

30  
PAGES OF  
EXPERT TUTORIALS

FREE 107 MINUTES  
OF JS VIDEOS

# designer

HTML

CSS

JavaScript

# DESIGN

AWESOME  
AUDIO

Add the power of sound to your projects

# FASTER

PROGRESSIVE  
WEB APPS

Get offline support with Google's Workbox

> 50 must-try tools for today's designers and developers



INTERACTIVE  
IMAGES

ANGULAR STATE  
MANAGEMENT

ES MODULES IN  
JAVASCRIPT

UI TILT EFFECT  
WITH SVG

# CODE QUICKER

- > Frameworks
- > Prototyping
- > Browser extensions
- > IDEs > Code playgrounds
- > Testing > Animation
- > Colour > Fonts

Introducing **STORM**, a hosting platform that makes your agency and websites run smoother. 80% of our customers have seen a 25% improvement in site speeds.

# SUPER CHARGE YOUR HOSTING

**STORM**

Developed by  
 **NimbusHosting**

[nimbushosting.co.uk](http://nimbushosting.co.uk)

# Welcome to the issue

## 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



**Steven Jenkins**  
Editor

## 50 tools to make you better



Every designer and developer needs the right tools in their creative and/or coding arsenal. The pros know all the best tools to use and that's exactly who has chosen our collection for this issue's lead feature. There is a cornucopia of talent that will make your everyday workflow, faster, easier, and, quite simply, better.

Included in the line-up are 50 of the best prototyping and build tools, frameworks, colour tools and fonts, web animation, methodologies and testing, IDEs, code playgrounds and must-try browser extensions. Have a good look through the collection and see what fits your workflow.

Audio and sound are crucial elements if you want a complete online experience. Imagine an app without simple notification pings or a site without music, sound effects and ambient noise. They will survive without sound, but add audio to visuals and you have a complete experience. Richard Mattka explains why and how to use sounds, and implement them with the help of the <audio> tag and the Web Audio API.

Elsewhere, we look at Progressive Web Apps – or PWAs, your choice :-). Find out what Google's Workbox collection of JavaScript libraries has to help make it easier to build web apps which work offline. Plus, we have our regular line-up of top tutorials and industry interviews. Enjoy the issue.

Sounds create the mood and set the pace of an interactive experience. Music enhances the suspense, and ramps up the intensity

## Highlight

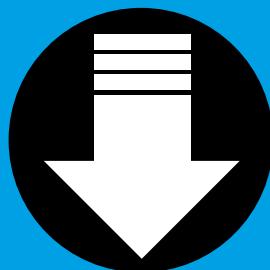


I've seen no innovation in web design in a long time. It's unfortunately become the most boring design medium of all

We fail add a little anarchy to design with their distinctive and striking style. **Web Designer** finds out more. [Page 36](#)

Follow us on Twitter for all the news & conversation [@WebDesignerMag](#)

Visit our blog for opinion, freebies & more [www.creativebloq.com](#)



**FREE - exclusive with this issue**  
**49 Designer resources**

**Video Tuition** - 107 minutes of Beginner's JavaScript video guides from StudioWeb ([www.studioweb.com](#))

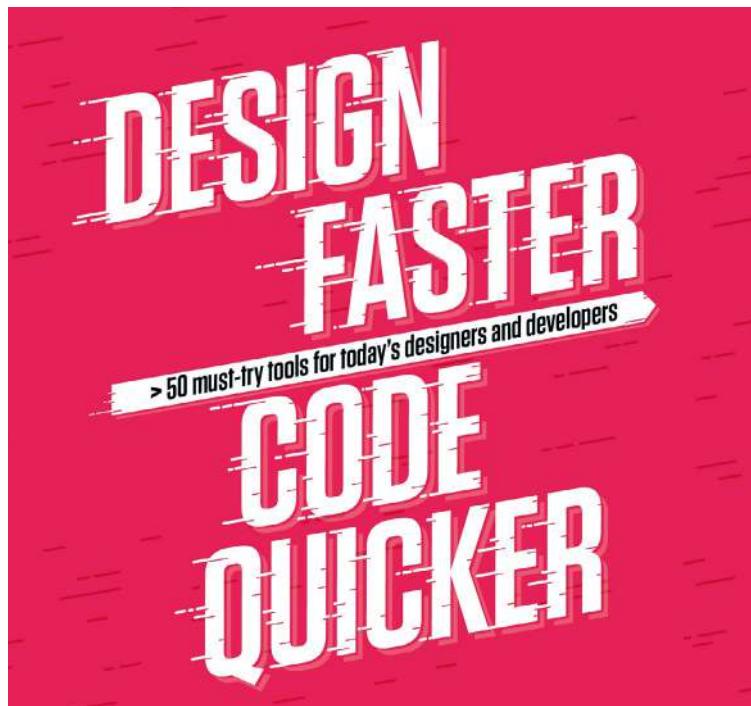
**Assets** - 14 Midnight Photoshop filters and 6 Hi-res postcard and card mockups from Sparklestock ([sparklestock.com](#))  
- Tutorial files and assets



[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...



**Steven Roberts**

Steven is a frontend designer and developer from Stockton-on-Tees. He's currently leading UX, Design and frontend at Asemblr.com. This issue he's taken a deep dive into the tools, frameworks, methodologies, extensions and playgrounds as used by the pros and recommended for you.

Page 44

Choosing the right tools to work with is essential for any designer or developer. Discover what you should be using to perfect your workflow

## Simon Jones



Simon has led software engineering teams for many years, and is tremendously excited by the potential for service workers – the technology behind Workbox – to bridge the gap between native and web development. **Page 74**

## Mark Shufflebottom



Mark is a Professor of Interaction Design at Sheridan College. In this issue, Mark will be showing you how to bring images to life by capturing the user's mouse movements and triggering reveal animations. **Page 54**

## Richard Mattka



Richard Mattka is an award-winning interactive technologist. This issue he explores the power of sound in web experiences. Dive into the Web Audio API, libraries and resources to take your projects to the next level. **Page 66**

## Frank Kagumba



Frank is a frontend developer specialising in WordPress CMS, HTML5, CSS3 and JS. In this issue, he shows you how to employ advanced WebGL techniques to reveal text on your webpages using ThreeJS. **Page 62**

## Daniel Crisp



Daniel Crisp is a senior frontend developer currently working on projects for Google. This issue he shows how to set up and use a Redux data store within an Angular 5 application that uses lazy-loading. **Page 80**

## Neil Pearce



Neil is a front-end developer and designer who has worked in the industry for as long as we can remember. This issue he will be demonstrating how you can use CSS, SVG and a little bit of jQuery to create a tilting UI hover effect. **Page 58**

## Tam Hanna



Tam quickly realised the benefits of modularisation during his work in the embedded space. He has always been interested in object-oriented JS, and considers ECMAScript 6's modularisation API a must-discover. **Page 16**

## Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he recreates a host of techniques as inspired by the top-class sites seen in Lightbox. **Page 16**

Future PLC Richmond House, 33 Richmond Hill, Bournemouth, Dorset, BH2 6EZ

## Editorial

Editor **Steven Jenkins**  
steve.jenkins@futurenet.com  
01202 586233

Designer **Harriet Knight**

Group Editor in Chief **Amy Hennessey**  
Senior Art Editor **Will Shum**

## Contributors

Bryant Chou, Tobias Van Schneider, Mark Billen, Leon Brown, David Howell, Steven Roberts, Mark Shufflebottom, Neil Pearce, Frank Kagumba, Richard Mattka, Simon Jones, Daniel Crisp, Tam Hanna, Philip Morris, Rob Mead-Green, Will Shum

## 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

Senior Advertising Manager **Mike Pyatt**  
michael.pyatt@futurenet.com  
01225 687538

Account Director **George Lucas**  
george.lucas@futurenet.com

Account Director **Chris Mitchell**  
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

## Subscriptions

Email enquiries [contact@myfavouritemagazines.co.uk](mailto:contact@myfavouritemagazines.co.uk)  
UK orderline & enquiries 0344 848 2852  
Overseas order line and enquiries +44 (0) 344 848 2852  
Online orders & enquiries [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
Group Marketing Director, Magazines & Memberships  
**Sharon Todd**

**Circulation**  
Head of Newtrade **Tim Mathers**

## Production

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

## Management

Brand Director **Matthew Pierce**  
Chief Operations Officer **Aaron Asadi**  
Commercial Finance Director **Dan Jotcham**  
Editorial Director **Paul Newman**  
Head of Art & Design **Greg Whittaker**

**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 9060

ISSN 1745-3534

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 mill holds full FSC (Forest Stewardship Council) certification and accreditation.

All contents © 2018 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 2008865) 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.

## Follow us!

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



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**  
Chairman **Richard Huntingford**  
Chief financial officer **Penny Ladkin-Brand**  
Tel +44 (0)1225 442 244

# 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**

# 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](mailto:steve.jenkins@futurenet.com) • [@WebDesignerMag](#) • [www.creativebloq.com](http://www.creativebloq.com)

## 08 What's new in Firefox 60?

The Mozilla browser gets an update and offers better support. **Web Designer** takes a 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 The future of eCommerce

Webflow CTO and cofounder Bryant Chou talks about what lies ahead for shopping online

## 16 Lightbox

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

## 28 Home comforts

When a travel firm needed a new website design it went local. **Web Designer** finds out more

## 36 Breaking the mould

Distinctive and opinionated. **Web Designer** talks to the talented trio at Wefail

## 44 Design faster, code quicker

A collection of 50 must-try tools and extensions for the modern day designer and developer

## 66 Awesome Web Audio

Discover why you need audio and how to add the power of sound to your projects

## 74 Progressive Web Apps

Learn how to add offline support to your apps with the help of Google's Workbox JS libraries

## 92 Hosting listings

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

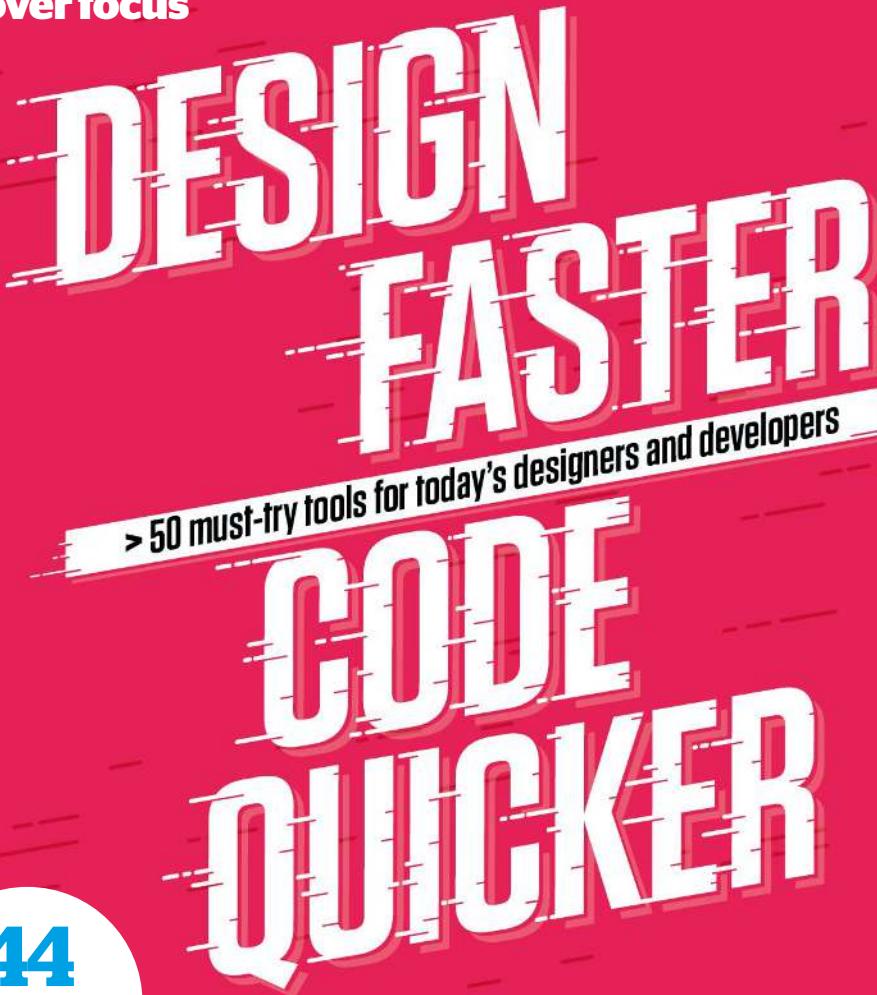
## 94 Course listings

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



**66**  
The power of sound  
Why and how to add web audio



**74**  
Progressive Web Apps  
How to add offline support with Workbox

# FileSilo

96 Get the latest must-have resources and videos

- 107 minutes of Beginner's JavaScript video guides from StudioWeb
- 14 Midnight Photoshop filters
- 6 Hi-res postcard and card mockups



