conflict with the default text colour also being black can be avoided by setting the default text colour to white.

```
body{
  background: #000;
  color: #fff;
}
```

## 5. Opening words

Thanks to Step 3, each of the text within the 'opening' container is now contained within a span element. This step sets each of these words to be invisible by default and to have a margin on their right side for spacing. Most importantly, two animations are applied – 'animateOpen' with a duration of eight seconds, and 'flash' with a duration of 0.1 seconds.

```
.opening span{
  position: relative;
  opacity: 0;
  animation:
    animateOpen 8s ease-in-out infinite,
    flash .1s ease-in-out;
  margin-right: 1em;
}
```

## 6. Animation delay

The effect requires each of the text items inside the opening container to start at different times. This is achieved by using the 'animation-delay' attribute combined with the 'nth-child' selector to provide unique timings. Our example sets delays for each fourth child span and the three child elements between them.

```
.opening span:nth-child(4n) { animation-
delay: 3s; }
.opening span:nth-child(4n-1) { animation-
delay: 2s; }
.opening span:nth-child(4n-2) { animation-
delay: 1s; }
.opening span:nth-child(4n-3) { animation-
delay: 0s; }
```

## 7. Animation definitions

Two animations are required from Step 5. The 'animateOpen' animation changes the word from fully visible to invisible, while the 'flash' animation uses a white text shadow to appear as a flash around the text that fades out. The duration of these animations are defined through the way they are called in Step 5.

```
@keyframes animateOpen {
  from { opacity: 1; }
  to { opacity: 0; }
}
@keyframes flash {
  from { text-shadow: 0 0 32px white; }
  to { text-shadow: 0 0 0 white; }
```

01/05

# ONLINE ACADEMY

Online academy provides profound understanding of haircutting skill.

Doing one course after another students immerse in the skill and systematically accumulate knowledge with the help of video lessons and lectures.

Information is provided visually and in detail, which gives our students the opportunity to master the basic elements without a trainer and

~~read more~~



## Anna Eshwood Ltd

[annaeshwood.com](http://annaeshwood.com)

Designer: Big Drop - [bigdropinc.com](http://bigdropinc.com)

Development technologies WordPress, MooTools, jQuery, Vimeo

“Promoting Anna Eshwood’s Matrix 9 system for coaching hairdressers, this chic site blends glossy photography with video-based tutelage”



#F3F2F6



#E0B298



#A3392D



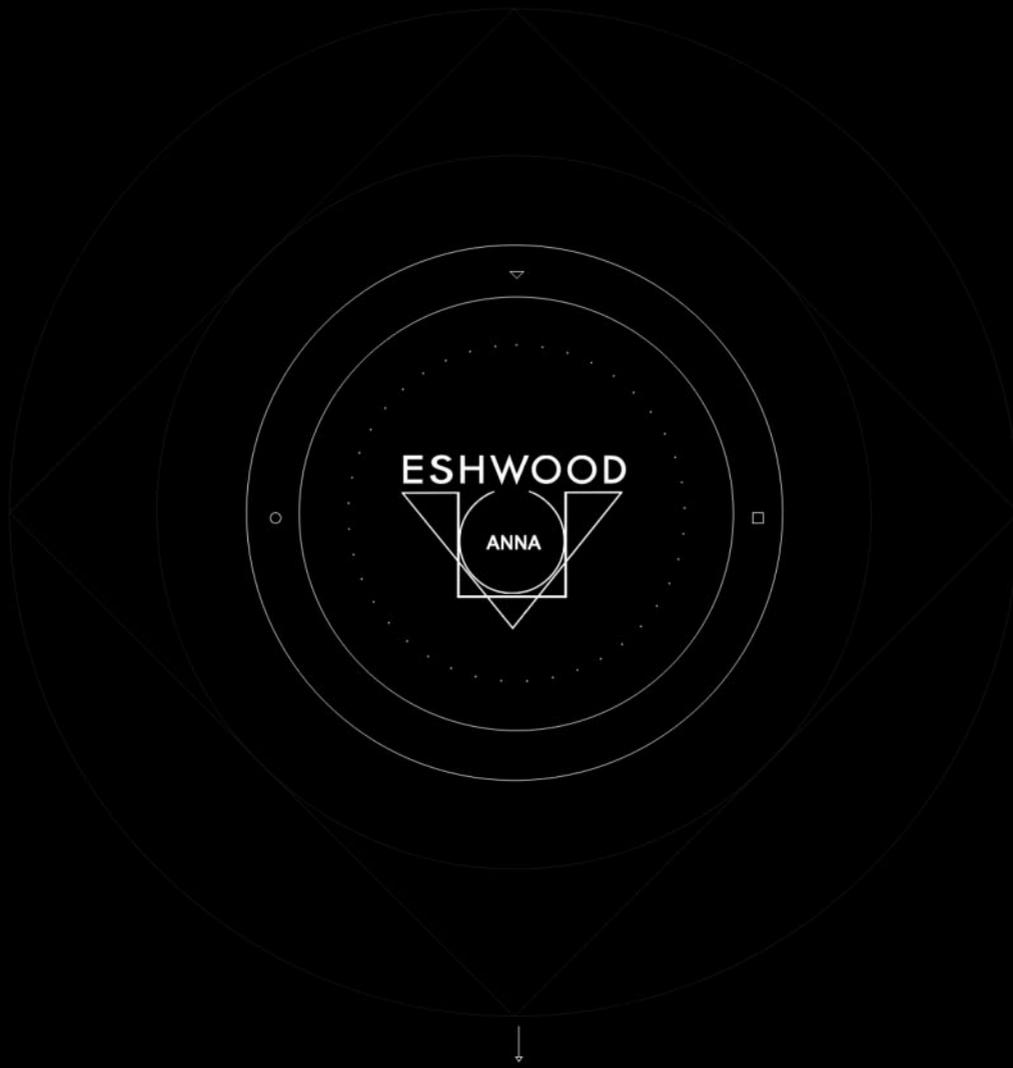
#010101

**ABCABC** ABCABC ABCABC  
**123456** 123456 123456

**Above**  
Bebas Neue by Ryoichi Tsunekawa at Dharma Type is used in Regular, Book and Bold typefaces throughout.

abcABC  
1234567890

**Above**  
Fira Mono by Carrois Apostrophe in Regular provides the angular styling to subheadings and paragraphs.



**Above**  
Opening after a skippable video intro, Anna Eshwood features a darkly stark header screen with animated SVG logo.



**Above**  
Eschewing much template colour beyond black and white, a look book section stacks big, bold fashion photography.

# Create an interactive expanding hover button effect with CSS

Use CSS features to create informative user interface components that respond to user interactions

## 1. Initiate HTML document

The first step is to initiate the HTML document. This consists of the HTML container, which stores the head and body sections. While the head section is primarily used to store the link to the external CSS file, the body section will store the visible page content that is going to be created in the second step.

```
<!DOCTYPE html>
<html>
<head>
<title>Hover More</title>
<link rel="stylesheet" type="text/css"
media="screen" href="styles.css"/>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. HTML content

The HTML content consists of a span container that has the 'button' and 'expand' classes applied to it. Two additional elements are stored inside this container; the first stores the default content, while the second contains content to display when being hovered. The presentation rules for these are defined later via the CSS.

```
<span class="button expand">
  <span>&#x2193;</span>
  <span>More...</span>
</span>
```

## 3. Initiate CSS

Now the HTML has been completed, it's time to create a new file called 'styles.css'. This file is loaded from its reference in the HTML head section. The first rules in this file will set the HTML document and body to cover the full browser window with a black background and without any visible border spacing.

```
html, body{
display: block;
width: 100%;
height: 100%;
padding: 0;
background: #000;
}
```

## 4. Button container

The element is meant to flow alongside regular text content, hence being assigned an inline-block display. The other important attribute is the overflow, which is set to hidden so that only one of the inner content containers are visible. This step also assigns the default colour and height of the button.

```
.button{
display: inline-block;
border: 1px solid #fff;
color: #fff;
height: 2em;
overflow: hidden;
}
```

## 5. Button expand containers

The 'button.expand' (without space) references the parent container element that has both the button and expand classes assign to it, while the '>\*' references each first level child within. These child elements are set to cover the full size of their parent, with a half character sized padding on all sides. Importantly, a transition for one second is applied to margin, enabling any changes to this to be animated.

```
.button.expand > *{
display: block;
height: 100%;
padding: .5em;
text-align: center;
transition: margin 1s; }
```

## 6. Button hover

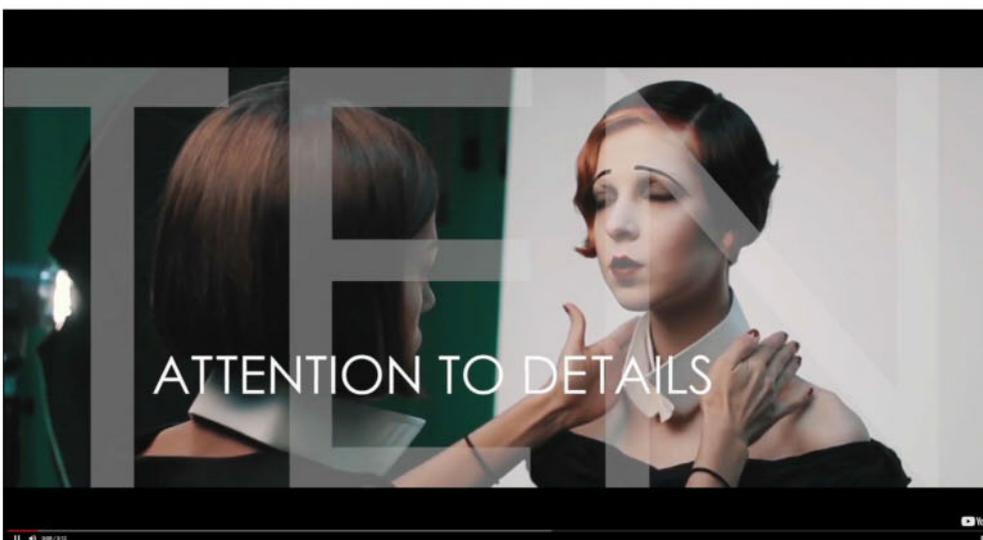
The hover selector is used on the 'button expand' container to set an update to the inner container containers. This update makes their top margin set to -1.6em, which produces the result of pulling their position up by this amount. With the transition applied to their margin in the previous step, this change will appear animated over a duration of one second.

```
.button.expand:hover > *{
margin-top: -1.6em;
}
```

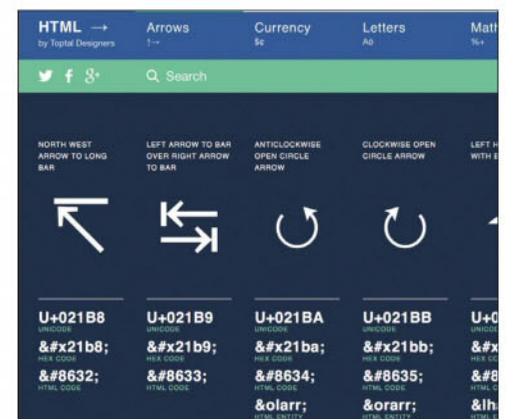
## 7. Arrow symbols

There are many HTML symbols that you can use for your button icons without the need to load a dedicated symbol library. These can be added with special HTML codes or via CSS. The website link provided in this step provides a comprehensive list of useful symbol icons that you could use for your project.

[bit.ly/2AbKb3m](http://bit.ly/2AbKb3m)



**Above**  
Embedded fullscreen YouTube and Vimeo videos provide technical guidance across a range of hairdressing courses.





# À l'heure du lait

[www.ferme-laitiere-france.com/en](http://www.ferme-laitiere-france.com/en)

**Designer:** Sweet Punk – [sweetpunk.com](http://sweetpunk.com)

**Development technologies** WordPress, HTML5 Canvas, Vimeo, MP3 audio

“This digital documentary site follows a day in the life of a French dairy farm, telling the story of milk from every angle”



E DU LAIT

ENCH DAIRY FARM

HE MOVIE • ▶



#B0C8D7



#5D6154



#C4A058



#A29DA5

abcABC  
1234567890

**Above**

The Narziss font family by Hubert Jocham is used in its regular "Drops" typeface to provide elegant headings.

abcABC  
1234567890

**Above**

Montserrat by Julieta Ulanovsky at ZkySky is applied to the styling of <h2> tags and button labels.

# Create an animated ring effect for titles

Use CSS styling and animation to create an animated ring effect for page content titles

## 1. HTML document definition

Start the project with a new HTML file. This step creates the HTML document container, which stores the head and body sections. The head section is primarily used to store the link to the external CSS file, while the body section will store the visible page content that will be created in the next step.

```
<!DOCTYPE html>
<html>
<head>
<title>Animated Ring Border</title>
<link rel="stylesheet" type="text/css"
media="screen" href="styles.css"/>
</head>
<body>
*** STEP 2 HERE
</body>
</html>
```

## 2. HTML content definition

The HTML content consists of a container for the title effect that is parent to the title followed by span elements to be used for the borders. The span elements each contain an inner span element that will be used for the effect. It is important that the title is the first element – for purposes of both SEO and referencing via CSS.

```
<div class="rings">
<h2>Title</h2>
<span><span></span></span>
<span><span></span></span>
<span><span></span></span>
</div>
```

## 3. Initiate CSS

The HTML is now complete; now create a new file called 'styles.css' – the file referenced from the HTML head section. This step defines formatting for the HTML document and body to cover the full browser window with a black background, and without visible border spacing. A white content colour is also defined for content to inherit as its default.

```
body,html {
display: block;
width: 100%;
height: 100%;
padding: 0;
margin: 0;
color: #fff;
background: #000;
}
```

## 4. Rings container

The title and its ring components are stored inside a container with the 'rings' class. This step sets the container's size and relative positioning, so that its child elements can be sized and positioned in relation to wherever the container is. A major advantage of this is being able to change the size and position of the container without concern for the child elements.

```
.rings{
position: relative;
display: block;
width: 25vw;
height: 25vw;
overflow: hidden; }
```

## 5. First level children

Each first level child within the rings container will have a set of default formatting applied to them. These will set their positioning to absolute, along with being positioned in the top left corner of the rings container. Their height is set to cover the full size of the rings container, as well as having the 'rotate' animation applied to them.

```
.rings > * {
position: absolute;
top: 0;
left: 0;
height: 100%;
width: 100%;
animation: rotate infinite linear; }
```

## 6. Second level children

The second level children are the span elements inside the first level span elements. These need to be styled with the visible border style and set to relative, so that they are positioned in relation to their parent span. A border radius of 100% is then applied to make the element appear as a circle, along with individual colours for each corner.

```
.rings > * > * {
display: block;
position: relative;
width: 100%;
height: 100%;
border: 1px solid #ccc;
border-radius: 100%;
border-top-color: rgba(255,255,255,.75);
border-right-color:
rgba(255,255,255,.5);
border-bottom-color:
rgba(255,255,255,.25);
border-left-color: rgba(255,255,255,0);}
```

## 7. First child

The first child inside the rings container is the title, so this step specifically references this element using the nth-child selector to apply its unique formatting. Absolute positioning is applied to position the text, along with setting its height and top position. A calculation is used for the top position – but you could change this to a manual percentage position such as 45%.

```
.rings > *:nth-child(1){
position: absolute;
text-align: center;
color: #fff;
height: 1em;
top: calc(50% - 1em);
}
```

## 8. Ring borders

The second, third and fourth children inside the rings container are used for the border effect. The visual presentation formatting for these were defined in Step 6, but they still require unique formatting for their size, position and animation duration. A margin is used to position them centrally in accordance to their size.

```
.rings > *:nth-child(2){
animation-duration: 8s;
}
.rings > *:nth-child(3){
animation-duration: 12s;
margin: 10%;
height: 80%;
width: 80%;
}
.rings > *:nth-child(4){
animation-duration: 20s;
margin: 20%;
height: 60%;
width: 60%;
}
```

## 9. Animation definition

The final step is to define the animation that the ring elements use – as applied in Step 5. This animation sets two frames for the elements to animate between; starting at a rotation of 360 degrees, they rotate backwards to 0 degrees rotated. The browser will automatically calculate the speed of each animation frame applied to the ring elements according to the animation duration, which was applied in Step 8.