Never miss  
an issue  
**Subscribe**  
Turn to page 34 now

Save on the cover price and get  
the mag delivered to your door



 The state of  
eCommerce, on the  
other hand, seems  
mired in the past 

COMMENT – Bryant Chou - p11



36

**Profile:** Wefail  
Adding a little anarchy to design



16

**Lightbox:** Crypton  
Introducing simple but smart effects

Visit the **WEB DESIGNER** online shop at

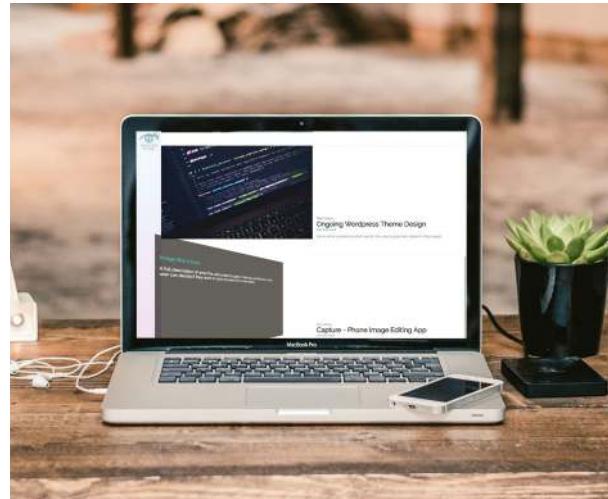


**myfavouritemagazines**  
myfavouritemagazines.co.uk

for the latest issue, back issues and specials

## Tutorials

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



54 **Add interactive directional images**  
Use the Direction-Reveal library to respond to user movement and animate images in different directions

58 **UI tilt effect with SVG**  
Introduce animated hover effects to landing pages using CSS, JavaScript and SVG

62 **Code animated text with WebGL**  
Employ the power of WebGL and ThreeJS to create an effect that blows text away in the wind

## Web Workshop

52 **Ripple effects and transitions**  
[themustafacelik.com](http://themustafacelik.com)

Add engagement with awe-inspiring interactive effects for background images

## Web Developer

74 **Progressive Web Apps**  
Introduce offline support to your applications the easy way with Google's Workbox JS libraries

80 **Angular state management**  
Learn how you can use the power of RxJS to manage your application's state with Redux

86 **Understand how ES modules work**  
ECMAScript 6 brings a new native modularization API. Find out how to use it in your projects

# Header

The tools, trends and news to inspire your web projects

## What does Firefox 60 have to offer?

The popular browser introduces a new set of features and support. **Web Designer** takes a look at what's new

**I**t's no secret that Google Chrome is the world's number one browser. But popular as it is, it still only has just over 57 per cent worldwide market share. Admittedly that is a huge chunk of the market, and around 40 per cent more than its nearest competitor, Safari. Think of this in real terms and even five or ten per cent still serves millions of users. So when one of the major browsers, that isn't Chrome, gets an update it is still big news. And as designers and developers, it is still important to know what browsers beyond Chrome are doing.

Firefox sits in the major browser department – and is *the* popular choice at **Web Designer** – but we love them all of course. Firefox 60, the latest update at the time of writing, includes some noteworthy new additions. ES modules bring an official, standardised module system to JavaScript, but support has not been available across all browsers. Firefox has now joined those which do offer support, so now is the time

to start using them as intended. There is a great post by Lin Clark at <https://mzl.la/2Gjkqy4>, which gives a great overview of what ES modules are all about and you can find out more about how they can be used on page 86. For a more technical view of what's going on be sure you pay a visit to <https://mzl.la/1PY7nnm>.

Authentication API. Again not sexy, but still vital. To find out more about what the API is about head to <https://mzl.la/2jPvDwJ>. And to find out what Firefox 60 offers, in turn, head to <https://mzl.la/2rAAI01>.

This one is not such a big deal but we always like an update that adds something simple and potentially attractive – it's the

**“The latest Firefox update now supports the Same-Site attribute when setting cookies”**

So what else does Firefox 60 bring to the masses? Not the sexiest of subjects, but still vitally important, is security. The latest Firefox update now supports the Same-Site attribute when setting cookies. This helps prevent a malicious site from deep-linking to a page. If you want to read more about this head over to <https://mzl.la/2lsDr5t>. Sitting in the same arena is the Web

designer in us; Firefox 60 adds support for the paint-order CSS property – the only major browser which currently offers it. You can find out more at <https://mzl.la/2rBtisr>.

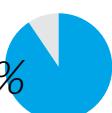
If you are a Firefox lover, or a diligent designer and developer, you can check out the release list for number 60 at <https://developer.mozilla.org/en-US/Firefox/Releases/60>.

## STAT ATTACK SEARCH ENGINES

Google is the dominant force, but how dominant?

### Google

90.61%



Number one, as might be expected

### Bing

3.24%



Microsoft's offering not quite as popular

### Yahoo

2.09%



The old stalwart still the third most popular

### Baidu

2.04%



Popular-ish Google alternative in Asia

### Yandex RU

0.68%



Ever used this search engine? We haven't

Source: [gs.statcounter.com](https://gs.statcounter.com) (correct as of April 2018)

**CB** CREATIVE BLOQ  
[creativebloq.com](http://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.



**Subscribe today and save 20%**  
<https://bit.ly/2qLxVl4>

# Sites of the month



01.



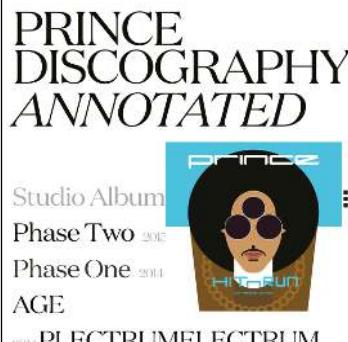
02.



03.

04.

**01. Amazonia Font**  
[amazoniafont.com](http://amazoniafont.com)  
Inspired by the exotic wildlife of the Amazon. Choose from the alphabet for more.



**02. Barovier & Toso**

[barovier.com/en](http://barovier.com/en)  
Typography speed effects, plus stretch and snap image effects.

**03. Célia LOPEZ**

[celialopez.fr](http://celialopez.fr)  
Multiple scrolling backgrounds, 3D objects, interactive expanding backgrounds.

**04. Prince discography**

[princeestate.com](http://princeestate.com)  
A simple pop-up window on hover presents music, images and video from the maestro.

## Graphics

Robert Romanowicz

[bit.ly/2K5bgqc](http://bit.ly/2K5bgqc)

Gorgeous illustrations from the very talented pencil of Robert Romanowicz.



## Colour picker

05.05.18

[bit.ly/2I8dUeu](http://bit.ly/2I8dUeu)



## Typesetter

Scotch

[bit.ly/2wrLw5D](http://bit.ly/2wrLw5D)

A clean, modern serif font that offers a nod to a historic past. Great for headlines.

ABCabc  
0123456

## WordPress

Kings and Queens

[kingsqueens.ancorathemes.com](http://kingsqueens.ancorathemes.com)

A simple, straightforward single-page theme with neatly themed sections. Includes eCommerce options.



# webkit

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

## Grid to Flex

### Flexbox fallbacks for popular UI solutions that use CSS Grid.

CSS grid is AMAZING! However, if you need to support users of IE11 and below, or Edge 15 and below, grid won't really work as you expect ([more info here](#)). This site is a solution for you so you can start to progressively enhance without fear!

Things to Note: CSS grid allows for specified styling using the vertical area as well as horizontal. Flexbox doesn't allow for this, so we'll be using fixed heights in these examples.

Feel free to play around with the CSS! To swap between implementations, use the toggles.

## Page Layouts

### Grid to Flex

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

CSS Grid is the way forward for layouts, however for full adoption browser support is always an issue. This is where Grid to Flex steps in and helps out. If you need support for IE11 or Edge 15 or below, so you have a complete solution, this is the answer. Flexbox is the fallback option and the site provides option for the more obvious layouts.



#### Calendar

[ui.toast.com/tui-calendar](http://ui.toast.com/tui-calendar)

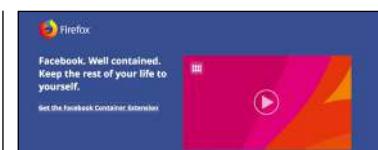
A fully featured JavaScript schedule calendar that is customisable. View daily, weekly, monthly dates, and resize your schedule to fit. Plus lots more.



#### Tone.js

[tonejs.github.io](http://tonejs.github.io)

A framework for creating interactive music in the browser. Provides synths, effects and intuitive musical abstractions built on top of the Web Audio API.



#### Facebook Container

[mzl.la/2IUqvT8](http://mzl.la/2IUqvT8)

Want to make it harder for Facebook to track your activity around the web. Then this Firefox extension/add-on from Mozilla is the place to get started.

## TOP 5 Web conferences - July 2018

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



### Angular Camp

[angularcamp.tech](http://angularcamp.tech)

A two-day developer-driven conference and one-day workshops, brought to you by the community.



### ScotlandJS

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

Want to learn about JavaScript and get a bit of whisky and gin tasting in as the same time? Then sign up now.



### Design & Content

[content.design](http://content.design)

A conference where designers and content strategists team up. Get a day of workshops and two days of talks.



### FullStack 2018

[bit.ly/2G0pw12](http://bit.ly/2G0pw12)

Get three intensive days of learning and sharing from 68 experts who're keen to pass on their skills and knowledge.



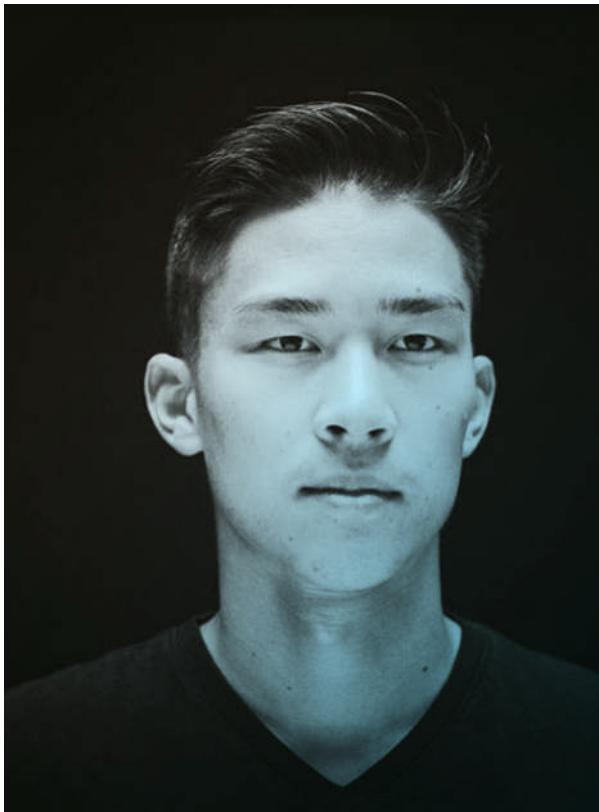
### Curry On 2018

[curry-on.org/2018](http://curry-on.org/2018)

Set in Amsterdam, Curry On is 'a conference focused on the intersection of emerging languages and challenges'.

# eCommerce needs a shot in the arm

Can code-free platforms provide a much-needed catalyst to innovation?



**Bryant Chou**

CTO and cofounder, Webflow  
[Webflow.com](http://Webflow.com)

“ Every store you visit, whether online or IRL, should represent a wholly unique experience. An experience infused with brand attributes in everything ,”

**I**t's 2018. We're at the height of innovation – and tomorrow will see us reach all-new heights. Technologies like self-driving cars, true artificial intelligence (AI), anthropomorphic robots, and privatised space travel are all right around the corner. They're so imminently possible that we're already seriously considering how these technologies stand to completely transform our civilisation. The state of eCommerce, on the other hand, seems mired in the past – 1997, to be exact.

A prime example could be the comparison of a couple of reviews of top eCommerce platforms. Back in 1997 CNET ran a review stating: "For people who've been hanging back wondering if they should put their small business on the web, Viaweb's product says, Yes, you can get involved and you don't need a rocket scientist to do it". Fast forward to 2018 and PC Mag ran a review stating something very similar: "Shopify makes it easy to get off the ground with basic features and the ability to fine-tune features at a later date. Shopify is our Editors' Choice for shopping cart tools for its intuitive and easy-to-navigate dashboard, and its ease of use for first-time merchants".

It's almost unsettling how similar these descriptions sound.

Over 20 years later, the emphasis remains on ease of use, speed, and accessibility to beginners. All noble goals that help entrepreneurs worldwide get their businesses up and running fast – and driving a transition to a more diverse, localised, yet global economy.

But there's not much said about the output of eCommerce website tools: the experience store owners create for shoppers, which – as any UX designer will argue – has a powerful influence over the success of a store and its underlying brand.

When we do talk about the output of eCommerce tools, it tends to be about how beautiful and modern the templates are. It is about how easy it is to customise a theme thanks to its built-in drag and drop editor, and how easy it is to edit the HTML/CSS through another built-in editor.

All of which creates a 'design process' that can be boiled down to the something simple. Once the user is happy with a theme, all they need to do it hit the Publish button to apply.

Now, I'm not here to excoriate template-based site builders. There's a clear, market-confirmed place for such tools in the world. Webflow itself offers a variety of templates that are ready to publish to the web once the content's ready.

But this single sentence highlights a gap in the eCommerce site build process today. I call it the 'experience gap', because it results in the flattening, or homogenisation, of the shopper's experience.

Ideally, every store you visit, whether online or IRL, should present a wholly unique experience. An experience infused with brand attributes in everything from the colour scheme to the voice of the promotional banners, from the staff's outfits to the clarity and precision of the wayfinding system.

Think of your favourite brick-and-mortar shopping experiences. Whether it's a tatty, well-used bookstore-café or a freshly opened Anthropologie, the experience is unique, multi-sensory, and most of all, memorable. You think about these places and you can feel the bookstore's dog-eared paper between your fingers, smell the wood pulp and leather – or hear Lana del Rey pumping through the state-of-the-art sound system, almost taste the perfume of Anthropologie.