```
@keyframes rotate {
from { transform:rotate(360deg); }
to { transform:rotate(0deg); } }
```

# PUT A PAUSE IN YOUR DAY

With so many demands from work, home and family, there never seem to be enough hours in the day for you. Why not press pause once in a while, curl up with your favourite magazine and put a little oasis of 'you' in your day?



**PRESS PAUSE**  
ENJOY A MAGAZINE MOMENT

To find out more about Press Pause visit:  
[pauseyourday.co.uk](http://pauseyourday.co.uk)



# DIRECTING TRAFFIC

---

BRINGING ATTENTION TO INDIA'S CHILD EXPLOITATION PROBLEM  
IS NOT EASY IN ANY MEDIA. MONTREAL STUDIO FFUNCTION  
TELL US ABOUT HOW IT TRANSFORMED POIGNANT DATA INTO  
AN ENGAGING ONLINE STORY

**(UN)TRAFFICKED**

[bharatyatra.online/untrafficked](https://bharatyatra.online/untrafficked)

by

**FFUNCTION**

<https://ffctn.com/en/index>

@ffollow



Let's begin by getting philosophical for just a moment. If somebody asked you what it was that made you human, gave you that sense of reality, life and the world around you – what might you say? Particularly in this age, a good answer might be data and its importance when interpreting and defining everything. Digital ubiquity has only underlined that fact and highlighted how much data informs our communications, documents our existence and allows us to share that experience with those we care about. But data isn't always terribly glamorous and often the art in online design, if it can be called artistic, is in making data palatable and engaging enough to inspire attention. Montreal studio FFunction has made it its business to do just this, focusing on the information at the heart of projects to forge engaging, award-winning solutions. Known for developing interactive data visualisations as well as data-driven applications, infographics, reports and motion graphics, the team applies a visual storytelling technique that transforms data into something users can relate to. Helping clients including Google, UNESCO Institute for Statistics, the Bill & Melinda Gates Foundation, National Geographic and the Canadian Cancer Society, FFunction has won the prestigious AIGA Justified and the Kantar Information is Beautiful Award for Top Studio. But it is its recent work for the Kailash Satyarthi Children's Foundation we examine this time, fronting a large-scale campaign to end child exploitation in India. The (UN)TRAFFICKED project is an interactive 'choose-your-own-adventure' for desktop and mobile, available in English and Hindi, that crucially intersperses up-to-date statistics on the state of Indian child labour and sexual abuse. So this is data delivery with a conscience and a purpose to highlight plight and ultimately change the world for the better. A big challenge indeed, so we leapt at the chance to do some data gathering of our own to learn how the talented FFunction team approached the task.

## SETTING THE TONE

The initial genesis of how (UN)TRAFFICKED would be realised was similar to how many client projects begin, with prior working knowledge. A year previous FFunction had been commissioned to create an interactive portrait of Children's Investment Fund Foundation's areas of work – chiefly children's health, nutrition, early learning, adolescent reproductive health and climate change. "This provided a way for the user to learn more about CIFF's work through the story of a child's life, from the moment of birth and then through several milestones in her life," starts Creative Director Audrey Lapiere. "Beattie's Story was very successful and the foundation was discussing creating a sequel that would focus on child protection and/or child labour." Those talks unearthed the idea of an interactive narrative akin to the 'choose-your-own-adventure' style of books, with gaming elements brainstormed to provide



Untrafficked highlights the plight of vulnerable Indian children in 'choose-your-own-adventure' style.

an engaging edge. "In the end, CIFF decided that this might actually be a good project for their partner, the Kailash Satyarthi Children's Foundation (KSCF), to lead instead, as they were making preparations for a huge international campaign dealing with the issue of child trafficking and sexual abuse. So CIFF made all the requisite introductions and we began to work together to develop an interactive digital story that would depict child labour, sexual abuse and commercial sexual exploitation in India." An immediate challenge here would obviously be getting the tone right for such a weighty subject matter, striking a balance between employing gaming elements while still treating a very loaded topic with enough sensitivity. "Obviously it was also vital that the story would resonate with an Indian audience, so we worked closely with the KSCF to make sure that we hit the mark aesthetically and in terms of narrative."

## INTERNATIONAL COLLABORATIONS

This project would revolve around the co-operation between three parties and FFunction dealing with both the CIFF and KSCF as clients. Such a setup would bring timezone differences between Eastern North America, India, and Western Europe that would make scheduling

## PROJECT STATS

### PROJECT DURATION

23 weeks

### ACTUAL PRODUCTION

12 weeks

### PEOPLE

#### Founder & Director

Sébastien Pierre

#### Creative Director

Audrée Lapiere

#### Art Director

Wim Bruyninckx

#### Graphic Design & UX

Chloé-Ève Levasseur

#### Developer

Sam Dupras

#### Project Manager

Jacqueline Smoak

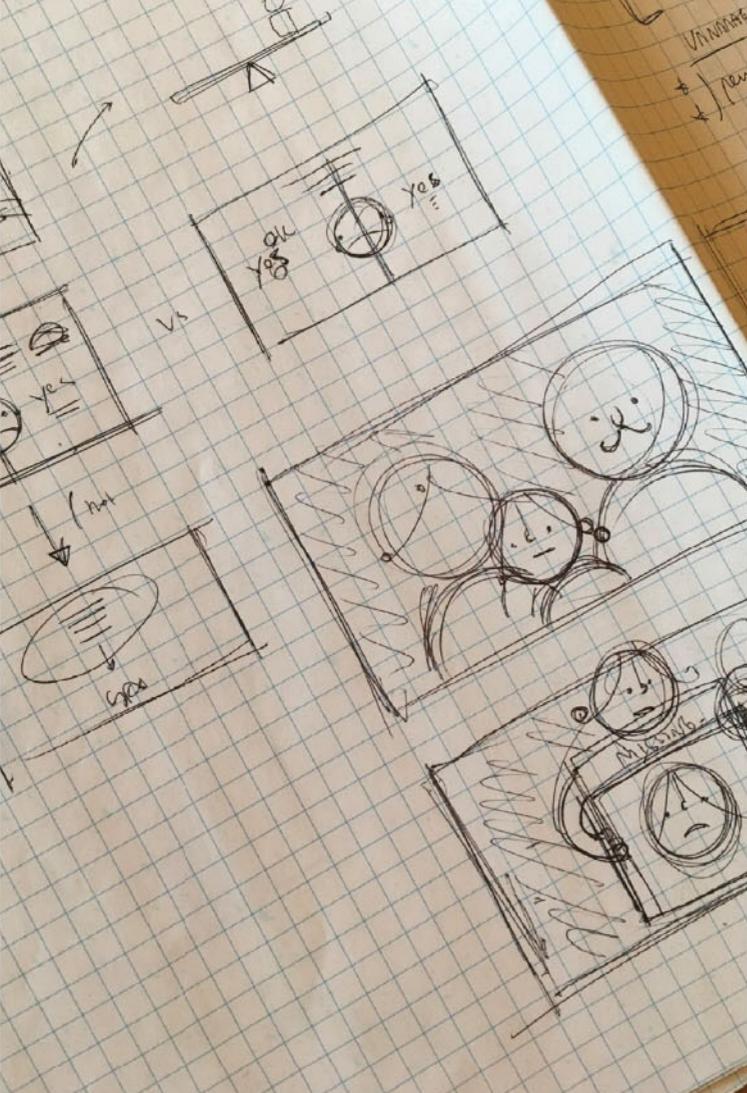
#### Marketing Director

Rebecca Galloway

#### Studio Dog

Byron





▶ meetings understandably tough. “Fortunately we had a very flexible client in the Kailash Satyarthi Foundation, who were willing to take meetings quite late in the evening with as much communication as possible done via email,” Project Manager Jacqueline Smoak explains. “Here we discussed everything from the flow of content to the styling of the character’s attire. We also used sharing tools such as a Google Doc for tracking the story’s content. This made collaboration easier.” The team knew from experience that getting multiple disparate

parties to buy into a basic format quickly would improve productivity, enabling them to work fairly independently and seek input on the project deliverables only. Early versioning on the narrative was performed via a Google Doc with feedback attached to that, before black-and-white concept work would explain story structuring, suggest calls to action and demonstrate how users could affect outcomes. Approval here would lead on to wireframing and copywriting, with the latter requiring a ‘bullet-proof’ translation process and special practical

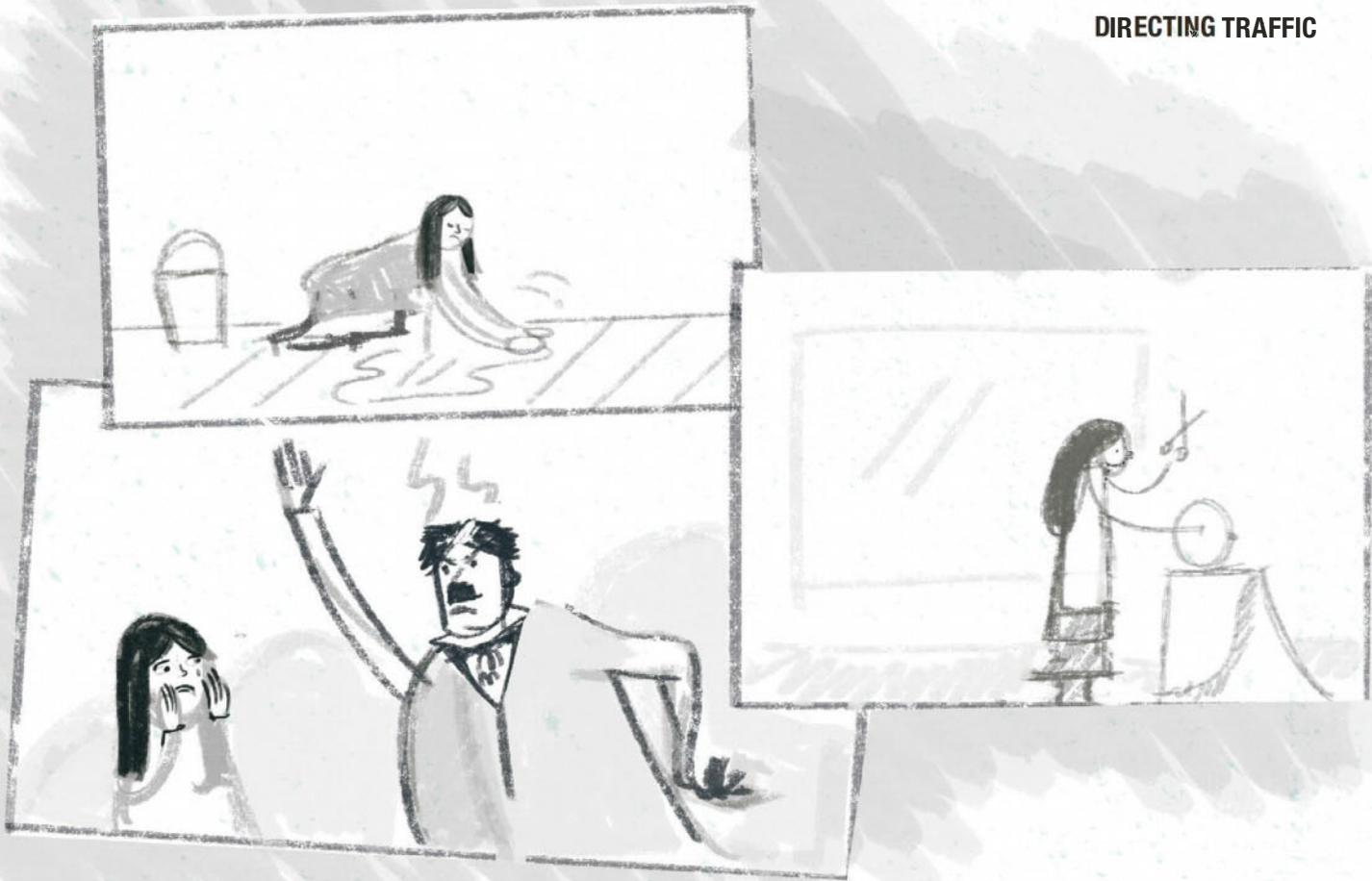


## NARRATIVES AND NUMBERS

For the uninitiated, the (UN)TRAFFICKED experience follows a 13-year-old Indian girl over the course of a life-changing week. FFunction worked hard to create a strong visual storytelling style that gives life to an array of characters that the girl meets during her journey. Every decision the user makes, whether on the behalf of the girl’s father, friend, placement agent, or employer, has a profound impact on her future. “With careful input from the Kailash Satyarthi Children’s Foundation, we focused on nailing down the narrative around this girl

with some degree of verisimilitude,” Audrée Lapierre, Creative Director, confirms. “We wanted to make sure we covered the different areas of child protection, like child labour, sexual abuse and commercial sexual exploitation through the various storylines in the interactive. We soon realized that we had many actors within the girl’s world making decisions that had a profound impact on her and yet she was the only one who had no voice. This powerlessness, or lack of agency, became an integral part of the overall concept of child protection.”

Such a compelling story was of course rooted firmly in the kind of factual data that FFunction considers to be its bread and butter. As a data visualization studio, their work most often starts from statistics and spreadsheets and this one proved to be no exception. “For this project, we were provided with a document of shocking facts and statistics of international, national and state crimes against children. We made sure to sprinkle some of those factlets at key points in the interactive, like anytime a user makes an important decision.”



**ABOVE: The rough black-and-white concepts were a core element of the project.**

measures in order to apply the Hindi text in the correct formatting and location. "Again, we used a Google Doc with an embedded table. Screenshots from the interactive were included as reference points, and we used a coding system to refer back to different sections of the game if necessary. These codes were removed for production, but were very helpful during the content development and translation process."

### DEVELOPING A STORY

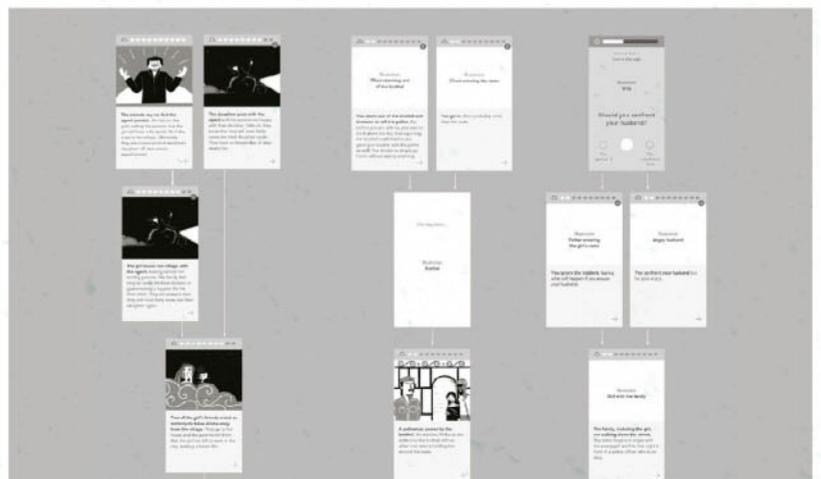
Concept conversations initially threw up the idea of an 8-bit style game that was discarded due to doubts over how receptive an Indian audience might be. FFunction therefore proposed instead a more minimalist, high-contrast and bright illustrative look featuring gif-style animations to add life. "Once we had a

solid enough storyline, our Art Director created a storyboard so that we could discuss with the clients what we would be illustrating and animating," Lapierre reveals. "The rough black-and-white concepts, the narrative document and the storyboard were the three core elements we had to get right before moving on with the design." That subsequent design work would be split into parallel workflows with UX Designer Chloé-Ève crafting user flow and interface elements in Sketch while Art Director Wim Bruyninckx tackled content storyboarding, illustration and animation. "One of the first visual deliverables was the moodboards because we wanted to get approval on the artistic direction before nailing down anything else. We proposed different illustration styles and colour palettes and got a feeling for what they liked and disliked. While the story was

being drafted, we created wireframes that explained the different elements of the user interface such as how you can select a name for the girl and a region that she grows up in, how you move from one screen to the next, how the timer appears when making a decision, etc." Wim would

## "WE PROPOSED DIFFERENT ILLUSTRATION STYLES AND COLOUR PAlettes AND GOT A FEELING FOR WHAT THEY LIKED AND DISLIKED"

map how content could be split into frames and identify illustrations and animations with storyboards that Chloé used to create flowcharts plotting the user journey and story pathways. Documentation here was then indexed in a Google spreadsheet with numbered ▶



## ADDRESSED IN ADVANCE

As is customary for most digital agencies worth their salt, FFunction is no exception when playing the long game on its projects. Offering certain warranty guarantees on any problems encountered after launch, the team prefers however to minimise those risks well before delivery time. “Rather than rely on repairing errors after an application has gone live, we choose to mitigate those integration issues early on, addressing as many variables in advance of product release as possible,” confirms Project Manager Jacqueline Smoak. “So during the planning and design phase, we always establish a full understanding and consensus between the studio and the client for variables such as application format, sharing/tracking requirements, file hosting and what types of updates may be needed in future. During early development we send a ‘test’ package that is essentially what they will receive for final implementation, minus all of the bells and whistles. A lot of style and integration issues are worked out during this phase. Then, during the final development stage, we send a Beta version of the package to the client to install on their testing servers. Since this includes all the functionality of the final version we have time to work out issues such as style leakage or social media sharing options. We do our final Quality Assurance directly on the client’s version to mimic the final production environment as closely as possible. All these measures serve to mitigate post-production issues and if minor issues arise post-launch, we have a warranty period to cover errors in coding.”

▶ pages that the developers could reference to keep track of page content. “Within our workflow we needed to find a solution to develop the illustrations quickly,” admits Bruyninckx. “We used a vector drawing program to lay out the rough shapes and then we exported this to a multi-layered bitmap for fine-tuning in image-editing software. The challenge for animating the different illustrations was mainly then a technical constraint around how to keep the file sizes low whilst retaining the textured visual style. The solution was to generate animated gifs with a really low frame-rate. Each animation also has a maximum of eight frames and the gif format was then easy to integrate it into the interactive.”

**“WE USED A VECTOR DRAWING PROGRAM TO LAY OUT THE ROUGH SHAPES”**

### ANIMATION CHALLENGES

In a technical sense, there would be no real ‘backend’ for the project with more emphasis being placed on the frontend

visual narrative so fundamental to delivering an engaging story. It was also true that the data requirements were deemed to be much less complex than other projects FFunction had previously worked on such as HP’s Earth Insights for example. In contrast, (UN) TRAFFICKED would prove to be largely illustration and text-based with statistics woven

throughout the experience. “The biggest development challenges were therefore more about getting the animations to feel well-integrated and appropriate, so like when the girl’s Happiness Meter loses a heart, we had to get the attention of the user so that they notice the heart disappearing and process what that means,” begins Web Developer Sam Dupras. “To do that, I had to make the whole title bar flash and adjust the delays of the animation. It was difficult to figure out exactly when it should happen relative to the copy showing and the speed that



the user reads. It's a delicate thing. Page transitions were also tricky to get just right. We tried a few different approaches until we settled for the dropping panels we have now. Managing the copy across so many pages was also a challenge. We decided to create a YAML file that contained all the copy with special characters for the girl's name, village and user's name, which worked out well."

## MARKETING A MESSAGE

Taking heart from a Kickstarter board game by 24-year-old graphic designer Nashra Balagamwala, highlighting arranged marriage in Pakistan, the team were confident of success. The media attention that 'Arranged' received around its engaging, interactive approach to another important human rights issue boded well for (UN)TRAFFICKED's launch. This final phase would hinge on getting people talking and acknowledging the message, so thoughtful marketing was paramount. "Once I had posted the project across FFunction's social media channels, I began to reach out to journalists who had written about similar projects." explains Marketing Director Rebecca Galloway. "Journalists who were already writing about these sorts of projects and would really understand immediately what (UN)TRAFFICKED is trying to achieve. They got it straight off the bat and the media approaches began to bear fruit quickly. Additionally, the Kailash Satyarthi Children's Foundation did a big publicity push around Universal Children's Day held on November 20th each year. It provided an ideal hook for the media to use, and garnered plenty of interaction on social media." While some of FFunction's projects have largely targeted policymakers and journalists, the team also asserts that (UN)TRAFFICKED



To keep file sizes low animated gifs with a really low frame-rate were used.

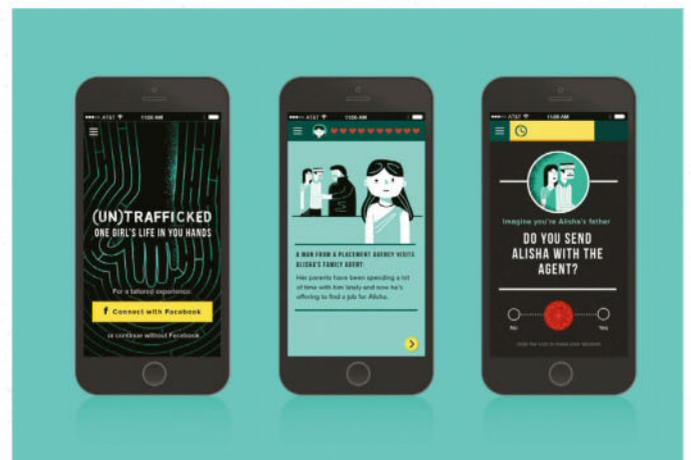
is crucially for the people and especially for Indian society.

Its availability in both English and Hindi speaks to that quite literally and despite India's diversity across spoken languages and dialects, this provides a real point of pride for reaching out successfully. "Here at the studio we've had great responses to the interactive from all over the world, from Paris to New Delhi to Rio de Janeiro. At the time of writing, the site has been visited more than 110,000 times, with the game being played the most in India, which is a great result. The game has also garnered significant acclaim on social media with over 100,000 people and organisations engaging with the Kailash Satyarthi Children's Foundation about the game and the campaign as a whole." ❀



**FFunction's Marketing Director Rebecca Galloway gives her verdict on which feature represents a personal highlight and qualifies why the team looks on it with such pride**

**"FOR ME, THE STRONGEST ELEMENT OF (UN)TRAFFICKED IS THE INTERPLAY BETWEEN THE ILLUSTRATIONS, SOUND DESIGN AND THE SUBTLE GIF-STYLE ANIMATIONS. OUR ART DIRECTOR WIM MANAGED TO AMALGAMATE THE PLAYFUL ASPECTS OF A CHILD'S WORLD AND THE SINISTER INTENTIONS OF THE PEOPLE AROUND HER; NOT AN EASY THING TO BALANCE, AESTHETICALLY"**



# SAVE UP TO 49% ON A SUBSCRIPTION THIS CHRISTMAS



FROM  
JUST  
£13.75

FROM £13.75 EVERY 3 MONTHS (€73.40 / \$92.88 PER YR)

*Biggest savings when you buy direct*

*Choose from a huge range of titles*

*Delivery included in the price*

**ORDER HOTLINE: 0344 848 2852**

**PLEASE QUOTE XMAS17 WHEN ORDERING BY PHONE**

LINES ARE OPEN MONDAY - FRIDAY 8AM TO 7PM AND SATURDAY 10AM TO 2PM (GMT)



**OFFER ENDS  
31 DECEMBER 2017**

*You might also like...*



**FROM £14.75 EVERY 3 MONTHS**  
(€99.03 / \$121.26 PER YR)



**FROM £12.65 EVERY 3 MONTHS**  
(€104 / \$137 PER YR)



**FROM £12.70 EVERY 3 MONTHS**  
(€104 / \$137 PER YR)



**FROM £13.70 EVERY 3 MONTHS**  
(€79.22 / \$100.62 PER YR)



**FROM £12.65 EVERY 3 MONTHS**  
(€104 / \$137 PER YR)



**THE PERFECT  
PRESENT FOR  
EVERYONE**

**SEE THE FULL RANGE AND ORDER ONLINE**

**[myfavouritemagazines.co.uk/xmas17](http://myfavouritemagazines.co.uk/xmas17)**



\*Terms and conditions: Savings calculated against the full RRP (single issue price x frequency). Dollar prices quoted are for the United States, other global territory dollar pricing may vary. This offer is for new subscribers only. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum term specified and will expire at the end of the current term. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. All gift subscriptions will start with the first issue in January 2018. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available on request. For full term and conditions please visit: [bit.ly/magtdc](http://bit.ly/magtdc). Offer ends 31st December 2017.

**WHO** AREA 17

**WHAT** Strategic planning, visual identity, user experience, web and mobile engineering, analytics and optimisation

**WHERE** Paris and New York

**WEB** [area17.com](http://area17.com)

---

**KEY CLIENTS**

Opéra National de Paris

ESPN

Facebook

Vice

Harvard Art Museums

Quartz



Engineered  
Design

---

---

**G**eorge Eid founded AREA 17 in a corner of a large loft in Williamsburg, Brooklyn. The space was called The Ranch and was primarily dedicated to George's filmmaking activities. 1999 marked the first year AREA 17 poked its nose out into the internet fray, working for the legendary hip-hop label Tommyboy Records. However, the agency in its current incarnation was officially founded several years later.

In 2003, AREA 17 was a new agency in search of its symbol. George contacted renowned interactive designer and founder of Elixirstudio, Arnaud Mercier. After a few days of non-invasive coercion techniques, Arnaud agreed to assist in the creation of AREA 17's visual identity.

Soon, it became clear that both George and Arnaud's ideas played well together. What started as the designing of a logo turned into the creation of a mutual philosophy. The collaboration continued on a project basis until 2005 when the duo decided to tie the knot and merge companies. Arnaud left his studio overlooking a small Mediterranean port in Marseilles to come work face-à-face with George at The Ranch in Brooklyn, overlooking a dumpster.

From the start of this new beginning, their eyes were set on being an international agency. With clients from all over the world, and the inherently borderless nature of their work, planning for a European studio started immediately. Paris, France – the city of love and the home of croissants – was the chosen location. The effort was spearheaded by Dominique Deriaz, a blood-in-blood-out Parisian, who had joined the agency in 2004 as a senior producer and information architect.

Since the internet boom of the late '90s and before her séjour in New York, Dominique had been working the web scene in Paris. She was looking to move back and opening the European studio was the perfect opportunity. In 2006, Dominique became a partner and AREA 17 officially opened its doors (and hearts) to Europe with its new Paris studio.

During the years that followed, George split his time between the Paris and New York studio. As the agency grew from both sides of the Atlantic, it became clear that stronger leadership was required in the New York studio. The reins were taken up by Kemp Attwood, who joined the agency in 2007 as a design director in the Paris studio. ▶

Design is woven into the DNA of **AREA 17**. From its inception, the ambition to transform communication using diverse channels and approaches has shaped this studio.

Today, few can match the breadth of understanding AREA 17 bring to the digital channels and beyond

Back in the day, Kemp and Arnaud were close collaborators at an agency in Vancouver and immediately before joining the Paris studio, he was directing designers at an agency in Portland, Oregon. After his two-year séjourn in Paris, he was ready to move back to North America and in 2010, he became a partner and AREA 17's New York studio gained a new leader.

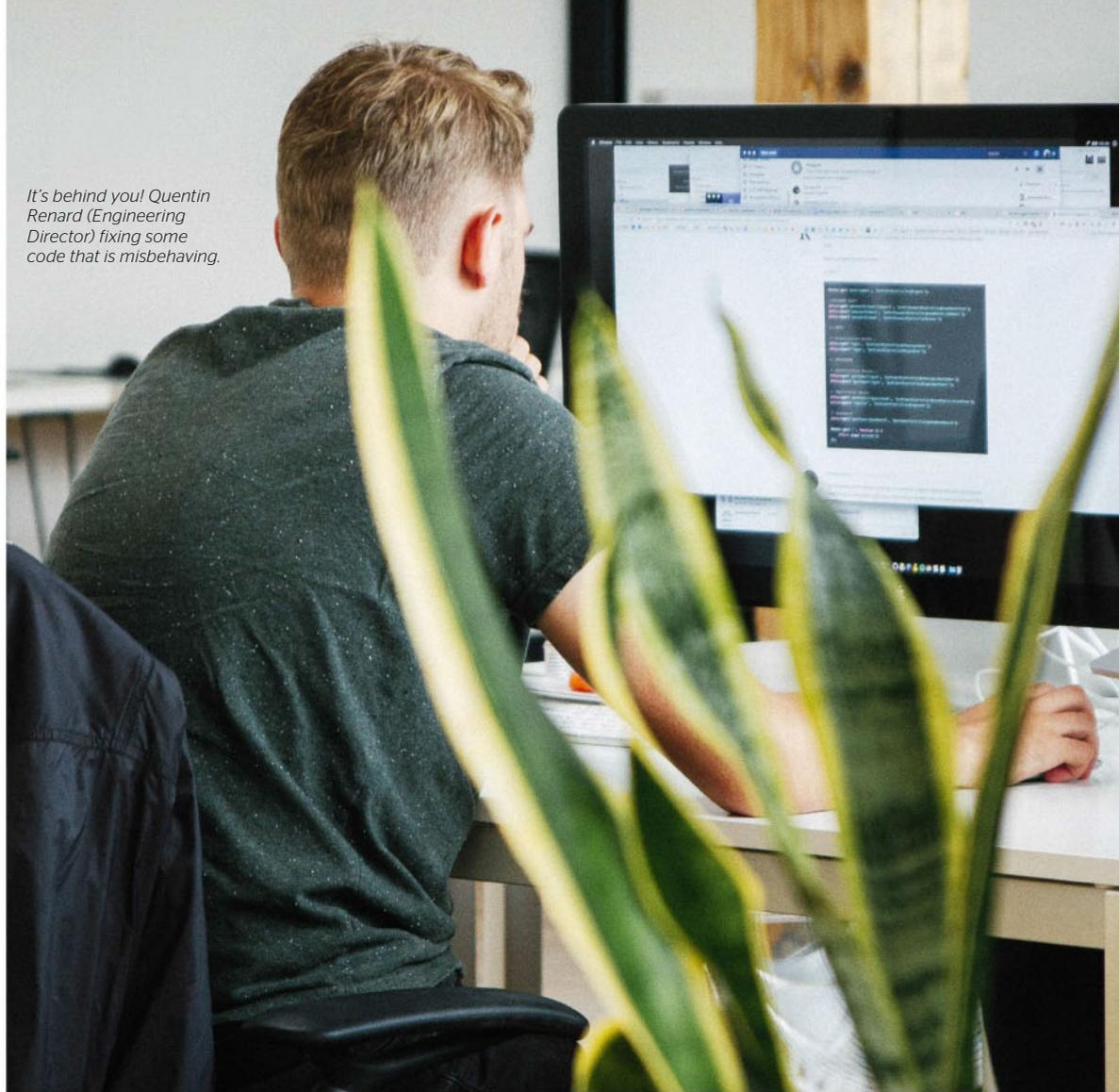
In 2011, Arnaud was diagnosed with chronic myelogenous leukaemia and died on September 26 that year. Widely considered to be among the most important and prolific interactive designers, his death was a great loss for the agency and the industry as a whole. An online collection devoted to his work is available at [arnaud.area17.com](http://arnaud.area17.com).

For AREA 17, naming their new studio was an important first step as George explained: "AREA 17 refers to the visual cortex of the brain where sensory data is received, patterns recognised, and images formulated. While there is a mechanical aspect to human vision, it is sensory memory that constructs most of what we see as individuals."

AREA 17 now has a website that uses the latest techniques to showcase the studio and its work. However, as George commented, it wasn't always like this: "Our first website was an original concept using a basic file structure to present the work. Echoing the early days of the internet and representing the fact that the work – not the website – should speak for itself. This first website was a big success. In 2008, the second version of our website was launched and received many accolades including best portfolio website awards."

George continued: "By 2015 we still had the same website and while we knew it was old, we felt that it stood the test of time, and ultimately we didn't have the time to redesign. By 2016 we realised that our phone stopped ringing and most of the business we received was from existing clients. A close friend of the agency opened our eyes by saying, "Of

*It's behind you! Quentin Renard (Engineering Director) fixing some code that is misbehaving.*



course you are not getting new business, your website is old and dusty. Your situation is 911 and you must relaunch ASAP!" Heeding his advice, we took down the website and within a few weeks, launched a one-page temporary website. The phone started ringing almost immediately. It was magical! Within a few more months, we launched our full site and 2017 has been our best year yet."

The new website was clearly a key component of the studio's marketing, but how else does AREA 17 gain new clients? "Most of our business comes through word of mouth and reputation," said George. "Many of our clients stay with us for years and recommend us to others. That said, we recognise that when most of your business is inbound, you are only working with clients that think of you.

**Most of our business comes through word of mouth and reputation. Many of our clients stay with us for years and recommend us to others**

While over the years we've been honoured by the fact that many great clients think of us, we also understand that there are many other great clients that may not be thinking of us. As our agency grows, we look to develop our outbound activities through thought-leadership and active outreach to clients we'd love to collaborate with."

How an agency approaches its business can be manifold. The work completed has to fulfil a number of criteria that not only meets financial needs, but also enables the studio to grow as a business. George outlined AREA 17's approach: "Over the last 12 years, we've actively kept our agency small in order to maintain quality of craft and effectiveness of output. We've always felt that as an agency, we are defined ▶

## TIMELINE



**2003**  
AREA 17 is founded in New York and creates a first of its kind music discovery and listening platform for Giant Step.  
**employees: 6**

**2006**  
AREA 17 opens its Paris studio and redesigns Madame Figaro, boosting its traffic and user engagement significantly.  
**employees: 14**

**2007**  
AREA 17 partners with Aol to start a three-year redesign of many of their properties including Aol Weather and Moviefone.  
**employees: 24**

**2008**  
AREA 17 creates the brand and website for ALLI, the alliance of action sports and its property the Dew Tour.  
**employees: 26**

**2009**  
AREA 17 partners with Vice to create Motherboard, The Creators Project, and VBS, the largest video platform of its kind.  
**employees: 28**

**2010**  
After the creation of its incubator, AREA 17 launches Slash, the premier art event website in Paris, and Krbb, a buy and sell platform.  
**employees: 30**

## BARNES FOUNDATION

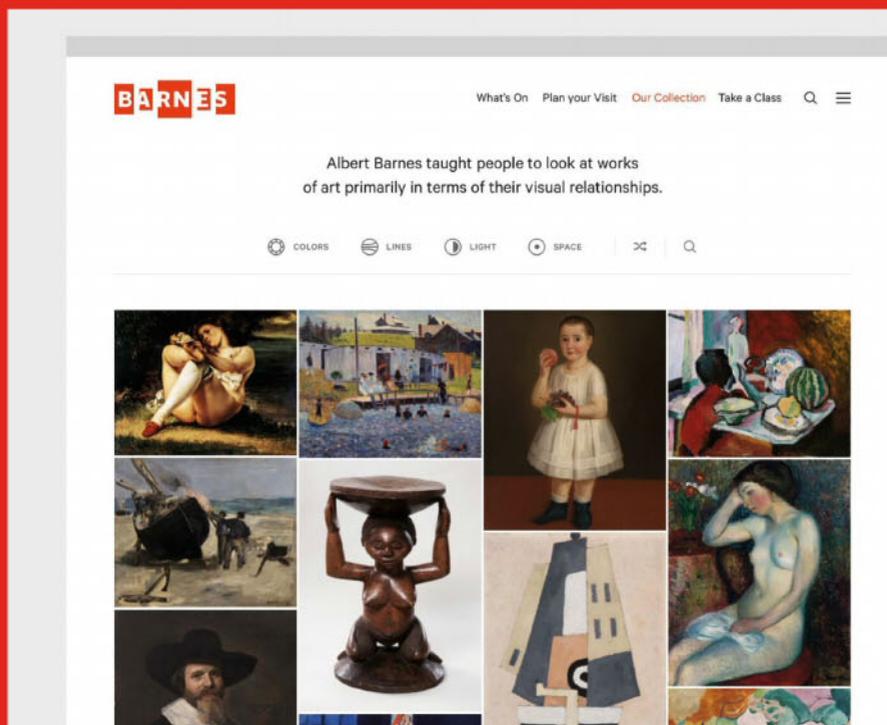
[barnesfoundation.org](http://barnesfoundation.org)

The Barnes Foundation is home to one of the most important collections of Post-Impressionist and early Modernist art in the world, and includes masterworks by Cézanne, Renoir, Matisse, Picasso, Van Gogh, and others. The world-renowned permanent collection is known not just for the artworks themselves, but for the presentation of these artworks in 'ensembles' – groups of paintings, decorative art and sculpture that share formal connections.

Following its move to Philadelphia in 2013, the museum had an opportunity to engage

more regularly with a broad local audience, while continuing to elevate its stature as a must-see destination for out-of-town visitors.

The previous website, designed shortly after the move, made it difficult to elevate programs outside of the permanent exhibition and, as it evolved, expressed a museum-centric rather than visitor-centric organisation of information. Together with the Barnes, AREA 17 helped reshape the story of the museum and create an experience where visitors could easily learn about



the museum, access current and upcoming programming and plan their visit.

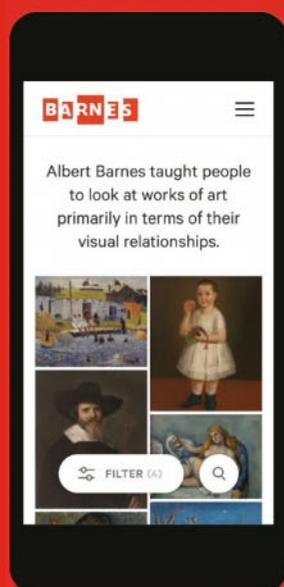
It was important that the redesign reflected the singular nature of the Barnes, and in this we were inspired by the beautiful architecture of the building itself, which achieves a unique blend of minimalism and warmth. By pairing a liberal application of white space with full-bleed, outsized artwork and imagery, the new site approximates the experience of being in the museum.

In support of the aesthetic direction, with each design decision

we asked ourselves if any element could be removed or simplified, such that the overall effect is a 'purity' of information. We limited the number of options on the homepage, reduced the main navigation – tucking less-essential critical pathways under a secondary menu – and designed pages that were dense with information into collapsible components to make it easy for visitors to scan all of the available options before diving deeper on a specific object or topic of interest. The response to the redesign has been overwhelmingly positive.



Above A bold design was adopted for the mobile incarnations of the site. Uncluttered pages let the artwork speak for itself.



Top The use of white space across the new site speaks to the unique architecture that houses the Barnes collection.

Above Inspired use of a collapsible navigation, enables visitors to see the collection.

### 2011

The industry mourns the loss of Arnaud Mercier and a permanent online collection devoted to his work is launched.  
**employees: 32**

### 2012

AREA 17 creates Facebook Stories, a UGC platform that becomes a primary marketing engine for Facebook.  
**employees: 34**

### 2013

Creates the event platforms for Paris Photo and Maison & Objet, the largest design and home decor exhibition in the world.  
**employees: 36**

### 2014

Redesigns the website for ESPN, for the first time in six years, transforming the sports behemoth into a modern, user-first experience  
**employees: 40**

### 2015

AREA 17 redesigns the website for Opéra national de Paris, increasing traffic and ticket sales, and winning the Strategies Grand Prix.  
**employees: 40**

### 2016

AREA 17 creates the video platform for Charlie Rose, presenting over 20 years of interviews with the world's most influential people.  
**employees: 41**

### 2017

AREA 17's long partnership with Pentagram on sites such as Adweek and Billboard culminates in the redesign of their own site.  
**employees: 48**

## THE NEW SCHOOL

[newschool.edu](http://newschool.edu)

Design and social research meet to form an intellectual hub at The New School in New York. But the university was not yet top-of-mind for many prospective students, despite having academic credentials which rival the best schools in the world. Further, the New School was not capitalising on the success of its constituent parts.

Despite Parsons' status as an elite design school, few can grasp how profound an impact its placement within a full-service university has on its progressive approach. So, when The New School came to us with a mandate to "deliver one university," we were thrilled at the challenge.

The New School knew that a lot of their

prospective students were self-starters, but weren't always sure which academic path to take. So, we helped those students explore fields of study through a card-based visual browse experience organised around high-level topics, not courses.

With light and fast interactions that allow users to get a quick overview and links to additional information, students get a bird's eye view of how they might carve their own academic path at The New School.

To tell the story of the university, we crafted a user experience with a strong central narrative and channels that made it easy for visitors to discover its individual schools and programs.

We built a robust

content-publishing platform so The New School can feature the groundbreaking research and work done by students, alumni, and faculty. And with a mandate to deliver technology that can be used across other sites in the future, we developed a modular front-end that opens the door for consistent experiences

across The New School's digital platform.

The look and feel of the user interface reflects the values of a university looking not just to expand boundaries, but to redraw them. We brought their progressive ethos to life online with a bold, open and airy system that extends the type-driven visual identity by Paula Scher

and the rest of the team from Pentagram.

While invigorating us as designers, the dynamic brand required intense focus to control screens and devices. To solve this, we designed site-wide templates as a foundation for content, then collaborated on page-specific effects that reflected the spirit of the university.

by the clients we collaborate with and the excellent work we produce together.

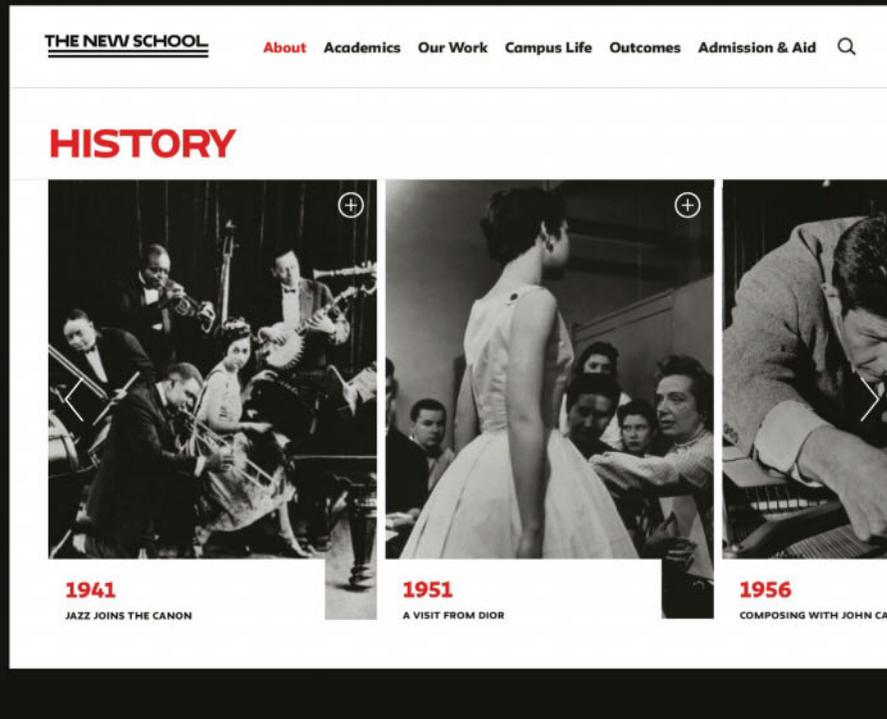
"Choosing each client we work with is paramount to maintaining our mission because our clients' success is our success. First and foremost, we look to collaborate with clients that have success on their horizon – offering great services and products through a team that is able to bring their own mission to fruition. But further, it is important to us that they have design as part of their business strategy (as we do not want to be in the business of educating clients on its value). And finally, that they are bringing some sort of good into the world, including the support of arts, culture, science and education. While there have been times where we've needed to take on projects because of financial reasons, we pride ourselves on the fact that most clients we have worked with are representative of our values, ambition, and mission."

Like most studios, AREA 17 strive to ensure they continue to develop

world-class original work for their client base. However, does the studio have an ethos that is enshrined in the work they create? George commented: "It is difficult to pinpoint a single project that is representative of our ethos as we believe that it is the entire body of our work – as a whole – that best represents the complexities and intricacies of who we are. It is a strong belief of ours that every project we work on must be representative of our values, ultimately pushing our mission forward.

"It is a common idea at AREA 17 that we are building a single platform across all projects – with common features being standardised and client-specific ones representing their differentiating factors. Internally we build developer toolkits that allow us to rapidly develop standard features so that we can spend the lion's share of our time on the ones that'll make them different, that'll ensure our clients' success."

How AREA 17 approach the innovative work they produce offers an insight into



Above A long history in design education was also made a critical component of the new site's design.

Senior Design Director David Lamothe deep in thought. We're not sure what he's listening to. Let's hope it helps him come to a decision.





**Above** The use of bold typography – from Pentagram – ushers visitors across the site to discover not only the available courses, but also the research programs that the school develops.

Architect, Designer, Interface Engineer, Application Engineer, and QA Tester. Additional support team is added as needed for larger projects.

“From an overall philosophy perspective, we are technically agnostic. However, tactically speaking we develop platforms using open source frameworks, libraries, and components. From a language perspective, we specialise in Ruby and PHP (Laravel).

“Internally we develop standardised toolkits and boilerplates that ensure quality, consistency, and rapidity – and allow us to rapidly develop custom solutions as quickly as a packaged one. As for design, we are Adobe-centric, designing mostly in Illustrator (see our design techniques: [guides.area17.com/design-techniques](https://guides.area17.com/design-techniques)). However, we are currently experimenting with other tools such as Sketch (see page 66). For prototyping, we use various tools depending on project needs such as Framer, Invision, Principle and Axure.”

George also commented: “One of our founding principles was to create a new kind of design vernacular that is native to the internet. So, in our early days, when everyone was focused on Flash, we doubled down on HTML/CSS/JS. Our belief was that Flash was a media, not the internet itself, and if we were to design an internet-specific vernacular, then we must stay away from designing media (which includes broadcast design and other types of interactive design).

“This decision served us well and today our agency is on the forefront of web-based mobile technologies. As for jQuery, we do not use it a lot as it is bloated, however we do invest quite a bit into modern JS frameworks such as Angular, React, and Vue.”

Turning to design for mobile, George explained how AREA 17 approaches these design technologies: “While we have a great deal of experience building native apps, our focus is on mobile web technology. We see similarities in the current web app vs. native app debate as the Flash vs. HTML debate of years past. Back then, we committed ourselves to HTML/CSS/JS as we believed its semantic nature was more suited to the strengths of the internet. That decision positioned us well as Flash died off.

“As with Flash, native apps are closed systems. While native apps allow for a high degree of customisation, fluidity and functionality that takes advantage of the smartphone ecosystem (notifications, ▶

**In our early days, when everyone was focused on Flash, we doubled down on HTML/CSS/JS. Our belief was that Flash was a media, not the internet itself**

what drives this studio. George outlined their workflow: “We work on many large-scale projects of different sizes. Excluding strategy only or design sprint projects, it is rare that we work on a full-service project that is less than 4-5 months. An average project is 6-7 months and a large-scale one is 9-12 months.

“It is important to note that this doesn’t necessarily mean that we are not quick to market – on the contrary. When we’re approaching a large-scale project, we most often break it down into phases that reach beyond the first public launch. Our first phase focuses on MVP (Minimal Viable Product) and the feedback mechanisms that give us qualitative and quantitative feedback, allowing us to measure success and identify the priorities for subsequent phases.”

George also said: “We work in small interdisciplinary teams, ensuring that each person is vested in the project’s success. At minimum, a project team includes the following roles: Product Director, Technical Director, Producer, UX



**GEORGE EID**  
 Founder and CEO

*Some people just don't like their photo being taken. A very shy member of the AREA 17 team.*

*“With our commitment to HTML/CSS/JS in the past, today we are committed to web apps because they are open systems that take full advantage of the semantic nature of the internet. As an agency, we’ve made significant investment into web-based technology for mobile to behave more like a native app”*

data storage, etc.), Google and Facebook are both seeking ways to bring these capabilities to web apps. Google has created the Progressive Web App architecture that enables native app features on a web app and Facebook has created React Native to allow developers to use JavaScript (React) to create native apps for both iOS and Android.

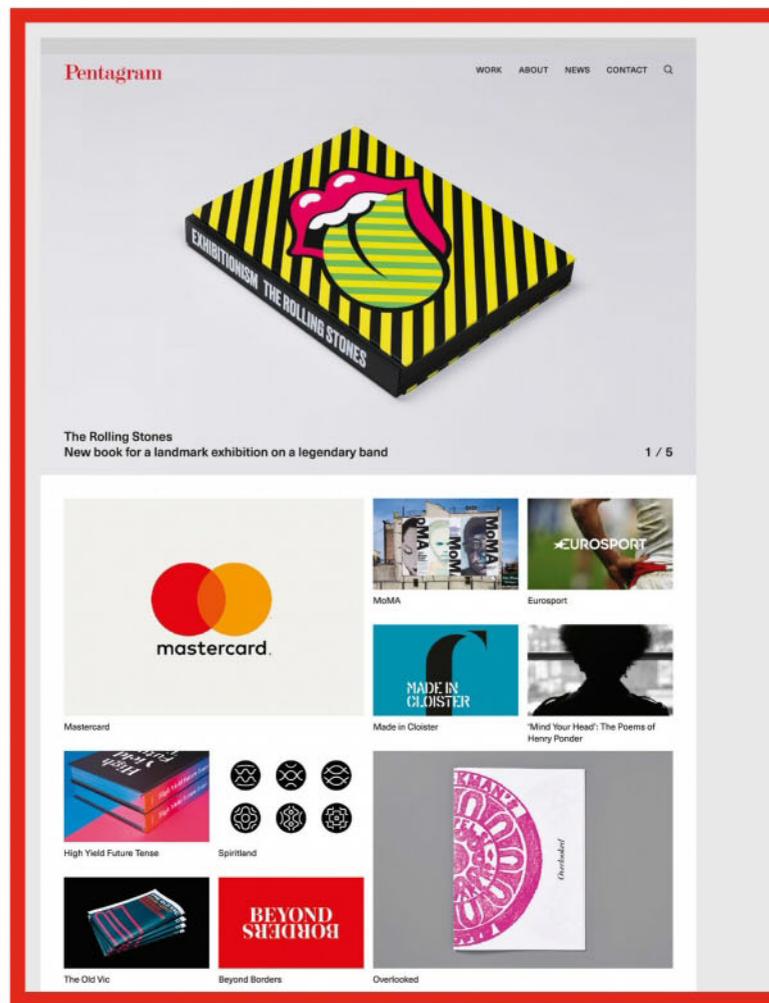
“Also, with our commitment to HTML/CSS/JS in the past, today we are committed to web apps because they are open systems that take full advantage of the semantic nature of the internet. As an agency, we’ve made significant investment into web-based technology for mobile to behave more like a native app. We’ve committed to Single Page Apps that allow for a website – whether

on desktop or mobile device – to behave more like a native app with transitions and fluidity, but with the added bonus of ubiquity. Moving forward we will commit ourselves to Progressive Web Apps and build up our skills with React and React Native so that we can build one app that can then be both web and native.

“As for responsive design, we see this as a starting point. Of course, the design must visually respond to a device. However, a mobile-first approach recognises the importance of other technical subjects such as mobile performance, fluidity, ubiquity, and security. And, of course, we believe that context matters when designing an omni-channel experience, with different usage patterns and expectations across different devices and channels, including offline ones.

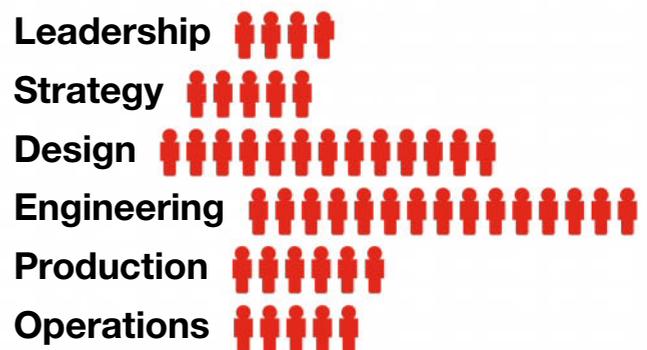
“We tend to adopt new technologies slowly as our goal is to be on the cutting edge of technology, not the bleeding edge. Plus, it is of the utmost importance to us to deliver consistency on time and on budget. Our engineering team meets twice a month to discuss new technologies and, as needed, new ones are adopted and integrated into our boilerplates and toolkits.”

And, of course, the rise of social media networks has meant designing for these channels must be an integral component of any development. George said: “While



we are not a marketing agency, we see search, social and syndication as a key part of our work. The work we do is semantic by nature and must be machine readable. If we are going to design the overall experience, it must extend beyond the walls of the domain and across the internet as a whole, even extending into offline environments with connected things and places. When launching a platform, we have an

## AGENCY BREAKDOWN



## PENTAGRAM

[pentagram.com](http://pentagram.com)

As the world's largest independently owned design firm, Pentagram needs little introduction. Working across a broad spectrum of clients, Pentagram design is woven into the fabric

of our culture and ubiquitous in our everyday lives.

Having enjoyed a long partnership with the firm, beginning with the website redesign for Adweek in 2011 and continuing through the years on projects that include Billboard Magazine, Library of America and Charlie Rose, we felt uniquely positioned to elevate the firm's website to meet their own high standards of design and performance.

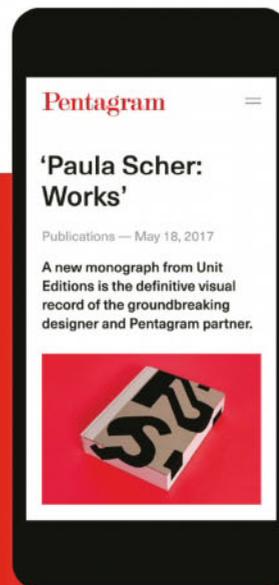
"As a consultant myself, I can't imagine a worse client than one with 20 equal owners, each of whom are opinionated and strong-minded. AREA 17 not only did a great job on our site, but structured a process



*Above* Pentagram's site also offers regular columnists who bring a depth of design understanding that is reflected in AREA 17's design for these components.

that led to interesting discussions, clear decisions, and a result that we can claim as the best expression of our thinking and work." — Michael Bierut.

Our approach focused on consolidating information into fewer entry points, providing more context and curation throughout and designing a browse and search experience that makes it easy to find and



discover relevant work from within the firm's expansive portfolio. Befitting the values of the firm, the work comes first and the visual design and information architecture aim to elevate the firm's output. Functionally-focused, elegant animations provide hints on where to go next — making both navigating the site and browsing the work effortless and enjoyable.

With an archive of nearly 1,000 projects across a range of disciplines, it was crucial that the CMS allows partners to easily customise each case study. A modular system with fixed and flowing underlying grids, individual projects can be optimised for the visuals that are best suited to it.

Visually-driven, all case study images open into a full-screen slideshow and for select projects, a secondary 'story' view splits the screen into independent narrative and image panels. Not only is the website a showcase for Pentagram's prolific output, but a hub for all of their rich contributions to the industry. Champions of design, the website includes a stream of regular columns, speaking engagements and podcasts.



*Above* The modular design of the CMS was critical to enable case studies to showcase the work of each client.

exhaustive checklist to ensure adherence to best practices for performance, search, social, and other types of syndication."

The breadth and depth of the work AREA 17 has created is thanks to the talented people that make up the studio. George outlined how valuable people are to AREA 17: "We are made of people and it's the success of our team as individuals that makes up the success of our agency. We look for talented people who have a consistent record showing commitment to their craft. But further, we look for curious, helpful, and committed people: people who love to explore, learn, and iterate; people who seek to collaborate with others, respecting the interdisciplinary nature of our work; people who like to help others and be helped; people who like to finish things 100 per cent. Ego has no place at our agency."

As a dynamic studio driven to not just create new innovative work, but craft whole new experiences, what does the future hold? "Over the years, we've kept ourselves small in order to maintain the highest level of quality output. And as a

small team, we invest a lot into the professional growth of each individual. Because of this commitment to quality and professional development, many who join AREA 17 stay around for a long time. In recent years, we've realised that we've become quite senior and because we like to keep project teams small, it has left little room for junior talent.

"In order to maintain our agency for the future, we must look to the role of apprentices within our team structure and that means growth. However, we believe that the world doesn't need another large digital agency and if we are going to grow, we must grow our soul.

"In 2017, we've looked internally to 'service design' ourselves and the programs we offer. This will act as a foundation to our growth, as we introduce multiple levels of talent within the agency, and ensure that it doesn't mean multiple levels of quality output.

"We are not looking to diversify, but rather become more focused as we grow. So that we can truly affect change within our ever-changing industry and be a

wonderful part of the digital community as we all build the future of the internet and how it becomes an extension of our daily lives, not a distraction from it."

The approach AREA 17 takes to each piece of work they create is akin to engineering. Understanding how form and design integrate to create experiences in the digital space, is a skill few studios master to the level AREA 17 have achieved. Code becomes not only functional in their hands of AREA 17, but also engaging and beautiful. ■



**area17.com**  
**Founder** George Eid  
**Year founded** 2003  
**Current employees** 48  
**Locations** Paris, New York  
**Services** Strategic planning  
 Visual identity design  
 User experience design  
 Web and mobile engineering  
 Analytics and optimisation



*The*  
**NEW KINGS** *of*  
**JAVASCRIPT**

MORE FRAMEWORKS THAN EVER ARE DEMANDING OUR ATTENTION. HERE WE TAKE A LOOK AT THE FIVE BIGGEST AND HOW TO GET THE BEST OUT OF THEM FOR YOUR NEXT PROJECT



```
import Vue from 'vue';
import Vuex from 'vuex';
```

```
Vue.use(Vuex);
```

```
const state = {
  count: 0,
};
```

The state is held as an object literal. Vuex updates this object as the data changes and all other logic will refer back to it.

```
const getters = {
  evenOrOdd: currentState => currentState.count % 2 === 0 ? 'even' : 'odd',
};
```

Getters can restructure the state to provide information for the different components. This is great when combining with ES2015 array methods, such as map or filter.

```
const mutations = {
  increment(currentState, amount = 1) {
    currentState.count += amount;
  },
};
```

Mutations are the only way to update the state. Vuex can identify the changes that have occurred within a mutation method and modify the state appropriately.