These are places where you don't just want to *buy* some of their wares. You want to *buy into* the whole paradigm, become part of the experience.

At Webflow, we believe you can bring that to the web. Without sacrificing most of the ease.

We believe that you should be able to build a bespoke eCommerce store that's shaped by its product lines and manifests the brand in ways that move you. That you should be able to take control of the look and feel of every single element of your site – without having to master some arcane or proprietary markup language.

We believe there should be a platform that offers the best of both worlds: intuitive store management for the entrepreneur, balanced by speedy but powerful design tools for the designer. And we're going to build it: [webflow.com/ecommerce](http://webflow.com/ecommerce).

CONTACT US AT: steve.jenkins@futurenet.com | @WebDesignerMag

# webkit

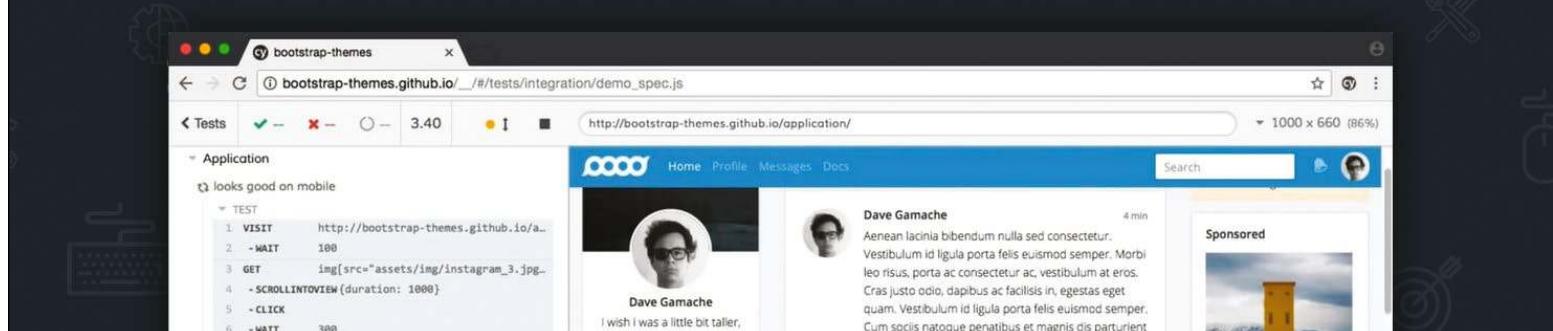
Discover the must-try resources that will make your site a better place

## The web has evolved. Finally, testing has too.

Fast, easy and reliable testing for anything that runs in a browser.

\$ npm install cypress or [Download Now](#)

Install Cypress for Mac, Linux, or Windows, then [get started](#).



## Cypress

cypress.io

Testing is a process that every website needs to go through, but the question is what do you use to do it? Cypress claims to offer 'fast, easy and reliable testing for anything that runs in a browser.' So what does it have to offer? A complete end-to-end testing experience. Setup (no dependencies), execute tests in real-time and view results via its Dashboard.



**PROBOT**  
GitHub Apps to automate and improve your workflow

Use pre-built apps to extend GitHub, and easily build and share your own.

### Probot

<https://probot.github.io>

This is a framework for building GitHub Apps in Node.js. Probot enables you to code and create your own apps to automate and improve your workflow.



### react-web-animation

[react-web-animation.surge.sh](https://react-web-animation.surge.sh)

A collection of React components that expose the Web Animations API. Contains a number of examples, source code and documentation.



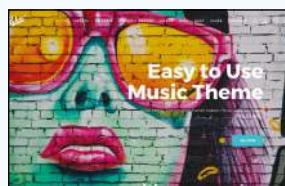
### Simple Print

[pdf.fivefilters.org/simple-print](https://pdf.fivefilters.org/simple-print)

This is a free service that provides an easy way to produce printable PDFs from web articles. Enter the URL of the article and press the button.

## TOP 5 WordPress themes

Need to get a good-looking website up and running quickly? Try one of these themes



### Phase

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

In a band? Want to start your own record label? This theme has all the elements needed to make it happen.



### Buzznote

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

A minimalist and easy-to-use theme. Big imagery coupled with serif and sans serif fonts equals a good-looking blog.



### Brewski

[bit.ly/2K8kWQW](http://bit.ly/2K8kWQW)

A bar-and-beer themed theme with big landing page images, neat animations and a shop layout for selling your beers.



### Benri

[bit.ly/2Ia0kat](http://bit.ly/2Ia0kat)

Not quite sure what you want? This multi-purpose theme has a host of layouts to choose from and is fully customisable.



### Baroque

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

An architectural-based theme with a fullscreen image slider. Big, bold typography and a clean design complete the look.

# generate

The conference for web designers

## LONDON

# 19-21 SEPTEMBER 2018

### 3 DAYS OF INSIGHT AND INSPIRATION

TICKETS ON SALE NOW

[www.generateconf.com](http://www.generateconf.com)  
**#generateconf**

BROUGHT TO YOU BY   The voice of web design 

# Semplice

**Semplice is the first fully customisable portfolio system based on WordPress. Built by designers, for designers.**



Web Designer talks to co-founder  
Tobias van Schneider  
[semplice.com](http://semplice.com)

#### **What's the story behind Semplice? How did it come into existence?**

My co-founder, Mike, and I originally built Semplice for our personal portfolios. We were dreaming about a perfect design portfolio, completely customised with beautifully branded case studies for each project. No other portfolio tool out there offered what we needed, so we built it ourselves. We imagined other creatives felt the same way; as a designer you have a specific vision for your portfolio, but not necessarily the time or coding knowledge to make it happen. With Semplice, you can make it happen.

#### **What sets Semplice apart from other portfolio builders?**

Many portfolio builders are focused on maintaining the status quo with trendy templates. Semplice is focused on building with pride. You have the freedom and power to create a completely custom website, no coding required. If you do know how to code, you can take it even further. Semplice is built on WordPress, so you have the power of the WordPress community behind you.

#### **What plans do you have for the future of Semplice?**

Just last year we launched our latest product, Semplice 4. We stripped Semplice down and built it again from scratch, streamlining the experience and adding a ton of helpful new features. Now we continue releasing new Semplice 4 updates, like our new navigation options or recent Unsplash integration, that make it even better. Aside from that, we have a big project underway for Semplice. It's focused on our belief that portfolios should be a living thing, continually updated as a designer evolves in their career.

## 5 key features

**Semplice 4 boasts a range of modern features for the independent designer, artist or illustrator**

### 1. Visual content editor

Semplice is built on WordPress, but you won't see the WordPress CMS. The visual drag and drop content editor provides a live view of your page as you create it.

### 2. Mobile ready

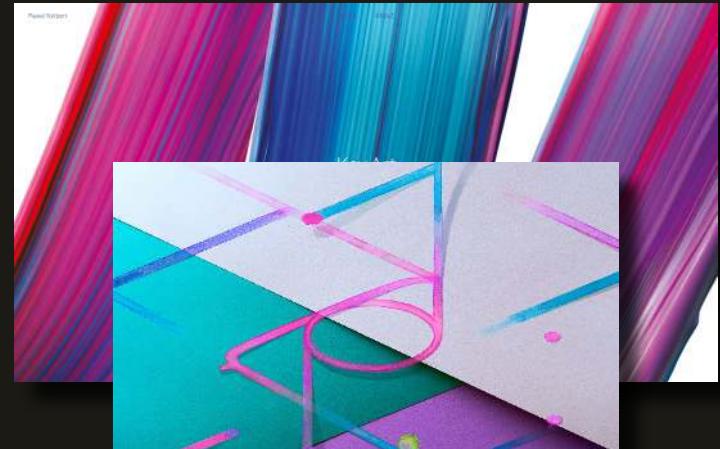
Your Semplice site is responsive by default, but you can also adjust mobile breakpoints straight from the desktop editor for seamless presentation at any screen size.

## PORTFOLIO SHOWCASE

**Chances are, designers and studios you admire already use Semplice**



**MICHELA PICCHI**  
[michelapicchi.com](http://michelapicchi.com)



**PAWEŁ NOLBERT**  
[nolbert.com](http://nolbert.com)



**VERENA MICHELITSCH**  
[verenamichelitsch.com](http://verenamichelitsch.com)



**NOÉMIE LE COZ**  
[noemielecoz.com/hello/](http://noemielecoz.com/hello/)

### 3. Fully brandable

Nothing is off limits with Semplice. Every element of your portfolio is customisable, from navigation to button style to footer. You can even upload custom fonts.

### 4. Animations and transitions

Add page transitions or motion to any piece of content. Choose triggers (on load, click, hover or scroll) and effects to create an elegant and interactive experience.

### 5. Efficient workflows

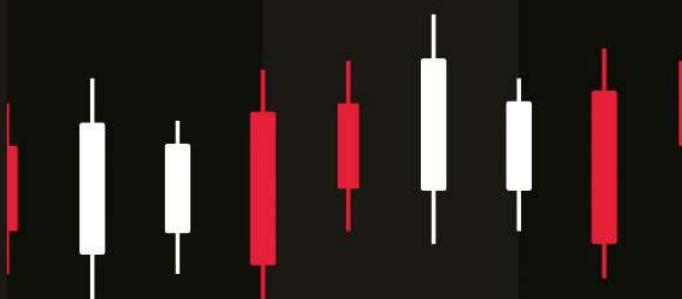
Create site-wide presets for heading and paragraph text to work more efficiently. The Studio edition enables you to save content sections as Blocks and repurpose them.

+\_+

↑

# I'm good at making forecasts

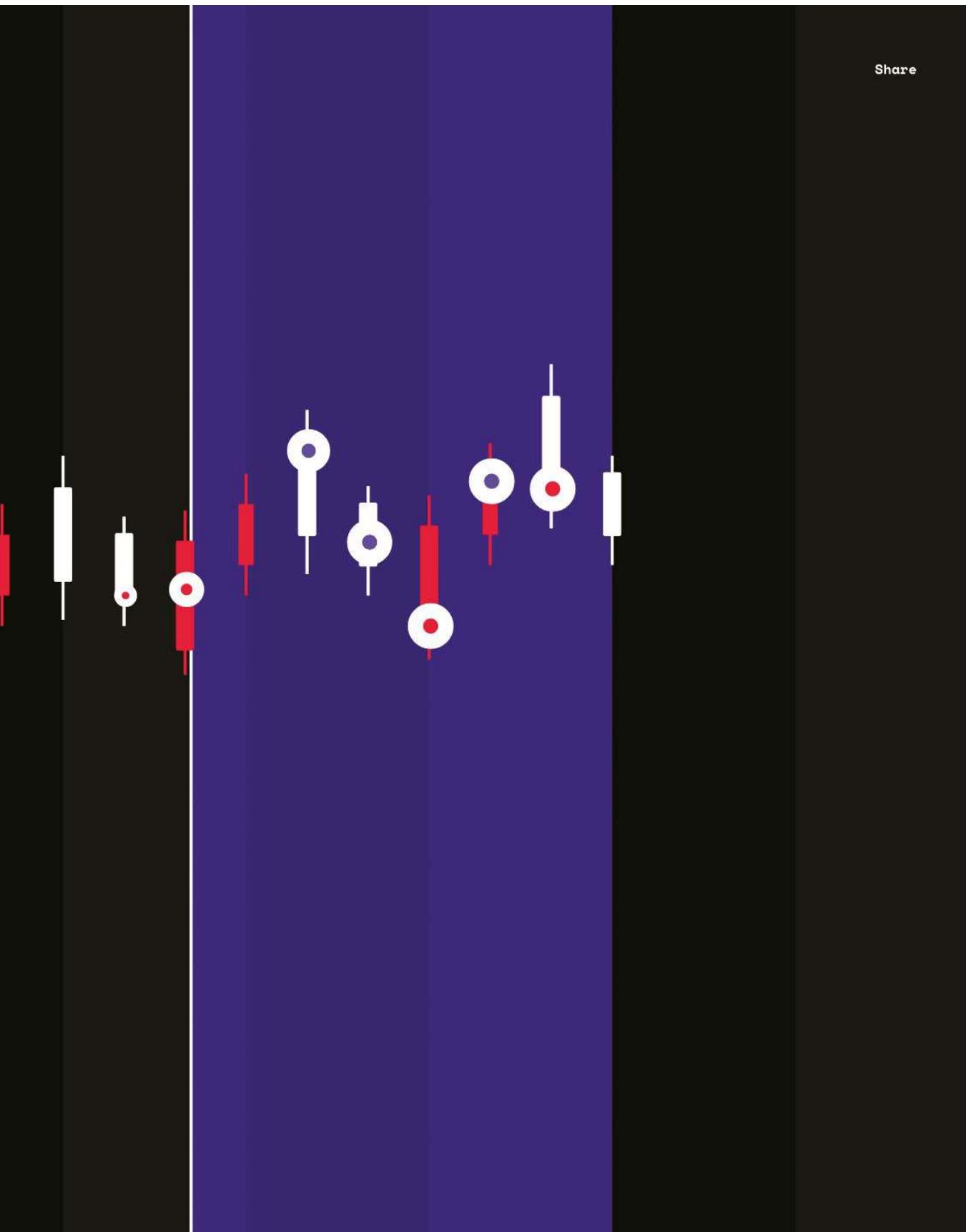
I analyze thousands of trading transactions and reveal patterns using my machine learning algorithms. Then I predict changes in certain points and signal to sell or buy currency.



Technology  
Team lookup  
Contacts

↓

**“This dynamic Ruby-built site represents a cryptocurrency bot that utilises machine-learning algorithms to forecast profitable trading”**



# Hello! I'm Crypton |

A cryptocurrency trading bot  
based on machine learning.

[Technology](#)  
[Team lookup](#)  
[Contacts](#)

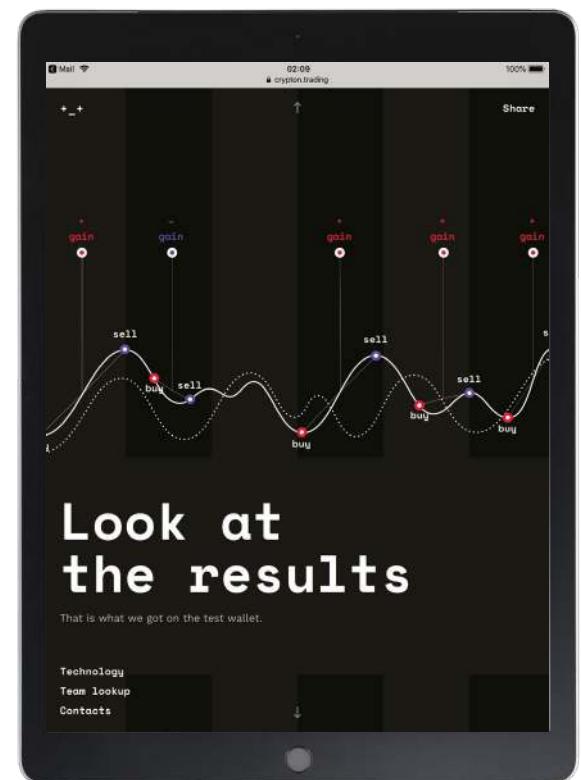
**Above**

The opening splash header is very utilitarian, favouring little more than stark typography to underline the robotic theme



**Above**

Scrolling the page triggers a series of data visualisations that thoughtfully illustrate Crypton's various feature highlights



**Above**

On smaller screens the site reforms its layout accordingly and enjoys more control over the scrolling animations

# Code an animated typing effect with narrative face

Present webpage content using a typing effect, complete with an animated face

## 1. Document initiation

The first step is to initiate the structure of the webpage. This consists of the HTML container responsible for storing the head and body sections. While the head section's main responsibility is to load the external CSS, the body section will store the HTML content created in step 2.

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

## 2. HTML content

The HTML content consists of a container that uses the 'typing' class. This will be used by the CSS to apply the typing effect to any child elements. The child content element is made from a h1 tag, but you could use another element such as 'p' to create the element as a paragraph.

```
<div class="typing">
  <h1>Put your title here...</h1>
</div>
```

## 3. CSS initiation

Create a file called 'styles.css'. The first step of the CSS defines the document and body containers to cover the full browser window without any visible border spacing. The default colours for the black page background and white text colours are also set in this step. Content elements within the page will inherit the colour set in this step as their default colour.

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

## 4. Typing children

All children within the typing container are set to display over one line without the use of text wrapping. Most

importantly, these child elements have the 'typing' animation applied to them. This animation is set to play over 5 seconds with 50 frame snapshots - allowing for the staggered typing effect.

```
.typing > *::before{
  overflow: hidden;
  white-space: nowrap;
  animation: typingAnim 5s steps(50);
}
```

## 5. Face eyes

The effect is also accompanied by an animated face that appears to narrate the typed text. This step creates the eyes of this face as a CSS 'virtual' element using the 'after' selector. The eyes are placed relative to the parent text, with its content set as two dot text characters.

```
.typing > *::after{
  content: ". .";
  display: block;
  position: absolute;
  top: 1em;
  left: .35em;
}
```

## 6. Face mouth

Like with the eyes, the face's mouth is made from a CSS virtual element - this time using the 'before' selector. The mouth is positioned in relation to the parent text element, as well as having a border radius to appear with a rounded shape. The 'typingSpeak' animation is applied; it will last for 0.5 seconds using two frames of animation. With the animation being repeated five times, the total

animation time will be 2.5 seconds.

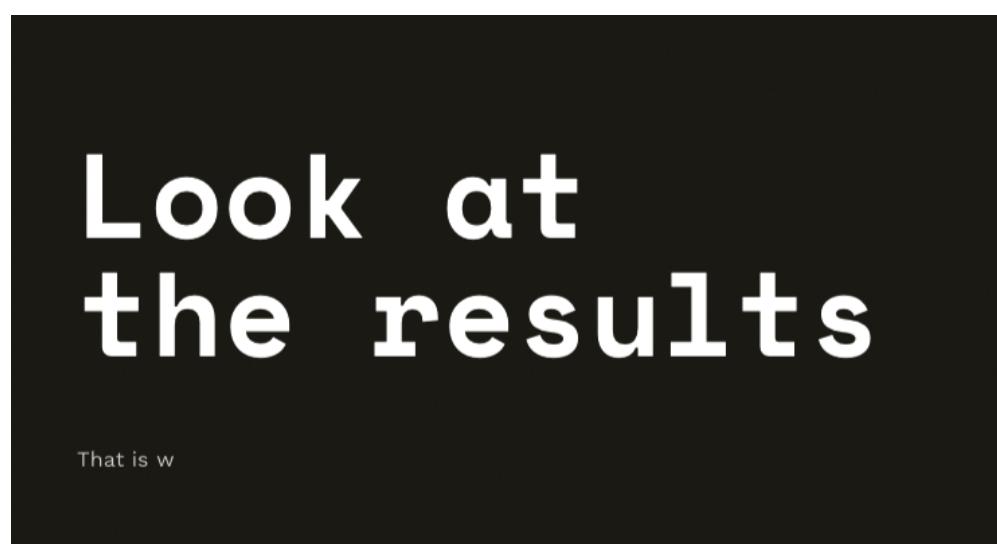
```
.typing > *::before{
  content: "";
  position: absolute;
  display: block;
  top: 2.1em;
  left: .25em;
  width: 1em;
  height: .1em;
  border-radius: 100%;
  background: #fff;
  animation: typingSpeak .5s steps(2);
  animation-iteration-count: 5;
}
```

## 7. Animation definitions

This step defines the animations referenced by elements created in previous steps. The 'typingAnim' animation that has been used for the typing effect changes its element from no width to full width. The 'typingSpeak' animation used for the face's mouth changes its element from appearing flat to more open.

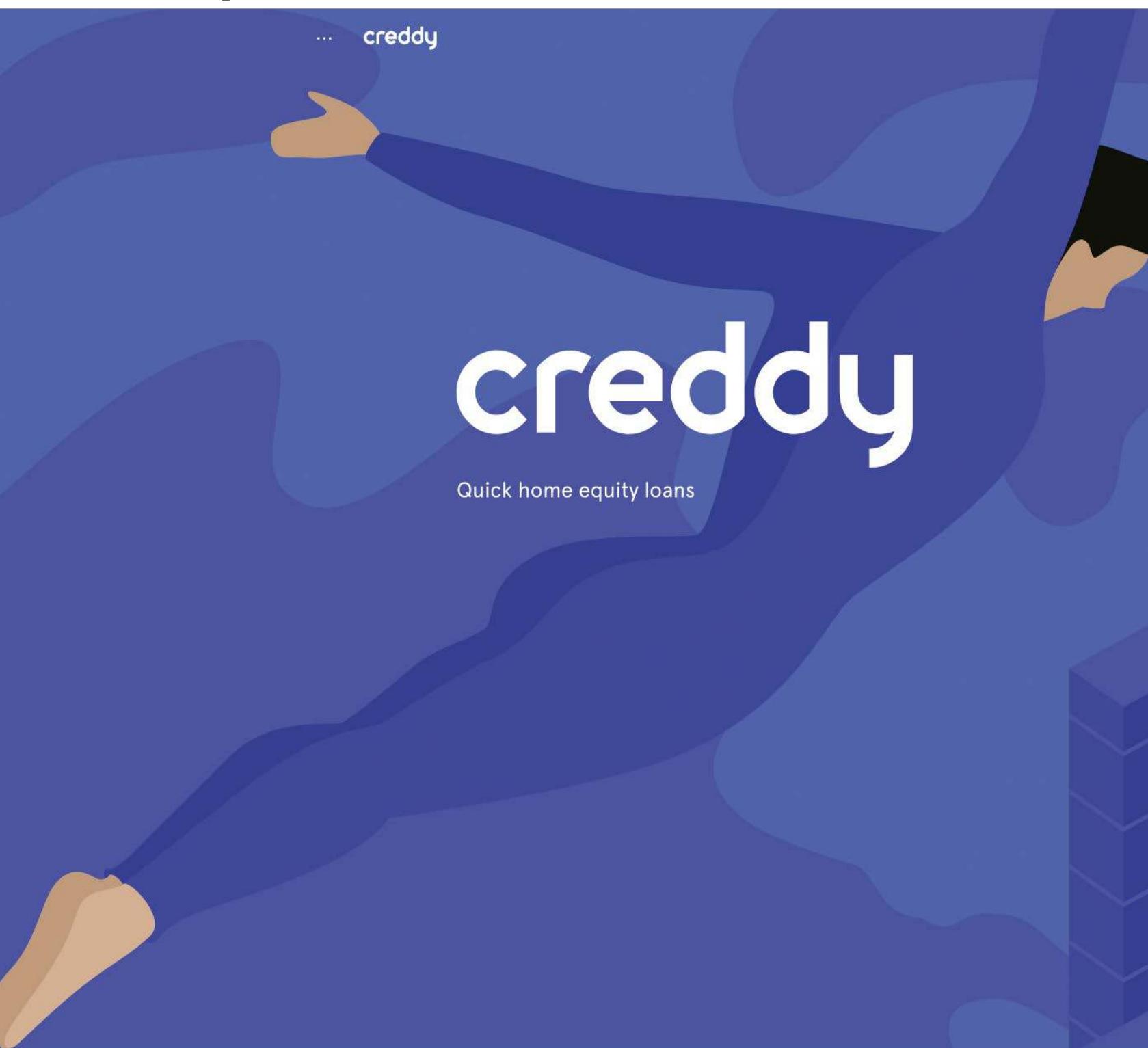
```
@keyframes typingAnim{
  from { width: 0 }
  to { width: 100% }
}

@keyframes typingSpeak{
  0% { width: 1em; height: .1em }
  100% { width: 1em; height: .5em; }
}
***
```



**Creddy** <https://en.creddy.ru>

Designer:  
**Red Collar** <http://redcollar.digital>



Take a loan



Become an investor

“A modern p2p online equity lender of Russian origin, Creddy’s multi-lingual site unfolds colourful stylised SVG animations on scroll”



#### Colours

#4C6AD6	#182EC4
#BC9A80	#FA4E40

#### Tools

PHP, Bitrix,  
HTML5, JavaScript

#### Fonts

abcABC  
1234567890  
*abcABC*  
1234567890  
**abcABC**  
**1234567890**

The Aperçu Pro font by Colophon is featured in Regular, Italic and Bold weights across page headings and quotations.

**abcABC**  
**1234567890**

Museo Sans Cyrillic 500 by Jos Buivenga and Irina Smirnova for exlibris styles the site's paragraphs.