```
const actions = {
  increment: ({ commit }) => commit('increment'),
  incrementIfOdd: ({ commit, state }) => {
    if((state.count + 1) % 2 === 0) {
      commit('increment');
    }
  },
  incrementAsync: ({ commit, state }) => new Promise((resolve, reject) => {
    setTimeout(() => {
      commit('increment', 10);
      resolve();
    }, 1000);
  });
};
```

Actions are the best way of performing asynchronous mutations to the state. They can return promises that allow them to be dispatched in sequence.

## VUE.JS

# MANAGE STATE WITH VUE

## TAKE THE STRESS OUT OF MANAGING MULTIPLE DATA STORES

### BEST FOR

- Lightweight applications with a small footprint
- Beginners - all templates are valid HTML

As with any component-based library, managing state in Vue can be tricky. While the application is small, it's possible to keep things in sync by emitting events when values change. However, this can get brittle and prone to errors as the application grows, so it may be better to get a more centralised solution in from the start.

If you're familiar with Flux and Redux, Vuex works much the same. State is held in one centralised location and is linked to the main Vue application. Everything that happens within the application is reflected somewhere within that state. Components can select what information is relevant to them and be notified if it changes, much like if it was part of its internal state.

A Vuex store is made up of four things – the state, getters, mutations and actions.

The state is a single object that holds all the necessary data for the

entire application. The way this object gets structured depends on the project, but would typically hold at least one value for each view.

Getters work like computed properties do inside components. Their value is derived from the state and any parameters passed into it. They can be used to filter lists without having to duplicate that logic inside every component that uses that list.

The state cannot be edited directly. Any updates must be performed through mutation methods supplied inside the store. These are usually simple actions that perform one change at a time. Each mutation method receives the state as an argument, which is then updated with the values needed to change.

Mutations need to be synchronous in order for Vuex to understand what

has changed. For asynchronous logic – like a server call – actions can be used instead. Actions can return Promises, which lets Vuex know that the result will change in the future as well as enabling developers to chain actions together.

To perform a mutation, they have to be committed to the store by calling commit() and passing the name of the mutation method required. Actions need to be dispatched in a similar way with dispatch().

It's good practice to have actions commit mutations rather than commit them manually. That way, all updating logic is held together in the same place. Components can then dispatch the actions directly, so long as they are mapped using the mapActions() method supplied by Vuex.

To avoid overcomplicating things, the store can also be broken up into individual modules that look after their own slice of the state.

Each module can register its own state, getters, mutations and actions. State is combined between each module and grouped by their module name, in much the same way as combineReducers() works within Redux.pport.

**STATE IS HELD IN ONE CENTRALISED LOCATION AND IS LINKED TO THE MAIN VUE APPLICATION**

## LAZY LOAD ROUTES

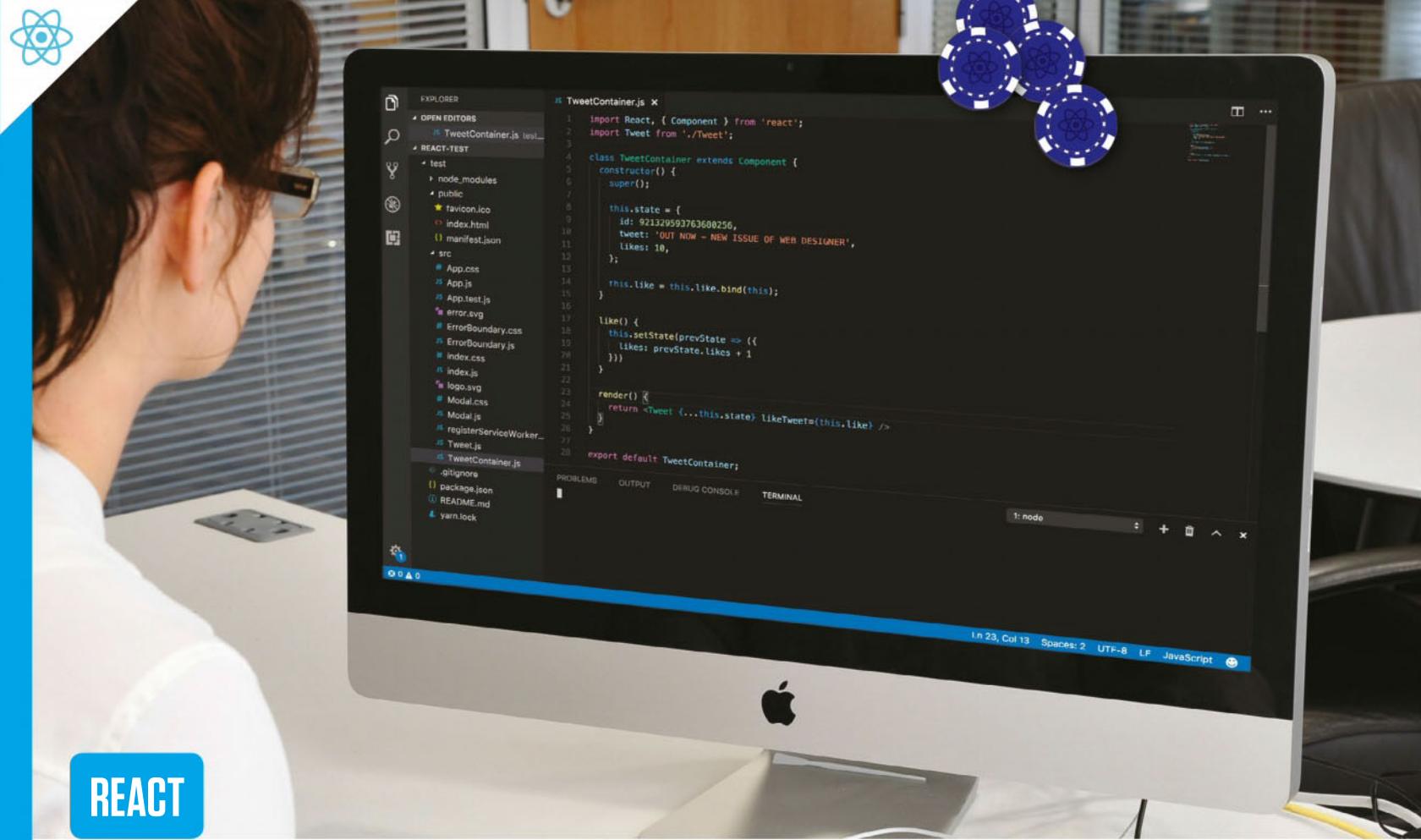
### SPEED UP THE FIRST LOAD TO IMPROVE PERFORMANCE

By default, the entire contents of the application end up inside one JavaScript file, which can result in a slow page load. A lot of that content is never used on the first screen the user visits. Instead it can be split off from the main bundle and loaded in as and when needed.

Vue makes this process incredibly simple to set up, as vue-router has built-in support for lazy loading.

```
const AsyncAbout = () =>
import('./About.vue');
const router = new
VueRouter({
  routes: [
    { path: '/about', component:
AsyncAbout }
  ] })
```

Vue supports using dynamic imports to define components. These return Promises, which resolve to the component itself. The router can then use that component to render the page like normal. These work alongside code splitting built in to Webpack, which makes it possible to use features like magic comments to define how components should be split.



# REACT

## BEST FOR

- ★ Sites and applications with complex view logic
- ★ Quick prototypes with a low barrier to entry



**REACT AND JSX ENCOURAGE THE CREATION ON MULTIPLE SMALL COMPONENTS TO KEEP THINGS AS SIMPLE AS POSSIBLE**

## USING CONTAINER AND PRESENTATIONAL COMPONENTS

### KEEP THINGS SEPARATE TO AVOID OVERLY COMPLEX CODE

As with any project, it's important to keep a separation of concerns.

All React applications start off simple. As they grow, it can be tempting to keep adding logic to the same few components. In theory, this simplifies things by reducing the amount of moving parts. When problems arise, however, these large components become prone to errors that are difficult to debug.

React and JSX encourage the creation on multiple small components to keep things as simple as possible. While breaking the interface down into smaller chunks can help with organisation, having a further separation between how a component works and what it looks like provides greater flexibility.

Container and presentational components are special names given to this separation. The container's job is to manage state and deal with interfacing with other parts of the application such as Redux, while the presentational component deals solely with providing the interface.

```
class TweetContainer extends Component {
  constructor() {
    super();
    this.state = {
```

```
id: 921329593763680256,
tweet: 'New issue of Web Designer',
likes: 10,
};
this.like = this.like.bind(this);
}
like() {
  this.setState(prevState => ({
    likes: prevState.likes + 1
  }));
}
render() {
  return <Tweet {...this.state} likeTweet={this.like} />
}
}
```

A container component will often be in charge of a small section of the UI, like a tweet. It will hold all the workings of that component – from storing state, like the number of likes, to the methods required for interaction, such as a mechanism for liking that tweet.

If the application makes use of external libraries, include at this point. For example, Redux's "connect" method would provide the container with a way of dispatching actions to the store without worrying the presentational component.

```
const Tweet = ({tweet, likes, likeTweet}) => (
  <div>
```

```
Tweet: { tweet }
Likes: { likes }
<button
  onClick={likeTweet}>Like</button>
</div>
);
```

Containers will never render their own UI and will instead render another component – the presentational component.

This component will be passed props that detail all the information needed to render the view. If it needs to provide interactivity, the container will then pass down methods for this as well, which can be called like any other method.

Having this separation encourages developers to keep things as simple as possible. If a container is starting to grow too large, it makes it easy to break off into a smaller set of components.

If the inner workings of a component, such as its state, needs to change, this technique allows the presentational component to remain unaffected. This also means this component can be used somewhere else in the application without needing to adjust how it functions. As long as it keeps getting served the same data it will continue to work.



# RENDER WITH PORTALS

## MOVE CONTENTS OUTSIDE OF THE ROOT COMPONENT

React 16 introduced the ability to return lots of different types of data from a component. While previously it had to be either a single component or "null", the latest version allows strings, numbers, arrays and a new concept called "portals".

The return value of a render() method decides what React displays, which is shown at that point in the component hierarchy. Portals allow React to render any of these return types outside of the component they were called from.

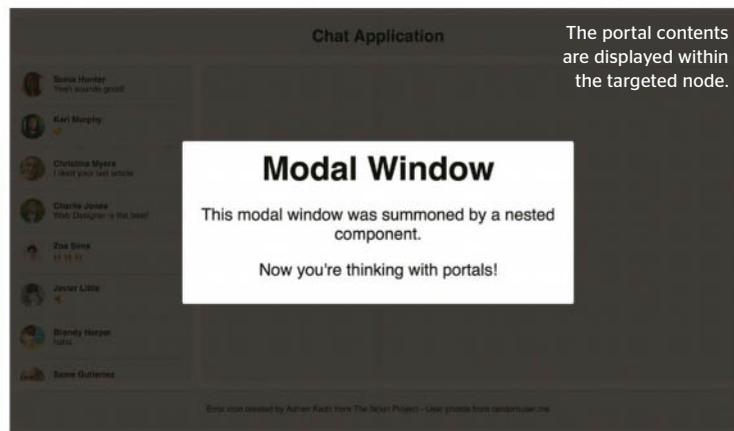
These can be other parts of the page completely separate from the main application. They still form part of React and work just the same as any component, but are able to reach outside of the normal confines of the root container.

A typical use case of this technique would be to trigger modal windows. To get correct positioning, overlay and accessibility requirements out

of a modal it ideally needs to sit as a direct descendant of the <body>. The problem is, the root of a single page application will likely take up that position. Components managing modals will either need to trigger something in the root component, or render it out of place.

```
class Modal extends Component {
  render() {
    return ReactDOM.createPortal(
      this.props.children,
      document.getElementById('modal'),
    );
  }
}
```

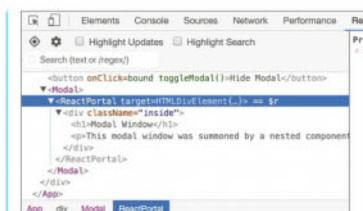
Here the Modal component returns a portal. The create function for it takes two arguments – what needs to be rendered and where it should render it. The second parameter is a regular DOM node reference, rather than anything specific to React. In this example, it references a <div> at the top of the DOM tree that is a sibling of the main app container. It is possible to target any node, visible or not, as



The portal contents are displayed within the targeted node.

with any JavaScript. To use it, another component can summon Modal just like any other component. It will then display its contents in the targeted node.

Because React events are synthetic, they are capable of bubbling up from the portal contents to the containing component, rather than the DOM node they are rendered in. In the modal example, this means that the summoning component can also handle its state, such as its visibility or contents.



Above React DevTools will still show a portal inside of its containing component.

# ERROR BOUNDARIES

## KEEP ERRORS FROM DESTABILISING AN APPLICATION

Unhandled errors can cause havoc in a JavaScript application. Without catching them as they happen, methods can stop executing half way. This can cause unpredictable behaviour if the user continues and is a bad experience all around.

Previous versions of React did not cope with these situations well. If an error occurred in a nested component, it would leave its parents in limbo. The component state object would be stuck in the middle of performing an operation that could end up locking up the interface.

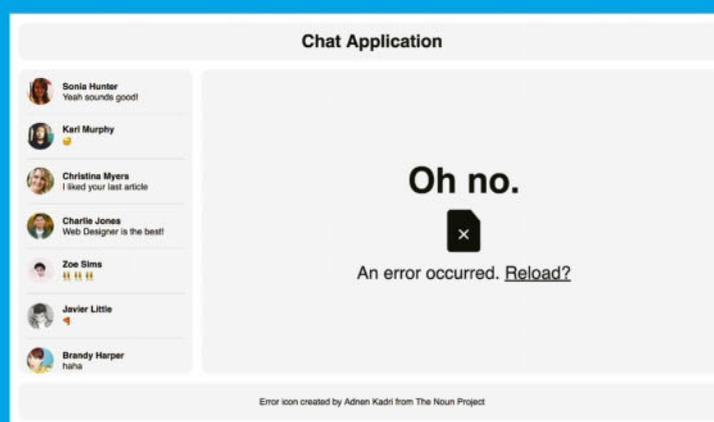
As of version 16, the way React handles errors has changed. Now an error inside any component would unmount the entire application. While that would stop issues arising with an unstable state, it doesn't lend itself well to a good user experience.

To avoid this, we can create a special component called an error

boundary to ring-fence parts of the application from the rest. Any errors that happen inside children of the boundary will not cause issues to those outside of it.

```
class ErrorBoundary extends Component {
  constructor(props) {
    super(props);
    this.state = { errored: false };
  }
  componentDidCatch(error, stackTrace) {
    this.setState({ errored: true });
  }
  render() {
    if (this.state.errored) {
      return (
        <h1>An error occurred</h1>
      );
    }
    return this.props.children;
  }
}
```

Error boundaries work a lot like typical catch blocks in JavaScript. When an error occurs somewhere inside the component tree, it will be



caught by the componentDidCatch() method, which receives the error thrown along with a stack trace. When that gets called it is an opportunity to replace the tree with a fresh interface – typically an error message.

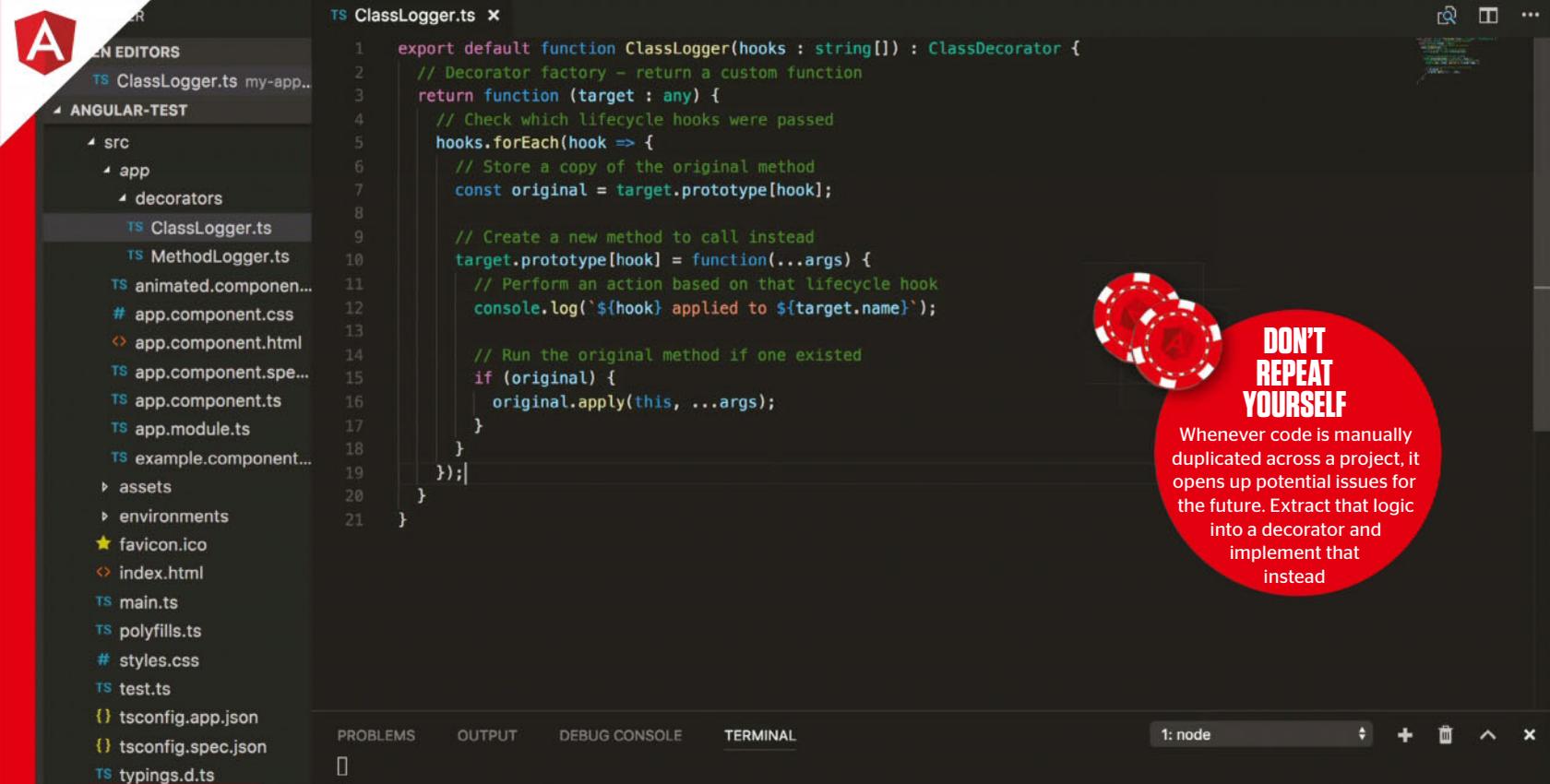
Since it only renders its children, this component can wrap others to catch any errors that happen within it.

The components chosen for this will vary by application, but error boundaries can be placed wherever they are needed, including inside other boundaries.

Error boundary components shouldn't be too complicated. If an error occurs inside of a boundary, it will bubble up to the next boundary up. Failing that, it will unmount the whole application as usual.

### REPORT ERRORS IN componentDidCatch()

Use the values provided to componentDidCatch() to report any errors to an external error logging platform to keep track of any issues



# ANGULAR

- BEST FOR**
- Large projects in need of structure
  - Applications with lots of changing data



**ANGULAR IS BUILT EXCLUSIVELY ON TOP OF TYPESCRIPT, SO IT IS IMPORTANT TO UNDERSTAND HOW TO UTILISE IT CORRECTLY**

## CREATE CUSTOM DECORATORS

### REUSABLE BLOCKS OF CODE TO DROP IN WHERE NEEDED

TypeScript is a superset that sits on top of JavaScript. It supplies features such as static typing, classes and interfaces that are lacking in the native language. This means that when creating large applications developers can get feedback on how best to work with external code and avoid unnecessary bugs.

Angular is built exclusively on top of TypeScript, so it is important to understand how to utilise it correctly. Combining the strengths of both provides a solid foundation for the application as it grows. There are not many better techniques to demonstrate this than with decorators.

```
function ClassLogger(hooks : string[]) : ClassDecorator {
  return function (target : any) {
    hooks.forEach(hook => {
      const original = target.prototype[hook];
      target.prototype[hook] =
        function(...args) {
          console.log(`${hook} - ${target.name}`);
          if (original) {
            original.apply(this, ...args);
          }
        };
    });
  };
}
```

Decorators are special functions designed to supply behaviour to whatever it is applied to. Angular makes extensive use of them to provide hints to the compiler, like with `@Component` on classes or `@Input` on properties.

The aim is to make these functions as reusable as possible and are often used to provide utility functions, such as logging. In the example above, `ClassLogger` is supplied to a component to log to the console when certain lifecycle hooks are fired. This could be applied to any component to track its behaviour.

The `ClassLogger` example above returns a function, which enables us to customise the behaviour of the decorator as it is created. This is known as the decorator factory pattern, which is used by Angular to create its own decorators.

```
@Component({
  selector: 'decorated-component',
  template: '<div></div>'
})
@ClassLogger(['ngOnInit'])
class DecoratedComponent {}
```

To apply a decorator, it needs to be positioned just before what it is decorating. Because of the way they are designed, decorators can be stacked on top of each other, including

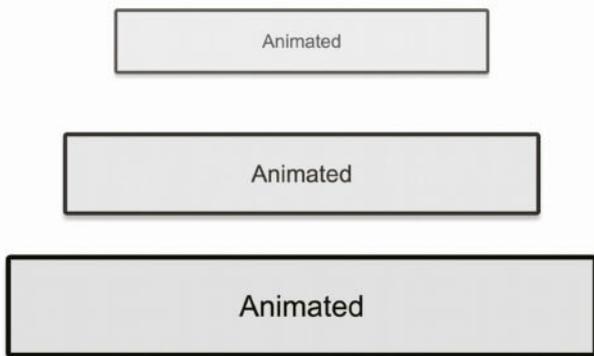
Angular's own. TypeScript will chain these decorators together and combine their behaviours.

Decorators are not just limited to classes. They can be applied to properties, methods and parameters inside of them as well. All of these follow similar patterns, but are slightly different in their implementations.

```
function MethodLogger(target, key, descriptor) {
  const method = descriptor.value;
  descriptor.value = function(...args: any[]) {
    console.log(`${key} was called with ${args}`);
    return method.apply(this, args);
  };
}

class ExampleComponent {
  @MethodLogger
  loggedMethod(a : any, b : any) {}
}
```

This is an example of a plain method decorator. These take three arguments - the object targeted, the name of the method, and the descriptor that provides details on its implementation. By hooking into the value of that descriptor we can replace the behaviour of the method based on the needs of the decorator.



## PLATFORM-LEVEL ANIMATIONS

### FIND OUT HOW TO CREATE PERFORMANT AND CONTROLLABLE ANIMATIONS WITH EASE

Animations are a great way to introduce a friendly side to an interface. But trying to control animations in JavaScript can be problematic. Adjusting dimensions like height is bad for performance, while toggling classes can quickly get confusing. The Web Animations API is a good approach, but working with it inside Angular can be tricky.

Angular provides a module that enables components to be animated by integrating with the properties already within the class. It uses a syntax similar to CSS-based animations, which gets passed in as component metadata.

```
@Component({
  [...]
  animations: [
    trigger('hidden', [
      state('true', style({
        transform: 'scale(0)'
      })),
    ],
    transition('true => false',
      animate('250ms ease-in')),
    transition('false => true',
      animate('250ms ease-out'))
  ]
})
```

Each animation is defined by a “trigger” — a grouping of states and transition effects. Each state is a string value that, when matched, applies the associated styles to the element. The transition values

define different ways the element should move between those states.

In this example, once the value bound to “hidden” evaluates to “true”, the element will shrink out of view.

Two other special states are also defined — “void” and “\*”. The “void” state relates to a component that was not in the view at the time and can be used to animate it in or out. The wildcard “\*” will match with any state and could be used to provide a dimming effect while any transition occurs.

```
template: '<div
  [@hidden]="hidden"
  (@hidden.
  done)="hasBeenHidden($event)"
  >Animated</div>',
```

Inside the template, the trigger is bound to a value within the component that represents the state. As that value changes, as does the state of the animation. That bound value can be supplied either as a plain property or as the output of a method, but the result needs to evaluate into a string that can be matched against an animation state.

These animations also provide callbacks such as when they start or stop. This can be useful for removing components that are no longer visible.

### USING WILDCARDS AS VALUES

Asterisks can also be used in places where a value is not known at the time of creating the animation, such as the width of inline elements

## SERVER RENDERING

### SERVE APPLICATION CONTENT TO USERS QUICKER

HTML parsers struggle with JavaScript frameworks. Web crawlers are often not sophisticated enough to understand how Angular works, so they only see a single, blank element and not the whole application.

By rendering the application on the server, it sends down an initial view for the users to look at while Angular and the rest of the functionality downloads in the background. Once the application arrives, it silently picks up from where the server left off.

The tools needed to achieve this in Angular are now a native part of the platform as of version 4. With a bit of set up, any application can be server rendered with just a few tweaks.

```
@NgModule({
  [...]
  imports: [
    BrowserModule,
  ],
  withServerTransition((
    appId: 'web-designer-example'
  )),
  ServerModule,
  AppModule
],
  bootstrap:
  [AppComponent]
})
```

Both server and browser builds need their own modules, but share a lot of common logic. Both need a special version of BrowserModule, which allows Angular to replace the contents on-screen when it loads in. The server also needs ServerModule to generate the appropriate HTML.

Servers also need their own entry points where they can bootstrap their unique behaviours as necessary. That behaviour depends on the app, but will also likely mirror much of the main browser entry point.

```
“apps”:
[
  [...]
  {
  [...]
  “outDir”: “dist-server”,
  “main”: “server.main.ts”,
  “platform”: “server”,
  }
]
```

If using the CLI, that also needs to be aware of how to build the project for the server by pointing to the new entry point. This can be triggered by using the “-app” flag when building for the server.

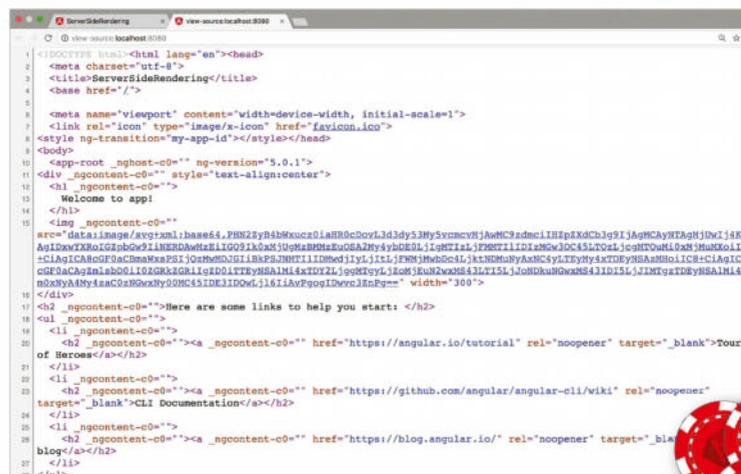
The application is now ready to be server rendered. Implementations will vary based on the server

technology used, but the base principles remain the same. For example, Angular provide an Express engine for Node, which can be used to populate the index page based on the request sent. All the server needs to do is serve

that file. Server rendering is a complex subject with many edge cases. For more information, get the documentation can be found at [angular.io/guide/universal](http://angular.io/guide/universal)

### CHECK THE ENVIRONMENT

Calling anything browser-specific on the server like the “window” object will throw an error. Check by using `isPlatformBrowser()` from `@angular/common`



## POLYMER

### BEST FOR

- Combining with other platforms and frameworks
- Working with JS standards

## WORKING WITH FORMS

### HOW TO TAKE THE PAIN OUT OF SUBMITTING VALUES

Custom elements are part of the browser. Once they are set up they work like any native element would do on the page. Most of the time, Polymer is just bridging the gap between now and what custom elements will be capable of in the future, along with bringing features like data binding.

One place where custom elements shine is their use as form inputs. Native input types in browsers are limited at best, but provide a reliable way of sending data. In cases where a suitable input isn't available – such as in an autocomplete field, for example – then custom elements can provide a suitable drop-in solution.

As their work is performed within

the shadow DOM, however, custom input values will not get submitted alongside regular form elements like usual. Browsers will just skip over them without looking at their contents.

```
<iron-form>
<form action="/submit">
<my-input></my-input>
</form>
</iron-form>
```

One way around this is to use an `<iron-form>` component, which is provided by the Polymer team. This component wraps around an existing form and will find any values either as a native input or custom element. Provided a component exposes a form value somewhere within the element, it will be detected and sent like usual.

### KEEPING THINGS ACCESSIBLE

Make sure custom form elements do not break conventions like tab order and labelling. Aim for them to function as much like native as possible

### Name Search

Do you accept?

SUBMIT

In cases where a custom element does not expose an input, it's still possible to use that element within a form provided that it exposes a property that can be bound to

```
<form action="/submit">
<my-input
value="{{formValue}}"></my-input>
<input type="hidden"
value="{{formValue}}"
name="formValue" />
<input type="submit" />
</form>
```

If `<my-input>` exposes a property like "value" to hook into we can pull that value out as part of a two-way binding. The value can then be read out into a separate hidden input as part of the main form. It can be transformed at this point into a string to make it suitable for form transmission. Forms not managed by Polymer that would need to make use of these bindings, the Polymer team also provide a `<dom-bind>` component to automatically bind these values.

## BUNDLING COMPONENTS

### KEEP REQUESTS LOW AND SIZES SMALL

One of Polymer's biggest advantages is that components can be imported and used without any need for a build process. As optimised as these imports may be, each component requires a fresh request, which slows things down. While HTTP/2 would speed things up in newer browsers, those who do not support it will have a severely degraded experience. For those users, files should be bundled together.

If a project is set up using the Polymer CLI, bundling is already built in to the project. By running "polymer build", the tool will collect all components throughout the project and inline any subcomponents they use. This cuts down on requests, removes unnecessary comments and

minifies to reduce the file size. It also has the added benefit of creating separate bundles for both ES5 and ES2015 to support all browsers.

```
polymer-bundler my-input.html >
bundled-component.html
```

Outside of Polymer CLI, applications can still be bundled using the separate Polymer Bundler library. This works much like the CLI, but is more of a manual process. By supplying a component, it will sift through the imports of the file, inline their contents, and output a bundled file.

Polymer Bundler has a few separate options to customise the output. For example, developers can choose to keep comments or only inline specific components.

## UPGRADING TO POLYMER 3.0

### STAY AHEAD WITH THE LATEST RELEASE

The philosophy behind Polymer is to "use the platform" – instead of fighting against browser features, work with them to make the experience better for everyone. HTML imports are a key part of Polymer 2, but are being removed from the web components specification moving forward.

Polymer 3.0 changes the way that components are written to work with more established standards. While no breaking changes are made with the framework itself, it's important to know how the syntax changes in this new version.

First thing to note is that Polymer is migrating away from Bower as a package manager. To keep up with the way developers work, npm will become the home

of Polymer, as well as any related components in the future.

To avoid using HTML imports, components are imported as JavaScript modules using the existing standardised syntax.

The major difference inside a component is that the class is now exported directly. This enables the module import `<script>` tag to work correctly. Any other components can be included by using ES2015 import statements within this file.

Finally, templates have been moved into the class and work with template literals. A project by the Polymer team called lit-html is working to provide the same flexibility as `<template>` tags along with the efficiency of selective DOM manipulation.

```
class RightNow {
  constructor() {
    this.update();
    setInterval(() => this.update(), 1000);
  }
}
```

This component holds a Date object for the current time, which is being updated every second. This is then being accessed from within the template.

```
update() {
  this.now = new Date();
}
```

```
.html x
<template>
  <require from="resources/value-converters/nice-time"></require>
  <div class="right-now">${now | niceTime}</div>
</template>
```

The template first needs to require the custom converter as part of the component. It then displays the time while piping it to the new converter.

```
.js x
import class NiceTimeValueConverter {
  view(date) {
    const options = {
      hour12: true,
      hour: "2-digit",
      minute: "2-digit"
    };
  }
};
```

The "toView" method in the converter receives the value each time it updates. The return value creates a localised, formatted string to display the time.



Most will be one way – from the model to the view. But they can also work the other way. The same logic applies, but by using "fromView" instead of "toView", values can be adjusted before they are returned back to the model. A good use-case for this would be to format user input directly from the bind on the element. In this example, it will capitalise every word that is entered, which may be useful for a naming field.

```
<li repeat.for="task of tasks | filter | sort">
```

They can also be chained together, which encourages the creation of composable converters that can have different uses across the application. One converter could filter an array of values, which then passes to another that sorts them.

```
<li repeat.for="task of tasks | filter:done">
```

Converters can also be given simple arguments that can alter the way they behave. Instead of creating different converters to perform similar filtering, create one that takes the type of filter to be performed as an argument. While only one argument is allowed, they can be chained together to achieve the same effect.

## AURELIA

### BEST FOR

- Simple applications with little setup
- Developing alongside web standards



## VALUE CONVERTERS

### LEARN TO CHANGE HOW PROPERTIES DISPLAY AND FUNCTION

Sometimes, when developing components, the values being stored do not lend themselves well to being displayed in a view. A Date object, for example, has an unhelpful value when converted to a string, which requires developers to make special conversion methods just to show values correctly.

To get around this problem, Aurelia provides a mechanism to use classes

to change values, known as value converters. These can take any kind of value, apply some kind of processing to it, and output that changed value in place of the original.

They work similar to pipes in Angular or filters in template languages like Twig.

```
<input type="text" value.
  bind="value | capitalise" />
```

## USING FRAMEWORK-LEVEL FORM VALIDATION

### LET AURELIA TAKE THE STRAIN OF CHECKING VALUES

Validation is an important part of any application. Users need to be putting the correct information into forms for everything to work correctly. If they do not, they should be warned of the fact as early as possible.

While validation can often be a tricky process, Aurelia has support for validating properties built right into the framework. As long as form values are bound to class properties, Aurelia can check that they are correct whenever it makes sense to the application.

```
import { inject } from 'aurelia-framework';
import {
  ValidationControllerFactory,
  ValidationRules } from 'aurelia-validation';
@inject(ValidationControllerFact
ory)
```

```
export class ValidationForm {
  constructor(controllerFactory) {
    this.controller =
    controllerFactory.
    createForCurrentScope();
  }
}
```

Aurelia provides a ValidationController, which takes instructions from the class, looks over the associated properties and supplies the template with any checks that have failed.

```
ValidationRules
  .ensure('firstName')
  .required()
  .withMessage('first name
  required')
  .on(ValidationForm);
```

Each controller requires a single ValidationRules class that defines what's to be checked. These are all

chained together, which enables the controller to logically flow through the checks dependant on the options that are passed. Each ruleset begins with a call to ensure(), which takes the name of the property being checked. Any commands that follow will apply to that property.

Next are the rules. There are plenty of built-in options like required() or email() that cover common scenarios. Anything else can use satisfies(), which takes a function that returns either a Boolean or a Promise that passes or fails the check.

After the rules come any

customisations of that check, for example the error message to display. Rules provide default messages, but these can be overridden if necessary. Finally, calling on() applies the ruleset to the class specified. If it is being defined from within the constructor of the class, it can be called with "this" instead.

```
this.controller.validateTrigger
= validateTrigger.manual;
By default, validation will be fired whenever a bound property's input element is blurred. This can be changed to happen either when the property changes, or it can be triggered manually.
```

**EACH RULESET BEGINS WITH A CALL TO ENSURE(), WHICH TAKES THE NAME OF THE PROPERTY BEING CHECKED**



# web workshop

# Create multiple interactive backgrounds

Inspired by [www.serpentiform.bulgari.com/en/home/](http://www.serpentiform.bulgari.com/en/home/)

## Main navigation

The navigation is found in the top right hand corner and this brings a panel over the right of the screen, the particles are still over this layer and connects the menu with the rest of the design.

## Highlighted navigation

Two main areas of navigation are located on the left and right of the screen, rolling over these brings a full screen dramatic overlay onto the screen.

## Swirling particles

Moving around the image are light swirling particles, like dust moving through a shaft of sunlight, these particles are continuously generating and have short lifespans.

## Parallax interactive movement

The main image of the serpent jewellery and various layers of blurred fragments, respond to the user's mouse movement at different speeds on the screen, giving a feeling of depth to the design.

## Content loading

A preload screen holds the user until the images have been loaded for the interface, then all of the elements slide onto the screen as they fade in.

## EXPERT ADVICE

### Focusing users attention

The movement of the snake with the fragments of jewellery instantly focuses the user's attention on the specific content of this site. The parallax movement of the layers is immediately attention grabbing, this ensures the user is left in no doubt what this site's theme is, with the purpose to explore custom jewellery.



**<comment>**  
What our experts think of the site

## Connecting the exhibition with visitors

The Serpenti web app provides exclusive contents about the exhibition path. Using cutting-edge and user-friendly technology, visitors will be able to design a personalized Serpenti creation and see it projected in the exhibition space. The web app will include a 360° tour of the exhibition.

**Bvgari, Roma**

## Technique

### 1. Creating the parallax effect

To get the same parallax layer effect as Bvgari, the parallax.js library will be used. The first step is to get the content into the page. A container panel will hold the parallax scene that will respond to when the user hovers over this.

```
<div id="container" class="container">
  <div id="scene" data-hover-only="true" data-
    relative-input="true" class="scene">
```

### 2. Adding the layers

Creating the layers that move around requires each image to be placed inside its own div tag. The data depth parameter states how close each element is to the front of the screen, therefore giving a 3D parallax effect.

```
<div id="block">
  <div data-depth="2.00" class="layer"></div>
  <div data-depth="1.20" class="layer"></div>
  <div data-depth="0.70" class="layer"></div>
</div>
</div>
</div>
```

### 3. Styling this to work

Creating CSS to style the elements is the next important part of the process. The first is just to ensure that the background is a black colour with the width and height set to fill the browser. The default browser margin and padding are removed.

```
html, body {
  width: 100%;
  height: 100%;
  margin: 0;
  padding: 0;
  background: #000;
}
```

### 4. Filling the display

To get the same effect as Bvgari, both the container and scene are told to be the full width and height of the scene. The container's overflow is set to be hidden to prevent scrollbars if the parallax content moves off screen in either direction.

```
.container {
  width: 100%;
  height: 100%;
  overflow: hidden;
```

```
}
scene {
  width: 100%;
  height: 100%;
  position: relative;
}
```

### 5. Centre the parallax

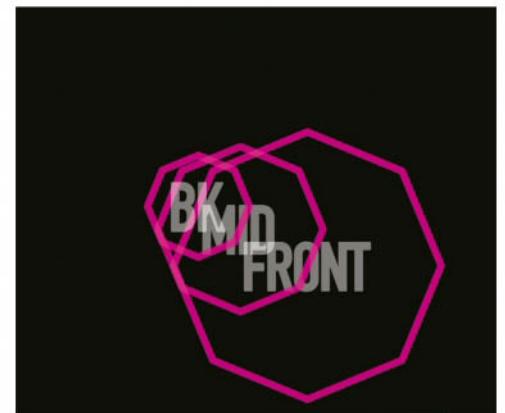
The parallax section is then centred onto the display for the horizontal and vertical axis to make the layers start from the centre of the screen space. This is achieved by centring the top left corner and then taking half of the width and height of the block off this position to place it in the centre.