+7 (495) 150-1-150

# Take a loan at creddy

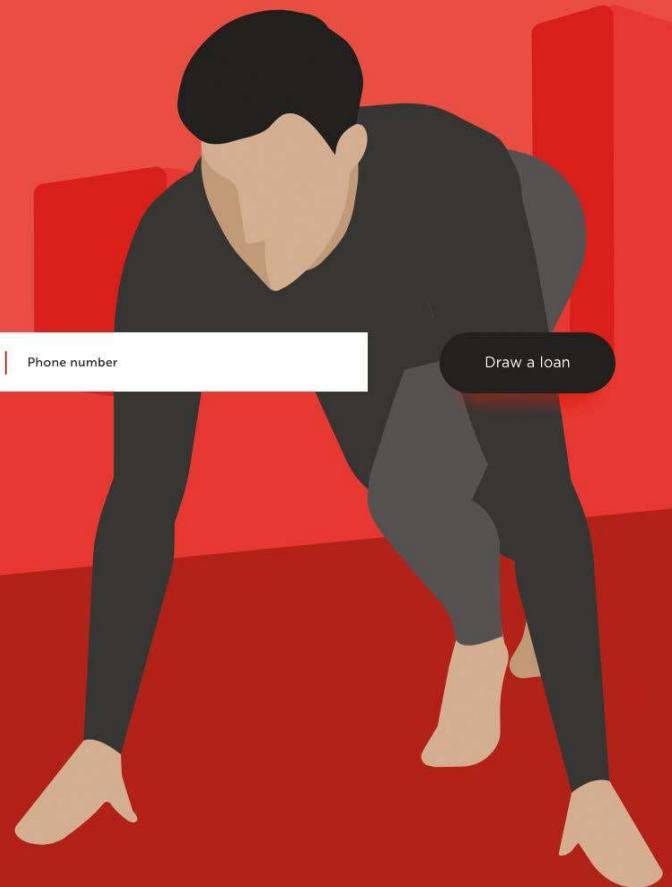
Full name

Email

Phone number

Draw a loan

I agree to the processing and use of my personal data

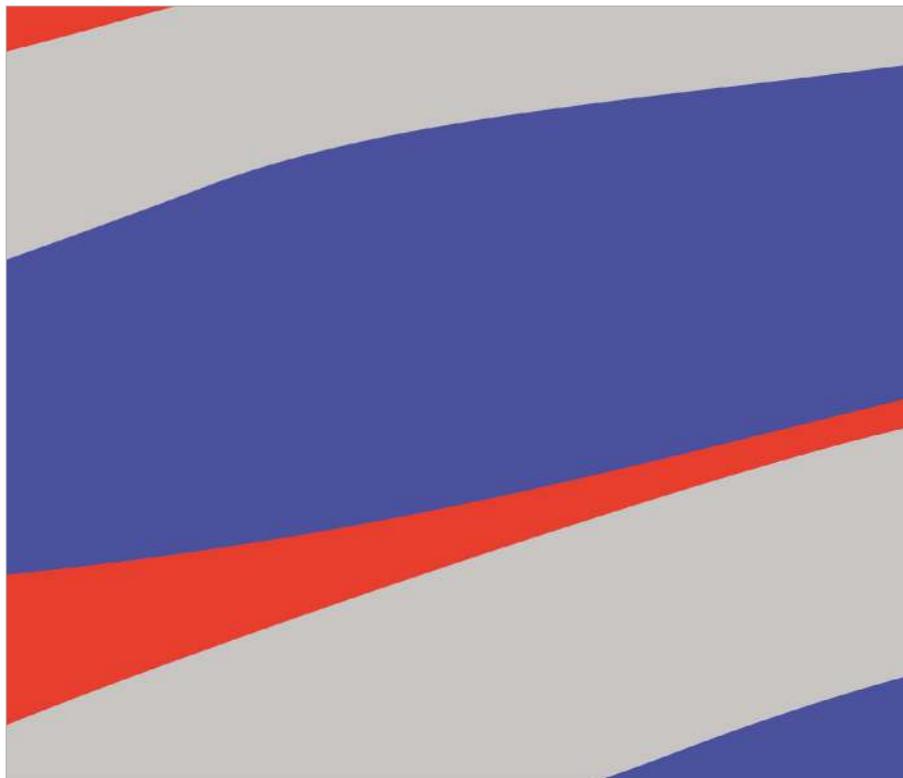


Loans against the pledge of real estate for any purpose

+7 (495) 150-1-150

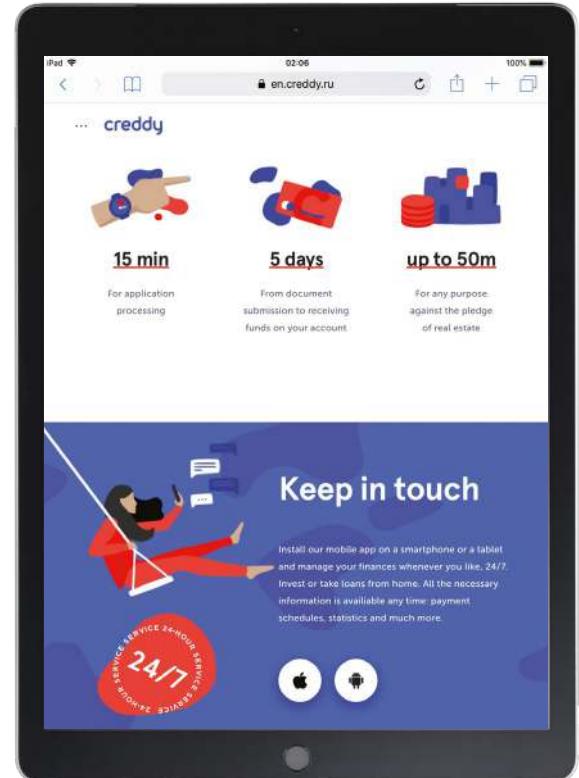
#### Above

Stacked <div> elements containing SVG assets combine over static backgrounds, using subtle motion techniques to add life



#### Above

Click the links to navigate sections and the browser windows fills with multi-coloured waves to smooth page transitions



#### Above

Responsive enough to look identical in smaller browsers, the site also promotes links to native apps for iOS and Android devices

# Create an animated header

Use CSS keyframe animation to create independently animated elements for your web page banners

## 1. Initiate HTML document

The first step is to initiate the document with HTML. This consists of a document container element used to store the head and body sections. While the head section is primarily used to load the external CSS, the body is used to store the content elements created in step 2.

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

## 2. HTML content

The HTML content consists of a head element to be used as a container for the content elements to show. Each content that is to be animated uses a 'data-animation' attribute to describe which animation is to be applied. This attribute is picked up by the CSS in a later step to apply the required animation.

```
<header>
  <h1 data-animation="in" data-
    title>Title</h1>
  <div data-animation="up"></div>
  <div data-animation="down"></div>
</header>
```

## 3. CSS start: header

Create a new file called 'styles.css'. The first step in this file defines how to display the header container. Its size is set in relation to the browser window size using the vw and ch unit measurements. Overflow is also set to hidden, so that content outside the defined size is clipped.

```
header{
  position: relative;
  width: 100vw;
  height: 50vh;
  overflow: hidden;
}
```

## 4. Inner elements and title

The elements inside the header section are displayed using absolute positioning; appearing in relation to the parent header. The element with the 'data-title' attribute is set to display 5 times bigger than the page font size. The 'left' property will display it in a specific position - 10% from the left, while 'z-index' will guarantee its appearance above all other content.

```
header > *{
  position: absolute;
```

```
}
```

```
header > [data-title]{
  font-size: 5em;
  left: 10%;
  z-index: 99999;
}
```

## 5. First div child

The div elements are used to represent where pictures will appear; the 'nth-of-type' selector is used to select the first div inside the header. Use an image of your choice with the 'background' property - this example sets a semi-transparent colour for default visibility. The background is set to cover the element size defined through the width and height properties.

```
header > div:nth-of-type(1){
  top: 5%;
  left: 10%;
  width: 50%;
  height: 50%;
  border: 1px solid #000;
  background: url(your-image.png)
  rgba(0,255,0,.25);
  background-size: cover;
}
```

## 6. Second div child

The same type of formatting is applied to the second div child inside the header. Unique positioning using top and left are applied, along with width and height for its size. The background colour defined by 'rgba' is for display purposes of this tutorial - so feel free to take this out when using your image.

```
header > div:nth-of-type(2){
  top: 20%;
  left: 40%;
  width: 50%;
  height: 50%;
```

```
background: url(your-image.png)
  rgba(0,0,255,.25);
  background-size: cover;
}
```

## 7. Animation application

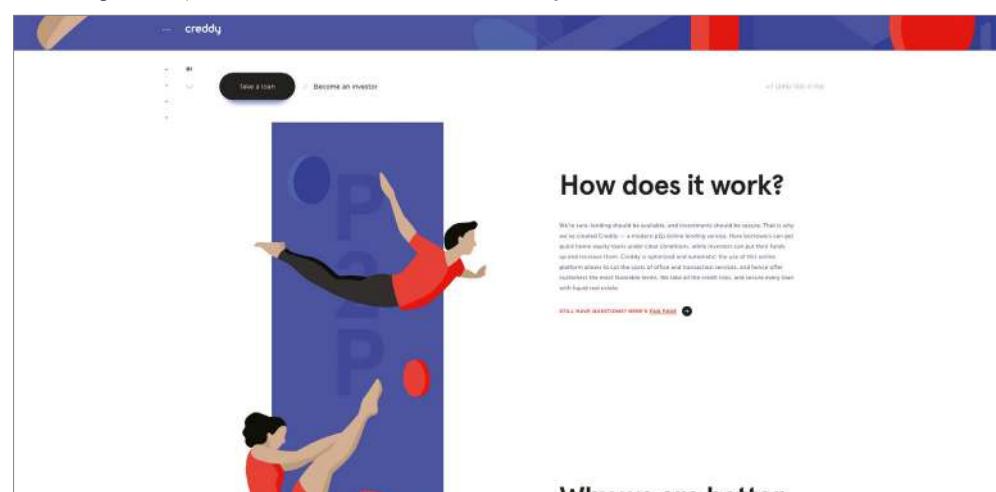
Elements that have the 'data-animation' attribute need to have an associated animation applied to them. This step searches for these elements inside the header container and applies the correct animation. The settings are made to make the animation last for one second using '1s'; alter this to change the speed of the animation.

```
header [data-animation="up"]{
  animation: openUp 1s forwards;
}
header [data-animation="down"]{
  animation: openDown 1s forwards;
}
header [data-animation="in"]{
  animation: openIn 1s forwards;
}
```

## 8. Animation definitions

The different animations are defined using just one keyframe - which will be the starting point for the animation to start 'from'. The final animation frame will be the default settings created for the element in previous steps 4 - 6. This allows the same animation to be applied to many items.

```
@keyframes openUp {
  from{ top: 100%; }
}
@keyframes openDown {
  from{ top: -100%; }
}
@keyframes openIn {
  from{ font-size: 0; }
```



## How does it work?

We're sorry, funding should be available, and instruments should be selected. That is why we've started Creddy - a modern crowdfunding lending service. More instruments will get added over time, so stay tuned! We're currently working on adding more instruments, so please check back soon.

[Still have questions? Check out our FAQ](#)

# HURU Backpack <https://huru.rocks>

Designer:  
Bachoo <https://bachoodesign.com>

**HURU**



“This boutique backpack brochure scrolls to transition between typography, photography, video clips and spinning product samples”

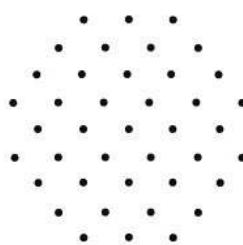
[HOME](#)

[ABOUT US](#)

[BUY HURU](#)

# THE ULTIMATE CITY & TRAVEL BACKPACK

[BUY HURU](#)



[FACEBOOK](#)

[E-MAIL](#)

[INSTAGRAM](#)

CREATED BY  BACHOO

## Colours



#CBAC90



#68986A



#DECDB9



#646A68

## Tools

PHP, jQuery, GSAP,  
WooCommerce

## Fonts

**ABCABC  
1234567890**

Integral CF Demi Bold,  
published by Connelly  
Fagen Type Design, is the  
display sans typeface used  
across the site's headings.

**abcABC  
1234567890**

PF DIN Text Pro Regular  
by Parachute is used to  
style the nuggets of text  
found within page  
paragraphs.

# Create an inverted text swipe animation effect

An effect for page titles where you can choose between changing the background style and content

## 1. Initiate HTML template

The first step is to define the template structure of the HTML document. This consists of the HTML document container, which is responsible for storing the head and body sections. While the head section is responsible for loading the external CSS stylesheet, the body section will be used to store the content elements 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>Typewriter Effect</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 individual headings with the 'swipe' class used as containers for the animated text. Each heading has two first level children - the first being visible by default, with the second being the text to transition into. The first element needs to have its text contained with another container for reasons of the CSS.

```
<h1 class="swipe">
  <span><span>Same text.</span></span>
  <span>Same text</span>
</h1>
<h1 class="swipe">
  <span><span>Some text.</span></span>
  <span>Different text.</span>
</h1>
```

## 3. CSS initiation

Create a new file called 'styles.css'. This step initiates the document to display with a red coloured background that covers the full space of the window. Additionally, the padding property is eliminated to ensure that no visible border spacing is presented around the sides of the web page window area.

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

```
background: #f00;
padding: 0;
}
```

## 4. Swipe container

Each container element using the 'swipe' class is required to use relative positioning and to be displayed with a set height. The height set to 1em will show the element with the same height as the text characters it will display. These settings are important for the CSS defined in following steps.

```
.swipe{
  position: relative;
  height: 1em;
}
```

## 5. Swipe children defaults

All of the first level children inside the swipe containers are required to have a set of default properties for the effect to work. These include absolute positioning so that they can be positioned at the top of their container. It's important for these items to be floated to the right for their animation to work.

```
.swipe > *{
  position: absolute;
  display: block;
  overflow: hidden;
  width: 100%;
  height: 1em;
  color: #000;
  background: #fff;
  white-space: nowrap;
  float: right;
  top: 0;
  margin: 0;
}
```

## 6. Swipe child one

The nth-child selector is used to apply specific styles to the first element that appears within the swipe container. While the outer container uses an animation to push the black box forwards, the inner container uses an animation to pull its inside text backwards. This keeps the text positioned in the same place while the black background swipes forward to reveal the second level child element.

```
.swipe *:nth-child(1){
  position: relative;
  color: #fff;
  background: #000;
  animation: outer 5s forwards;
}
.swipe *:nth-child(1) > *{
  display: block;
  z-index: 3;
  width: 100%;
  float: right;
  animation: inner 5s forwards;
```

## 7. Animation definitions

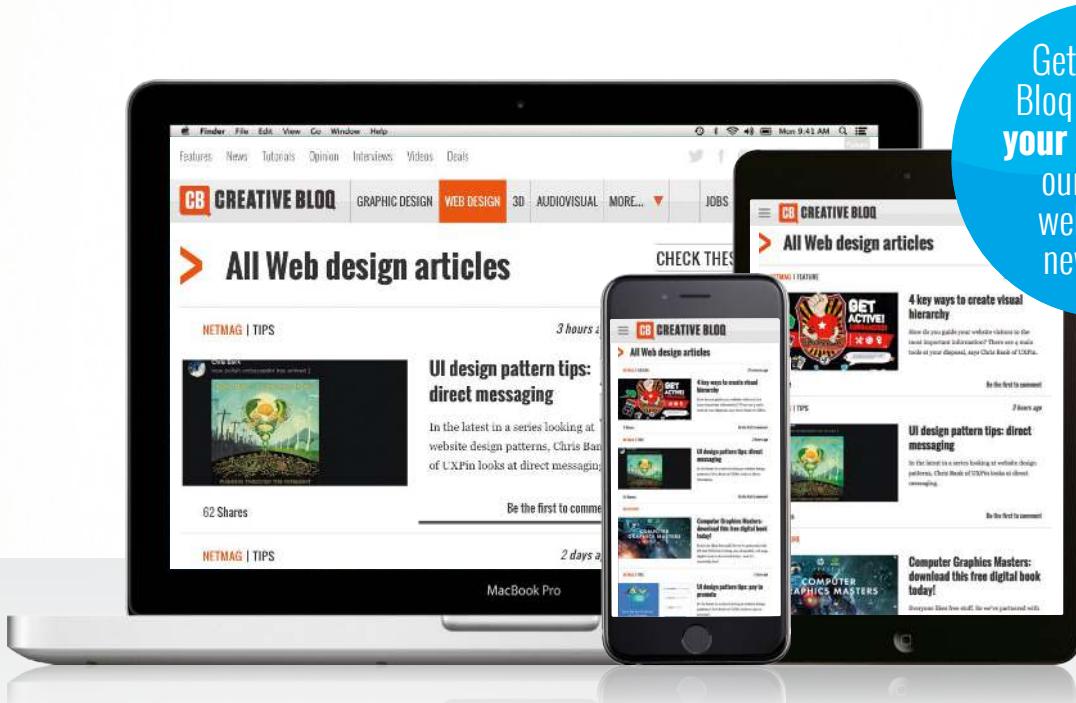
The final step is to define the animations required for step 6. The outer element is animated by changing its width from 100% to 0 - hence the importance of it being floated right in step 5. The inner animation needs to pull the text backwards, which is achieved by increasing its width to double the size of its container.

```
@keyframes outer{
  from { width: 100%; }
  to { width: 0; }
}
@keyframes inner{
  from { width: 100%; }
  to { width: 200%; color: #000; }
}
***
```



**THE ULTIMATE  
CITY & TRAVEL  
BACKPACK**

# The number one destination for **web design** news, views and how-tos.



Graphic design

Web design

3D

Digital art

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



# Home Comforts

When award-winning travel specialists Inside Asia Tours Ltd requested a new web experience celebrating exotic away days, they kept it local with a standout agency close to their hearts





## When to Travel

[insideasiatours.com/  
when-to-travel](http://insideasiatours.com/when-to-travel)

by

**Green  
Chameleon**  
[@CraftedbyGC](http://craftedbygc.com)

OVER THE MANY issues that Design Diary has appeared, featuring the many agencies and projects, a big theme of loyalty keeps popping up. There's very much a refreshing sense of devotion and partnership between client and agency, or agency and client too, which speaks volumes for how the web design industry seems to function. Instances where projects are 'pitched out' blind to a pool of fervent takers seem rare, avoiding a creative clamouring or stampede for business. In fact, if the 'pick me' process exists for an agency that succeeds in satisfying a client it might occur at the very start of a relationship

**"Our aim became to create an evergreen content hub that the client could add to and amend over time"**

that tends then to prevail. Often this is because a follow-up commission leads directly on from work the trusted agency is already attached to, which is precisely the case this month. For those who haven't spotted them already, Green Chameleon are a Bristol-based creative production studio that specialise in crafting experiences across digital, branding, packaging,

**Project duration**

6 months

**People involved**

*Nathan Riley*  
Design Director

  
*Paul Thomas*  
Developer

  
*Jess Caddick*  
Designer

  
*Mike Babb*  
Developer

  
*Tom Anderson*  
Creative Director

video and animation. Formed in 2012 and growing to boast a 12-strong team boasting designers, animators, UX specialists, developers and videographers, it's no surprise the talented team keep their impressive client list coming back. So when the people at Inside Asia Tours, a travel and tourism brand already working with GC for almost three years, wanted to deliver a new digital presence they knew where to go. In fact the previous 'Step Inside Asia' campaign between the two was featured in Web Designer 252's Lightbox, so IAT knew that 'When to Travel' would be in good hands. "This time the brief was designing a tool to showcase IAT's expert knowledge of local events, weather and generally the best time of year to visit the countries that they operate in," begins GC's Design Director, Nathan Riley. "Our aim became to create an evergreen content hub that the client could easily add to and amend over time whilst ensuring that the piece was visually striking and engaging for potential customers." ▶



## Closer calls

As we know, a strong defining factor for the production process of the 'When to Travel' project has been the connection between client and agency. It's clear from speaking to them that Green Chameleon takes considerable pride in the on-going working relationship it enjoys with Inside Asia Tours, partaking in a variety of campaigns and projects. "We're always thrilled when they come to us with a new brief as their in-depth knowledge and passion for travel makes working with them and coming up with concepts a real pleasure," enthuses Design Director Nathan Riley. "The focus of this last project was always in highlighting IAT's expert knowledge of the locations and the events that take place there, so working closely with them to ensure this was always

kept at the forefront of the creative was extremely important." Of course, it's all very well and good working so close and talking things out but only when the discussions can be productively applied. Creative design inspires such an emotional kind of evaluation and assessment of what feels right, that it demands some formal techniques to make a myriad of opinions practically useful. "Along with face-to-face meetings we present a lot of design work with InVision and always encourage clients to leave feedback via the commenting system. This enables us to keep track of changes and evolutions of the design as the project goes on, it also gives us a good record to refer back to if any issues crop up later down the line."

### Close communications

If the first little part of an initial brief was to represent the client's expertise in an effective way, another big focus was promotional and not purely in typical form. A second objective for the campaign would be as a tool that could also be used as part of the client's exhibition and trade show stand, making the need for versatile access essential. "With this in mind we knew we'd spend a lot of time refining the experience across mobile and tablet devices ensuring that when demos were given to potential customers the experience was as smooth as possible." All these requirements and early ideas were established immediately when the project kicked off with a brainstorming session to discuss possible avenues and get IAT's thoughts on their brief. "Following this the GC team went away and started putting together look-and-feel concepts along with user flows to explore how the site might work in more detail, and the overall art direction. During the early stages we try to bring in the development team as much as possible to ensure that what we are designing is buildable, however challenging.



## • • • • Site Highlight • • • •

**Designer Director Nathan Riley sticks his neck out and reveals the team's opinion on a favourite feature for showing the project's true colours.**

"Although a relatively small piece of the puzzle, the draggable timeline at the bottom of the month pages was a feature that seemed simple by design but the usability issues once prototyped became apparent, screen real estate being one of the biggest challenges. The resulting functionality, being able to drag and scroll through the months whilst the line animates was a real labour of love but we're glad we persevered and completed it for the final build."



and makes sense from a technical perspective. This also helps to inform the technologies and development approach and brings everyone involved onto the same page, which helps massively in keeping the project running smoothly, the scope agreed and defined, and deadlines hit." Having worked previously as part of a continuing client-agency relationship, both sides would, of course, know that strong and close communication would be easy enough. IAT are based just around the corner from Green Chameleon's Bristol office, so regular catch-up meetings were convenient and deemed vital to the producing a successful site for giving users easy access to the best events and weather at selected destinations.

### Evoking reactions

"As always we started with a kick-off meeting where ideas were thrown about between designers, strategists and developers," elaborates Creative Director Tom Anderson. "The content types to include were clear at this stage so the discussions revolved around how best to create an interesting but intuitive user

## "A website provides the opportunity to design not just how it looks but how it feels"

interface for the destination information being conveyed. The real challenge was how we cohesively tied that info together with locational-based weather and then split that content up between different countries. As we were dealing with events and dates we all settled on some form of timeline, which allowed users to view events sequentially, but at any time they could toggle to the weather for that time and destination. This seamless navigation between the two core content threads is where we focused our efforts and eventually spawned the final outcome." Once the GC team had sign-off on the initial art direction both sides had agreed, then a second stage of design commenced. Here the focus was the rollout of the design across all the inner pages pertaining to mobile and desktop views, before a final stage tackling the interaction design. "We would spend considerable time producing animations

of page transitions and load effects to help guide the development of the site," Nathan recalls. "We're strong believers that a website provides the opportunity to design not just how it looks but how it feels to a user and therefore the emotions and reactions it can evoke are revealed more vividly than from just a flat design."

### Visual assurances

Given that the project would represent a travel brand so versed in the value of good sightseeing, it's no surprise that engaging visuals would be pushed up front. The graphical work poured into honing the frontend would therefore utilise the collective talents of the whole GC crew, with a preference for keeping the deeper development team well informed of where things were headed. "As with all our projects it's a real team effort, we generally ►

use rapid visual prototyping in all our projects to start to generate a unique look-and-feel concept," Tom explains. "While developing a clear visual direction we carefully plan out the user's experience in tandem. Using wireframes and visual prototyping along with animated interaction prototypes we construct the final product. We're careful to include developers throughout the decision-making process in order to keep the designs realistic." From this work the team would have something tangible on the visuals to take back to the folks at IAT and gauge their reactions. Only after getting the OK and the necessary agreements here would allow GC's developers to start writing the code for running the show. "The development stage kicks off after all designs and mock-ups have been officially signed off," says Nathan. "As the site relies on a Laravel backend to allow the client to upload and manage content, we realised the task at hand of uploading over 200 events and weather details would take considerable time so we focused on delivering the CMS first whilst the frontend team push forward with their tasks with an eye to connect the two parts up in the final stages of the build."

### Arresting development

This Laravel-based content management system was custom-built specifically with the purposes of assisting the client. From here IAT would have the necessary control most clients

**"A particular challenge was adding the little animated touches to the Google Map powering the weather pages"**

demand when it comes to managing old and new content in-house throughout the life of the site. "On the backend, there's a content management system that was custom-built in Laravel so the client can manage all the site's content," reiterates Senior

Developer, Mike Babb.

"The data from that CMS is exposed to a React frontend that was so quick we actually had to slow things down a little in order to allow the nice transitions and animations to take place! We spent a long time reviewing and tweaking them so that every single one felt just right. Slow enough to have time to complete and flow nicely, but

# Together but apart

The period after development and delivery of 'When to Travel' revealed a curious additional challenge for taking the project live. As part of Green Chameleon's agreed service remit and relationship with client Inside Asia Tours, the campaign sites are actually hosted on GC's own servers. This is because each, including this one, are all built with different frameworks and technologies that are separate to IAT's main commercial website. "So one of the challenges faced here was the setup of a reverse proxy, which is essentially a set of server rules that allow users and search engines to see the campaign site as part of the IAT domain and therefore gain all of the

SEO value generated, even though it is hosted somewhere else," explains Tom Anderson, Creative Director. "This approach allowed us to continue offering maintenance and updates on the site without interfering with their main website. Plus, generally we have a very integrated relationship with Inside Asia Tours, which includes offering aftercare as well as continued support on digital marketing campaigns as part of an ongoing arrangement. An example of a previous campaign site we also created for them is 'Step Inside Asia', which was a resounding success, and there is more to come this year that we're all very excited about!"

not so slow that you ever feel you're like you're waiting for them to finish to get to the content you clicked to see."

Moving back towards the frontend coding, the majority of the site's animations were created using GSAP, the GreenSock Animation Platform. A popular choice for

many of the agencies we cover here, GSAP was GC's firm studio favourite for adding necessary creative flourish to 'When to Travel' and solving a few issues in the process. "A particular challenge was adding the little animated touches to the Google Map powering the weather pages, as we couldn't implement them directly with the Google Maps API and instead had to work the GreenSock magic on a cleverly positioned layer over the top of the map. The result is a beautiful, customised map bursting with animated interactions. Another significant challenge was getting the site's homepage to perform well. The background effects for the three screens of the country selector were originally JavaScript writing to HTML5 Canvas elements, but some browsers and devices really struggled to render the effects smoothly enough to meet our standards. In the end, we scrapped the JavaScript and Canvas approach and implemented them using some creative CSS trickery instead to



reach an almost identical-looking end result that performs much better."

## Final destination

Looking at the final launch site it's clear any development headaches were overcome

beautifully to meet the six-month project time. Looking back, the main project objectives were to create a tool for IAT to improve their search engine rankings while providing a rich, and most of all useful, brand experience for their prospective customers and existing customer base. So with that in mind, the GC guys did a certain amount of work in the lead up to delivery to meet the client's expectations and promise the site maximum exposure. "This tool needed to be scalable and have a long-term viability," Tom Anderson reflects. "The underlying goal of the project was to ultimately drive sales by achieving these objectives. We worked on the media distribution plan with the client as well as preparing our own distribution and promotional plan internally. Due to the

creative nature of the project and it being very visually unique a lot of the hard work is taken care of." That said, when asked about whether that hard creative work has ended in success, the answer is understandably subjective. Given their considerable experience, GC know it's always just as challenging to measure quantifiable statistics for the success of a project such as this, especially when talking about direct sale conversions. "Sales for IAT on average take 12-16 months to convert and rely upon multiple touch points all doing their job," concludes Anderson. "However, unique site visits are rising week on week and low bounce rates and high page visit durations suggest that the site is performing very well on the consumer side. The fact that this resource is an ever-expanding database of information that IAT can add to, increasing its value to the travel community and ultimately helping IAT establish a foothold in the search engine ranking across multiple long- and short-tail keywords, is also a testament to the success of the project." ■





Free  
WordPress  
guide  
worth  
£12.99



## Subscribe today and receive your free gift!

Every issue, delivered straight to your door



### Never miss an issue

13 issues a year, and you'll be sure to get every single one



### Delivered to your home

Free delivery of every issue, direct to your doorstep



### Get the biggest savings

Get your favourite magazine for less by ordering direct

## What our readers are saying about us...

"I've been a reader of Web Designer Magazine since the early days."  
@sixrevisions via Twitter

"Love the latest magazine that I purchased today"  
@navigation\_web via Twitter

"My favourite magazine!!!"  
@eduardomurillo via Twitter

# Pick the subscription that's right for you



## Subscribe and save 20%

- ✓ Automatic renewal – never miss an issue
- ✓ Pay by Direct Debit

Recurring payment of £34 every six months,  
saving 20% on the retail price



## One year subscription

- ✓ Great offers, available world-wide
- ✓ One payment, by card or cheque

A simple one-off payment ensures you never miss an issue for one full year. That's 13 issues, direct to your doorstep



Instruction to your Bank  
or Building Society to pay  
by Direct Debit

Originator's reference

7 6 8 1 9 5

Name of bank

Address of bank

Account Name

Postcode

Sort Code

Account no

Please pay Future Publishing Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Future Publishing Ltd and, if so, details will be passed on electronically to my Bank/Building Society. Banks & Building Societies may not accept Direct Debit instructions for some types of account

Signature

Date

UK £69 (saving 20% on the retail price)

Europe €81  USA \$103.20

Rest of the world \$103

## Pay by card or cheque

### Pay by Credit or Debit card

VISA  Visa  Mastercard  American Express

Amex

Card number

Expiry date

### Pay by Cheque

I enclose a cheque for £

Made payable to  
Future Publishing LTD

Signature

Date

## Your information

Name

Address

Telephone number

Mobile number

Email address

Postcode

Please tick if you want to receive any communications  
from Future and its group companies containing news,  
special offers and product information.