```
#block {
  width: 600px;
  height: 600px;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```

### 6. Link the library

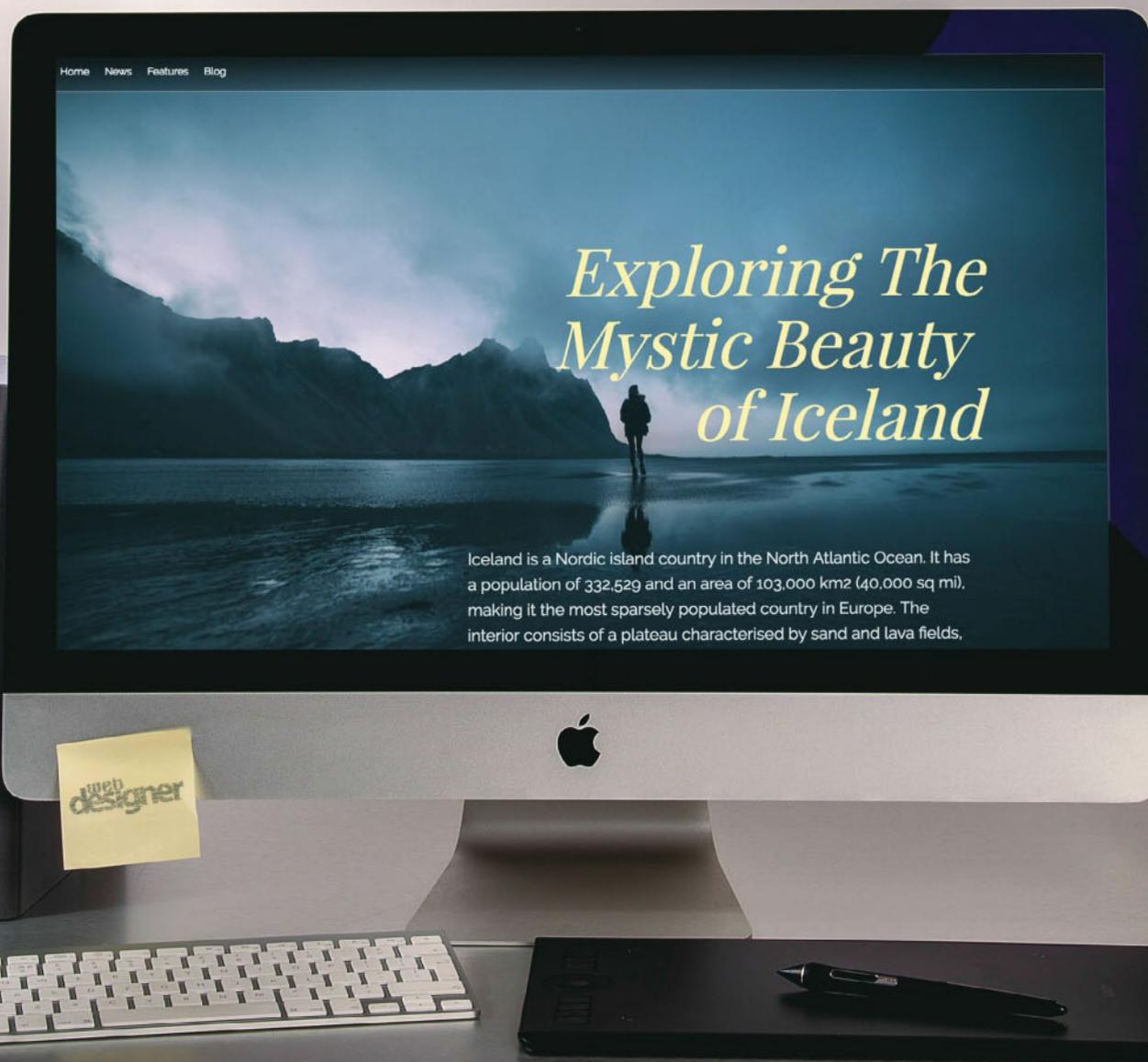
The final step is to link the library from a content delivery network (CDN) and then create the parallax scene with a minimum of code. Saving the page to finish and viewing it in the browser will show three layers of interactive movement to the mouse position.

```
<script src="https://cdnjs.cloudflare.com/
  ajax/libs/parallax/3.1.0/parallax.min.js"></
  script>
<script>
  var scene = document.getElementById('scene');
  var parallax = new Parallax(scene, {
    selector: '.layer'
  });
</script>
```



# Build complex layouts with CSS

Employ the power of CSS Grid Layout to create eye-catching magazine-style page designs



Layout for the web has always been limited, there really has never been anything truly dedicated to making content easy to display in a format that makes sense.

Back in the '90s when the web first took hold, designers who were used to laying out content on a grid, found the only way to get the same result in the browser was to use the table. By merging cells, complex layouts could be achieved. Sadly, this played havoc with accessibility and SEO. Fast forward and today there is all manner of ways to place content with floats, flexible boxes and so on.

CSS3 has really pushed the capabilities of the web forward and since 2014 there has been CSS Grid Layout. This currently has 75 per cent browser support, so it's getting to the point when it's time to give it serious consideration. Zurb's Foundation 6 framework has got on board and is using that to power its grid.

This tutorial will start by giving a brief introduction to the subject with three simple layouts that give the basics of how it works. The remainder of the tutorial will be spent creating a magazine-style feature that will show you how to make it responsive for medium and small screen sizes. The tutorial has already completed the HTML and most of the design in CSS for you, so here our focus will be entirely on the grid elements.

## 1. A simple primer

Open the file 'grid1.html' from the start folder in the project files. The HTML layout for this has five div tags, simply named item1-5 as CSS classes. These will be the content placed into the grid. The container wrapping this will define the grid, which will have four columns.

```
<div class="container">
  <div class="item1">item 1</div>
  <div class="item2">item 2</div>
  <div class="item3">item 3</div>
  <div class="item4">item 4</div>
  <div class="item5">item 5</div>
</div>
```

## 2. The CSS for the grid

Looking in the head section you can see that the 'container' has been told to be laid out as a grid, with auto height for the rows, while the columns are to be set to four with each set to 25 per cent of the browser. Looking at this in the browser you will see that each item is automatically assigned the next available grid position.

```
.container {
  display: grid;
  grid-template-rows: auto;
  grid-template-columns: repeat(4, 25%); }
```

## 3. Define grid positions

Now take a look at the 'grid2.html' file. It's the same as the first file, except here 'item1' and 'item2' are given specific positions. The first is positioned in row 1 and ends before row 2. The column starts at 1 and ends at 3 so it spans two columns. The second starts at column 3 and takes the next two columns. The remaining items fill the next available grid slots.

```
.item1 {
  grid-row-start: 1;
  grid-row-end: 2;
  grid-column-start: 1;
  grid-column-end: 3;
}
.item2 {
  grid-row-start: 1;
  grid-row-end: 2;
  grid-column-start: 3;
  grid-column-end: 5;
}
}
```

## 4. Use a grid template

Open 'grid3.html' and look at the body of the HTML. You'll see that there is a layout with a header, sidebar, main content section and a footer. You can add more text into the content to see what happens when this is placed in. The grid will use a template feature to make these sections into a layout.

```
<div class="container">
  <div class="header">Header</div>
  <div class="sidebar">Sidebar</div>
  <div class="content">Content</div>
  <div class="footer">Footer</div>
</div>
```

## 5. Define the template

Look at the CSS for the container. It is again defined as a grid. The top row will be 200px high, the middle will be auto-sized and the last row will be 100px high. The columns are set to be 33 per cent wide and to autofill the rest. The template states the header will fill both columns. The next row will be the sidebar in the first column and the content in the next. The footer goes across both.

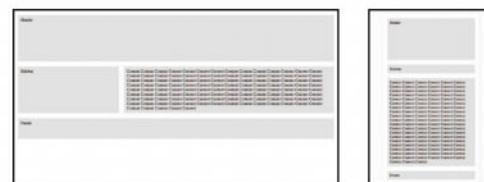
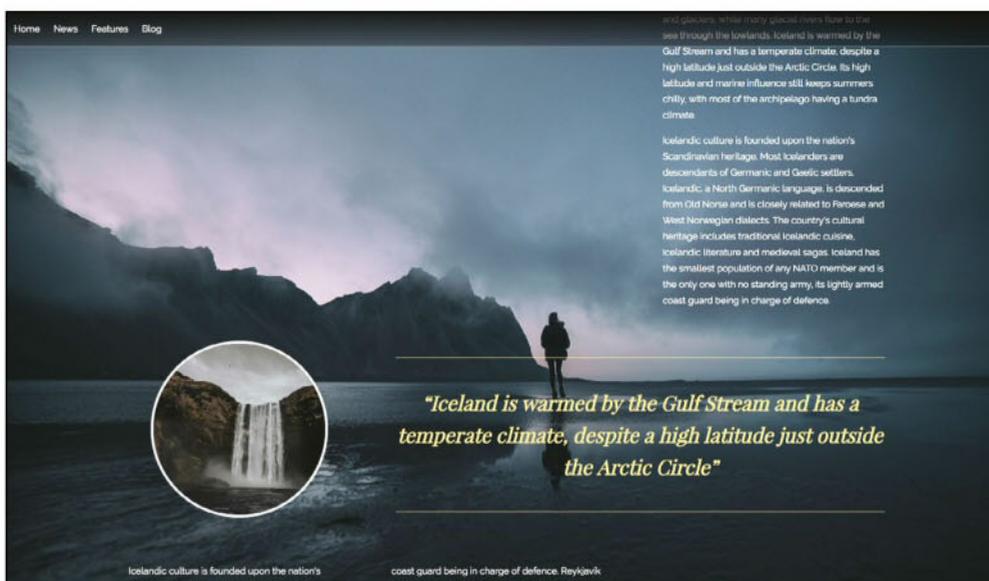
```
.container {
  display: grid;
  grid-template-rows: 200px auto 100px;
  grid-template-columns: 33% auto;
  grid-template-areas:
    "header header"
    "sidebar content"
    "footer footer"; }
```

## 6. Link the template to the class

To link up the class to the template, the code shown here defines this. Each grid area is named and the link is created. The content isn't shown here for space reasons, but it's in the 'grid3.html' document. Look at this in the browser to see the layout of the grid. Because two columns are defined the template needs two areas stating in each of the inverted commas.

## Grid Template Area names & classes

Sadly the Grid Template Area names do not map directly onto classes and the various classes have to be assigned to which grid area you want them to be associated with.



### Top left

A simple webpage layout is created using the grid-template-areas to define that the header and footer span two columns, while the sidebar and webpage content take up one column each.

### Top right

By simply redefining the grid layout in a media query all the elements can reflow with just a few adjustments.

### Left

When the first grid is finished, the design fits just right on a desktop display, even fitting in well with background image elements - as can be seen here.

# Tutorials

## Build complex layouts with CSS

```
.header {  
  grid-area: header;  
}  
.sidebar {  
  grid-area: sidebar;  
}  
.footer {  
  grid-area: footer; }  
}
```

### 7. Making it responsive

To make 'grid3.html' responsive a media query is inserted and the top row is kept at 200 per cent, while the remaining rows will automatically be sized. There is only one column, the full width, so the template has one word in each inverted comma to define the layout. These can easily be reordered without shifting any of the HTML.

```
@media screen and (max-width: 699px) {  
  .container {  
    display: grid;  
    grid-template-rows: 200px auto;  
    grid-template-columns: 100%;  
    grid-template-areas: "header" "sidebar"  
"content" "footer";  
  }  
}
```

### 8. Working on a real layout

Now open 'index.html' - all of the HTML for the content has already been created, as has some of the CSS for the design elements. Add this grid to the style tags in the head section. Doing so creates a three-column grid with the template for each section. Note the full-stop for the empty grid sections.

## Article2's column count

You may have noticed that 'article2' has a column count attribute - this is the CSS3 Multi-column Layout for text and has nothing to do with the columns in the grid!

```
.container1 {  
  width: 80%;  
  margin: 0 auto;  
  display: grid;  
  grid-template-rows: auto;  
  grid-template-columns: 33.3% 33.3% auto;  
  grid-template-areas: "header header  
header" ". standfirst standfirst". .  
article1" "focus pullquote pullquote"  
"article2 article2 .";  
}
```

### 9. Linking up the template

As with the previous step, this links the header with the template. The header is told to span all three columns of the grid, then the standfirst is set to take two columns and have an empty column on the left. If you check the browser, that column is filled because the remaining content autofills the next available space - it won't do this when it's all set up, however.

```
.header {  
  grid-area: header;  
}  
.standfirst {  
  grid-area: standfirst;  
}
```

### 10. Adding the next areas

Now the first article, the pullquote and the focus image are placed into the design. The pullquote and image are side by side on the same row. At this stage article2 hasn't been placed so it's taking the first available space on the grid which is next to the standfirst.

```
.article1 {  
  grid-area: article1;  
}  
.pullquote {  
  grid-area: pullquote;  
}  
.focus {  
  grid-area: focus;  
  text-align: center; }  
}
```

### 11. Finishing the first grid

Adding the second article's CSS enables all of the first grid to be placed correctly. Looking at this in the browser will show that the layout works with the background image and creates the kind of layout seen in magazines, when the designer works around a large background image.

```
.article2 {  
  grid-area: article2;  
  column-gap: 65px;  
  column-count: 2;  
}
```

### 12. A white background

Before starting the second grid, you may wonder why there is a need for two grids. The reason being is that this grid is going to have a white full width background so this CSS will wrap the second grid. This is to give this section, the feeling of a second page in the design.

```
.whitebg {  
  padding: 100px 0;  
  background-color: #ddd;  
}
```

### 13. Second grid

The second grid is simpler than the first. There are three columns, with an automatic height on the rows. The content is going to fill a column each so there is no need to define the template areas. However when the tablet design is added, this needs to switch to two columns, so a reflow is required and the names will be important.

```
.container2 {  
  width: 80%;  
  margin: 0 auto;  
  display: grid;  
  grid-template-rows: auto;  
  grid-template-columns: 33.3% 33.3%  
auto;  
  grid-template-areas: "img1 img2  
article3";  
}
```



#### Right

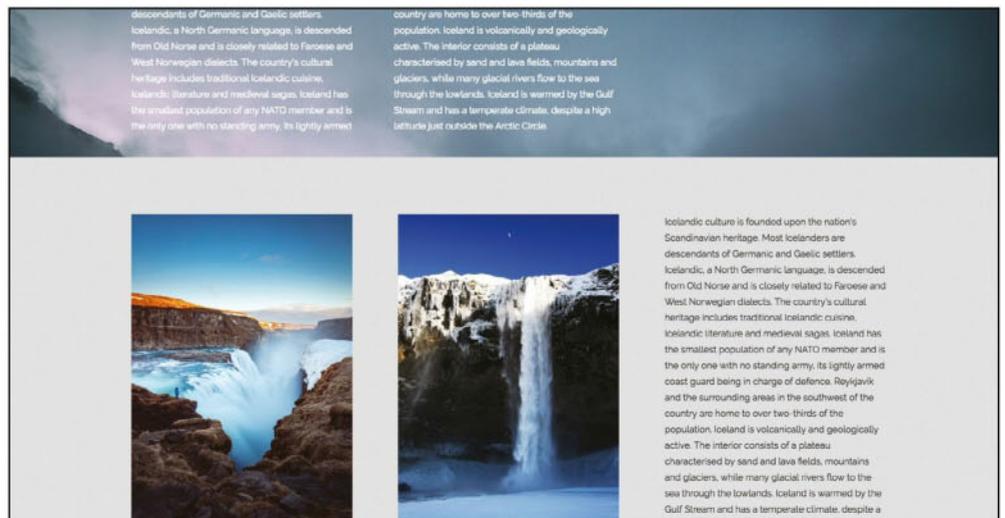
With both grids placed on the screen, the design can be scrolled through as everything is in the correct order.

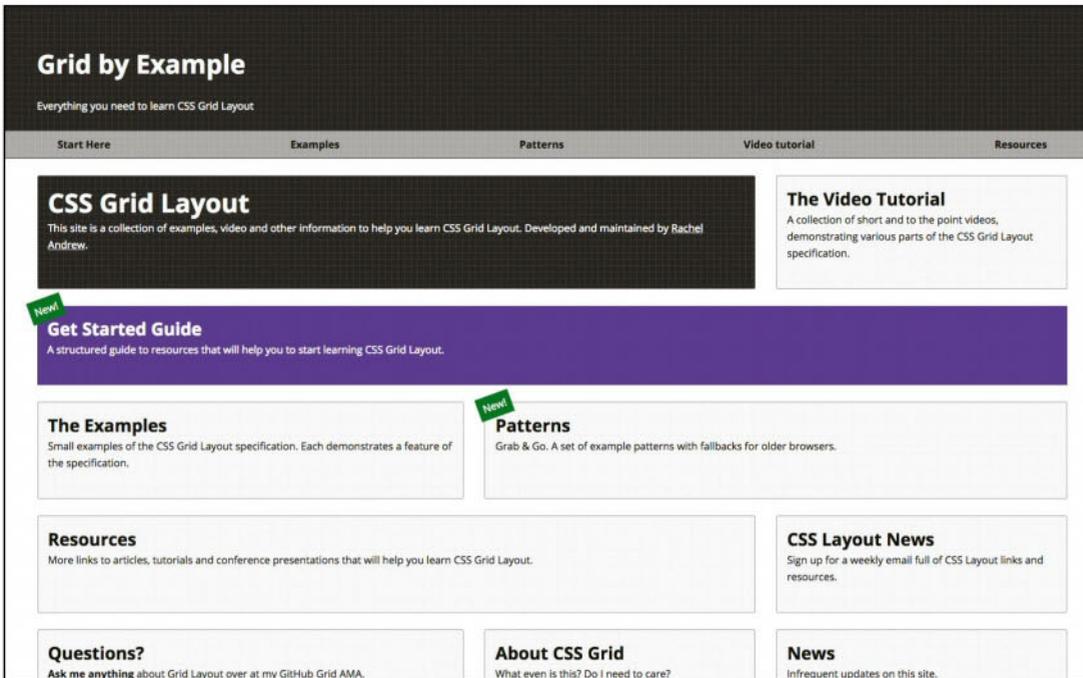
#### Top left

The tablet-sized design is in place with just a reflow of both of the grids on the page. Adjusting these grids is one of the easiest ways to redesign for smaller screens.

#### Top right

Just telling the grids to have one column is virtually all that needs to be changed for the smaller mobile-sized screens.





### Going further with CSS Grids

The tutorial here has only been able to show a small amount of what is possible through CSS Grids. The Mozilla Developer Network has really great documentation that will help you ([developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout)), however it's not always found in a step-by-step guide. Rachel Andrew, who has been a long-time advocator of the CSS Grid Layout has some great tech talks on YouTube, one ([youtube.com/watch?v=N5L1SLqBmQ](https://www.youtube.com/watch?v=N5L1SLqBmQ)), will give you a good overview of the subject area. Rachel has also put together a great website resource ([gridbyexample.com/](https://gridbyexample.com/)) called Grid by Example and this is a very practical approach to dealing with the layout.

## 14. Adding the columns

Each of the CSS classes in the second grid is told to link up with the relevant column, as defined in the grid template. The article text colour is changed just to make it stand out against the lighter background of this section. With only the footer to complete, the design of the magazine-style layout is almost in place.

```
.img1 {
  grid-area: img1;
}
.img2 {
  grid-area: img2;
}
.article3 {
  grid-area: article3;
  color: #333;
}
```

## 15. The foot of the page

Finishing off the page will place a full width div across the screen, which will be filled with a black colour and the text is just centred. This completes the desktop version of the design, but moving the screen down below 1200px wide and the site starts to break.

```
.footer {
  background-color: #000;
  color: #999;
  text-align: center;
  padding: 50px 20px 100px;
}
```

## 16. Medium screen design

A media query is inserted here to look after the design when the width of the browser is less than 1200px. The code for Steps 17 and 18 will be placed inside the brackets where the comment is. This will be a case of changing both of the grids' layout structure.

```
@media screen and (max-width: 1200px) {
  /* code here */
}
```

## 17. Reflowing the first grid

The first grid is set to now fill the full width of the browser with just two columns instead of three. The order of the sections is placed into the template, with the articles switching sides, since this fits in better with the background image at this screen size.

```
.container1 {
  width: 100%;
  grid-template-rows: auto;
  grid-template-columns: 50% 50%;
  grid-template-areas: "header header"
    "standfirst standfirst" "article1 ."
    "pullquote pullquote" "article2 focus";
}
.article2 {
  column-count: 1;
}
```

## 18. Another two columns

The second grid is also resized to take the full width of the browser and two columns added. The images are placed side by side on the row above the text so that it fits neatly onto the display. You can test this layout in the browser by resizing your browser below 1200px width.

```
.container2 {
  width: 100%;
  margin: 0 auto;
  grid-template-rows: auto;
  grid-template-columns: 50% 50%;
  grid-template-areas: "img1 img2" "article3
  article3";
}
```

## 19. Designing for mobile screens

Any browser that has a width of less than 769px will get the code that is added in the final steps. All we need to do here is ensure that each of the grids has a single column layout so that the content can be viewed properly within the smaller space.

```
@media screen and (max-width: 768px) {
  /* final steps code here */
}
```

## 20. Single columns grid

Now the first grid is set to a single column of 100 per cent of the browser's width and the order of the sections is added in the template areas. Check to see how the first part of the page is working on mobile-sized screens.

```
.container1 {
  width: 100%;
  grid-template-rows: auto;
  grid-template-columns: 100%;
  grid-template-areas: "header" "standfirst"
    "article1" "pullquote" "focus" "article2";
}
```

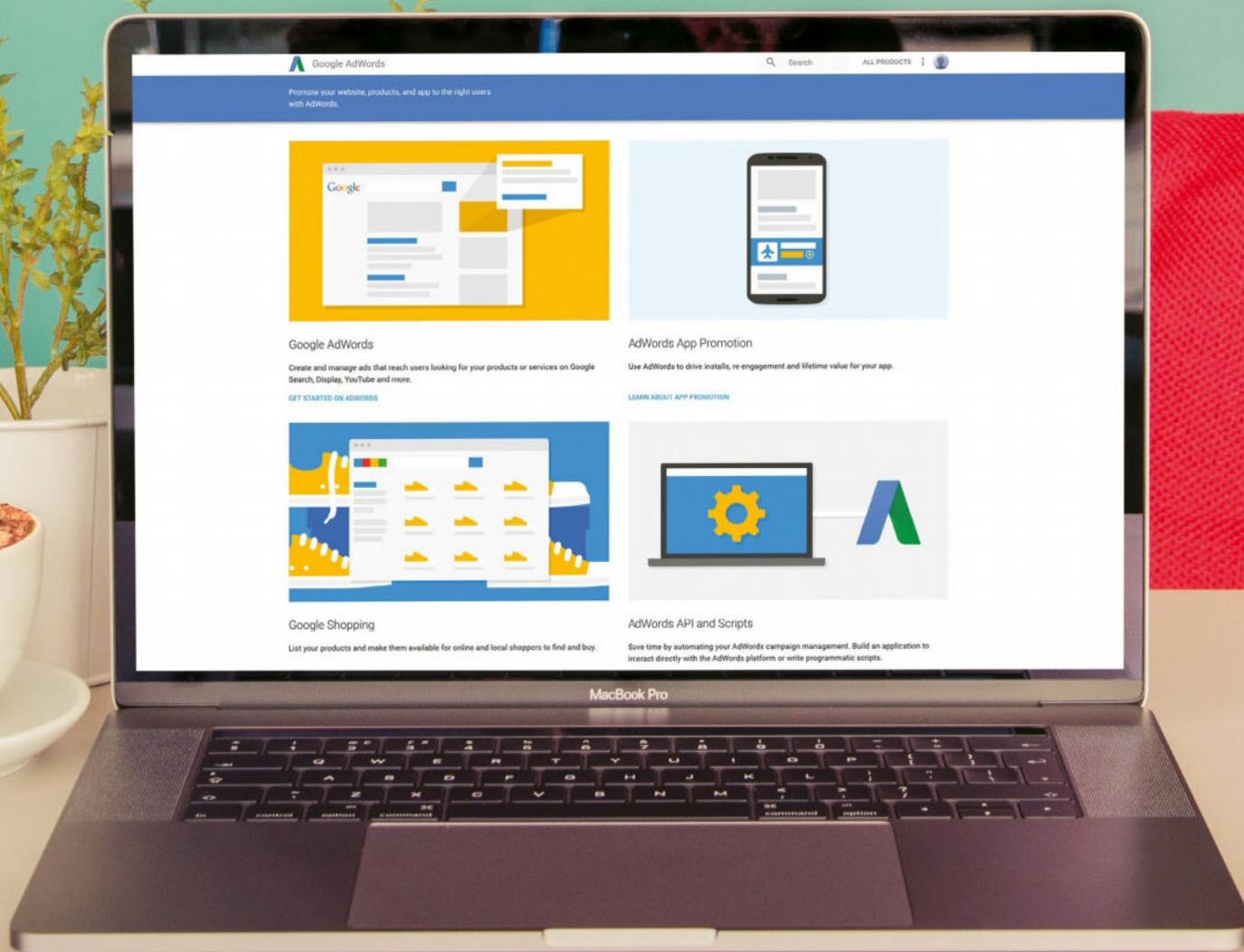
## 21. Finishing the layout

Here, the second grid is also made to fill a single column and the layout of sections is defined. Now save the finished design and view it across different-sized screens to see the full layout capability of the CSS Grid and how easy it was to just reorder the content for different widths.

```
.container2 {
  width: 100%;
  margin: 0 auto;
  grid-template-rows: auto;
  grid-template-columns: 100%;
  grid-template-areas: "img1" "img2"
    "article3";
}
```

# Optimise content for a better ranking

Discover the ten steps that will help you understand what's required to influence Google rankings within your content

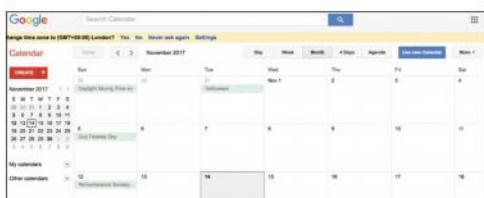


In today's content-driven world of digital marketing, it is a marketer's goal to persuade a content-saturated audience to consume their particular content, share that content with their network, or make a purchase.

By the time you have finished reading this tutorial you will have a better understanding of how you can ultimately improve your rankings amongst this competition, boost your visibility and capitalise on the right audience through your content.

Ultimately your content is going nowhere unless people can search for and find it, so content should be written with an understanding of factors such as SEO and aimed at the right audience groups. The steps below will highlight how to drive more traffic to your site, building a connection between your brand and your audience.

You may be in a crowded field, but you have a lot of opportunity to expand and make the space your own.



### 1. Get organised

Creating an editorial calendar enables you to prioritise your content tasks, getting into a cadence with publishing blog posts and social pushes. More importantly, it keeps your audience engaged by preventing your content from stagnating or being repetitive.

Organisation is absolutely crucial for success, although it won't always be easy to stick to! Start by setting up a calendar, then brainstorm ideas (what questions can you answer, what themes are relevant, how can you inspire). Breakdown your content into social posts, blog posts, onsite copy, referrals, video... and so on. And layer on top of this your promotion, since content rarely succeeds on its own, and plan your frequency (daily, weekly, monthly).



### 2. Buyer personas – understand who your audience is

The focus of your content should be aimed at the audience group that contributes to the success of your business. Content will therefore be different based upon where the user is along their journey, a first-time visitor for example has no loyalty to the brand; hence the focus should be on inspirational content to convert – content and metas – and brand values to nurture.

Whereas existing clients have a different set of priorities and needs, therefore content needs to be served differently. Nurturing them, encouraging advocacy and facilitating the ability to upsell are the keys in driving both customer satisfaction and further sales.



### 3. Learn from the competition

By reviewing the top performing organic content around your targeted keywords, you can take advantage of what

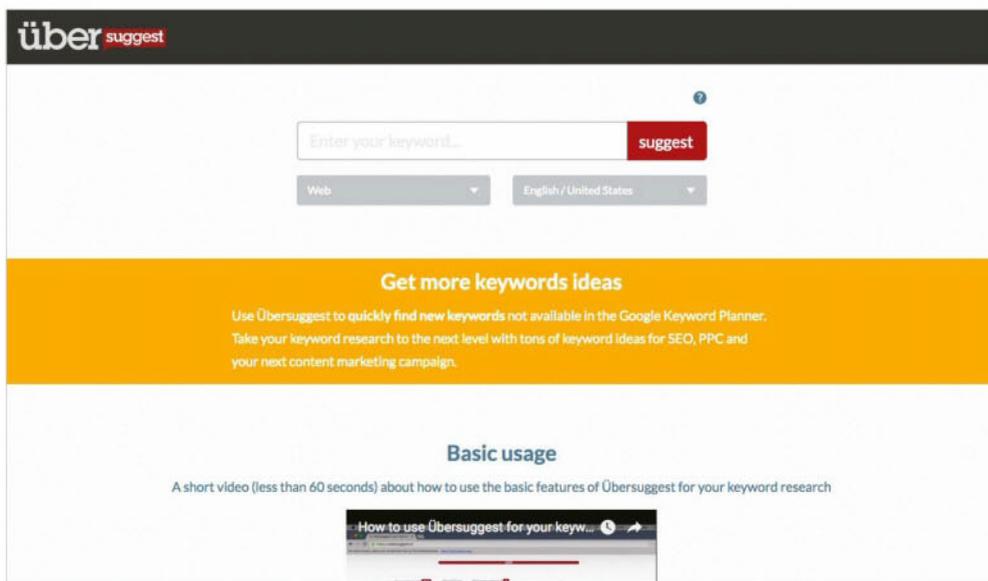
works and then replicate it. Understanding the competition's focus on content and keyword terminology will also help you to close the gap between those terms you are not taking advantage of and ultimately increase the amount of traffic coming to your own domain.

### 4. Make sure you follow the 70%/20%/10% content rule

70 per cent of your content should be specific to your business vertical and services (Hygiene), because your audience needs that information to convert. It's your day-to-day, always-on messaging; include industry news, reviews and offers.

20 per cent of your content should be inspirational (Hub), in order for your visitors to share it. It aims to develop a shared passion between you and your

**Create question and answer FAQs**  
Some customers are primed for inspiration and some need support, think about your customers' needs and gain more visits from long-tail keyword variants in your content.



**Above** Check out Thesaurus.com to see which of your keywords can be substituted with synonyms.

**Left** Take your keyword research to the next level with a host of keyword ideas for SEO, PPC and content marketing with Übersuggest (ubersuggest.io).

# Tutorials

## Optimise content for a better ranking

audience, keeping them coming back for more. You can do that by creating unique content that exceeds your visitors' expectations – think of inspiring solutions, solving problems and answering the 'What' 'Why' and 'How' of your business, and how it can best serve your visitors.

Ten per cent of your content should be risky enough that it scares you a little (Hero), although it can attract enormous numbers of visitors, and in which you invest a lot of time and effort. This may be content related to the most important event or product launch of the year.

### 5. The content and SEO overlap

An easy way to ensure the success of your content marketing efforts and improve the search ranking of a particular post is to optimise the on-page elements by applying key SEO techniques.

None of your content is going anywhere unless people can search for it and find it, therefore content should be written with an understanding of SEO goals. This includes a healthy solid foundation, content on the website targeting the right keywords and relevancy within the metas, such as: title tags, headers, meta descriptions, image alts and a good URL structure.

With these working together you will be able to rank for the right keywords, with visitors that actively convert since the success of each depends on the other.



### Upscale your existing content

Upscale previously indexed and popular pages such as previous trends, reviews and events. By upscaling the content, search engines will look favourably on its page placement, while also re-engaging your audience and increasing dwell time.

### Code Sample

```
<a href="http://www.same-domain.com/" title="Keyword Text">Keyword Text</a>
```

### 6. Internal links and backlinks

Internal links help search engines identify the site's most important content, provide context and help your audience by directing them to the most relevant pages. Make the most of anchor text within your links, with variations of your target keywords to boost rankings.

The number of backlinks (incoming links, directed at your website) is another positive indicator to search, in particular websites that carry similar content (topical relevance). Start link building both internally and externally with your best content to improve ranking.

### 7. The impact of a blog

Incorporating a blog into a website can have a huge impact on the overall website's search engine rankings. Not only does it add keyword-rich pages, but it also increases the potential for incoming links from high

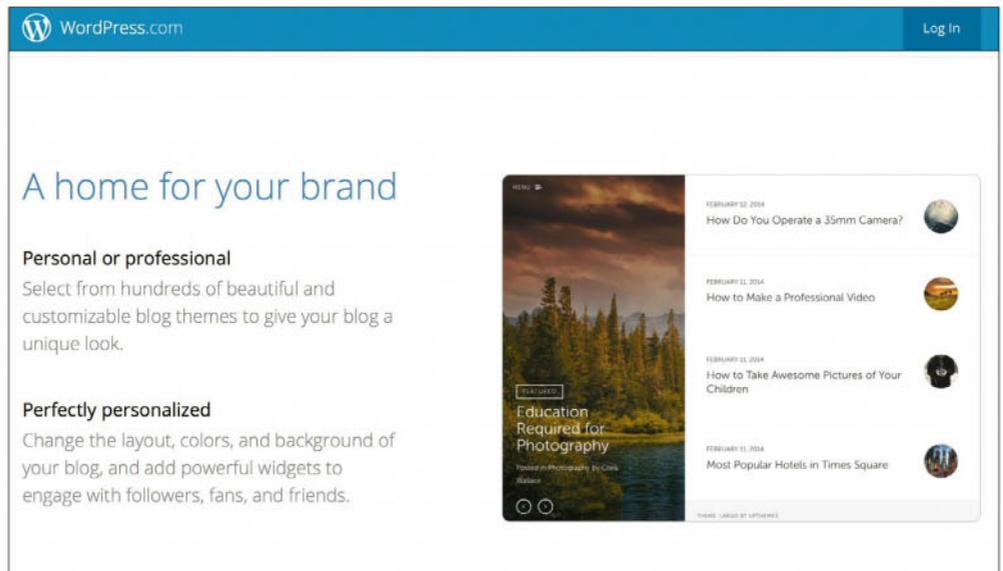
quality websites, as well as advocacy from your audience as they share that content with their networks.

Aim for original and high quality content, since this will go a long way towards satisfying both the needs of your audience and the search engines.

And remember, given the number of distractions that your audience faces daily, strong headlines are key to encouraging their inspiration to click within a crowded marketplace. You can achieve this by understanding your audience's pain points – solve their problems in a way that seems tailored to them, and give them both actionable insights and inspiration.

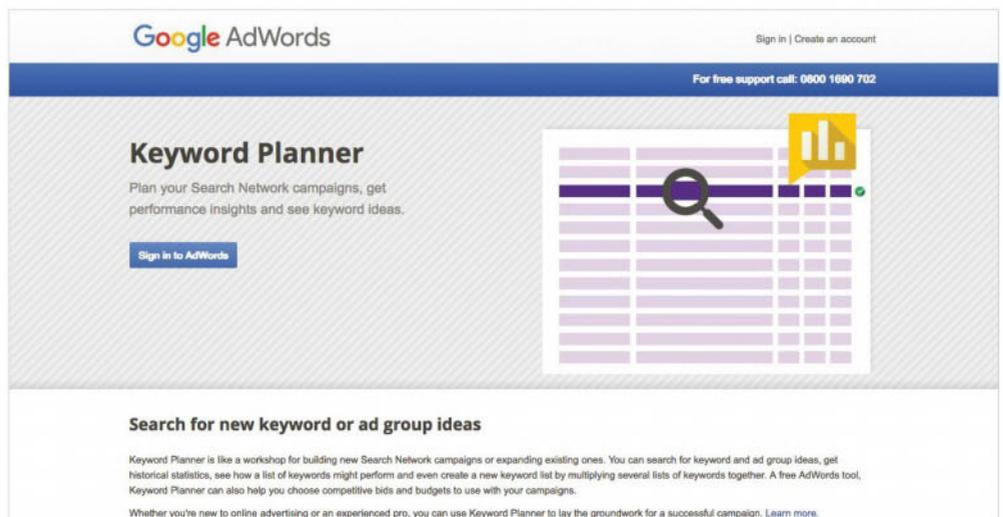
### 8. Remember visual content

Visual content – such as images, video and infographics – can be a very powerful tool for any brand looking to communicate more effectively with its readers, especially



**Above**  
Your brand should also have a selection of social media channels. Make sure that you use them to reinforce your brand message – it really helps.

**Right**  
To establish a baseline list of keywords you can use, employ the help of Google AdWords Keyword Planner.





**Fit content around your keywords**

Keywords in your web content make it possible for people to find your site via search engines; they define what your content is about. Ranking for the right keywords can make or break your website, by researching your market's keyword demand you can learn which terms and phrases to target within your content.

Start by brainstorming your ideal customers search terms and establish a baseline list of keywords with the AdWords Keyword Tool. Go further by substituting in synonyms with Thesaurus.com and gain new keyword ideas from Wikipedia articles around your keyword list. Take advantage of UberSuggest to generate variations, and spend some time on Quora, since it will highlight how people speak in their niche communities.

since it can have a very positive impact on your brand's reach, engagement, and sales.

Start by splitting up body text with compelling imagery (include image Alts) to encourage your audience to finish reading and boost your search engine ranking. Infographics also help to collate content into compelling and easily intelligible visual displays.

Your site visitors will also expect you to offer video content, show your personality as you connect with them with information that meets their needs.

How-to videos, demonstrations and customer testimonials are all opportunities to consider.