Please post this form to  
Future Publishing Ltd, 3 Queensbridge, The Lakes, Northampton, NN4  
7BF, United Kingdom

Order securely online to receive your free gift  
[myfavouritemagazines.co.uk/WBDPS18G](http://myfavouritemagazines.co.uk/WBDPS18G)



Speak to one of our friendly  
customer service team

Call **0344 848 2852**

This offer will expire on  
**10 July 2018**

\*Terms and conditions: This offer entitles new UK Direct Debit subscribers to pay just £34 every 6 months plus receive a bookazine worth £12.99. Gift is only available for new UK subscribers. Gift is subject to availability. Please allow up to 40 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. Prices and savings quoted are compared to buying full-priced print issues. You will receive 13 issues in a year. Your subscription is for the minimum term specified and will expire at the end of the current term. You can write to us or call us to cancel your subscription within 14 days of purchase. 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... UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: [www.bit.ly/magterms](http://www.bit.ly/magterms). Offer ends 10.07.18



# BREAKING THE MOULD

Wefail are distinctive. An agency of just three talented designers working in independent spaces, create striking digital experiences that hark back to the Gold Rush days of web development. Using complementary talents, this agency thinks and acts differently



# C

ompact and bijou amply describes Wefail. An agency of just three creative minds, Wefail have been able to push the boundaries of what is possible with today's digital technologies. As an agency that doesn't operate conventionally – don't ask them for a meeting – Wefail have their own unique perspective.

Co-founder Martin Wefail explains how he came to start an award-winning agency: "I've always loved computers. In the '80s I got my first Sinclair Spectrum and my mind was blown by the realistic graphics and how you could make a little man jump through dangerous levels with super high detail made out of massive blocks of primary coloured pixels.

"But I think it was later on the Atari ST that I began drawing pictures, then I obsessed over Photoshop on my first PC for years. Eventually I saw someone using Macromedia Flash and knew I had to have a go at that. By about 2002 I met Jordan Stone on a Flash forum and Wefail came along in 2003.

"I'd always been the person that loved art class in school. This led to art college and many years later – when I was learning Flash

## Who Wefail

**What** Interactive online campaigns, animated film and app development

**Where** Manchester, Los Angeles and a disgruntled Edmonton

**Web** [wefail.com](http://wefail.com)

## Key Clients

Shawn Colvin

Eminem

Jackass

Bob Schneider

American Horror Story

on my own – I found a free course in Manchester that taught you the basics of Adobe Macromedia apps. It was that class that really pushed me on."

As all agency owners know, naming their new business can be a tortuous experience. "Agency name came first," Jordan explains. "Martin definitely shepherded us along to 'wefail' after we brainstormed and piggybacked back and forth." Martin also comments: "By the time I met Jordan he already had his personal site SoFake. It was a negative name at a time when agencies were super positive about themselves, I loved the self-depreciation and Wefail came from that, though some people still don't read 'we fail' in the name and pronounce it 'wuhfal'. People are strange."

The website Wefail has created is striking. Martin outlines their approach to its design and how it has evolved: "I think it's important to make a new site now and again to keep it fresh. But for an agency like ours, it's also important to keep up with tech. If we still had our original Flash site up for Wefail.com the majority of users couldn't see it on their phones. There are other agencies that have always worked in the traditional way of a scrolling HTML page and that no doubt has a longer shelf life, but where's the fun in scrolling down a page of text?"

Each agency approaches its marketing differently depending on the type of clients they are looking to work with. For Wefail, the level of commitment their clients demonstrate is often a deciding factor as Martin explains.

"When we first came out we'd have clients lined up each year. But once ad agencies got spooked by 'the death of Flash' they fled for the hills and we suddenly found ourselves asking where everybody had gone. It's still very much word of mouth for us.

"We'll release new work and the next client might see it and get in touch. We've actively chased potential clients many times before,

## Breaking the mould

and I always find that if you have to aggressively chase them down, chances are they're going to back out somewhere down the line and leave you with a half-built job. Always best if they come to us. At least you then know they're committed."

How an agency chooses which clients to work with and which campaigns to work on has a number of factors. Often, how interesting the work on offer is will be the overriding decision, or if a client can add to the agency in some way. For Wefail, they have been lucky enough that the clients they have attracted are diverse and have offered engaging projects to work on.

"I don't think we've ever had the chance to pick and choose," says Martin. "We've been lucky, though, in that every job we've been offered has been something fun for a good client, from American Horror Story to Jackass, The Simpsons, Eminem, Amazon Prime, EMI musicians, they've all been clients that fit our style of work."

"Our latest work for Amazon Prime came through our very first client, Bob Schneider. We'd made an animated music video last year for him and a year later the musician Shawn Colvin saw the video and asked Amazon to get in touch to ask if we'd build her an animated film for her new album of lullabies. It's always great to have our work bring in new clients because they already understand our style, as we're not exactly mainstream."

"Yes, there's work we turn down, because of the nature of the things we make we've had some really out-there pitches from people. One I remember wanted us to build a live virtual auditorium where users would walk in and take a seat in a 2,000-seat stadium and could look around at each other, then a corn-based snack would walk out on to the stage and do a show before asking selected members of the virtual theatre to come on up and pick a prize in real time. This was 12 years ago, too. It's not like there was any way at all to make such a thing, there still isn't. We said no."

As an unconventional studio that pursues unusual design challenges, is there a project that embodies the Wefail philosophy? Martin continues: "I think our latest job for Amazon Prime defines how we work. It's a 20-minute animated music video for Shawn Colvin. A 20-minute piece of animation is an incredibly long time for two people to make, and we only had four months to complete it."

"Coming from animated web design we had to use all the tricks we'd learnt from working on the Web so the whole movie is coded in AnimateCC. Cameras, environment, puppet animation, everything was put together the same way we'd approach building an animated Flash website. The primary difference being the output to video and not having to worry about



**"I can't remember the last site I saw that I told anyone about and I think that's down to both the tools and the designers. We seem to have fallen into a one-page scroller habit and if that's all that's being made then why ask for more. I've seen no innovation in web design in a long time. It's unfortunately become the most boring design medium of all"**

Martin Wefail, Founder and Director

interaction. We really had to adapt for that job and all from two separate locations on opposite sides of the world. The workflow was perfect, though, and we're really happy with how it came out."

With Jordan also commenting: "I like to think we work in a box, together, trying to satisfy each other. If we can manage to do so simultaneously then that's great for us. Now if other people like it too, that's also great."

As no two projects are the same, the timeframe to complete the work can, of course, vary. "It's four to five months," Martin reveals. "Now, some of that time will inevitably be downtime. If the client pauses and needs to approve a direction, then we have to wait on that approval. It's the pauses that soon add up and you'll find a two-month contract creeps into three."

"I'd say always add 30 per cent more time on to your estimate for things that are out of your control. Who is assigned to a project depends on the format. On the Shawn Colvin music video for instance, it was just Jordan and myself working away on it around the clock. However, if it's an HTML5 web job then it's more Sebastian (DeRossi) and me that will take the wheel. The way we use JavaScript is pretty complex in our own little way, building a Flash-style site with all that movement and interaction is really not an easy task. Sebastian's a genius with CreateJS and makes the impossible work."

With a background in Flash developments, Wefail have gravitated towards the latest animation design tools. Jordan outlines their current toolset: "Adobe Flash and Adobe Photoshop used to carry us (and I used Cool Edit Pro for sound stuff) but nowadays we rely on AnimateCC and CreateJS for websites."



Welcome Who? History Contact  
About Work Pitch Manifesto



## Wefail

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

The new, award-winning, HTML5 JavaScript-powered Wefail site, built for your telephone, tablet, laptop and desktop. The art direction was created in Photoshop, then animations and environment put together in AnimateCC. Text-to-speech voices and background music were made in an audio editor before being crunched down to a loadable size.

Once this was complete, the overall CreateJS output was taken apart and supercharged in a text editor. Finally, a large chunk of time went into the site being tested and debugged on as many devices as possible, with responsive scale being handled by breakpoint.

We'd wanted to build an entertaining little site that used animation and audio to create an environment that worked on phones through to desktops. Something that brought all the visuals of our Flash work up-to-date through JavaScript and hopefully opened doors to new clients that want something different to what they're currently being offered in web design. There's a reason sites like this don't come along often, they're really hard to make. It took three months to put together between two of us, but it was worth it to see how far we could push the idea and create something original.

**Top** Striking to say the least, animation and sound meld to create an immersive environment no matter what device the site is viewed on

**Middle** Showcasing what can be done with AnimateCC and a large dose of imagination, Wefail flex their design muscles on each page

**Bottom** A tour de force of multimedia with art, animation and audio perfectly balanced to deliver an immersive experience



### Shawn Colvin The Starlighter

[youtu.be/xBnsWhPvrHY](https://youtu.be/xBnsWhPvrHY)

When we first started this project I really couldn't see the end. A 20-minute music video built with illustrations we had to put together in-house before animating in AnimateCC and setting up camera rigs, 3D rotation and depth of field all in 5K.

To make it harder on us we also decided to base the whole thing on a stage in a theatre, so the animated lullaby stories would be based within a parent environment of a theatre full of people.

On top of that we'd also rotate the stage, so the camera floated around to the backstage view, which meant designing both a front and a back view for hundreds of moving pieces.

We never make things easy for ourselves and halfway through I felt like we'd never complete it, but if you keep moving forward each day you eventually see the finish line. The video is a promotion piece for the new album of lullabies by Shawn Colvin and it now resides on Amazon Prime US.



**Left** Stunning art direction coupled with fluid animation illustrates the diversity that Wefail bring to the design community

**Below** Current animation techniques and tools bring to life a bygone era of miniature theatre reimagined for the iPhone generation





Martin also explains: "Yes, the same. But I'm a big fan of Procreate and Adobe Draw on iPad, so I sketch and make illustrations (all of the Shawn Colvin imagery was hand-drawn with Procreate) on the iPad then I take that over to Photoshop to slice.

"The usual workflow for me would be to draw a character on the iPad. Import that to Photoshop on the desktop. Cut each limb and place them in their own layer. And finally import those pieces to AnimateCC before setting them up with individual pivot points on their own layers to create the complete character. They

are then ready to animate and pass over to Jordan for import to the environment. Quite a lot of work but once you have that puppet set up you can reuse it throughout the overall animation."

With such a long history and experience how does Wefail see the evolution of the digital space and the tools including HTML5, CSS3 and JavaScript developing?

"I'd say they've already been shaping web design for the past six years, to be honest they're the only tools used by the majority and I don't think they're actually being pushed when it comes to fun interaction and leftfield ideas," says Martin.

He continues: "I can't remember the last site I saw that I told anyone about, and I think that's down to both the tools and the designers. We seem to have fallen into a one-page scroller habit and if that's all that's being made then why ask for more. I've seen no innovation in web design in a long time. It's unfortunately become the most boring design medium of all."

"One problem is that to access design now you absolutely need to know how to code at root level. In our day we had Flash and a huge amount of help with code snippets and the way the app itself held your hand and introduced you to the basics. One person could sit in their bedroom and build the most amazing foundation for a site within a day. I'd like to see AnimateCC picked up more with the output being CreateJS for Canvas. It's so much like the old days of designing in Flash that it's actually fun to work with."

Sebastian agrees: "Having the ability to use familiar tools such as AnimateCC is a massive advantage. With AnimateCC exporting JavaScript, old ActionScript 3.0 users like Martin can do everything... well, almost everything he



## Timeline

### 2003

Wefail is founded. Our first Flash site (<http://archive.wefail.com/wefail/>) launches our careers.

*Employees: 2*

### 2003

Bob was our first client and we still work with him today, a lovely man.

*Employees: 2*

### 2005

The handsome photographer Matthew and our infamous zoom engine.

*Employees: 2*

### 2007

Two people and a three-week deadline, a tough job, but one we wanted to pull off.

*Employees: 2*

### 2011

Our hipster platform game that gained six-million users on launch.