**9. Create a promotion plan**

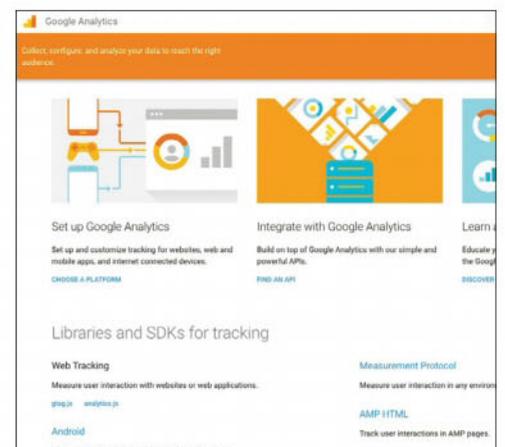
Place a promotion plan behind your best content to improve audience engagement. This includes:

- Paid traffic that offers immediate results: Focus on the content synergy between advertising text, landing page copy and SEO metas. This will improve placement, by improving the quality score.
- Social media that offers great brand awareness: Review your brand guidelines, tone of voice, targeted demographic reach and audience needs and wants.
- Influencers who can share your content and increase your outreach to engaged communities: Understand where your audience is and let them know you exist.

**10. Measuring the success of your content is crucial**

Assess both the positive and negative metrics within Google Analytics, continually evaluating your content's appearance in search and how users ultimately interact with it. Build on the positives; learn from the negatives.

This includes 'Soft' metrics that are focused on measuring engagement, interaction and brand awareness. Look for bounce rates (high and low), time on page (high and low averages) and page views (in particular, redirection). And 'Hard' metrics, such as lead generation, shares, the number of people added to your list, and sales impact. The aim is to continuously learn from each content creation, setting and reviewing goals for each, and remaining agile.



# Animate SVG with JavaScript

Learn how to create crisp clean animated graphics with the help of the powerful GSAP library from GreenSock



**T**here's so much that can be achieved natively in the browser using CSS3 or the Web Animations API, in JavaScript.

Simple animations and transitions are well suited to CSS3 - whereas more complex animations need to be accomplished using JavaScript. The problem with the Web Animation API is browser support and the current feature set, being a fairly young specification this will improve in the coming years.

In order to combat this feature and browser support deficit we can turn to animation libraries, such as GreenSock (GSAP). GSAP gives us the ability to create complex animations and multiple timelines, with the ability to animate almost any element or property/value pair - all achieved with a simple and intuitive syntax.

In this tutorial we're going to use the GSAP library to create a few animations. We'll be using a landscape vector drawing created by Steven Roberts in order to animate the different elements in SVG. The main and most complex animation we'll be creating will be transitioning the scene from day to night, we'll see how easily we can chain together multiple transitions to create complicated animations. We'll also be creating a few simple animations we'll be running constantly.

### 1. Document setup

To start we need to fork the GreenSock Pen in order to trial their premium plugins (<http://srt.lt/Xt6Ho5>). For the tutorial we'll be using an SVG which has already been optimised and pasted into our HTML editor. However, if you're using your own SVG you'll need to make sure that all of the elements have unique IDs.

<http://srt.lt/WyKf>

### 2. Create the first timeline

GSAP offers two timeline types: TimelineLite and TimelineMax. The TimelineMax version offers access to additional features, such as the ability to repeat animations, as well as playing them in reverse among others. Our first timeline will be the water, which we will

repeat infinitely and yoyo back and forth.

```
var animation_water = new TimelineMax({
  repeat: -1,
  yoyo: true
});
```

<http://srt.lt/M5Fz1D>

### 3. Create the first animation

In order to animate the water we have another path in our SVG, hidden with a '0' opacity. We'll utilise the morphSVG plugin ([greensock.com/morphSVG](http://greensock.com/morphSVG)) to transform our original water path into the new water path. We'll move the water element '12px' down on the y-axis. The two numbers at the end of the property represent the delay and the start times respectively.

```
animation_water
  .to("#water", 2, {
    y: 12,
    morphSVG:"#water-2",
    ease:Linear.easeNone
  }, 0, 0)
;
```

<http://srt.lt/Gc4T8>

### 4. Reusable properties

Since we will be reusing a number of properties and values a number of times we're going to create variables for these properties.

```
var animation_ease = Linear.easeNone;
```

<http://srt.lt/wA8XnA>

### 5. Console logging

The GSAP library offers us the ability to get a number of properties from any given timeline. We can log these in the console to make sure that everything is working as we expect it to.

```
console.log(
  'animation_water duration: '
  + animation_water.duration()
  .toFixed(2)
```

```
);
http://srt.lt/Zo9j8
```

### 6. Cloud timeline and console log

For every element we wish to animate separately and constantly we need to create a new timeline. We also want to log that timeline in the console as we go.

```
var animation_cloud = new TimelineMax({
  repeat: -1,
  yoyo: true
});
console.log( '\n' +
  ...
  animation_cloud.duration().toFixed(2) + '\n'
);
```

<http://srt.lt/D1nOs>

### 7. Cloud animation

Now that we have our timeline ready, we can create our cloud animation. The section of animation which takes the new properties can handle multiple property/value pairs separated using commas.

## GreenSock (GSAP) animation tweens

GreenSock (GSAP) offers a number of animation tweens. However, the two most common are the 'to' and 'from' tweens. The 'to' tween will animate from the element's original properties; whereas the 'from' tween will animate to the element's original properties.

```
.to('element', time, {
  animation properties and easing
}, delay, start-time);
.from('element', time, {
  animation properties and easing
}, delay, start-time);
```



**Above**  
These images show the transitioning of the background gradient and the overlay combined to create the day to night animation.

# Tutorials

## Animate SVG with JavaScript

Our cloud animation only needs to be subtle, so we only need to change the values by a small amount.

```
animation_cloud
  .to("#cloud", 3, { x: -2, y: 1, scale:
    0.95, rotation: 1, ease: animation_ease}, 0,
    0)
  ;
```

<http://srt.lt/Q4J4H>

### 8. Create the night time animation

Next, we'll start to create our day-to-night animation. We'll create a variable for the cycle time and the day. The 'yoyo' setting in GSAP also enables us to delay the animation before repeating.

```
var day_night_cycle_time = 15;
var animation_delay = day_night_cycle_time /
2;
var animation_toNight = new TimelineMax({
  repeat: -1,
  yoyo: true,
  repeatDelay: animation_delay
});
```

<http://srt.lt/N1b0>

### 9. Animate the overlay layer

Inside our SVG we have an overlay layer made of a rectangle covering the entire image with the same background gradient as our night-time background. The overlay applies the 'multiply' blend mode in order to darken the entire image. Our animation simply transitions the opacity of this element.

```
animation_toNight
  .to("#nighttime-overlay",
    day_night_cycle_time, {
```

## GSAP docs & forums

The GreenSock site ([greensock.com](http://greensock.com)) is an invaluable resource for learning how to best use GSAP. It includes learning resources, examples and a showcase of websites built using the library.

```
opacity: 1,
ease: animation_ease
}
, 0 , 0)
;
```

<http://srt.lt/r8M5A>

### 10. Animate the gradient

GSAP offers other tweens on top of the common 'to' and 'from' types. The tween type we need in order to animate our gradient is the 'staggerTo' tween. We can also use the 'cycle' property in order to rotate the colour wheel around to our new values.

```
.staggerTo('#daytime-gradient stop',
  day_night_cycle_time, {
  cycle:{
  stopColor: ['#060414', '#416584']
  },
  ease: animation_ease,
}, 0 , 0)
```

<http://srt.lt/pMdX>

### 11. Animate the Sun

We can keep adding animations to our 'toNight' animation. This time we'll add a new 'to' tween in order to set our Sun. We'll set the display time to be a fraction of the cycle time in order to animate the Sun before the moon. GSAP enables us to set almost any attribute. We'll use this in order to animate the 'cx' and 'cy' properties to below the hill on the right.

```
.to('#sun', day_night_cycle_time / 1.25, {
  scale: 0.9,
  attr:{cx:"753", cy:"697"},
  ease:animation_ease}
, 0, 0)
```

<http://srt.lt/n3q7A>

### 12. Animate the Moon

We'll use the same technique we used to animate the Sun out of view (see Step 11 above) in order to animate the Moon into view. We could achieve this using one tween, of course, but in order to create a faux arc we'll do this in

two parts. In both parts we're also going to apply a new value to the scale property.

```
.to('#moon', day_night_cycle_time / 2, {
  scale: 0.9,
  attr:{cx:"174.5", cy:"202.5"},
  ease:animation_ease}
, 0 , 0)
```

<http://srt.lt/Hk4i>

### 13. Animate the Moon - Part two

The second part of the moon animation waits for the first section to finish before it begins. Note: these two parts of the moon animation are chained together inside the animation code along with the other day-to-night properties that we're using.

```
animation_toNight
  ...
  .to('#moon', day_night_cycle_time / 2, {
  scale: 0.9,
  attr:{cx:"410.5", cy:"114.5"},
  ease:animation_ease}
  , day_night_cycle_time / 2, 0)
  ;
```

<http://srt.lt/w2o2F1>

### 14. Animate the stars

The only part left of our day-to-night animation are the stars. We'll animate the stars into view by transitioning a number of properties. The first of them is to simply bring them into view by animating their opacity.

```
.to('#stars', day_night_cycle_time/2,
  {opacity: 1},
  day_night_cycle_time/2,
  0)
```

<http://srt.lt/oK6kD>

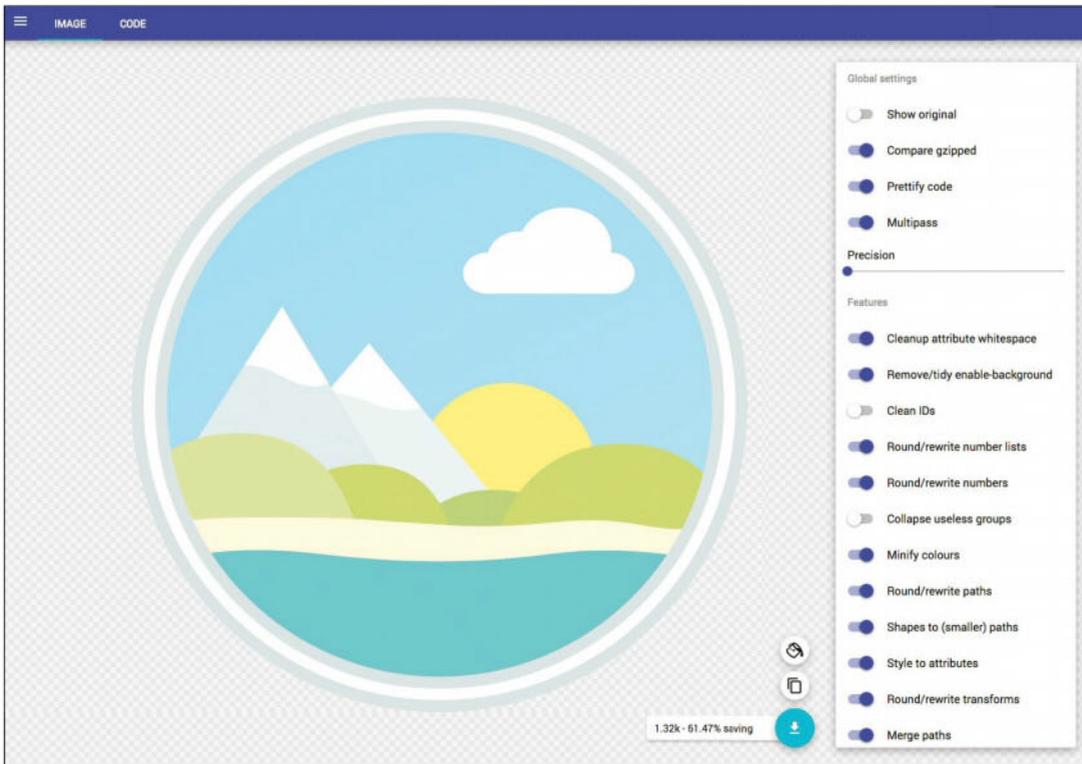
### 15. Animate the stars - Part two

Next we'll use the 'from' tween in order to move the stars up and rotate them from a negative angle as they animate into view. We're using some simple maths in order to calculate our animation time and delay, all based on our 'day\_night\_cycle\_time' variable.



#### Above

We've now completed our day-to-night animation, consisting over animating the background gradient, an overlay layer, the Sun, the Moon and the stars!



### SVG preparation

Preparing your SVG code for animation is an important step in the process. There's a few things we need to do before we try to animate an SVG:

#### Optimisation

There's a number of ways you can optimise your SVG code. The most effective, however, is Jake Archibald's SVGOMG, which offers a visual rendering of the optimised image.

#### Unique identifiers

Elements inside of the SVG which are going to be animated need a way to be targeted. Make sure all of the elements have a unique ID or an identifiable class name!

#### Performance

Animation on the web is expensive in terms of performance, so keep your SVG as simple as possible. The GSAP library includes a number of optimisations, but it can only do so much.

```
.from("#stars",
  day_night_cycle_time - (day_night_cycle_
time / 4),
  {y: 150, rotation: -15, ease: animation_
ease},
  day_night_cycle_time / 4,
  0)

```

<http://srt.lt/qOA4>

### 16. Create the stars timeline and the console log

Now we've created our day-to-night animation we can create another constant animation to make our stars blink. We'll create the new timeline and then log the timeline duration in the console.

```
var animation_stars = new TimelineMax({
  repeat: -1,
  yoyo: true
});

```

<http://srt.lt/n7MOK3>

### 17. Animate the stars

Now we've created the timeline ready for animation, we need to create our blinking animation. The animation is really simple - all we want to do is reduce the opacity value. Thanks to the 'yoyo' property the opacity will animate on and off and so will make the stars look like they are blinking.

```
animation_stars
  .to("#stars", 0.5,
    {opacity: 0.5, ease: animation_ease}
  , 0, 0)
;

```

<http://srt.lt/N6wK>

### 18. Delay the blinking

In the last step we're targeting the stars group in order to apply our blinking animation, however it would look much better if the stars were to blink one at a time instead of together at the same time. We achieve this by targeting each star separately and applying a different animation.

```
animation_stars
  ...
  .to("#star-two", 0.5,
    {opacity: 0.5, ease: animation_ease}
  , 1.25, 0)
  .to("#star-three", 0.5,
    {opacity: 0.5, ease: animation_ease}
  , .75, 0)
  ... ;

```

<http://srt.lt/T1e9p>

### 19. Additional elements - snow!

That's it! Our day-to-night cycling animation is finished and it looks awesome, but we don't have to stop there. Since the image is in SVG we can easily add new elements to our landscape. Let's add some snow. We'll do this using two separate layers. Each layer has a collection of ellipses large enough to cover the landscape and then the same collection repeated above.

```
<g id="snow-bottom-layer" ...>
  ...
  <ellipse .../>
</g>
<g id="snow-top-layer" ...>
  ...
  <ellipse .../>
</g>

```

<http://srt.lt/r7yX8B>

### 20. Create the snow timelines

We create two separate timelines for our snow in order to be able to animate them over different durations. We'll also log their durations to the console.

```
var animation_snowTop = new TimelineMax({
  repeat: -1,
  repeatDelay: 0
});
var animation_snowBottom = new TimelineMax({
  repeat: -1,
  repeatDelay: 0
});

```

<http://srt.lt/P3o9jD>

### 21. Animate the snow

In order to animate our snow layers we want to move the two layers along the vertical axis. By differing their durations we will get the appearance of the layers moving at different speeds. The animation works by moving the collection of ellipses along the vertical axis until the second collection is in place of the first. We then repeat the animation.

```
animation_snow
  .to("#snow-top-layer", 7,
    {attr: {transform: "translate(24 -108)"},
  , 0, 0)
  ;

```

<http://srt.lt/R3pZb>

### Collection of Tutorial Pen's

<http://srt.lt/MeGb7G>

# web workshop

## Create a splash image with scrolling content

As seen on [pomwonderful.com](http://pomwonderful.com)

### Main splash image

Product packshot used as the splash image placed in the centre of the page, making it highly noticeable to all visitors of the page.

### Introduction title

The splash image is accompanied by a bold title that explains the main message to be associated with the web page's splash image.

100%  
POMEGRANATE  
JUICE



ANTIOXIDANT  
SUPERPOWER

Every pure, delightful sip of POM Wonderful unleashes the antioxidant power of our handpicked, California-grown pomegranates.

### Icon navigation

These icons activate JavaScript functionality to allow the user to scroll to the next and previous content sections on the page.

### Section container

Section areas set with relative positioning allow their children, using absolute positioning, to be placed in relation to their location.

### Section text content

Each section has text content that describes the purpose of the background image. These are controlled with absolute positioning.

# Create a splash image with scrolling content

↓ **DOWNLOAD TUTORIAL FILES** [www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

## EXPERT ADVICE

### Content positioning

Conflict with the splash image can be avoided by using a combination of relative positioning for the section containers and absolute positioning for child content. Use non-absolute measurement units, such as percentage (%) and view width (vw) content positioning that adapts to different screen sizes.



**<comment>**  
What our experts think of the site

## Page content layout design

It's important for the page content to remain readable without this type of element becoming an obstruction. Not only should the image be relevant to the content, but the content should also be designed to accommodate the image. How can your web page content be made to flow around the splash image?

**Leon, freelance web developer and trainer**

## Technique

### 1. Initiate HTML document

The first step is to initiate the HTML document. This consists of an HTML container for storing the head and body sections of the page. While the head sections stores links to load the external CSS file, the body section will store the visible content that will be created in Step 2.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<meta name="description"
content="description">
<title>Fixed Image, Scrolling Background</
title>
<link rel="stylesheet" type="text/css"
media="screen" href="styles.css"/>
</head>
```

### 2. Main HTML content

The page content consists of a span element with ID of 'image' to be used for the image presentation, along with several 'sections' for storing the scrollable content areas. All sections are stored inside the main container to provide advantages for presentation control.

```
<span id="image"></span>
<main>
<section>
<h1>The content 1</h1>
</section>
<section>
<h1>The content 2</h1>
</section>
<section>
<h1>The content 3</h1>
</section>
</main>
```

### 3. CSS initiation

With the HTML now complete, create a new file called 'styles.css'. This file will be used to store the CSS presentation rules. The first rules will set the page's HTML container and body to cover the full space of the window without border spacing.

```
html, body{
display: block;
width: 100%;
height: 100%;
padding: 0;
```

```
background: #000;
}
```

### 4. Main content sections

All content sections will share a common set of formatting; primarily set to fit the full width of their container and the full height of the browser window. You can change these rules to suit the requirements of your particular project.

```
main section{
display: block;
width: 100%;
min-height: 100vh;
}
```

### 5. Main section colours

Individual colours are added to each section for the purpose of being able to distinguish them within this example. The nth-child selector is used to keep the HTML clean without the need to apply either a unique ID or class attributes.

```
main section:nth-child(1){background: red;}
main section:nth-child(2){background: green;}
main section:nth-child(3){background: blue;}
```

### 6. Splash image style

The splash image needs to be presented in a size and location relative to the browser window regardless of the screen resolution. This is achieved using vh (view height) and vw (view width) measurement units. Setting the background image and size using 'cover' will automatically resize the image.

```
#image{
display: block;
position: fixed;
width: 25vw;
height: 50vh;
top: 25vh;
left: 37.5vw;
z-index: 9999;
background: url(image.svg);
background-size: cover; }
```





# SKETCH

## 21 TIPS FOR BETTER DESIGN

SKETCH IS A LEADING TOOLS FOR DESIGNING USER INTERFACES AND WELL-LIKED IN THE INDUSTRY. HERE WE REVEAL A SELECTION OF ESSENTIAL TECHNIQUES, TIPS, SHORTCUTS & PLUGINS

Sketch. First Adobe Photoshop's rival, then Adobe XD's rival, and now InVision Studio's rival as well. A tool surprisingly resilient for an app that's never released a Windows version, although it may not seem so surprising when you learn not only what it can do, but how it does it. It's

an app for designing user interfaces, but its powerful, intuitive tools makes the design process so artfully smooth you'll quickly understand why and how it paved the way for a new generation of design apps. Here are 21 tips that will make you a power user in minutes.

## 1 MAKE YOUR DESIGN COLLABORATIVE WITH LIBRARIES

Can other designers also use your Symbols? Absolutely!

Sketch Libraries are incredibly easy to use, we can only speculate as to why they've only been added to Sketch in the last couple of months. All you need to do is to create the Symbols in a separate .sketch file, and then leave that .sketch file somewhere accessible, such as on Box, Dropbox or Google Drive.

Other designers can then import the Sketch Library into their Sketch documents by navigating to File → Add Library... After that, Symbols from that Sketch Library will be accessible via the usual Insert → Symbol method. Sketch Libraries are a terrific way to enable design collaboration in teams.



**“WHAT SETS SKETCH APART FROM OTHER DESIGN TOOLS IS HOW ACCESSIBLE ITS TOOLS AND USER INTERFACE ARE. DESIGNING WITH SKETCH FEELS SO FLUID, ESPECIALLY WHEN YOU THROW ITS MINIMALIST KEYBOARD SHORTCUTS INTO THE MIX. IT'S LIKE DESIGNING WITH YOUR MIND, YOU'RE RARELY SLOWED DOWN BY TABS AND HIDDEN MENUS.”**

**DANIEL SCHWARZ**  
DESIGN EDITOR AT SITEPOINT  
[MRDANIELS.CH](http://MRDANIELS.CH)

When the Sketch Library is updated, any .sketch file that's using that Library will receive a notification requesting if you'd like to update the document. Pretty cool, right?

## 2 TEST A RESPONSIVE DESIGN USING RESIZING

You have a container, which can be either a Group or an Artboard, and various layers inside that container. Let's say that this container is a website header with a floated logo and navigation - the logo floated on the left, and the navigation floated on the right. When you resize the container, you want the logo to still be floated to the left, and the navigation to the right. Enter Resizing, where contained elements can be floated to a container edge, helping you to test responsive designs.

Once again, this all happens from the Inspector, under the "Resizing" tab. There are four possible locations that you can pin from:

- Left: object will be fixed to the left-hand side.
- Right: object will be fixed to the right-hand side.
- Right and top: object will be fixed the top and right sides.

Pretty simple really, but there's a snag. When you resize containers ordinarily, the contained elements are resized accordingly. You can fix this by activating the "Fix Width" and "Fix Height" options, which will stop this from happening. Centralised objects will remain centralised as well. Magic!

## 3 SEPARATE YOUR PROJECTS BY PAGE

After a while, your canvas will start to feel cluttered despite the fact that it's essentially unlimited. Large designs require organisation, which is why we can logically separate Artboards into different Pages. The best way to organise Artboards into Pages is by platform. If you were designing, for example, an Android and iOS version of a mobile app, you could have a separate Page for each. The reason for this, is that when you use prototyping and design handoff tools, you'd want to export the right Artboards to the right Project, so separating them in this way makes logical sense. With web designs, I typically create things like Symbols and logos on their own Page

## 4 AUTOMATE YOUR ASSET EXPORTS

Once you have defined your export options, you won't have to do that again, and this is true whether you're actually exporting image assets from Sketch or you're exporting to a design collaboration app, such as InVision, Marvel, Sympli or Zeplin.

Here's what you need to know: File format matters. SVG is better for the web because it's a vector format, which means that you can scale it up and down without losing quality. SVG formats are usually smaller in file size too, which makes webpages load faster. We use PNG for mobile apps, although to cater for Retina/HDPI screens, we export images @2x and @3x, depending on the device that we're catering for. You can define as many options as needed, all at once.

Just click the "Make Exportable" button at the very bottom of the Inspector and list the different export options. And then, whenever you need to re-export, hit Cmd+Shift+E. Your export settings will now be saved, but you will also be given the option to export only one, or some, or all of the layers you have chosen.

## 5 ORGANISE IMAGE ASSET EXPORTS INTO FOLDERS

Developers don't want to sort through an entire folder of exported image assets, especially with large designs. Breaking them down into folders is more organised and we can do this from Sketch. Simply rename the layer using this format: "subfolder/asset.png" to auto-create folders in your export. Tip: did you know that you can drag a layer from the layer list onto your desktop to export it? It'll be a @1x PNG file by default.

## 6 ANALYSE COLOUR CONTRAST WITH THE STARK PLUGIN

Although it's not as trendy as concepts such as prototyping and animation, colour contrast is a big deal. Poor colour contrast results in limited accessibility, especially for users who have limited eyesight (something which affects 285 million of us worldwide). The WCAG (Web Content Accessibility Guidelines) sets worldwide standards for colour contrast, which can be somewhat difficult to understand for somebody that's never heard of them before.

The Stark Plugin ([getstark.co/](http://getstark.co/)) not only helps you to compare two colours against these WCAG 2.0 standards, to make sure that your contrast is sufficient enough, but also test your design against the eight different types of colour blindness. The key things to check are when two backgrounds are adjacent to one another; buttons against backgrounds; and most importantly, text against backgrounds. Font size factors into this as well, as smaller text is harder to read, so considering that there's so much to think about here, Stark is handy to have in your toolbox.

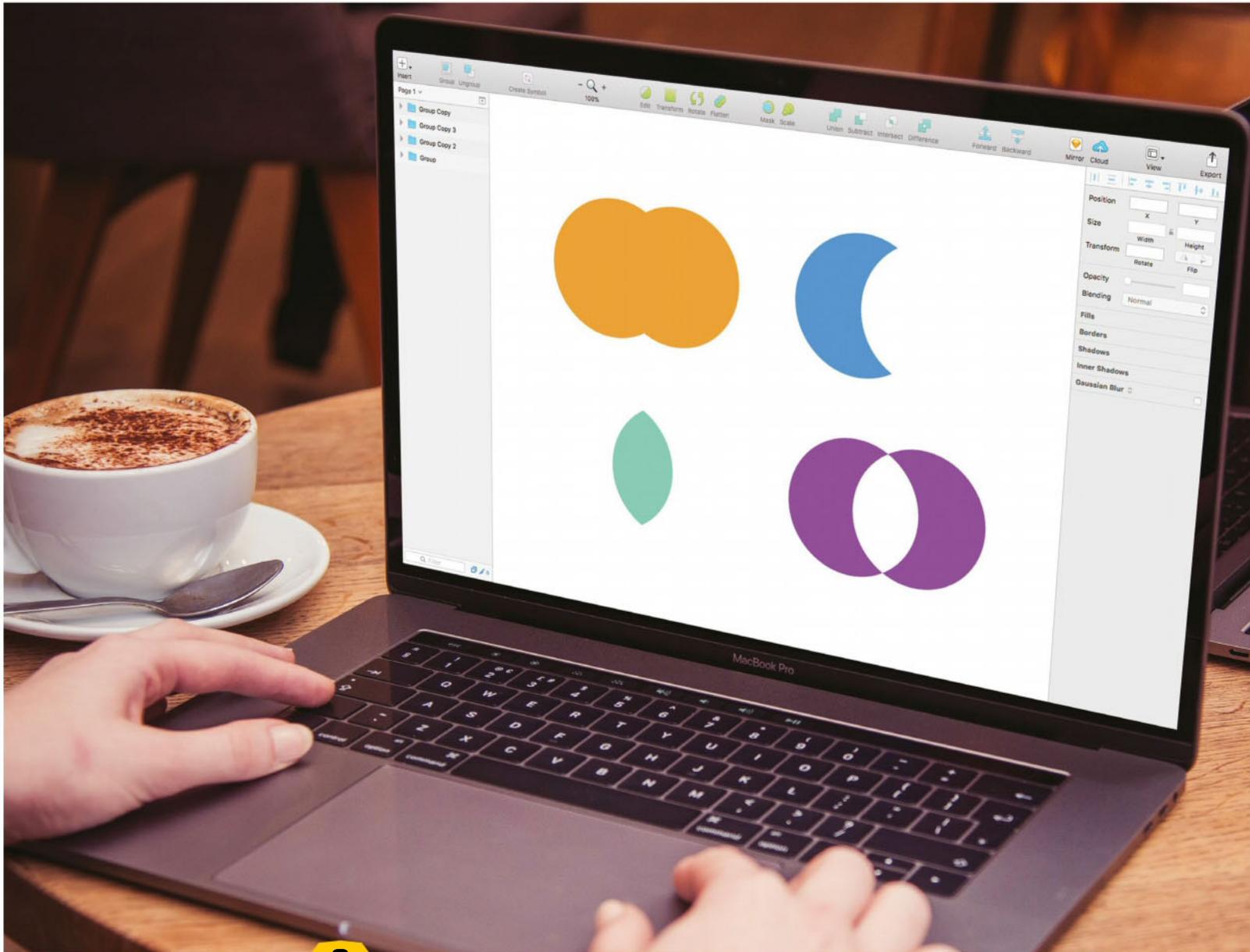
Stark opens in a new window (Plugins → Stark → Show Stark). Press ↑ and ↓ to cycle through the types of colour blindness, ← and → to cycle through the Artboards in your document, or select two colours to analyse the colour contrast. Colour contrast is defined by two different levels/ratio (AA and AAA), and your colour contrast should be meeting at least the AA requirements.

## 7 DESIGN HANDOFF WITH THE ZEPLIN OR SYMPLI PLUGINS

Design handoff is typically built into prototyping apps such as InVision and Marvel, but they are only available to teams with enterprise-level subscriptions. Smaller design teams (or solo designers) won't need design collaboration on that scale, so dedicated design handoff tools, such as Zeplin and Sympli, might be more suitable. You'll have design handoff and collaboration, but at the cost of prototyping features, which might be fine depending on the way you or your team does prototyping (because there are also paper prototyping and free tools, such as Craft).

Like most tools of this calibre, you can export screens from Sketch to Zeplin or Sympli using their Sketch Plugin. Both tools work largely the same way. With Zeplin, use Cmd+Ctrl+E in Sketch to export screens; with Sympli, use Cmd+Y.

Both apps let you tag screens for added searchability (useful for narrowing down screens in the handoff interface), and automatically collect image assets you've marked as exportable in Sketch, ready for the developer to implement them. As with all competing tools, use the free trials and decide for yourself.



## 8

# COMBINE SHAPES WITH BOOLEAN OPERATIONS

Boolean operations let us combine shapes to make new shapes.

### STEP 01: CREATING TWO SHAPES

Let's use a circle as an example here. You can combine as many shapes as necessary with boolean operations, but for now we'll use only two. Create a circle, and then duplicate it (Cmd+D) to create another circle. Make sure that they overlap, so that we can use one of them as a mask and see the effects.

### STEP 02: SELECT A BOOLEAN OPERATION FROM THE MENU BAR

As you can see from the Sketch menu bar, there are four boolean operations to choose from: Union, Subtract, Intersect and Difference. Union is the simplest of all the boolean operations, where the two shapes are combined to create a new shape. With Subtract,

one of the shapes eats into the other shape or shapes, subtracting from them. With Intersect, only the section where the shapes overlap will be visible. With Difference, it's the complete opposite, only the sections where the shapes don't overlap are shown. For simple shapes, this is often faster than creating them freehand using the vector tool.

### STEP 03: MAKE IT CLEAR WHICH LAYER IS THE MASK

As I said before, you can combine more than one shape with boolean operations. One of them has to be the mask, and this is the one that appears at the "bottom", as indicated in the Layer List. Combined shapes act like a Group once combined, so whichever one is lowest in the hierarchy is the mask. You can reorder these layers to

change which one is the mask. Note: boolean operations technically consist of paths and subpaths, but the concept of masking might make boolean operations feel more familiar.

### STEP 04: LEARN THE KEYBOARD SHORTCUTS FOR QUICK-ACCESS

Naturally, like everything else in Sketch, there are keyboard shortcuts that you can use to access these boolean operations faster:

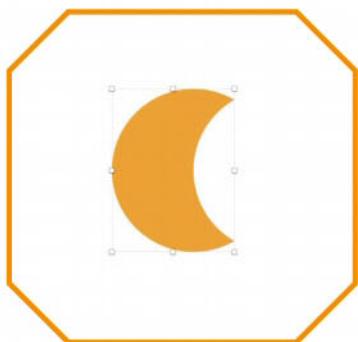
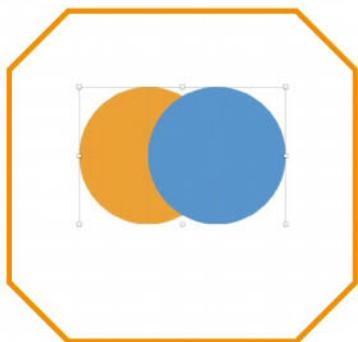
**Union:** `Cmd+Ctrl+U`

**Subtract:** `Cmd+Ctrl+S`

**Intersect:** `Cmd+Ctrl+I`

**Difference:** `Cmd+Ctrl+X`

If you have one of the latest MacBook Pros with a Touch Bar above the keyboard, these tools are also available for quick access via touch as well.



# 6 SKETCH PLUG-INS

## LAUNCHPAD

[animaapp.com](https://animaapp.com)

LaunchPad, in a nutshell, converts your .sketch files to HTML/CSS and publishes your websites to the web. After connecting your design to a domain and setting up any integrations supported by LaunchPad (such as Google Analytics), you'll have a working website without having to code. Here's what it actually supports:

- Responsive design
- Videos, links, forms
- Font-family fallbacks
- Hosting and domain linking
- SEO customisation (title/meta tags)
- Social media tags (Open Graph, Schema, etc)
- Built-in version control, in case something goes wrong
- Integrations like Google Analytics, Zendesk or MailChimp
- New: A/B testing functionality

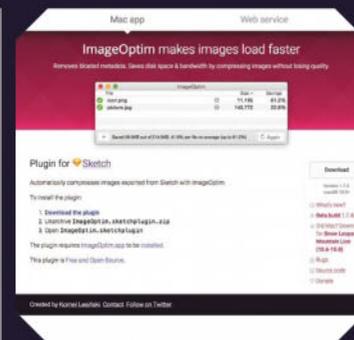


## SKETCH RUNNER

[sketchrunner.com](https://sketchrunner.com)

Sketch Runner is basically the macOS's Spotlight, but for Sketch. Power users swear by Sketch Runner, as it's the easiest and quickest way to access a tool or feature in Sketch without having to use the mouse quite so often. It also saves you from having to dig through drop-down menus to find the tools you don't need that often (these are usually the ones without a keyboard shortcut).

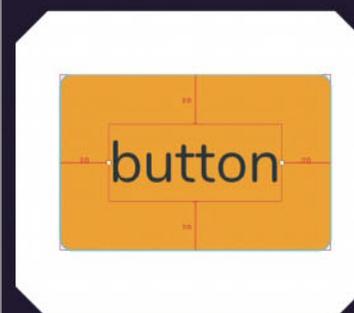
Open Runner with Cmd+, and run your last action with Cmd+Shift+.



## IMAGEOPTIM

[imageoptim.com/sketch](https://imageoptim.com/sketch)

While SVG optimisation is a big issue, for which Sketch offers its own SVGO Plugin (<https://github.com/BohemianCoding/svg-compressor>), you'll need ImageOptim's range of smart compression techniques to optimise PNG and JPEG files for the web. There's no special keyboard shortcut for ImageOptim, since it automatically compresses images when you export your assets. Just make sure to have the ImageOptim app open when you do this.



## SKETCH DYNAMIC BUTTON

<https://github.com/ddwht/sketch-dynamic-button>

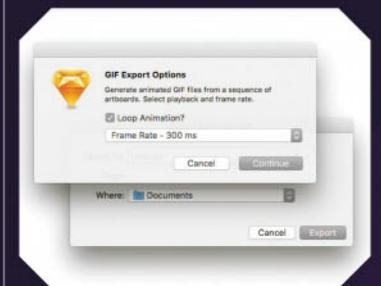
An oldie but a goodie, Sketch Dynamic Button allows you to define a certain amount of spacing on the inside of your buttons, known as padding to web developers. This means that buttons are resized in accordance with the text content inside them, and let's be fair, buttons are a pretty common and important aspect of any website design. It's really easy to use as well - simply create a text layer and then hit Cmd+J. For existing buttons, rename the containing Group in this format: 20:20:20:1020 and run the shortcut again, where 20 is the amount of inside spacing on the button itself.

## EXPORT MORE

<https://github.com/nathco/Export-More>

Export More takes all of your Artboards and then strings them together as an animated GIF, where you can then decide whether the GIF will play once or roll on a loop. Navigate to Plugins > Export More Artboards to GIF and choose the desired frame rate, and you're all set. It's super-simple!

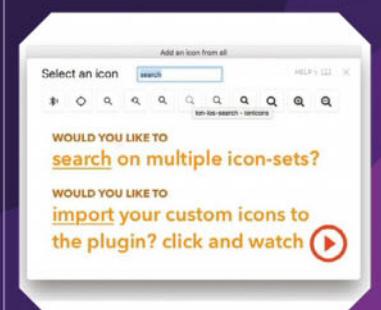
Navigating to Plugins > Export More > Artboards to ICNS will export a 1024 x 1024px Artboard into the various different Apple Icon sizes, as required by Apple.



## SKETCH ICONFONT

<https://github.com/keremciu/sketch-iconfont>

Icon fonts are a terrific way to save time and cut costs, with Ion Icons, Simple Line Icons, Material Design Icons and of course Font Awesome being some of the most used on the web. Sketch IconFont, assuming that you have the fonts installed on your computer, makes it really easy to insert these icons into your .sketch design as a scalable vector (SVG). After installing the icon fonts by downloading them from their respective websites, navigate to Plugins > Icon Font > Grid Insert [your desired icon name]. It's a huge time saver, especially for those that tend to default to a certain icon set.



COMBINED SHAPES ACT LIKE  
A GROUP ONCE COMBINED,  
SO WHICHEVER ONE IS  
LOWEST IN THE HIERARCHY  
IS THE MASK

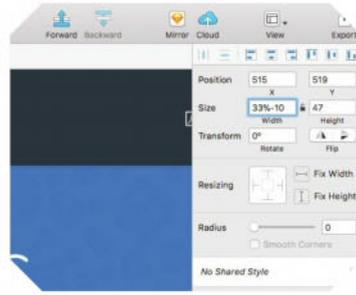


# SKETCH: 21 TIPS FOR BETTER DESIGN



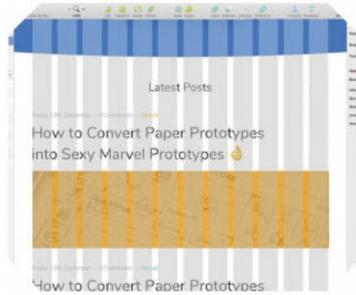
## 9 CREATE YOUR OWN SKETCH SHORTCUTS

You can create your own Sketch shortcuts. To set your own Sketch shortcuts, head to macOS Preferences → Keyboard → Shortcuts → App Shortcuts → +, and then choose Sketch as the Application and type the menu command you want to define a shortcut for. For example, "Collapse All Groups" (Cmd+Ctrl+C) is a useful way to quickly clean up your Layer List! Add as many shortcuts as you need.



## 10 DEFINE MEASUREMENTS WITH MATHS FUNCTIONS

When specifying measurements with the Inspector, you can use maths functions to save you from having to do the calculations in your head. Say that you wanted a Rectangle to span the full width of an Artboard minus 20px on each side (i.e. a 20px margin), you could define the Width as "100%-40px". With a 320px Artboard, the Rectangle's width would then equal 280px.



## 11 MIMIC CSS FRAMEWORKS

Web developers often use CSS frameworks such as Bootstrap and Foundation, which come with their own responsive grid systems. So to make designs more durable and flexible (and to protect development costs), designers using Sketch might want to replicate these grid systems using Layout Grids.

Navigate to View → Canvas → Layout Settings... to bring up the Layout Grid modal, then specify the numbers of rows and columns along with the overall width of the website container and any column/gutter widths. Layout Grids appear light grey on the canvas, although you can change this colour. Toggle them on and off with Cmd+Alt+L, your developer will thank you later!



## 12 KEEP YOUR COMMON COLOURS HANDY

Colours are reused throughout a design for visual consistency. When used correctly, colours mean something to the user and so the UX is improved. It's typical to have a certain colour for the brand, the body text, the headings and so on, and we can save these colours in Sketch for quick-use, so that we don't have to recall hex/RGBA values from memory. Whenever you're inside the colour chooser widget and you have the desired colour selected, click the "+" icon underneath "Document Colors" to save it to the document swatch. If you add colours to the "Global Colors" swatch, they'll appear in all Sketch documents. Drag the colours out of the colour chooser widget to remove.

## 13

THERE ARE A LOT OF KEYBOARD SHORTCUTS IN SKETCH. POPULAR ONES CAN BE FOUND IN THIS ARTICLE. YOU CAN SEE MORE AT [BIT.LY/2NPX8YK](http://BIT.LY/2NPX8YK)

Screen Design. Supercharged.

# CRAFT

EMAIL ADDRESS

GET CRAFT NOW



EXPLORE CRAFT

## 14 PROTOTYPE LIKE A PRO

Prototyping and design collaboration is all the rage right now. The ability to see your designs come to life with interactive hotspots, dynamic components and animations before development has even begun, is not only fun to watch, but lets teams and stakeholders test user experiences and leave feedback early on.

Being able to iron out the majority of creases before the user even experiences the app or website ensures that your conversions are higher, your customer complaints are lower, and your product launches have more impact.

One of the best ways to prototype directly in Sketch is arguably the Craft plugin by InVision ([invisionapp.com/craft](http://invisionapp.com/craft)), which supercharges your design process in a number of ways. Here's the rundown:

- Provide your design with realistic dummy data.
- Source stock images for your content.
- Duplicate content vertically

and horizontally.

- Create shared design libraries that are stored in the cloud.
- Enter "Freehand" mode for discussion and collaboration.
- Make your designs dynamic and interactive.
- Sync all of this into the core InVision App.
- Enables handoff, so developers can inspect design styles.

Craft exists as a vertical sidebar alongside the Inspector in Sketch. Despite its many features, the UI is fairly easy to use.

Marvel ([marvelapp.com](http://marvelapp.com)) is another prototyping app that has a Sketch plugin. Although it only serves as a way of exporting designs into the core Marvel app, there are some benefits of using Marvel over InVision - which is not as flexible as InVision - which has more prototyping features, for example - the learning curve is smaller and the Marvel interface feels simpler than InVision's.

Give both a try, they're fantastic!



## 15 TIME-SAVING SHORTCUTS

### LET THE KEYBOARD BE YOUR MOUSE

Actually, you'd be quite surprised what you can do with Sketch using only the keyboard. Many of the tools are single-letter shortcuts, and repetitive, mundane tasks are something of an urban myth now. Once again, here are the shortcuts you'll love:

- Rectangle:** *R*
- Oval:** *O*
- Line:** *L*
- Text:** *T*
- Artboard:** *A*
- Edit mode:** *Return*
- Move up in hierarchy:** *Esc*
- Change opacity of object:** *0-9*
- Rename:** *Cmd+R*
- Export:** *Cmd+Shift+E*
- Zoom in/out:** *Cmd and +/-*

### CMD/ALT/SHIFT COMBOS ARE YOUR NEW BEST FRIENDS

- Group:** *Cmd+G*
- Ungroup:** *Cmd+Shift+G*
- Move object:** *↑→↓←*
- Resize object:** *Cmd+↑→↓←*
- Move object by 10px:** *Shift+↑→↓←*
- Resize object by 10px:** *Cmd+Shift+↑→↓←*
- Click through to Object in Group:** *Cmd+left-click*
- Activate Smart Guides:** *Alt*

### Smart Guides with click-through:

*Cmd+Alt+left-click*

### Move object up/down in hierarchy:

*Cmd+Alt+↑↓*

### Increase/decrease font size:

*Cmd+Alt and +/-*

**Align left:** *Cmd+Shift+{*

**Align centre:** *Cmd+Shift+|*

**Align right:** *Cmd+Shift+}*

### MINIMISE YOUR WORKSPACE

What always frustrated me about Adobe Photoshop was how bloated and cluttered it felt. It was a multidisciplinary tool, used for both design and art, so it's no wonder.

Sketch, by contrast, is focused on UI design, so not only is it minimal by default, but you can toggle the toolbars on and off with a quick shortcut. This is useful for creating more space on the canvas when needed, or when you need to enter the "Just show me the design" mode. These are the Sketch toolbar shortcuts that you need to know:

### Show/hide Toolbar (at the top):

*Cmd+Alt+T*

### Show/hide Layer List (on the left):

*Cmd+Alt+1*

### Show/hide Inspector (on the right):

*Cmd+Alt+2*

### Presentation mode (hide everything): *Cmd+.*

### YOU SHOULDN'T FORGET ABOUT THESE SHORTCUTS EITHER

#### Slice: *S*

Creates a rectangular selection that can then be exported, which is useful for highlighting or screenshotting key sections. Be sure to activate the Trim Transparent Pixels option in the Inspector to remove any whitespace from your Slices.

#### Vector: *V*

Vector tools are there to help you freehand draw totally unique shapes – icons or illustrations, for example. You can use 1, 2, 3 and 4 on the keyboard once focused on a vector point to change the point style from the default Straight style, to Mirrored, Asymmetric or Disconnected.

#### Mask: *Cmd+Ctrl+M*

When a shape is used as a mask, it redefines any content that overflows that shape. One example of this could be when you need a square image to fit inside a circle – the circular mask hides the 90-degree angles of the square image. Handy.

MANY OF THE TOOLS ARE SINGLE-LETTER SHORTCUTS, AND REPETITIVE, MUNDANE TASKS ARE SOMETHING OF AN URBAN MYTH NOW



meaningful name (such as “Largest Heading”, for example).

Flexibility comes more naturally with Shared Styles than it does with Symbols, where you can simply change what you need to change in one instance of a Shared Style and those changes won't sync with the other instances automatically. The change is a manual one, where you'll need to hit the sync icon in that same drop-down menu.

By not hitting the sync icon here, you are making a singular change to an otherwise visually consistent Shared Style.

## 19 LIVE PREVIEW YOUR DESIGNS WITH SKETCH MIRROR

When mobile apps and websites are designed to be used by thumbs, it makes sense to test them with thumbs, which is exactly where Sketch Mirror comes in very handy. We need this to bring our designs off the big screen, and onto the little screen.

With Sketch Mirror for iOS, you can preview your Sketch artboards on your iOS device. Click “Mirror” from the toolbar, then select your device from the list of options. If you can't see your device, make sure that you have Sketch Mirror open on your device and that they're both on the same wireless network.

Tip: swipe right on the iOS Control Centre and tap the Record icon to record a user test or create a video demonstration, which you can then send to your teammates - or anyone else - in a quick message!

## 20 CREATE THAT SMOKEY iOS BLUR

When designing mobile apps for iOS, you'll often use a lot of their native UI elements and visual concepts, one of which is the smokey blur that appears as a backdrop to reduce visual distraction from any background elements. This effect blurs out the background when the Control Centre is open, for example.

You can create this effect by selecting the “Background Blur” option from the Inspector. It should be noted that this doesn't work if the opacity of the Fill is 100%. The lower the opacity, the greater the blur effect.

Background Blur differs from Gaussian Blur in that with Gaussian Blur, the edges of the blurred background are blurred too.

## 16 REUSE COMPONENTS WITH SYMBOLS

Colours aren't the only thing that we can reuse. In fact, we can reuse entire components. Take a main navigation, for example - you'll use that on most (if not all) screens, right? If you made a change to it, you'd want that change to be universal, right?

This is where Symbols come in.

Select the component that you want to make reusable and then click “Create Symbol” from the toolbar at the top. After that, navigate to Insert → Symbol and then click anywhere on the canvas to insert it.

You can choose to store the master instance of the Symbol on a separate Page, away from the main design. This is the instance of the Symbol that you edit when you want to make a universal change to it. Double-clicking on any other instance from the main design will redirect you to the master instance anyway.

## 17 NESTED SYMBOLS AND SYMBOL OVERRIDES

You can also embed an SVG image

into the page in CSS, using the same method as you would to attach any other kind of image to an element.

By using the SVG this way we can take advantage of the other CSS background properties allowing us to size, position and repeat our image as the background of an element.

To provide support we can use a trick in the way CSS is parsed by the browser and provide a fallback PNG right before we include our SVG.

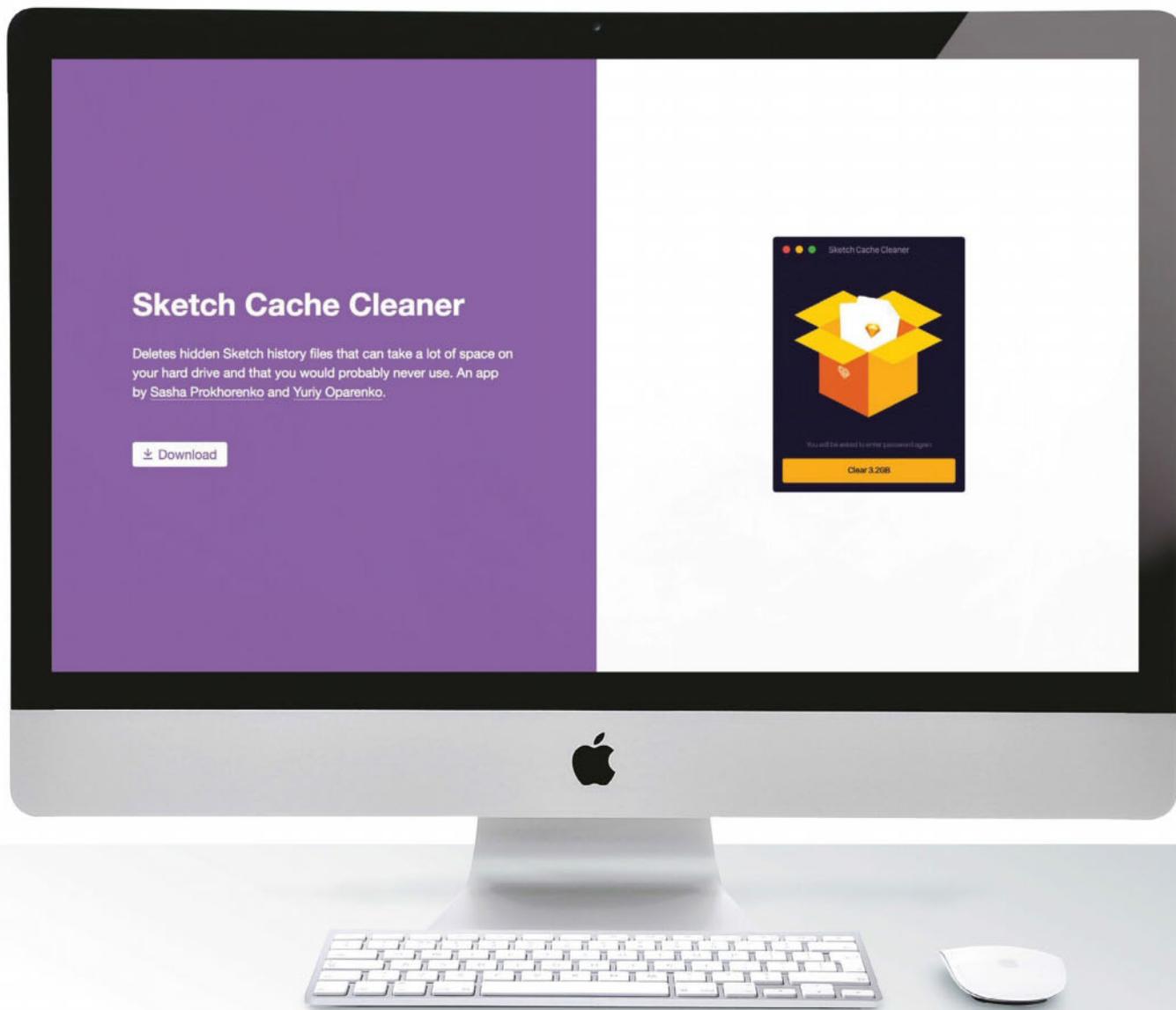
## 18 MAINTAIN DESIGN CONSISTENCY WITH SHARED STYLES

And finally there are Shared Styles, which are like Symbols but for layers. Consider your heading styles, button text styles, body text styles and so on - the styles that you'll reuse time and time again. Shared Styles are created in the Inspector.

To create Shared Styles, select the layer in question and then click the “No Shared Style” dropdown from the Inspector (or “No Text Style” if it's a text layer). Then, choose “Create new Text/Shared Style” from the list of dropdown options, then give it a

TO PROVIDE SUPPORT WE CAN USE A TRICK IN THE WAY CSS IS PARSED BY THE BROWSER AND PROVIDE A FALLBACK PNG





## 21 SPACE SAVERS

Sketch users often complain that .sketch files take up a lot of space on their computers. Here's what you can do to get around that problem.

### DELETING THE SKETCH CACHE TO SAVE SPACE

When a .sketch file is automatically saved as a backup, the new cached version doesn't overwrite the old cached version. Each cached version is stored somewhere safe, which is what enables Sketch's native version control features. While this is useful for those needing to roll back to an earlier version of their design sometimes, those versions eventually end up stealing all of your megabytes (or gigabytes). Sketch Cache Cleaner (<https://yo-op.github.io/sketchcachecleaner>) is an independent app that deletes all of

these versions when this cache becomes too large. Of course, you can turn off version control in Sketch's Preferences, but sometimes it's useful.

Instead, clean the cache only when you're sure there's nothing you'll need from it. If you're using a version control app such as Abstract ([goabstract.com](http://goabstract.com)) or Plant ([plantapp.io](http://plantapp.io)), then in that case it would serve you well to turn off Sketch's native version control.

### REDUCING THE SIZE OF SKETCH FILES

Even Sketch files themselves can become quite large, but Sketch now has a built-in feature that reduces their file size, which you can find at File → Reduce File Size. Some designers say that Reduce from the Flawless App team ([flawlessapp.io/reduce](http://flawlessapp.io/reduce)) is a little more efficient at this though, since it retains the colour quality better.

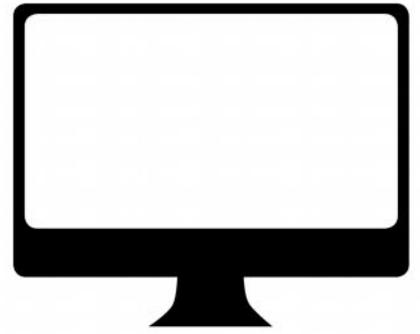
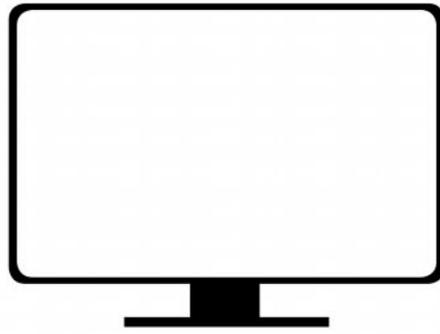
Reduce is free to download and sits in the macOS menu bar, making it quick and easy to access at any time..

### BEING A NEAT FREAK

Generally speaking, getting in the habit of cleaning up your .sketch documents as you design (or at regular intervals), will reduce their file size. By that I mean removing any redundant layers, Ungrouping (Cmd+Shift+G) any redundant Groups, and basically making your documents clean and easy to read. This not only benefits your file sizes, but also makes your work readable for anybody that views it after you, which might include developers who are inspecting your work so that it can be converted to code, or even another designer that you might be collaborating with on a large design.

GETTING IN THE HABIT OF  
CLEANING UP YOUR  
SKETCH DOCUMENTS AS  
YOU DESIGN (OR AT  
REGULAR INTERVALS),  
WILL REDUCE FILE SIZE

# BROWSER



**PLUGIN**

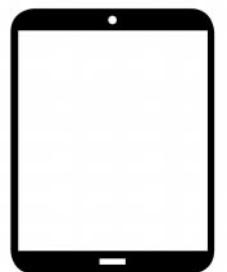
**TECHNOLOGIES**

**USING**

**OPEN**

**WEB**

**THE ARRIVAL OF FIREFOX 57, HAS ELIMINATED LEGACY ADD-ONS. HERE WE TAKE A CLOSER LOOK AT THE NEW TECHNOLOGY**





## About Mozilla Firefox



# Firefox

56.0 (64-bit) **What's new**  
Mozilla Firefox for Ubuntu  
canonical - 1.0

Firefox is designed by **Mozilla**  
**global community** working to  
the Web open, public and ac

Want to help? **Make a donati**

Licensing Information

End-User Rights

Firefox and the Firefox logos are trademarks of the Mozilla Foundation.

## A QUESTION OF COMPATIBILITY

The big browser vendors Mozilla, Microsoft, Opera and Google are all looking to abandon their native interfaces and replace them with ones based on classic web technologies. As with many other standards, theory and practice differ: while the APIs look very similar, some differences exist. For example, Mozilla's official recommendation, outlined at [developer.mozilla.org/en-US/Add-ons/WebExtensions/Porting\\_a\\_Google\\_Chrome\\_extension](http://developer.mozilla.org/en-US/Add-ons/WebExtensions/Porting_a_Google_Chrome_extension), advises that you start out by simply running your extension – if it doesn't work, then you need to look into the API documentation to find out more about the behaviour differences. Developers who have a crx file ( these are extension files used by Chrome ) can also upload it at [extensiontest.com](http://extensiontest.com) – this web-based tool will analyse the contents of the file and will point out common issues automatically.

# LET'S GET STARTED

## Recent versions of Mozilla's Firefox let you create plugins using Web technologies

One of the first real uses of plugins came in the form of WinAmp: the long-gone media player allowed developers to expand both the program's performance and its looks via third-party extension modules written in proprietary languages. Getting started with a web-based plugin for Firefox is not particularly difficult.

### 1. VERSION CHECK

Everything we've outlined here assumes you are working with Firefox version 56: the steps outlined should, however, also work with later versions of the browser. Should you find yourself on an unknown workstation, simply check the version number by choosing Firefox > About on a Mac or Firefox > More > Help > About Firefox on a PC.

### 2. CREATE A MANIFEST FILE

A browser extension identifies itself towards the browser by publishing a manifest file. Start out by creating an empty folder and placing a file called 'manifest.json' in it. Then, adjust its contents to start out as following:

```
{
  "manifest_version": 2,
  "name": "FutureTest",
  "version": "1.0",
```

"description": "This is a small test",

### 3. TIE INTO THE REST

Our plugin can connect itself to Firefox in various ways. For now, we will limit ourselves to tying in into a specific website, which requires the population of the manifest file with the code printed here.

```
  "icons": {
    "48": "icons/border-48.png"
  },
  "content_scripts": [
    {
      "matches": ["*://*.google.com/*"],
      "js": ["runthis.js"]
    }
  ]
}
```

### 4. POPULATE THE FOLDERS

Cook up a 48x48 icon of choice, and place it in the folder described by the manifest. Next, create a file called 'runthis.js' - it will be invoked by the

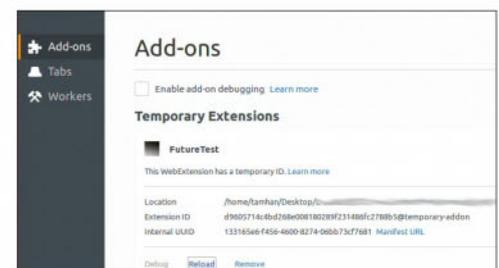
plugin engine whenever the launch of a matching web site is detected. Its code looks like this:

```
  alert("Hello Future!");
```

**TEST CAREFULLY**  
Practical experience shows that testing plugins with multiple versions of Firefox is recommended: as all kinds of odd behaviour can creep in between releases

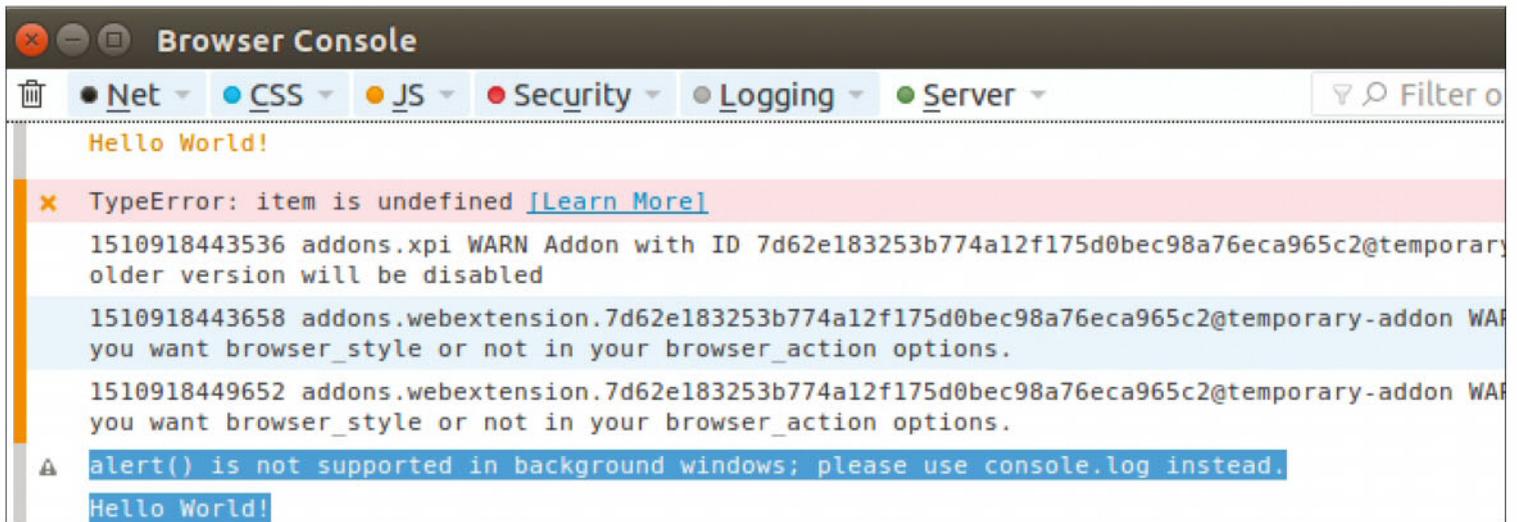
### 5. INTEGRATE THE ADD-ON

Plugins normally need to be signed by Mozilla. For testing, open about:debugging and click the 'Load temporary Add-On' button. The browser will react by popping up a common dialog window where the folder containing the example must be selected. When done, the program shows up as shown in the figure – open [www.google.com/test/](http://www.google.com/test/) to see the message box on the screen.



# EXPAND BROWSER FUNCTIONALITIES

Simply running a JavaScript file when a page pops up is boring. Let us integrate our plugin more deeply...



## 1. MODIFY THE MANIFEST

As seen in the exercise on the previous page, manifest files inform the browser about what an add-on wants to do. Creating a button or another form of GUI tie-in, therefore, requires the creation of an additional element, which replaces the content\_scripts block intended to kick off scripts as specific web pages get loaded.

```
“background”: {
  “scripts”: [“worker.js”]
},
“browser_action”: {
  “default_icon”: {
    “16”: “icons/page-16.png”,
    “32”: “icons/page-32.png”
  }
}
tamhan@TAMHAN14:~$ node -v
v6.10.3
```

## 2. IMPLEMENT SOME LOGIC

Mozilla’s plugin scripting engine runs the script file whenever it gets triggered: as we registered worker.js via the background script, it gets run as the add-on is loaded. Therefore, our first responsibility involves creating an onClicked handler to allow our program code to receive incoming button events.

```
function openPage() {
  alert(“Hello World!”);
}
browser.browserAction.onClicked.
addListener(openPage);
```

## 3. FIRE IT UP

The deployment process should be easy: create

the relevant icon files, and load the plugin using the temporary plugin button found in the Firefox debugging tab. Next, take a careful look at the address bar – our new plugin will pop up in a fashion similar to the one shown in the figure along with an error.

## 4. WHAT’S GOING ON?

Our example uses one of multiple internal APIs of the Firefox browser, which are documented in full under [developer.mozilla.org/en-US/Add-ons/WebExtensions/Browser\\_support\\_for\\_JavaScript\\_APIs](https://developer.mozilla.org/en-US/Add-ons/WebExtensions/Browser_support_for_JavaScript_APIs). Sadly, pages running in the background are not allowed to use the alert() function – Mozilla perceives that enabling this option would be an invitation for unscrupulous developers working on ‘browser nuisanceware’.

## 5. CREATE A BOOKMARK

Triggering the browser console is but part of the

## GET SIGNED

Firefox OS (the now defunct open-source operating system) introduced developers to Mozilla’s signing process: depending on the capabilities required, more or less checks had to be performed. In the case of plugins, the situation is a bit different – before an add-on can be installed by non-developer versions of Firefox, they need to be signed by Mozilla according to the process outlined at [developer.mozilla.org/en-US/Add-ons/Distribution](https://developer.mozilla.org/en-US/Add-ons/Distribution).

By default, Mozilla attempts to peddle its own Add-On repository during the signing process: if you do not want your add-on to appear in the official add-on list, be careful to disable the relevant checkbox during the submission process.





fun: our program would be much more interesting if it could interact with the browser API. As a first example, let us change the code of the event handler so that it creates a bookmark with a specified set of parameters.

```
function onCreated(node) {
  console.log(node);
}
function openPage() {
  var createBookmark = browser.bookmarks.create({
    title: "Tamoggemon's non-working homepage",
    url: "http://www.tamoggemon.com"
  });
  createBookmark.then(onCreated);
}
```

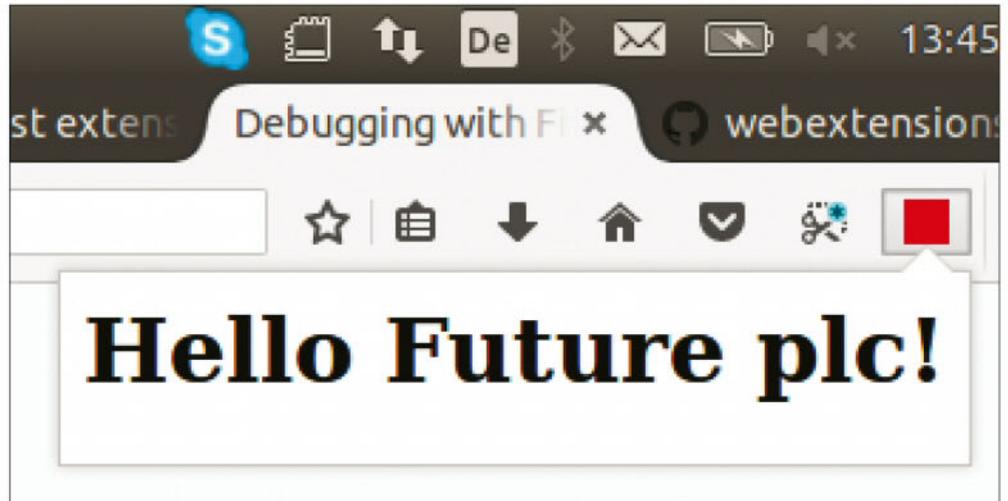
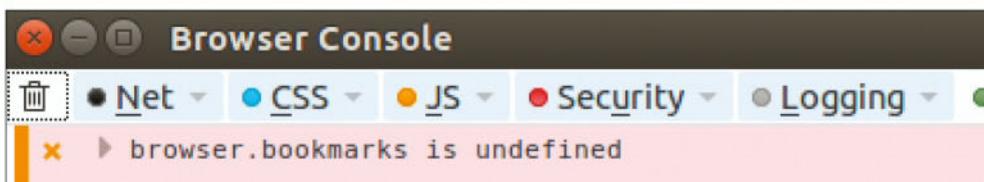
### 6. DELETE & RELOAD

Saving your changes in an editor of choice does not make them available to Firefox. The safest way involves first deleting the add-on in the debug pane, and then reloading it by hand. This does not restore buttons which were removed from the toolbar by right-clicking them: the only way to get them back is called toolbar settings.

### 7. AVOID BEGINNER'S MISTAKES

Running the add-on we just created yields the error shown in the figure. It is misleading: the namespace used is correct. However, our manifest lacks the necessary permission, which must be added before the extension can be run:

```
"permissions": [
  "bookmarks"
],
```



### 8. BE A NUISANCE!

Placing a button in the settings area is not the only possibility: the knob can also be set loose in the address bar of the browser. For this, simply create a page action element in the manifest – its content, by and large, is just as you would expect it to be.

```
"page_action": {
  "browser_style": true,
  "default_icon": {
    "19": "button/geo-19.png",
    "38": "button/geo-38.png"
  },
  "default_title": "Whereami?";
}
```

### 9. WITH A POP-UP!

As seen above, background workers do not have access to a particularly expansive GUI stack. One

small reprieve comes in the form of the pop-up attribute: it enables you to specify an element which runs whenever your GUI is clicked:

**SVG A-HOY!**  
Creating all of these symbols by hand can be annoying: a nice way to handle multiple resolutions in one strike involves the use of scalable vector graphics

```
"browser_action": {
  "default_popup": "page.html",
  "default_icon": {
    "16": "icons/page-16.png",
    "32": "icons/page-32.png"
  }
}
```

### 10. BRING IT UP

Create the referenced HTML file, and provide it with the markup shown in this Step. Please do bear in mind that the enabling of the pop-up attribute automatically disables the click event handler of the plugin.

```
<html>
<head>
</head>
<body>
  <h1>Hello Future plc!</h1>
</body>
</html>
```

# INTERACTING WITH PAGE CONTENT

## Managing information stored in the browser is part of the game

The real power of add-ons lays in their ability to interact with the web content that has been loaded. For example, a savvy developer could create a plugin which plucks data from a finance website, and performs some kind of analysis on it to help a trader work faster.

### 1. GET SOME RIGHTS!

Interacting with web content requires two types of permissions: first of all, the permission for tabs

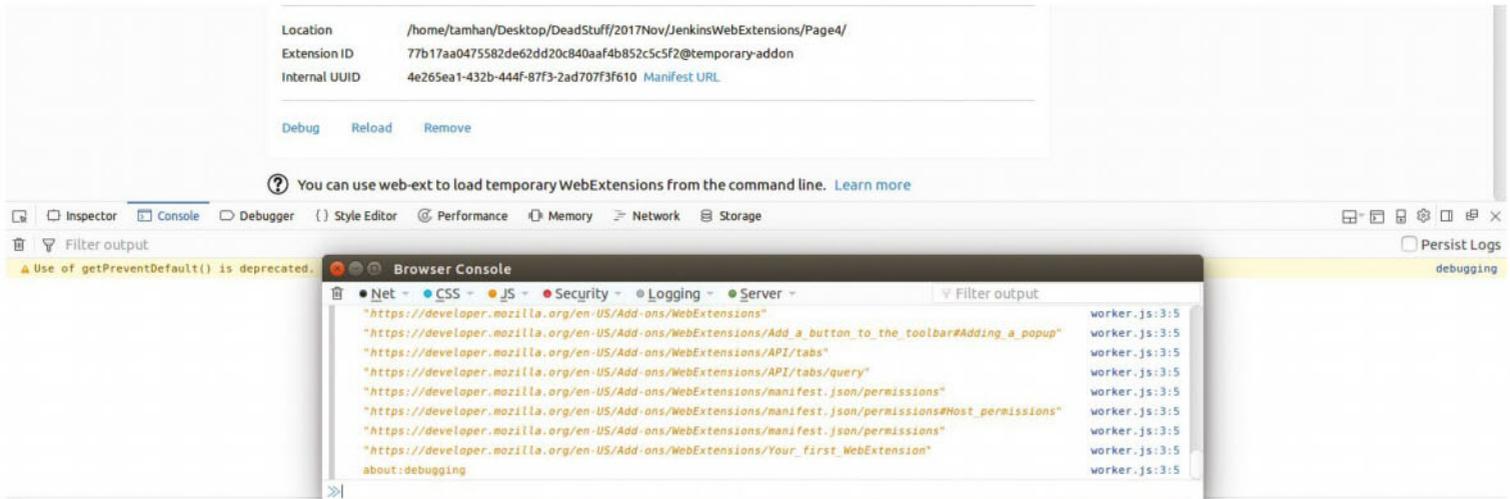
is required to enumerate the currently open windows. Secondly, a host-level permission is required for each group of websites, which the add-on wants to process:

```
"permissions": [
  "*/www.tamoggemon.com/*",
  "tabs"
],
```

### 2. CHECK PRESENCE OF OPEN PAGES

Mozilla's web API provides a list of currently-opened tabs. Let us give it a stab by creating a bit of code which prints the contents of all currently-opened tabs into the browser window.

```
function logTabs(tabs) {
  for (let tab of tabs) {
    console.log(tab.url);
  }
}
```



```
function onError(error) {
  console.log('Error: ${error}');
}
function openPage() {
  var querying = browser.tabs.
  query({});
  querying.then(logTabs,
  onError);
}
```

### 3. FIND AND FILTER

Seasoned web developers can stumble upon a little trap here: our output is shown in the Browser console, and will not appear in the web console. Another neat trick involves the query function, which can be provided with various filter parameters to limit the amount of tabs returned.

```
var querying = browser.tabs.query({url:
"moz-extension://*/*/*"});
var querying = browser.tabs.
query({currentWindow: true, active: true});
```

### 4. UNDERSTAND THE LIMITATIONS

We can now fire off a scripting payload into a target tab. This easy-sounding job gets complicated by a small detail – the DOM image presented to the script is called an 'xray image', which ignores script-made modifications. Let us try this out via the HTML file shown here.

```
victim.htm/c/
<html>
  <head> </head>
  <body>
    <script src="onlineworker.js"></
script>
  <p id="testp1">Test-P 1</p>
  </body>
</html>
onlineworker.js/c/
var p = document.createElement("p");
```

```
p.textContent = "This is a dynamic P.";
p.setAttribute("id", "softwarep");
document.body.appendChild(p);
window.tam = "Let us see";
```

### 5. PREPARE FOR TESTING!

Yours truly hosts the two example files at tamoggemon.com/test/2017/009 future/victim.htm – simply open the file in Firefox, and convince yourself that the new <p> element and the changed tam property show up in the DOM inspector and the debugger. This is of significance, since the above-mentioned XRAY view will impose a set of limitations on us.

### 6. FIRE OFF A SCRIPT

Firing off a program is slightly difficult: the executeScript function can take either a path to a file or a string containing actual code. For now, test the structure by creating a file called 'local.js' in the plugins root folder.

```
function logTabs(tabs) {
  for (let tab of tabs)
  {
    console.log(tab.url);
    var executing =
    browser.tabs.executeScript({
      file: "/local.js"
    });
    executing.then(onExecuted, onError);
  }
  . . .
function openPage() {
  var querying = browser.tabs.
  query({currentWindow: true, active: true});
  querying.then(logTabs, onError); }
```

**LORD OF THE ACL**  
Firefox knows about two dozen different permissions. Visit [mzl.la/2A318gl](http://mzl.la/2A318gl) to get a list, and scroll down to learn more about each one of them

**RIGHTS CHECK!**  
Not everything which can be crawled technically should be crawled. Check your data source's consent before releasing anything on the open market

### 7. INSPECT DYNAMIC PROPERTIES

With that, we can run a first test. Our JavaScript file sets a property called 'tam' in the universally-popular window object. In theory, it should be available also in the plugin. Sadly, this is not the case – as mentioned above, Firefox creates a 'clean copy' of the DOM at hand.

```
if(!window.tam)
  alert("Something is wrong!!");
```

### 8. CHECK FOR HARD PRESENCE

This firewall, however, has limits. For example, the software-generated <p> tag can also be inspected from the plug-in code. A good example to check that involves using the getElementById method as shown in the code accompanying that step.

```
if(document.getElementById("softwarep"))
  alert("New element detected");
```

### 9. USE THE CSS

Plugins are not limited to injecting JavaScript code. The extension can also add CSS to modify page design using the insert-css function in a fashion similar to the snippet shown here.

```
var insertingCSS =
browser.tabs.insertCSS(2,
{file: "content-style.css"});
insertingCSS.then(null,
onError);
```

### 10. THE KNIFE CUTS BOTH WAYS

Bear in mind that the 'firewall' effects demonstrated in this mini-tutorial are not limited to plugins accessing the DOM. The constraints also work the other way around: if your plugin sets properties, expect most of them to be invisible to 'normal' page scripts.

# CREATE THE IMPOSSIBLE

www.photoshopcreative.co.uk

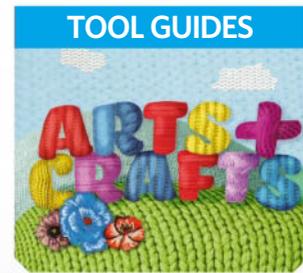
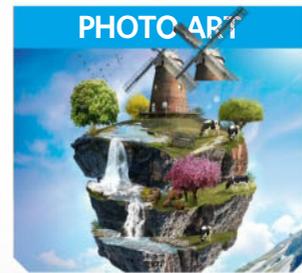


# Photoshop creative

Available from all good newsagents and supermarkets

ON SALE NOW

- Striking imagery • Step-by-step guides • Essential tutorials



# BUY YOUR ISSUE TODAY

Print edition available at [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)

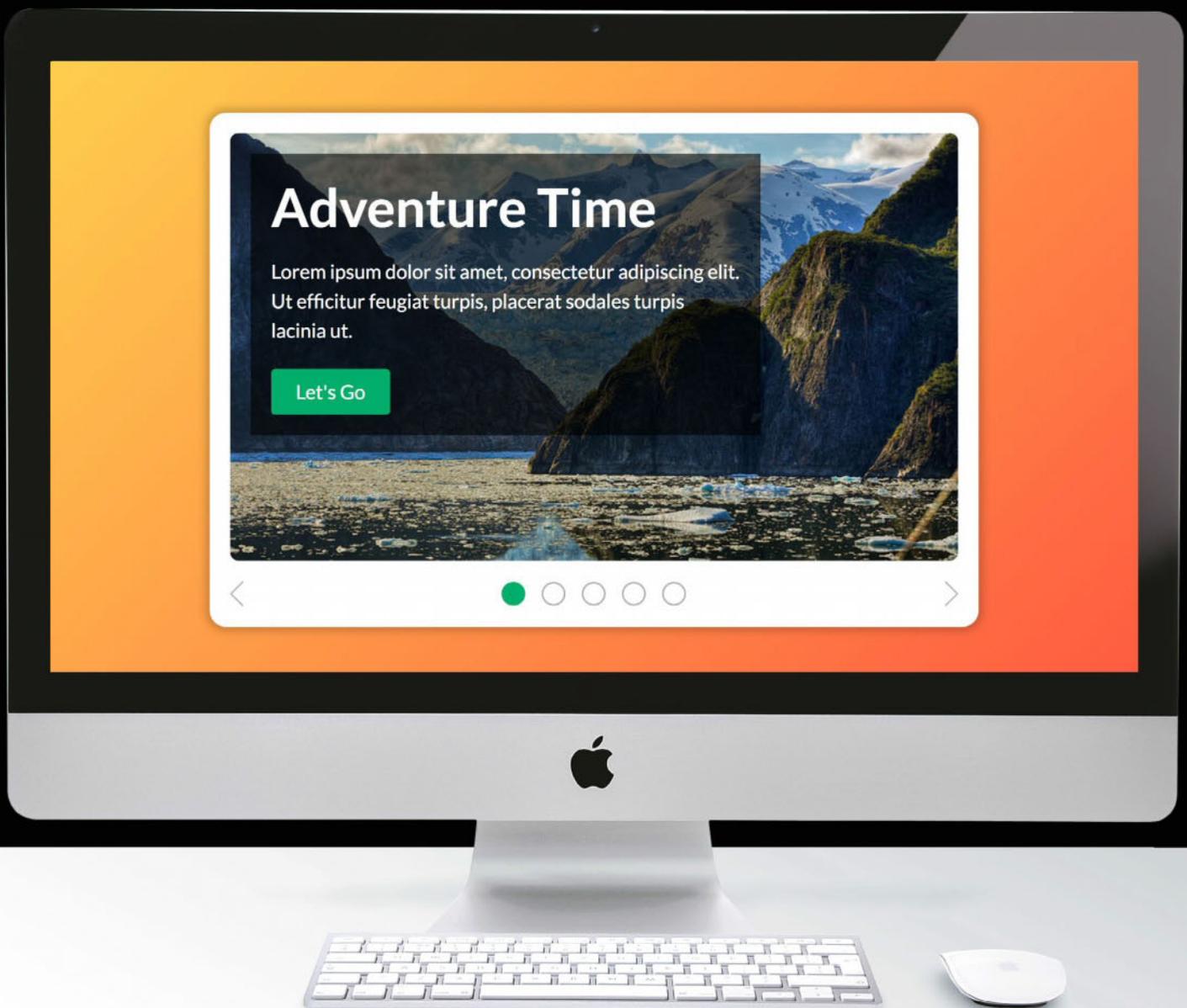
Digital edition available for iOS and Android

Available on the following platforms



# Create a flexible carousel component

How to create a simple, flexible carousel component in Vue.js, the JavaScript framework





Vue.js has come on leaps and bounds in 2017, becoming the sixth most forked project so far on Github, even ahead of Facebook's own ReactJS. It's safe to say

that it's fast becoming a mainstay in web development, and a reliable go-to JavaScript framework for use in projects. In this tutorial, we're going to be using Vue.js to create a simple carousel component. This component will accept a number of different properties, enabling you to tweak some basic settings such as transition speed, transition type, and whether the carousel should automatically transition slides.

To get started, download the project files from Filesilo and open the 'website-template' directory in your preferred text editor. Meanwhile in terminal, CD into 'website-template' and then run 'npm install' to install the projects Node.js dependencies. Finally, run 'npm run dev' to start up a development server so that you can view your project in the browser. Typically this would be at 'localhost:8080'.

## 1. Create carousel slide files

In 'src/components', create a new directory called 'app-carousel-slide' and in it two files: 'app-carousel-slide.vue' and 'component.app-carousel-slide.scss'. These will contain the first of two new Vue.js components that we will be creating, which when used together will create our carousel component.

## 2. Add carousel slide SCSS

From the filesilo package, copy the contents of '/support-files/step-02-slide.scss' into 'component.app-carousel-slide.scss'. This is the SCSS for the slide component, and uses the 'flex-grow' property to make sure each slide expands to fill its parent element.

## 3. Create the slide component

In the empty 'app-carousel-slide.vue' file, add the snippet below to create the structure of the Vue.js component. We'll use this as a foundation to build the carousel slide.

```
<template></template>
<script>
export default {
  name: 'app-carousel-slide'
}
</script>
```

## 4. Add a template slot for the carousel slide images

In the empty <template> element of the carousel slide, we'll add a 'div' element to represent the slide, along with a special slot element which we will name 'image'. In Vue.js, slots enable you to interweave your own content with the component's template without editing it. In this instance, they are used so we can later pass the image for the slide background, so the end result, when ultimately used, would look like <app-carousel-slide><img src="" alt=""></app-carousel-slide>.

```
<div class="c-app-carousel-slide">
  <slot name="image"></slot>
</div>
```

## 5. Add the slide text container

All that remains for the slide component is to build the text container. We will be using slots again, one for the larger slide title and one for regular text. We will also be using a Vue.js directive called 'v-if' to add logic, which only renders the text container if at least one of the slots is passed content. Add this snippet in 'c-app-carousel-slide', just before the 'image' slot.

```
<div class="c-app-carousel-slide__text-block">
  v-if="this.$slots['title'] || this.$slots['text']">
    <h1 class="c-app-carousel-slide__title" v-if="this.$slots['title']">
      <slot name="title"></slot>
    </h1>
    <div class="c-app-carousel-slide__text" v-if="this.$slots['text']">
```

```
<slot name="text"></slot>
</div>
</div>
```

## 6. Create carousel files

Back in 'src/components', create a new directory called 'app-carousel' and then within it two new files: 'app-carousel.vue' and 'component.app-carousel.scss'. These will hold the second of the two Vue.js components: the main carousel itself.

## 7. Add Carousel CSS

Copy the contents of '/support-files/step-07-slide.scss' into the empty 'component.app-carousel.scss'. This is the SCSS for the main carousel component.

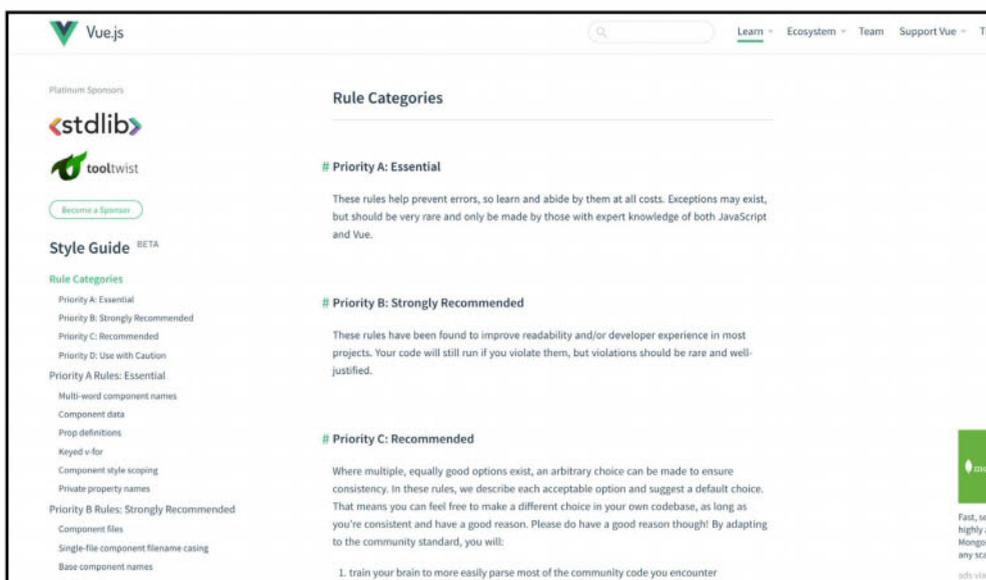
## 8. Create carousel component

Next in 'app-carousel.vue', we're going to build the structure of the carousel component. We're importing the 'applcon' component and the 'arrow' svg for later use in the carousel's next and previous icons. These work together with the 'svg-sprite-loader' dependency to generate a basic SVG icon system, which works using SVG 'symbols' and its 'use' element.

```
<template></template>
<script>
import appIcon from
  '@/components/app-icon/app-icon'
import arrow from
  '../assets/img/arrow.svg'
export default {
```

## Remember the commas

Often in the tutorial, you'll be adding content to JavaScript objects. Remember to separate each piece of content with a comma or Vue.js will throw an error.



### Top

The Vue.js team have recently released a style guide for the framework. This is full of useful examples, rules and recommendations which every Vue.js developer should check out.

### Left

The style guide is split up into four sections, providing rules which are considered "Essential", "Strongly Recommended", "Recommended", and "Use with Caution". Find out more at [vuejs.org/v2/style-guide](http://vuejs.org/v2/style-guide).

## Developer tutorials

# Create a flexible carousel component

```
name: 'app-carousel',
components: {
  appIcon
},
data() {
  return {
    arrow
  }
}
}
```

### 9. Build the carousel template

Let's start adding content to the empty template element. The main area of interest here is the 'c-app-carousel\_\_container' element, which we'll shortly calculate a width for based on the number of slides found within it. We'll then move the container using CSS transform:translateX and transitions to simulate slide movement.

```
<div class="c-app-carousel">
  <div class="c-app-carousel__wrapper">
    <div class="c-app-carousel__container">
      <slot></slot>
    </div>
  </div>
</div>
```

### 10. Add the carousel controls and arrows

We then need to add the HTML for the carousel controls container and the previous and next arrows; the latter using the icon system and svg imported in Step 8. Add these after the 'c-app-carousel\_\_wrapper' element.

```
<app-icon class="c-app-icon-arrow-prev
c-app-carousel__arrow" use="arrow" />
<div class="c-app-carousel__controls">
</div>
<app-icon class="c-app-icon-arrow-next
c-app-carousel__arrow" use="arrow" />
```

### 11. Create the carousel data store and add properties

We're going to add three new properties to the component's data store: 'slideTotal' will hold the total number of slides; 'activeSlideIndex' will record the index of the visible slide so it can be used to calculate the container's position; while 'autoInterval' will record the interval timer which will trigger an automatic slide transition. All of these are set to null, with the exception of 'activeSlideIndex', where the '0' value indicates that the first slide should be the default slide.

```
data() {
  return {
    arrow,
    slideTotal: null,
    activeSlideIndex: 0,
    autoInterval: null
  }
}
```

### 12. Calculate slideTotal

Add 'ref="container"' to the 'c-app-carousel\_\_container' element in the template and then add the snippet below as a property of the component object itself. 'ref' is used to give easy access to an element, which in this case is the container so we can count how many child elements (aka slides) it has. The presence of this logic in a 'mounted()' function means it is then automatically run when the component is first rendered.

```
mounted() {
  this.slideTotal =
  this.$refs.container.children.length;
}
```

### 13. Calculate container width

In the component, create a new object property called 'computed' and within it, a new function called 'containerWidth()'. We'll use this to calculate the width of the carousel container based on the 'slideTotal' figure.

```
computed: {
  containerWidth() {
    return this.slideTotal * 100 + '%';
  }
}
```

### 14. Create methods

Next, create another object property called 'methods' to store our carousel's functions. 'goToSlide()' is an easy way of setting 'activeSlideIndex' from Step 11, 'isControlActive()' returns true when a control's index matches 'activeSlideIndex', while 'nextSlide()' and 'prevSlide()' will simply cycle through the slides.

```
methods: {
  goToSlide(slideIndex) {
    this.activeSlideIndex = slideIndex;
  },
  isControlActive(controlIndex) {
    return controlIndex - 1
    === this.activeSlideIndex;
  },
  nextSlide() {
    this.activeSlideIndex === this.
    slideTotal - 1 ? this.activeSlideIndex
    = 0 : this.activeSlideIndex++;
  },
  prevSlide() {
    this.activeSlideIndex === 0 ? this.
    activeSlideIndex = this.slideTotal - 1
    : this.activeSlideIndex--;
  }
}
```

### 15. Calculate the carousel's container position

The carousel uses a percentage value with transform:translateX and CSS animation to simulate slide transition.

## Vue.js browser extension

When developing with Vue.js, we recommend that you also download the Vue devTools extension, which is currently available for Chrome and Firefox. It helps to visualise your Vue.js application.

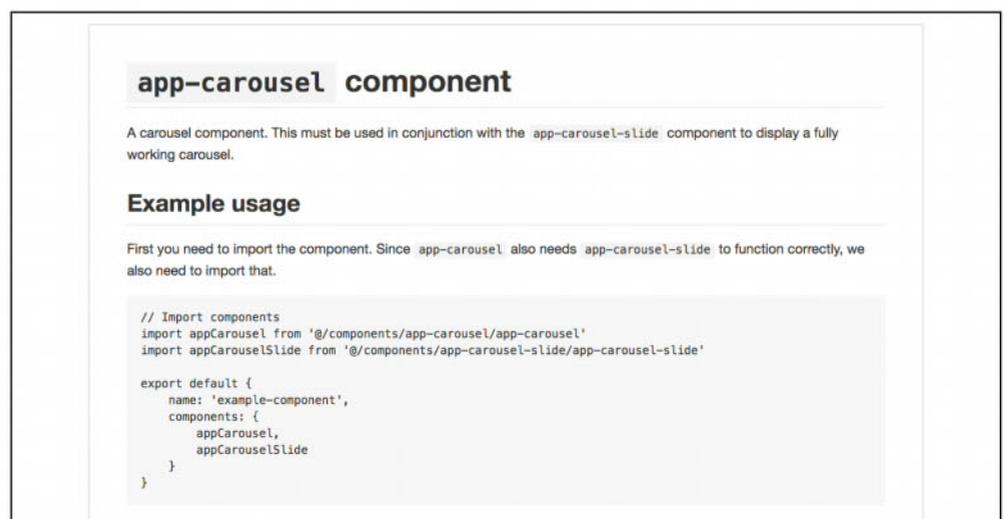


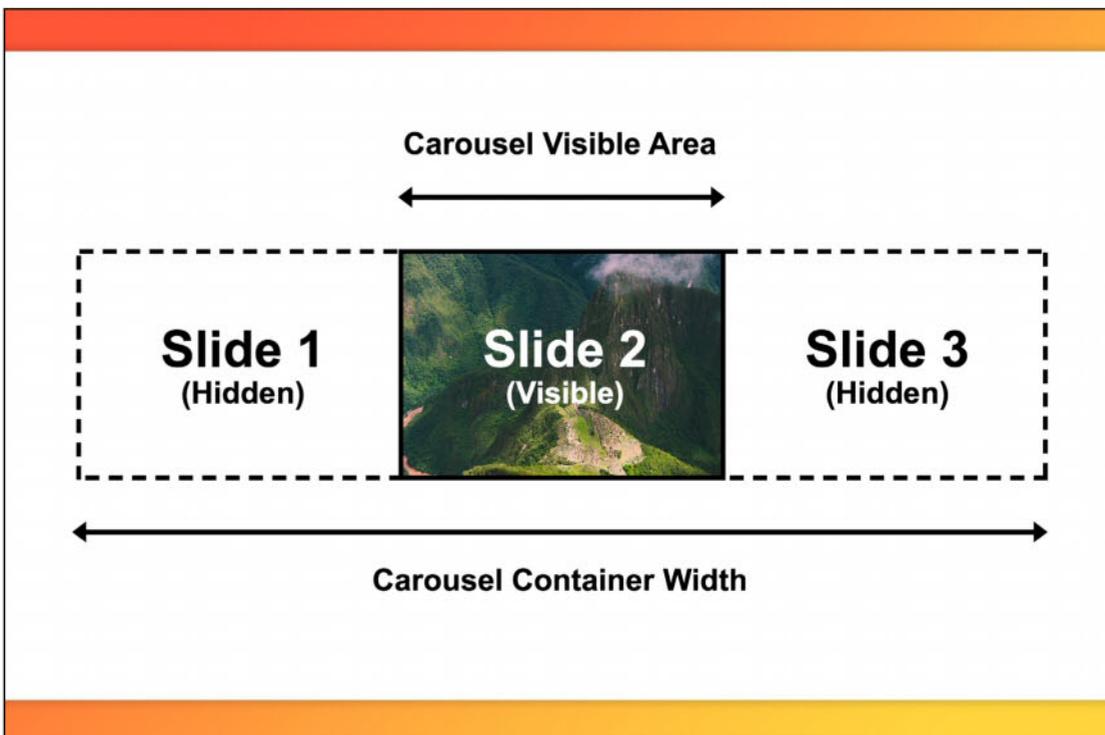
#### Above

In this test carousel Vue app, each component is documented. With some, such as the app-button component, the documentation is minimal. With others, for example the app-carousel-slide components, there's much more information which developers need to know.

#### Right

If working on a Vue.js project with other developers, you need to properly document a component so others can understand how it's used. Include a brief overview of its purpose, an example usage snippet and info on properties.



**How the carousel works**

Simply put, the trick is that the slides themselves don't move, but the container element which holds the slides in place does.

This container sits within a wrapper element, which has `overflow:hidden` applied so nothing outside its dimensions is visible. We then calculate the container width using the number of slides so it's always wide enough for its content to tile horizontally. For example, as seen in Step 13, if there are three slides within the container, its width would calculate to 300 percent.

Once the container is wide enough, and the slides are correctly positioned, it's just a matter of moving the container using `transform:translateX` into the wrapper element's visible area so that the correct slide displays.

Add the below snippet to the `computed` object so we can calculate this value.

```
activeSlidePosition() {
  return '-' + (100 / this.slideTotal)
  * this.activeSlideIndex + '%';
}
```

**16. Compose inline CSS**

Now we have all the values to correctly position the carousel slide container, we need to construct the CSS which we'll then add to its `style` attribute. We'll add this logic as another function in the `computed` object.

```
containerStyle() {
  return `width:${this.containerWidth};
  transform:translateX(${this
    .activeSlidePosition});`
```

**17. Bind inline CSS**

Add the below snippet to the `c-app-carousel__container` element in the template. This will bind the returned value of `containerStyle()` from the previous step to the `style` attribute of the carousel container, meaning that its CSS and therefore its position will automatically be updated when things change.

```
v-bind:style="containerStyle"
```

**18. Hook up next/previous arrows**

We now need to add logic to the next/previous arrows so that the correct method from Step 14 is called when each respective arrow is clicked. The `prevSlide()` snippet belongs on the `c-app-icon-arrow-prev` element, while `nextSlide()` belongs on the `c-app-icon-arrow-next` element. The `v-on` directive is just an easy way to set up event listeners in Vue.js, with `click` being the DOM event that we are targeting.

```
// Prev
v-on:click.native="prevSlide()"
// Next
v-on:click.native="nextSlide()"
```

**19. Generate carousel controls**

Let's generate the control elements and make them show the corresponding slide when clicked. Add the below element in `c-app-carousel__controls`. The `v-for` directive is used to create an amount of control elements matching the `slideTotal` variable, whilst the `v-bind` directive enables the `is-active` class only when the `isControlActive` method from Step 14 returns true. Finally, we're creating another event listener through `v-on` so, when clicked, the control calls the `goToSlide` method and passes its index, which should match the index of corresponding slide.

```
<div v-for="n in slideTotal"
  :key="n" v-bind:class=
  "{ 'is-active': isControlActive(n) }"
  class="c-app-carousel__control"
  v-on:click="goToSlide(n - 1)" ></div>
```

**20. Import components**

Let's now go back to the top level `app.vue` component and import everything. Just after the opening `<script>` tag, import the component `.vue` files:

```
import appCarousel from '@components/
  app-carousel/app-carousel'
import appCarouselSlide from '@components/
  app-carousel-slide/app-carousel-slide'
```

Next, amend the `components` object so it references these newly imported components.

```
components: {
  appButton,
  appCarousel,
```

```
  appCarouselSlide
}
```

Finally, in the `<style>` tag, import our new SCSS with the rest of the component imports.

```
@import "/components/app-carousel/
  component.app-carousel";
@import "/components/app-carousel-slide/
  component.app-carousel-slide";
```

**21. Add carousel and slides**

Finally, let's add our new carousel component and some slides to the main app. Still in `app.vue`, replace the 'Under Construction' with the snippet below. Each `<app-carousel-slide>` element represents a single slide. Add as many as you like, replacing the text or image where desired. Included are `test-photo-01.jpg` to `test-photo-05.jpg`. Once that's finished compiling, everything should now work. Huzzah!

```
<app-carousel>
  <app-carousel-slide>
  <template slot="title">My Slide
  </template>
  <template slot="text">
  <p>This is a carousel slide.</p>
  <app-button>Let's Go</app-button>
  </template>
  
</app-carousel>
```

**22. Add flexibility to the carousel**

Now we have a working Vue.js carousel, let's add some additional functionality so we can easily customise the

## Developer tutorials

### Create a flexible carousel component

```
<app-carousel>
  <app-carousel-slide>
    <template slot="title">My Carousel</template>
    
  </app-carousel-slide>
</app-carousel>
```

#### The simplicity of components

A major strength of creating components in Vue.js is that as a developer you should rarely need to know how a component is structured in terms of its inner HTML or CSS when using it within an app. Since everything is encapsulated within a custom element (For example: `<appcarousel>`), you only need know about any data you may pass to it via properties or slots, which should be documented in its readme.

A native equivalent for components has been making its way into browsers over the years as part of Web Components. We're still not quite there yet as many browsers still don't support this functionality (Edge and Safari for instance), but the future is looking increasingly component oriented.

duration of the slide transition, its timing property, declaring if slides should auto-slide, and if so how often. Reopen 'app-carousel.vue' and add the properties in the snippet below to the component object.

```
props: {
  transitionDuration: {
    type: String,
    default: '0.5s'
  },
  transitionTiming: {
    type: String,
    default: 'ease'
  }
}
```

#### 23. Amend containerStyle0

The values passed to these properties should make their way to the carousel's inline CSS from back in Step 17. Now let's amend the 'containerStyle' computed function to make sure that this happens.

```
containerStyle() {
  return `width:${this.containerWidth};
  transform:
  translateX(${this.activeSlidePosition});
  transition-timing-function:
  ${this.transitionTiming};
  transition-duration:
  ${this.transitionDuration};`
}
```

#### 24. Pass data to the carousel properties

The below snippet illustrates how we would pass data to these new properties to the `<app-carousel>` element in 'App.vue'. Once added, you should be able to pass

whatever values you wish. For example, a "3.0s" duration would result in a very slow slide transition!

```
<app-carousel
  transition-duration="0.25s"
  transition-timing="ease-in-out">
```

#### 25. Adding auto-slide props

For auto-slide, we need to add two additional objects to 'props' in 'app-carousel.vue'. 'auto' is either 'true' or 'false', which corresponds to if the carousel should continue to auto-slide. 'autoTiming' controls the time before auto-slide triggers, with the default value being 5000 milliseconds.

```
auto: {
  type: String,
  default: 'false'
},
autoTiming: {
  type: String,
  default: 5000
}
```

#### 26. Initiate auto-slide

Now we need to initiate auto-slide on component load. In the carousel's 'mounted()' function, after the existing content, check if the 'auto' property is set to 'true'. If so, create an interval which triggers the 'nextSlide()' method repeatedly once the 'autoTiming' value has passed.

```
if(this.auto === 'true') {
  this.autoInterval = setInterval(() => {
    this.nextSlide();
  }, parseInt(this.autoTiming));
}
```

#### 27. Cancel auto-slide method

Obviously, we need some way for the user to disable

auto-slide if they have expressed a desire to operate the carousel manually. The first step towards this is a new carousel method called 'cancelAutoSlide'. This will simply cancel the interval created in the previous step.

```
cancelAutoSlide() {
  clearInterval(this.autoInterval);
}
```

#### 28. Trigger cancelAutoSlide

If the user clicks an arrow or control element, it's reasonable to assume that they wish to operate the carousel manually, so let's call the 'cancelAutoSlide' method if any of these elements are clicked. To do this, simply add '+ cancelAutoSlide()' to each elements 'v-on' directive. See the snippet below for an example using the 'previous' slide arrow.

```
v-on:click.native="prevSlide() +
cancelAutoSlide()"
```

#### 29. Pass values to the auto-slide props

Finally, let's pass some values to the auto-slide properties we've created. Back in 'app.vue', add the below snippet to the `<app-carousel>` element to enable an auto-slide every three seconds.

```
auto="true" auto-timing="3000"
```

#### 30. Finish building the carousel component

Cancel the development server, or open a new terminal window, and run 'npm run build' to create a compiled, production-ready version of your Vue.js powered carousel component in the 'dist' directory.

OFFER ENDS  
31 DECEMBER 2017

# SAVE UP TO 49% ON THE PERFECT GIFT THIS CHRISTMAS



FROM £21 EVERY 6 MONTHS  
(€69.90 / \$77.40 PER YEAR)



FROM £21 EVERY 6 MONTHS  
(€85 / \$85 PER YEAR)



FROM £23 EVERY 6 MONTHS  
(€103 / \$105 PER YEAR)



FROM £24 EVERY 6 MONTHS  
(€81 / \$113 PER YEAR)



FROM £16.80 EVERY 6 MONTHS  
(€78 / \$108 PER YEAR)



FROM £25 EVERY 6 MONTHS  
(€89 / \$89 PER YEAR)



GUIDES & SPECIALS, BACK ISSUES AND GIFT  
VOUCHERS WILL MAKE GREAT GIFTS TOO!

Delivery included in the price

Free personalised e-card when buying for someone else

Buy as a gift or treat yourself!

Choose from a huge range of titles

SEE THE FULL RANGE AND ORDER ONLINE

[www.myfavouritemagazines.co.uk/xmas17](http://www.myfavouritemagazines.co.uk/xmas17)

ORDER HOTLINE: 0344 848 2852

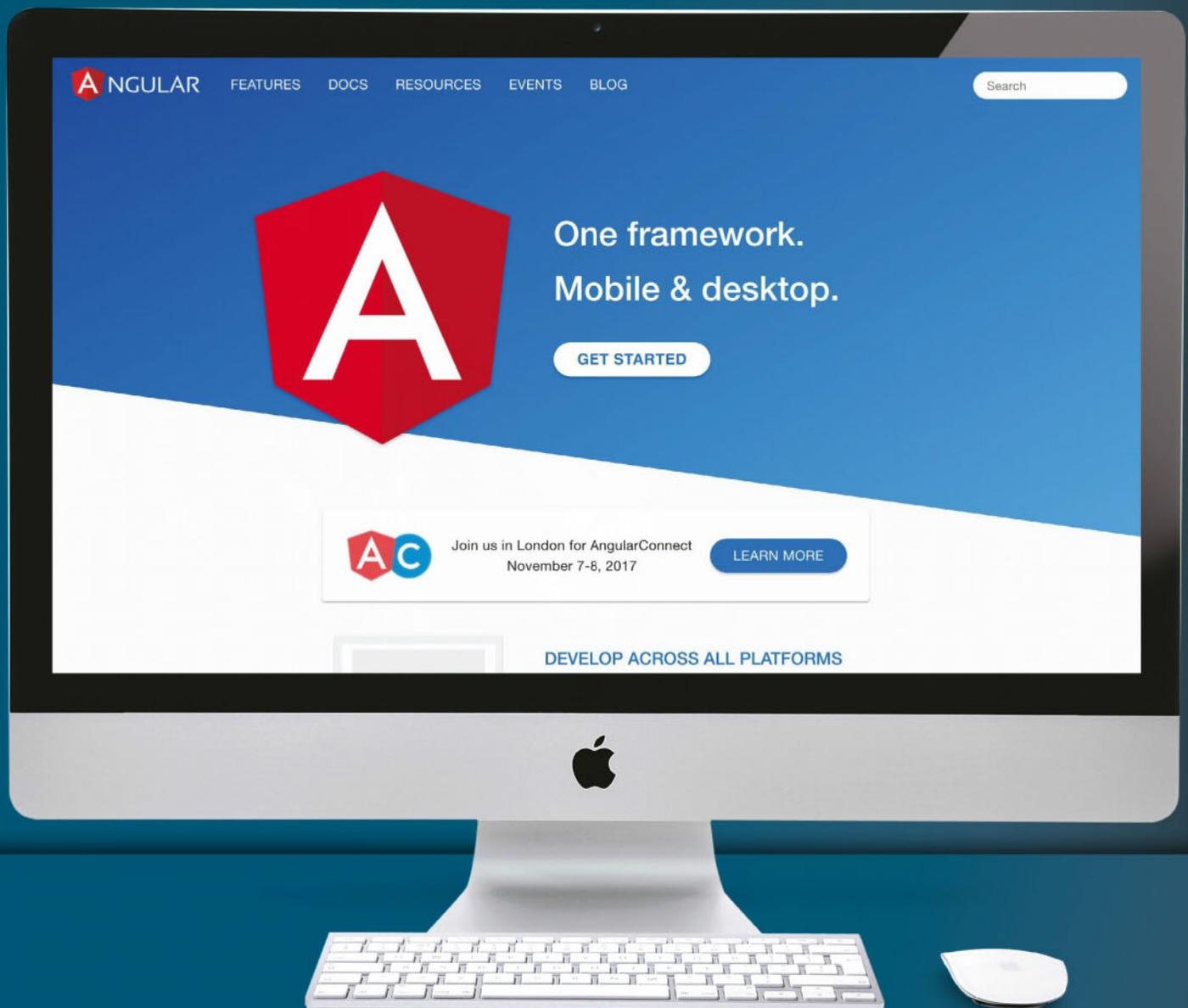
PLEASE QUOTE XMAS17 WHEN ORDERING BY PHONE

LINES ARE OPEN MONDAY - FRIDAY 8AM TO 7PM AND SATURDAY 10AM TO 2PM (GMT)

Terms and conditions: Savings calculated against the full RRP (single issue price x frequency). Dollar prices quoted are for the United States, other global territory dollar pricing may vary. This offer is for new subscribers only. You can write to us or call us to cancel your subscription within 14 days of purchase. Your subscription is for the minimum specified and will expire at the end of the current term. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available on request. For full term and conditions please visit: bit.ly/magtdnc. Offer ends 31st December 2017.

# Angular 5: What's new?

Google's recently-released latest version of Angular bestows a large variety of features upon the developer community





Getting AngularJS 5.0 ready turned out to be quite the challenge: The developer team went through a total of ten (!!!) release candidates before being

sufficiently satisfied with the results of their work.

Modifications to the new version centre around three tenets: Increasing the speed of resulting programs; reducing the bandwidth demands; and softening the (often shockingly steep) learning curve.

Furthermore, the Angular team at Google has used this opportunity to reflect updates done at some of its dependencies back into the main program. By far the most important change involves replacing the TypeScript compiler: Angular 5 uses TypeScript 2.4, which – in theory – would be worthy of an article of its own.

Older versions of Angular often brought a multitude of breaking changes, causing developers to delay updates. With the latest version, however, relatively few modifications of significance have been made – for most developers, getting the upgrade done should be a question of minutes or hours.

In short, there are a lot of new things to see. This tutorial will take a step-by-step approach to showing you some of the highlights: As always, further information is hidden away in a variety of boxouts.

## 1. Accelerated validation

Validating user input is important. Sadly, passing complex parameters into `updateOn` can be dangerous – complex validation criteria risk slowing down the application as a whole. The `updateOn` property enables you to specify which DOM events trigger an update (`blur`/`submit`), thereby saving processing resources.

```
<input type="text" name="userName"
  ng-model="user.name"
  ng-model-options="{ updateOn: 'blur' }"
  ng-keyup="cancel($event)" />
```

## 2. Use the Boost!

Creating Angular projects by hand is tedious. Fortunately,

`preserveWhitespaces?: boolean`

If `Component.preserveWhitespaces` is set to `false` potentially superfluous whitespace characters (ones matching the `\s` character class in JavaScript regular expressions) will be removed from a compiled template. This can greatly reduce AOT-generated code size as well as speed up view creation.

Current implementation works according to the following rules:

- all whitespaces at the beginning and the end of a template are removed (trimmed);
- text nodes consisting of whitespaces only are removed (ex.: `<button>Action 1</button> <button>Action 2</button>` will be converted to `<button>Action 1</button> <button>Action 2</button>` (no whitespaces between buttons);
- series of whitespaces in text nodes are replaced with one space (ex.: `<span>\n some text\n</span>` will be converted to `<span> some text </span>`);
- text nodes are left as-is inside HTML tags where whitespaces are significant (ex. `<pre>`, `<textarea>`).

the latest version of the Node.JS command line tools support Node.JS right out of the box. If the self-reflection of your version of the AngularCLI looks like the one shown in the screenshot, all is ready to go. If not, simply use NPM to order an update (or a re-install).

```
npm uninstall -g angular-cli
npm cache clean
npm install -g @angular/cli@latest
```

## 3. Ignore issues

A pretty odd problem occurs in relationship to the versioning output: In some cases, the self reflection feature shows outdated version information. Fortunately, the package.json file of generated project skeletons is usually perfectly up to date:

```
"dependencies": {
  "@angular/animations": "^5.0.0",
  "@angular/common": "^5.0.0",
  "@angular/compiler": "^5.0.0",
  "@angular/core": "^5.0.0",
  "@angular/forms": "^5.0.0",
  "@angular/http": "^5.0.0",
  "@angular/platform-browser": "^5.0.0",
  "@angular/platform-browser-dynamic":
  "^5.0.0",
  "@angular/router": "^5.0.0",
  "core-js": "^2.4.1",
  "rxjs": "^5.5.2",
  "zone.js": "^0.8.14"
},
```

## 4. Farewell, HttpClient

Firing off REST requests is the daily bread of web developers – Angular 5.0 deprecates the `HttpModule`, and brings in the `HttpClientModule` instead. Start out by editing `app.module.ts` to include the modules mentioned in the piece of code below.

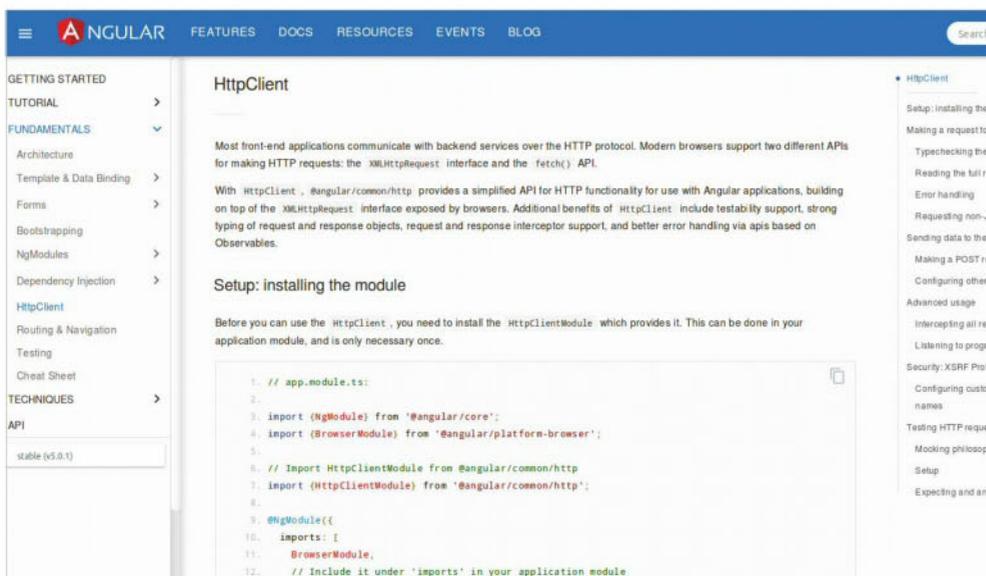
```
import {NgModule} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {HttpClientModule} from '@angular/common/http';
@NgModule({
  imports: [
    BrowserModule,
    HttpClientModule,
  ],
})
export class AppModule {}
```

## 5. Harness its power

The actual API of the new HTTP interface does not show significant differences to the old one: the most significant change is that some processes have been streamlined, removing the intermediary steps required for accessing data returned from the server.

## Avoid deprecated APIs

Angular's devs traditionally eliminate deprecated APIs in the next revision – if your program uses any of them, better start refactoring early.



**Left**  
Angular's documentation for the new HTTP client is a first-class example for how classes should be explained.

**Above**  
As of this writing, Angular CLI 1.5 is the latest version – it, furthermore, matches Angular 5.

## Developer tutorials

# Angular 5: What's new?

```
export class MyComponent implements OnInit {
  results: string[];
  constructor(private http: HttpClient) {}
  ngOnInit(): void {
    this.http.get('/api/items').subscribe(data
=> {
  this.results = data['results'];
});
}
}
```

## 6. Eliminate whitespaces!

Whitespace in templates is an issue as old as the existence of template engines: should you eliminate them, or should they be left in place? The easiest way to solve the problem is enabling the option in `tsconfig.js` – just bear in mind that it can negatively affect the performance of some DOM transforms.

```
“angularCompilerOptions”: {
  “preserveWhitespaces”: false
}
```

## 7. Protect your DOM ops

Due to the risk of DOM selectors being affected negatively, the `preserveWhitespaces` property can also be disabled on a per-component fashion. Just like with the global toggle, bear in mind that setting the property to false enables the stripping process.

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
  preserveWhitespaces: false
})
```

# Internationalisation is important

Failure to internationalise programs can have disastrous effects: in many cases, simply adding another locale enables you to open new markets not inhabited by competitors.

## 8. New Router events

Router classes act as the “hubs” of an Angular application: Thus, they are in a unique position to contain all kinds of logic related to keeping the application lifecycle in check. From now on, look forward to an additional eight events which enable you to tie into the framework more closely.

```
ActivationStart
ActivationEnd
ChildActivationStart
ChildActivationEnd
GuardsCheckEnd
GuardsCheckStart
ResolveStart
ResolveEnd
```

## 9. Beware of changes

While the eight new events might be fully implemented, be aware that their design is not finalised. For example, the API documentation for `ResolveStart` contains the – badly-sounding – warning shown in the figure accompanying this step.

```
class ResolveStart extends
RouterEvent”RouterEvent {
  \l “constructor”constructor(id: number,
url: string, urlAfterRedirects: string,
```

```
state: RouterStateSnapshot)
```

```
\l “urlAfterRedirects”urlAfterRedirects:
string
\l “state”state: RouterStateSnapshot
\l “toString”toString(): string
. . .
```

## 10. Compile in advance

When working with `ng build` or `ng serve`, passing in the `--aot` flag enables an AoT compiler, which reduces browser workload. As of Angular CLI 1.5, this compiler has been enhanced to provide additional support during the development process. The new CLI also implements the tree shaking process to reduce build sizes.

## 11. Problem avoidance, Part 1

Angular’s internationalisation features traditionally used the internationalisation API of the browser used to run the web application. Sadly, this process brought all kinds of interesting aspects to light: each vendor, after all, deploys his or her own internationalisation style.

## 12. Get locale information

The new version of Angular.JS sidesteps this problem by using data found in the Unicode Common Locale Data

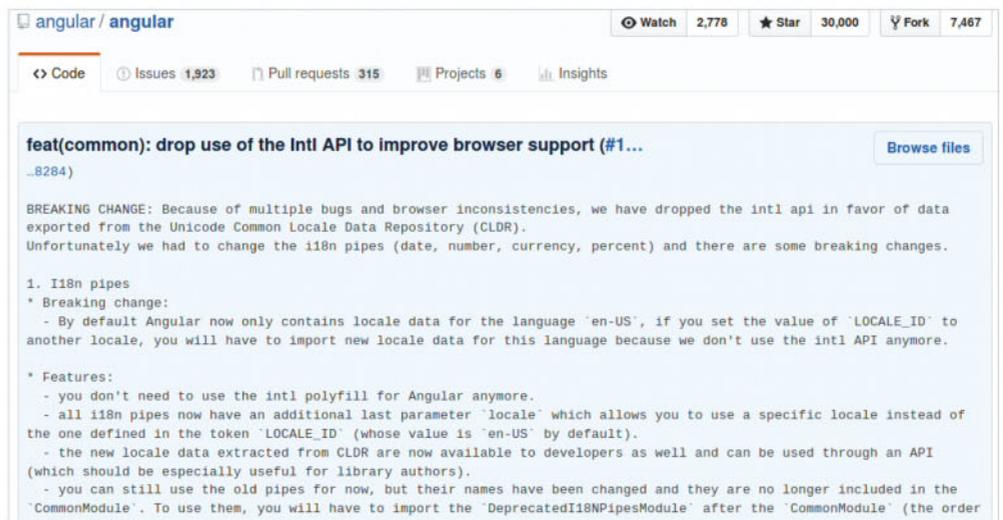


### Above

Always read the documentation carefully: The Angular team at Google reserves the right to change the way this event is handled.

### Right

The blue commit message of doom warns developers of severe impending changes to their work environment.



angular / angular

Watch 2,775 Star 29,995 Fork 7,465

Code Issues 1,923 Pull requests 314 Projects 6 Insights

Branch: master angular / CHANGELOG.md Find file Copy path

troyl docs(changelog): remove unnecessary revert item (#20303) dsdbef1 4 days ago

69 contributors and others

2365 lines (1632 sloc) 235 KB Raw Blame History

### 5.1.0-beta.0 (2017-11-08)

#### Bug Fixes

- **compiler:** don't overwrite missingTranslation's value in JIT (#19952) (799cbb9)
- **compiler:** report a reasonable error with invalid metadata (#20062) (da22c48)
- **compiler-cli:** don't report emit diagnostics when --noEmitOnError is off (#20063) (8639995)
- **core:** \_\_symbol\_\_ should return \_\_zone\_symbol\_\_ without zone.js loaded (#19541) (678d1cf)
- **core:** should support event.stopImmediatePropagation (#19222) (7083791)
- **platform-browser:** support Symbols in custom jasmineToString() method (#19794) (5a6efa7)

### An in-detail analysis

Should you feel like finding out more about the changes in AngularJS, visiting <https://github.com/angular/angular/blob/master/CHANGELOG.md> will reveal detailed technical information. A neat hint involves entering the string "BREAKING" into the search function of your browser - that way, you can easily jump to changes which require attention from developers maintaining existing Angular-based solutions. The links next to the various bullet points allow you to access additional information with minimal effort: in many cases, the development of features or breaking changes is preceded by long-running discussions to explain the motivations.

Repository. Unfortunately, deploying these new – and standardised – locale classes requires some changes in the application structure.

```
import { registerLocaleData } from '@angular/common';
import localeDe from '@angular/common/locales/de';
registerLocaleData(localeDe);
```

### 13. Parser optimisations

Traditional internationalisation comments provided a way for developers to add translation related markup to their web pages. Sadly, finding them required quite a bit of parsing intelligence. The Angular team has therefore deprecated them – use the ng-container instead.

```
<ng-container i18n></ng-container>
```

### 14. Advanced pipe-ing

Angular's internationalisation pipes enable you to send data for "localisation". With Angular 5.0, their behaviour receives significant changes. For example, developers working on currency values can now access a wide group of additional behaviours, as outlined in the code that you'll find here:

```
<p>{{ 10.7 | currency:'CAD' }}</p>
<!-- 'CA$10.70' -->
<p>{{ 10.7 | currency:'CAD': 'symbol-narrow' }}</p>
<!-- '$10.70' -->
<p>{{ 10.7 | currency:'CAD': 'code': '.3' }}</p>
<!-- 'CAD10.700' -->
```

### 15. Pipe terrorism

While the changes to the currency pipeline are somewhat manageable, users of the date pipeline will need to live with potentially breaking changes. Fortunately, the old pipes are retained in the DeprecatedI18NPipesModule to help soften the blow.

```
@NgModule({
  imports: [CommonModule,
    DeprecatedI18NPipesModule],
})
export class AppModule {
```

### 16. Mainlining @angular/service-worker

ServiceWorkers are nothing new, per se: These "mini-proxies" have been in beta stage for quite some time. The main change in AngularJS 5 is that ServiceWorkers have finally graduated to mainline – if you have a lot of static assets, look at implementing a ServiceWorker to reduce the transfer bandwidth.

### 17. Multi-named exports

Up until now, a 1:1 relationship has been established

README.md

build: passing circleci: passing BrowserStack: completed gitter: join chat Issue Stats Issue Stats npm package: 5.0.2

	Android	Firefox	Chrome	IE	iPhone	Edge
4.4	54	7	60	7	9	7
5.1			b	7	10	8
6.0			d	7	11	8.1
7.1						

Safari (7+), iOS (7+) and IE mobile (11) are tested on BrowserStack.

## Angular

Angular is a development platform for building mobile and desktop web applications using TypeScript/JavaScript and other languages.

### Quickstart

Get started in 5 minutes.

# Developer tutorials

## Angular 5: What's new?

The screenshot shows the TypeScript Playground interface. At the top, there's a navigation bar with 'Quick Start', 'Documentation', 'Download', 'Connect', and 'Playground'. Below that, a message says 'TypeScript 2.6 is now available. Download our latest version today!'. The main area is split into two panes. The left pane shows code using ES6 class syntax:

```
1 class Greeter {
2   greeting: string;
3   constructor(message: string) {
4     this.greeting = message;
5   }
6   greet() {
7     return "Hello, " + this.greeting;
8   }
9 }
10
11 let greeter = new Greeter("world");
12
13 let button = document.createElement('button');
14 button.textContent = "Say Hello";
15 button.onclick = function () {
16   alert(greeter.greet());
17 }
18
19 document.body.appendChild(button);
```

The right pane shows the equivalent code using ES5 function syntax:

```
1 var Greeter = /** @class */ (function () {
2   function Greeter(message) {
3     this.greeting = message;
4   }
5   Greeter.prototype.greet = function () {
6     return "Hello, " + this.greeting;
7   };
8   return Greeter;
9 }());
10 var greeter = new Greeter("world");
11 var button = document.createElement('button');
12 button.textContent = "Say Hello";
13 button.onclick = function () {
14   alert(greeter.greet());
15 };
16 document.body.appendChild(button);
17
```

### TypeScript without Angular

AngularJS traditionally lags TypeScript's development process by at least one, if not two versions. Developers who feel like experimenting with the latest features of the programming environment can visit [typescriptlang.org/play/](https://www.typescriptlang.org/play/) – it provides an in-browser-hosted version of the TypeScript transpiler. Alternatively, you can also set up a full-blown local TypeScript environment using a version of the interpreter closely matching your Angular environment: not being familiar with the underlying programming language is a mistake independent of the framework you use.

between components and names. The new version breaks this up – having multiple names can be helpful during upgrading procedures, or in extremely complex program architectures, for example.

```
@Component({
  . . .
  exportAs: 'matButton, matAnchor',
})
```

### 18. RxJS updated

AngularJS uses the ReactiveX library in the background. Just as with TypeScript, the reflection of updates can take some time – but RxJS has now been updated to version 5.5. This leads to slightly shorter code: The main benefit is that the new version simplifies compilation.

### 19. Internal restructuring

Reducing program coupling is always beneficial – Eric Sink's observations remain valid even though his legendary book on event-based programming has faded away. The Angular team has addressed this problem through an internal restructuring: the compiler package is no longer part of the core, which will help reduce the size of compiled programs.

### 20. Advanced animation triggering

AngularJS always supported developers with a wide range of animation facilities. The fifth version of the framework enhances this by including support for numeric triggers addresses – an animation can now be triggered if values change.

```
@Component({
  animations: [
```

```
trigger("counter", [
  transition(':increment', [ /*...*/ ]),
  transition(':decrement', [ /*...*/ ]),
])
```

### 21. Follow the upgrade guide

While this tutorial mentions quite a few changes and break points, it can not encompass everything. Fortunately, the AngularJS team provides an overview of the various changes – if you have existing code, definitely look at [angular-update-guide.firebaseio.com](https://angular-update-guide.firebaseio.com).

The screenshot shows the RxJS website. The top navigation bar includes 'Home', 'Manual', 'Reference', 'Source', and 'Test'. The main content area features the RxJS logo and the text 'The ReactiveX library for JavaScript.' Below this, there are sections for 'Install it', 'Learn it', and 'Full reference'. A sidebar on the left lists various RxJS classes and operators, including AsyncSubject, BehaviorSubject, Notification, Observable, ReplaySubject, Scheduler, AnonymousSubject, Subject, SubjectSubscriber, Subscriber, Subscription, ObservableInput, Observer, SubscriptionOrPromise, TeardownLogic, Rx.Scheduler, Rx.Symbol, ConnectableObservable, AjaxError, AjaxResponse, AjaxTimeoutError, AjaxRequest, GroupedObservable, Action, animationFrame, asap, async, and queue.



# Discover another of our great bookazines

From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes



Get great savings when you buy direct from us



1000s of great titles, many not available anywhere else



World-wide delivery and super-safe ordering



[www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
Magazines, back issues & bookazines.



## Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832

# HOSTING LISTINGS



## Featured host: tidyhosts

www.tidyhosts.com / 05603 674610

### About us

Tidyhosts has become a leading cloud hosting provider throughout the world. It boasts a feature-rich selection of services including domain registration, shared web hosting, WordPress hosting, hosted exchange email, cloud virtual servers and media streaming. Its passion and drive for

success has gained it a highly reliable and trusted reputation from its customers, making it the number one for hosting services. Founded in 2004 by three developers who wanted to build hosting infrastructure for developers - years on, the company has expanded to offer much more.

### What we offer

- **Domain names** - Simple domain registration with a large choice of TLDs
- **Shared web hosting** - Includes one-click application installers and a choice of Windows or Linux
- **Cloud virtual servers** - Take full control over your hosting, and install the software you need on your server
- **SHOUTcast hosting** - Start your own radio station with our easy to set up and use SHOUTcast hosting service

### 5 Tips from the pros

#### 1. Ensure that you choose the right domain

When choosing your domain, make sure it's easy to remember and resembles what you are offering. Customers are more likely to come back if they can remember your URL.

#### 2. Create clear, concise website content

Keep the content on your website clear, informative and, more importantly, relative! Avoid any duplication of content on different pages as this can affect your rankings on search engines.

#### 3. Utilise SSL certificates to stay secure

More sites are moving to SSL security

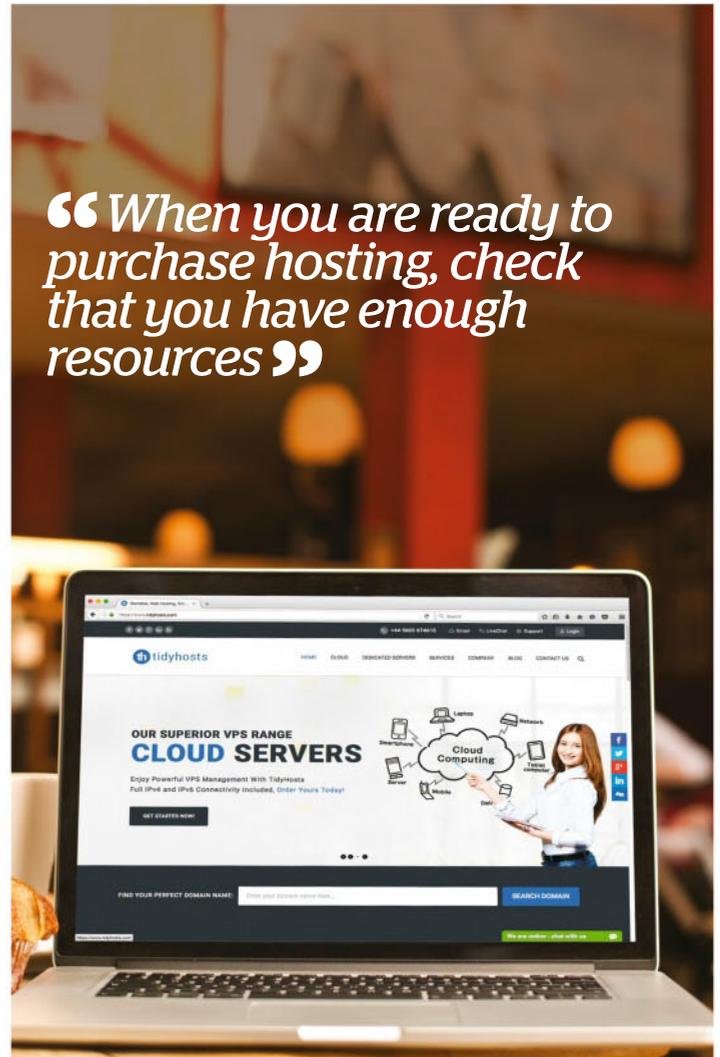
to protect their customers from the ever-increasing threats on the web. Search engines are now ranking sites with an SSL higher than those without.

#### 4. Make sure you choose the right plan

When you are ready to purchase hosting, check that you have enough resources, especially if you expect your website to grow quickly. Seek advice from tidyhosts if you are unsure.

#### 5. Use one-click installers when building a website

If you are new to building a website then we have a number of useful one-click application installers. These help you get up and running, including the popular WordPress system.



“ When you are ready to purchase hosting, check that you have enough resources ”

### Testimonials

#### Kelly Underwood

“ I've been with a number of web host providers in the past which have at some point let me down. I now realise the importance of using a host that is well known ”

#### John Corey

“ We have our email hosted with tidyhosts, which serves a small number of users in our office. We have found this solution is much more cost-effective ”

#### Jenny Brice

“ A great host with fantastic knowledge. I have only had to use the support channels a few times, but the replies I have had helped me instantly, so well worth it ”



## Get your listing in our directory

To advertise here contact Chris

[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

+44 (0)1225 687832

### Supreme hosting



[www.cwcs.co.uk](http://www.cwcs.co.uk)  
08001777000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and are passionate about what they do.

- Co-location hosting
- VPS
- 100% Network uptime

### UK-based hosting



[www.cyberhostpro.com](http://www.cyberhostpro.com)  
08455279345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow, then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

### Cluster web hosting



[www.fasthosts.co.uk](http://www.fasthosts.co.uk)  
08081686777

UK based and operating 24/7 from dedicated UK data centres. Fasthosts keep over one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



### Budget hosting



[www.hetzner.com](http://www.hetzner.com)  
+49 (0)9831505-0

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997, the company has provided private and business clients

with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position both nationally & internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



### All-inclusive hosting



[www.1and1.co.uk](http://www.1and1.co.uk)  
0333 336 5509

1&1 Internet is a leading hosting provider that enables businesses, developers and IT pros to succeed online. Established in 1988, 1&1 now

operates across ten countries. With a comprehensive range of high-performance and affordable products, 1&1 offers everything from simple domain registration to award-winning website building tools, eCommerce packages and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

### SSD web hosting



[www.bargainhost.co.uk](http://www.bargainhost.co.uk)  
0843 289 2681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

### Value Linux hosting



[patchman-hosting.co.uk](http://patchman-hosting.co.uk)  
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven eCommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

### Flexible cloud servers



[elastichosts.co.uk](http://elastichosts.co.uk)  
020 7183 8250

ElasticHosts offer simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Their team of engineers provide excellent support 24/7 over the phone, email and ticketing system.

- Cloud servers with any OS
- Linux OS containers
- 24/7 expert support



## Get your listing in our directory

To advertise here contact Chris

chris.mitchell@futurenet.com

+44 (0)1225 687832

# COURSE LISTINGS



Featured:

## Makers Academy

www.makersacademy.com

Twitter: @makersacademy

Facebook: MakersAcademy

### About us

Makers Academy is a fully immersive, 12-week computer programming boot camp. With their help, you will learn the principles of software craftsmanship and they'll also help you get your first job. They're Europe's number-one developer boot camp, running highly selective classes of the offline course every six weeks, and a

remote course every 12 weeks. They take a 'learn by doing' approach, through project-based work. Students are encouraged to work in pairs on coding challenges, with weekly tests, culminating in a final project. They help set up job interviews via their network of hiring partners including ThoughtWorks and Deloitte Digital.

### What we offer

#### • On-site:

12 week full-time coding course from the on-site campus in London

#### • Remote:

12 week full-time coding course remotely from home

### 5 tips from the pros

#### 1. Research all your options

Do your research into lots of different boot camps, read the reviews, read the student blogs and reach out to previous graduates and speak to them.

#### 2. Dabble in code

Although the course is for beginners, it's important that you've started to at least try to learn to code on your own.

#### 3. Prepare for the interview

We send you everything to prepare for the interview. Make

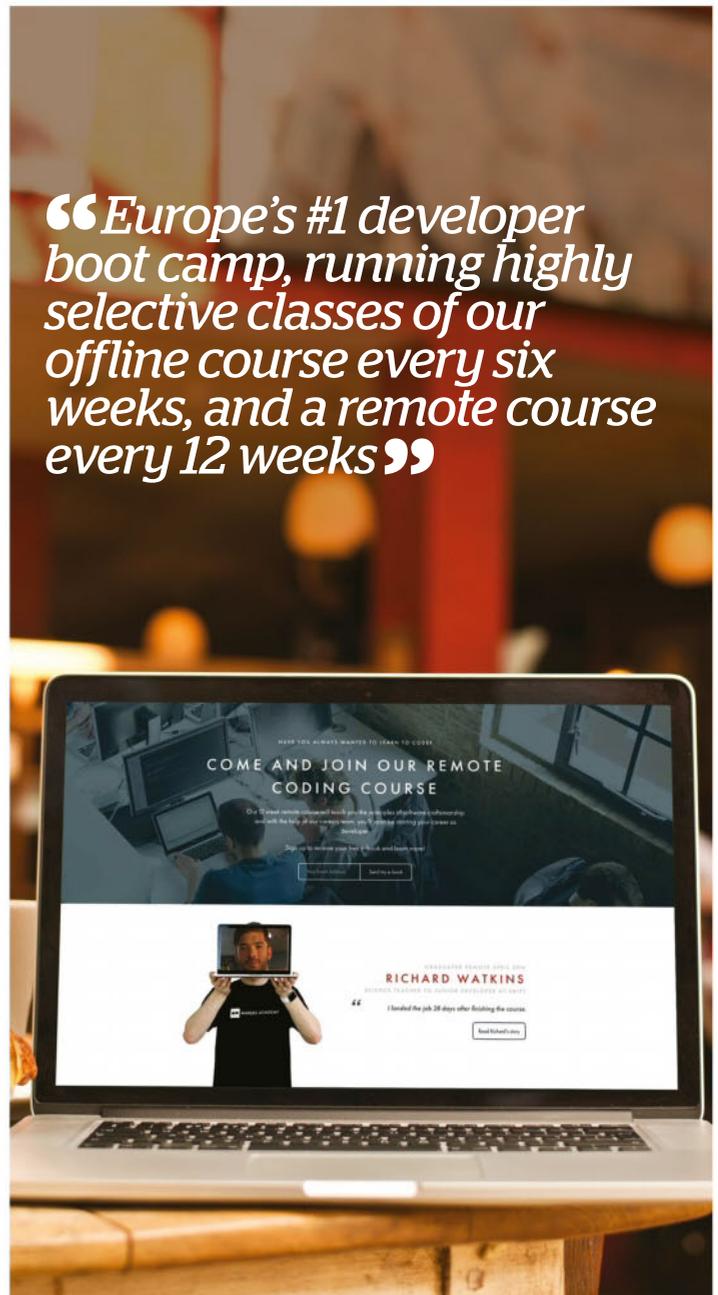
sure you go through all the resources and give yourself two weeks to prepare.

#### 4. Budget

The course is full-time for three months and it can take up to three months after to secure a job. It's important to financially plan for the period you won't be working.

#### 5. Visit us!

Book a visit and come visit us! Come see the Makers Academy HQ in person and learn more.



“Europe's #1 developer boot camp, running highly selective classes of our offline course every six weeks, and a remote course every 12 weeks”



#### Richard Watkins

Science teacher to junior developer at Shift

Makers Academy was frustrating and daunting but amazing and I wouldn't change any of it. I landed the job 28 days after finishing the course.



#### Ina Tsetsova:

Email campaign manager to graduate software developer at ThoughtWorks

I found a really nice community and I've met really cool people. I got a job quicker than I expected.



#### Hannah Carney

3D designer to junior developer at Play Consulting

Makers Academy not only focuses on your learning for code, but they also focus on your well-being. Work feels like fun and I've finally found a job I love.



## Get your listing in our directory

To advertise here contact Chris

[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

+44 (0)1225 687832



### UDEMY

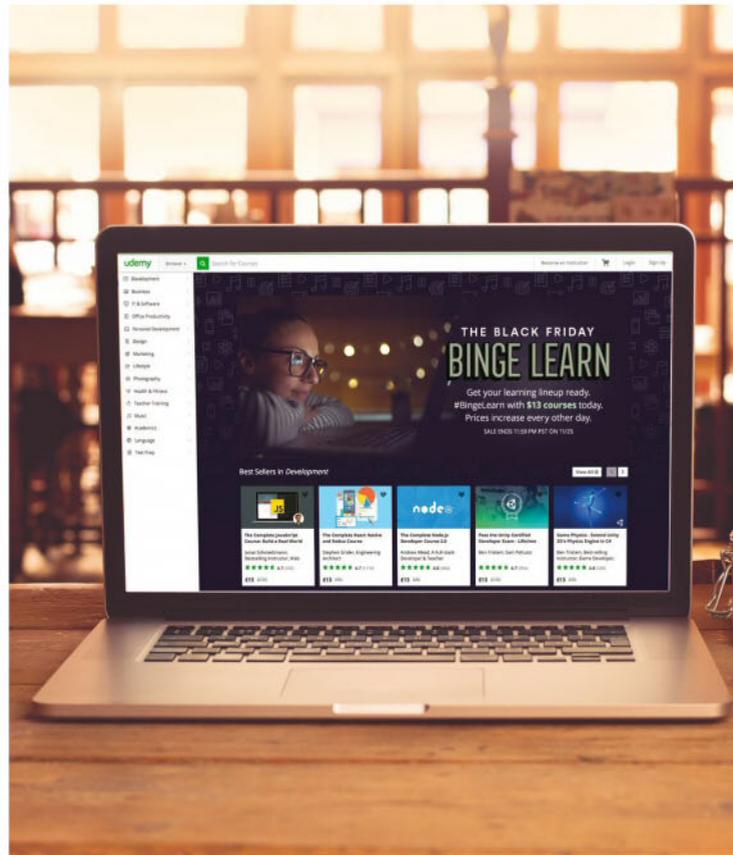
[www.udemy.com](http://www.udemy.com)

Twitter: @udemy

Facebook: udemy

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet, he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses including HTML, CSS, UX, JavaScript and web development.

**40,000+ courses:** There is a course for every designer and dev  
**Self-paced learning:** Learn how to code at your own pace



### WE GOT CODERS



[www.wegotcoders.com](http://www.wegotcoders.com)  
[hello@wegotcoders.com](mailto:hello@wegotcoders.com)

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training course that covers all that is required to become a web developer, with highly marketable full-stack web development skills.

- Classroom-based training
- Real-world work experience
- Employment opportunities

### FUTURELEARN



[www.futurelearn.com](http://www.futurelearn.com)  
[feedback@futurelearn.com](mailto:feedback@futurelearn.com)

Choose from hundreds of free online courses: from Language & Culture to Business & Management; Science & Technology to Health & Psychology. Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access



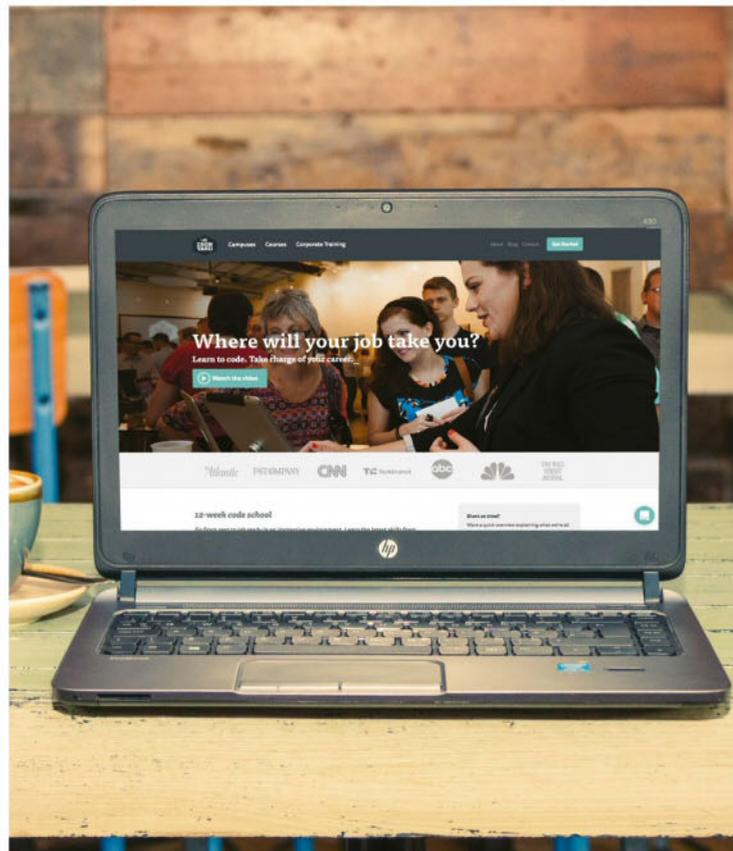
### THE IRON YARD

[www.theironyard.com](http://www.theironyard.com)

Twitter: @TheIronYard  
Facebook: TheIronYard

The Iron Yard is one of the world's largest and fastest-growing in-person code schools. It offers full-time and part-time programs in Back-End Engineering, Front-End Engineering, Mobile Engineering and Design. The Iron Yard exists to create real, lasting change for people, their companies and communities through technology education. The in-person, immersive format of The Iron Yard's 12-week courses helps people learn to code and be prepared with the skills needed to start a career as junior-level software developers.

**12-week code school:** Learn the latest skills from industry pros  
**Free crash courses:** One-night courses, the perfect way to learn



### GYMNASIUM



[www.thegymnasium.com](http://www.thegymnasium.com)  
[help@thegymnasium.com](mailto:help@thegymnasium.com)

Gymnasium offers free online courses designed to teach creative professionals in-demand skills. Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

- Gain real-world skills
- Get expert instruction
- Career opportunities

# Free with your magazine

Instant access to these creative resources...

## Essential assets and resources

Get textures, fonts, backgrounds and more



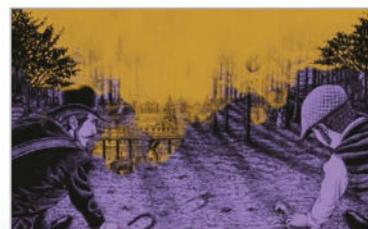
## Exclusive video tutorials

Learn to code/create with HTML, CSS, JS & PHP



## Tutorial project files

All the assets you'll need to follow our tutorials



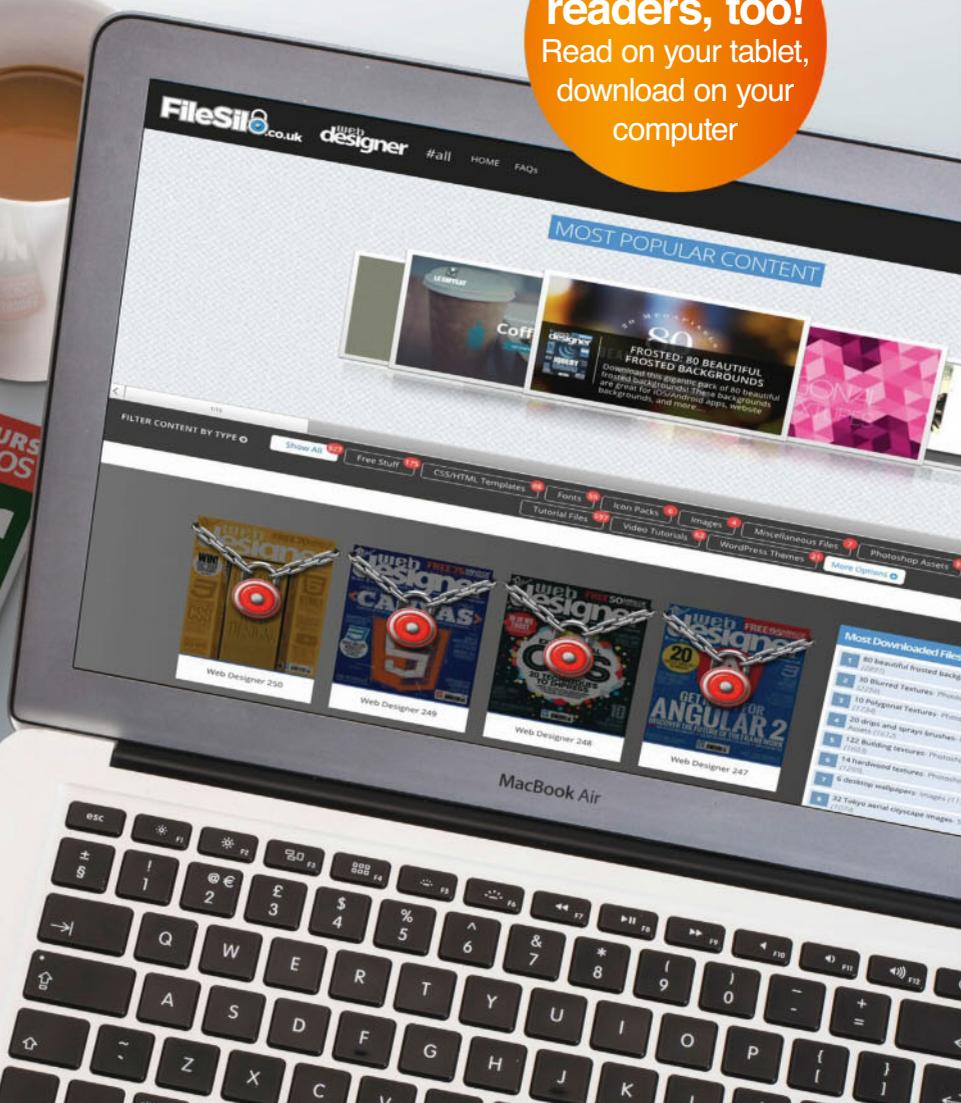
## Plus, all of this is yours too...

- All-new tutorial files to help you master this issue's HTML, CSS and JavaScript techniques
- Over 5 hours of expert JavaScript video from Pluralsight ([www.pluralsight.com](http://www.pluralsight.com))
- 90 Gradient triangulate backgrounds ([www.sparklestock.com](http://www.sparklestock.com))
- Real Photo product box mockups

Log in to [www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

Register to get instant access to this pack of must-have creative resources, how-to videos and tutorial assets

Free for digital readers, too!  
Read on your tablet, download on your computer



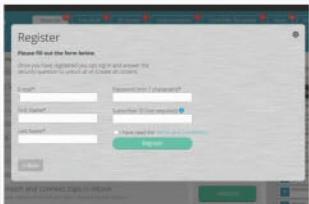


The home of great downloads – exclusive to your favourite magazines from Future!

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, in one place

## Get started

Everything you need to know about accessing your FileSilo account



**01** Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.



**02** You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.



**03** If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at the address below.  
[filesilohelp@futurenet.com](mailto:filesilohelp@futurenet.com)

# An incredible gift for subscribers



## Subscribe today & unlock the free gifts from more than 50 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

**Over 60 hours of video guides**

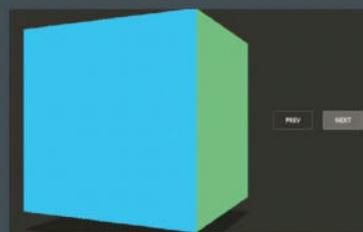
Let the experts teach you to create and code

**More than 400 tutorials**

Get the code you need to get creative

**Over 210 creative assets**

Templates, fonts, textures and backgrounds



## Head to page 32 to subscribe now

**Already a print subscriber? Here's how to unlock FileSilo today...**

Unlock the entire Web Designer FileSilo library with your unique Web ID – the eight-digit alphanumeric code printed above your address details on the mailing label of your subscription copies – also found on any renewal letters.

**More than 900 reasons to subscribe**



# NEXT MONTH

---

# HOT 101

---

A COLLECTION OF MUST-KNOW DESIGN  
& DEVELOPMENT TIPS, TECHNIQUES AND  
INSPIRATION TO MAKE 2018 A SUCCESS

## THE PATH TO CREATIVE CODING

Best data sources and APIs, multiple screen experiences, WebVR and more

## GET STARTED WITH FACEBOOK ORIGAMI

Learn how to start prototyping, previewing and building modern interfaces

## WEB FORMS AND USER EXPERIENCE

Discover the best practices and techniques for creating online forms

Visit the **WEB DESIGNER** online shop at

**Future** myfavouritemagazines  
myfavouritemagazines.co.uk

for the latest issue, back issues and specials

ALL IN YOUR NEXT  
**WEB DESIGNER**  
Issue 270 on sale  
Thursday 11th January 2018

**(mt)**

**DRAW**  

---

**WIRE**  

---

**DESIGN**  

---

**CODE**

Media Temple web hosting delivers premium performance and support so that creative professionals and enterprises can focus on bold ideas, not web servers.

[mediatemple.net](https://www.mediatemple.net)

# Lease an Apple Mac!

## Cheaper than your daily coffee

- Including accidental damage cover
- Upgrade when you want
- Own at the end
- 3 years service



 **iMac**  
with Retina display

21"  
FROM **£2.15** +VAT  
PER DAY



 **MacBook Pro**

13"  
FROM **£2.20** +VAT  
PER DAY

Discover the benefits of leasing computers for business at  
[www.hardsoft.co.uk](http://www.hardsoft.co.uk) Email [info@hardsoft.co.uk](mailto:info@hardsoft.co.uk)

**HARDSOFT**  
Established for over 25 years

 **Authorised Reseller**