*Employees: 3*

### 2012

A puzzle site linked to American Horror Story. Users had to unlock each section room-by-room.

*Employees: 3*

### 2013

Monster's Socks was our first app. It won best children's fiction at the BolognaRagazzi Digital awards.

*Employees: 3*

### 2015

The Ratboy HTML5 interactive site built for mobile and desktop.

*Employees: 3*

### 2016

Our HTML5 game. Swipe to slap and feel alive!

*Employees: 3*

### 2017

The HTML5 rebirth of our studio site.

*Employees: 3*

## Agency breakdown

<b>1x</b>	Martin: Design, some development
<b>1x</b>	Jordan: Development, some design
<b>1x</b>	Sebastian: All development



did before. AnimateCC's IDE is so flexible that the old Flash masters at Wefail have no real reason not to use CreateJS to build mind blowing experiences like they did in Flash's 'golden' days. I am just happy to help them out!

"As with most Wefail projects, not only do we try to use the latest and greatest versions of the CreateJS Libraries, but we also use advanced features that most developers are unaware of. When we started wefail.com we decided that we wanted it to be fast, fast on both desktop and mobile devices, so a lot of care was taken when putting it all together. StageGL focuses on content that WebGL excels at rendering, which was perfect for the experience we wanted for wefail.com. Of course there are other secret optimisations that we did but Martin has forbidden me to share them here... I have said too much already."

Mobile is a key component of the web experience. What is Wefail's approach to designing and developing for different devices?

"We now design a site for mobile and then adapt it for desktop," Martin explains. "Our workflow will be getting the framework together and functioning as it should do on phones, then working through breakpoints up to the largest desktop screens. For wefail.com there was a lot to consider, things like the rasterised falling text-cards had to be intelligible on an iPhone 5's screen but also had to hold together on a 5K monitor. If we make a site I want to know I can use it the same way on all devices and desktop. I hate when you visit a

**"Be passionate with your work! Even if it's not the most exciting work, find something about it that will challenge you. Nobody likes a grey cloud. Martin likes to come up with the craziest ideas and my initial response is 'no way' and after a few hours I challenge myself to figure out a way to get it done"**

**Sebastian DeRossi, Developer**

## Rat Boy

[ratboy.co.uk/home/](http://ratboy.co.uk/home/)

Our first client site using JavaScript to create something that's more relatable to the old days of Flash. The site was based on a little piece of ActionScript that made a character move left and right through a parallax environment.

The first hurdle was to rework that code in JavaScript and with CreateJS, and get it working on Canvas so we could have it run on mobile. The tech side of this job went great. Sebastian fortunately proved how magnificent he was at interpreting ActionScript concepts into HTML5-based script.

It was interesting to see the response on social media from teenagers that haven't grown up with experiential sites and most of them decided it was some sort of witchcraft.

However, it was all down to some very hard work to make it all look very simple. Things like dynamic Tour Dates being displayed with Canvas were not so easy and we had to figure out ways to pull live data in without breaking out of the animated stage. Lots of work to make something look simple.

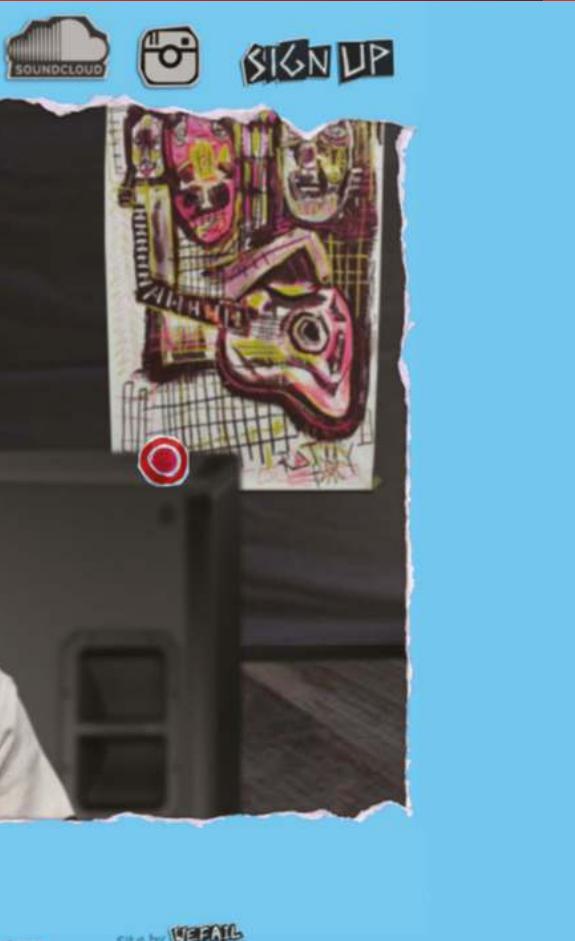




**Top** Wefail show their ample technical skills with this site that seems lightweight, yet packs a mighty technical punch no matter which screen it is viewed on

**Middle** Fast and intuitive was the guiding design principle behind this site. Click or tap a button to see more seamless animations

**Bottom** The immersive scrapbook feel of the graphics perfectly aligns with the artist and their music



site on mobile and they've put a video of it on there because they didn't get it working on smaller screens."

The partnership that is the foundation of Wefail is unusual. Looking back what advice would they give to anyone starting out? "Wefail has always been just Jordan and myself until I stumbled on CreateJS and began talking with Sebi and unfortunately for him he was too valuable to us, so he helps out a ton with the wizardry of CreateJS," says Martin.

"My god, advice for people getting into interactive web design? Become a vlogger instead, it's way easier and the money seems to be outrageous. Seriously though it's such a difficult job to make animated fun sites now that run on mobile, I'd say look to partner up with someone that covers your weaknesses. So, if you're a great animator, team up with a great coder, make something fun and post it up. God knows we need someone to shake things up or we're all going to be looking at one million variations of a one-page scroller with a sad little CSS animation for decades to come."

Sebastian also comments: "Be passionate with your work! Even if it's not the most exciting work, find something about it that will challenge you. Nobody likes a grey cloud. Martin likes to come up with the craziest ideas and my initial response is 'no way' and after a few hours I challenge myself to figure out a way to get it done. It's really rewarding to push yourself and usually your client, employer or co-workers will always respond positively."

And what of the future? Martin concludes: "We've just finished the Shawn Colvin music video and that was four months tied to the computer day and night, so at the moment we're dusting ourselves off and recharging.

Ideally I'd love for us to get a music video to animate or a cool little JavaScript site that's fun and aimed at leaving an impression on the user.

"There's not many studios left that make off-the-wall sites, as mobile pushed everyone into an HTML template that was guaranteed to work on all resolutions and give you a painless workflow. That in turn became what clients expect, to get a simple HTML page that followed the design of every other site being made at the moment.

"Technically it works, but as far as a memorable experience goes it's empty. I think we've lost the art of making people laugh or getting some kind of emotion out of them with a website. Everything's become so serious and follows the same structure, but being serious doesn't always make an impression. We need to step away from the one-page scroller."

Wefail strive to develop digital experiences that are unforgettable. In an environment that seems to have lost its excitement and ability to experiment, Wefail fly the flag for rich, edgy immersive spaces that hail the early days of the Web, when anything was possible. If there are champions for a return to experimental design, then it's Wefail.



## WEFAIL

### Founders

Jordan Stone and Martin Hughes

### Year Founded

2003

### Current Employees

3

### Location

Manchester, Los Angeles and a disgruntled Edmonton

### Services

HTML5 Design

App Design

Motion Design

TV and Movie Animation

Joy Makers and Heart Breakers

# DESIGN FASTER

> 50 must-try tools for today's designers and developers

# CODE QUICKER

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

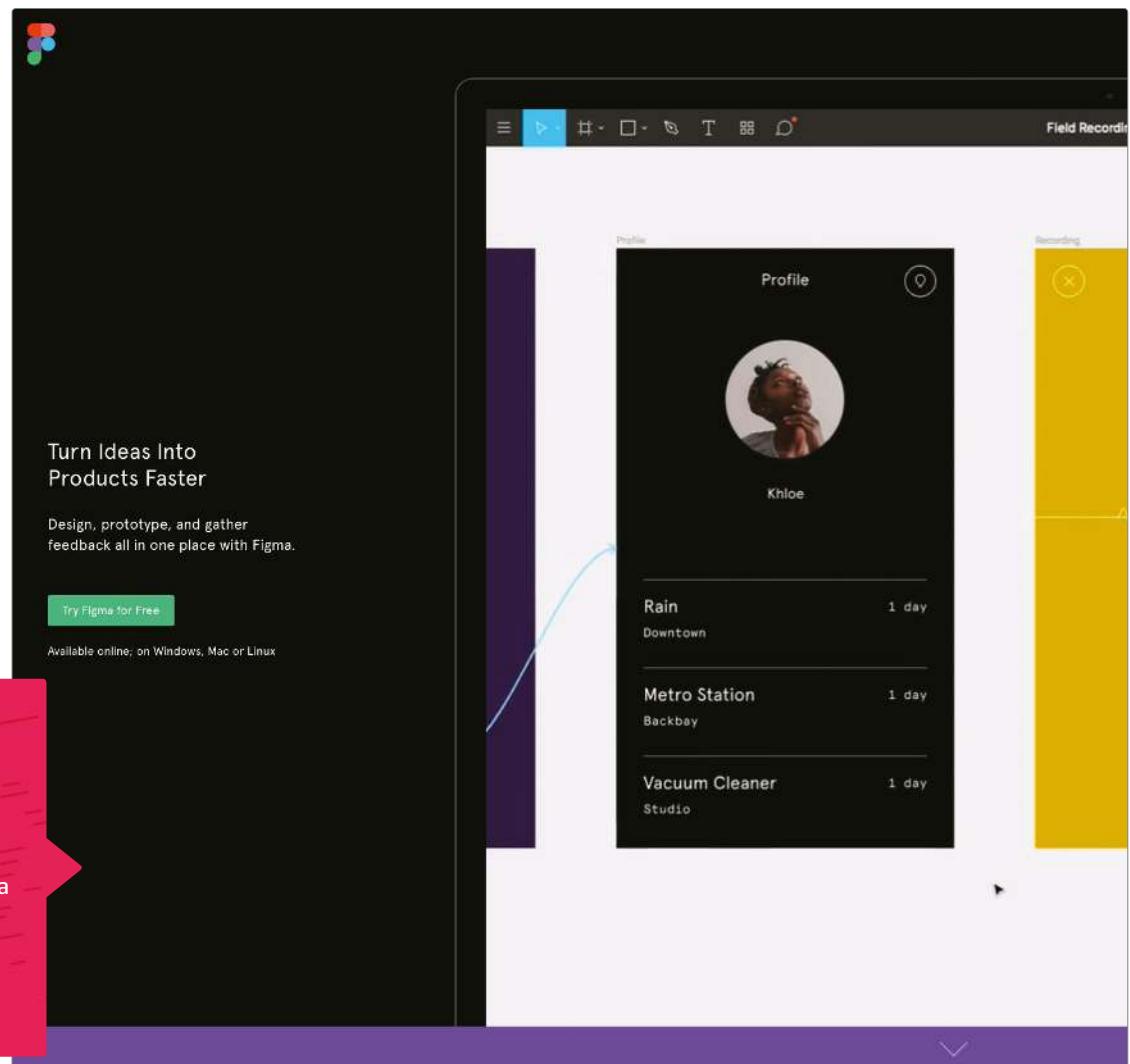
# PROTOTYPING AND BUILD TOOLS

**BIG HITTER****1 INVISION**[invisionapp.com](http://invisionapp.com)

InVision offers a way to string together individual static designs in order to create user flows. The tools are simple and easy to use, and allow syncing directly from Photoshop and Sketch before linking designs together. With powerful collaboration and comment features it's clear to see why so many designers use it.

**FREE WITH PAID PLANS****MUST TRY****2 FIGMA**[figma.com](http://figma.com)

Figma touts some impressive features, but the browser-based editor and cloud saving is the difference between most other design and prototyping tools. Figma 2.0 introduced new prototyping tools, bringing the abilities of multiple Adobe apps into one desktop and browser-based tool!

**FREE WITH PAID PLANS****3 MARVEL**[marvelapp.com](http://marvelapp.com)

Marvel, much like Figma, provides an all-in-one experience for design, prototyping and collaboration. The design UI is simple and beautiful, while offering all the tools you need. Marvel's prototyping tools enable both high and low fidelity prototypes to be created easily.

**FREE WITH PAID PLANS****4 AXURE**[axure.com](http://axure.com)

Axure is a wireframing and prototyping tool primarily aimed at software development. Axure isn't trying to offer everything. Instead, it offers really powerful tools for rapid prototyping, diagramming and documentation and it does them well.

**PAID WITH FREE TRIAL****5 GULP**<https://gulpjs.com>

Gulp is a frontend build tool used to perform any number of functions from compiling SASS to building SVG sprites. Gulp will save you time and perform a lot of the repetitive tasks required, enabling you to concentrate on writing code and building applications.

**FREE****6 GRUNT**[gruntjs.com](http://gruntjs.com)

Another frontend build tool that's used in the same way as Gulp with the aim of saving you time and effort when building the frontend of any website or app. These tools have become an important part of frontend development due to their ability to perform repetitive tasks.

**FREE****7 YARN**[yarnpkg.com](http://yarnpkg.com)

Yarn is a package manager. It works much the same as Node Package Manager (npm). Yarn caches every package making it incredibly fast to download and install dependencies, and with all of the same packages as npm so you can easily make the switch.

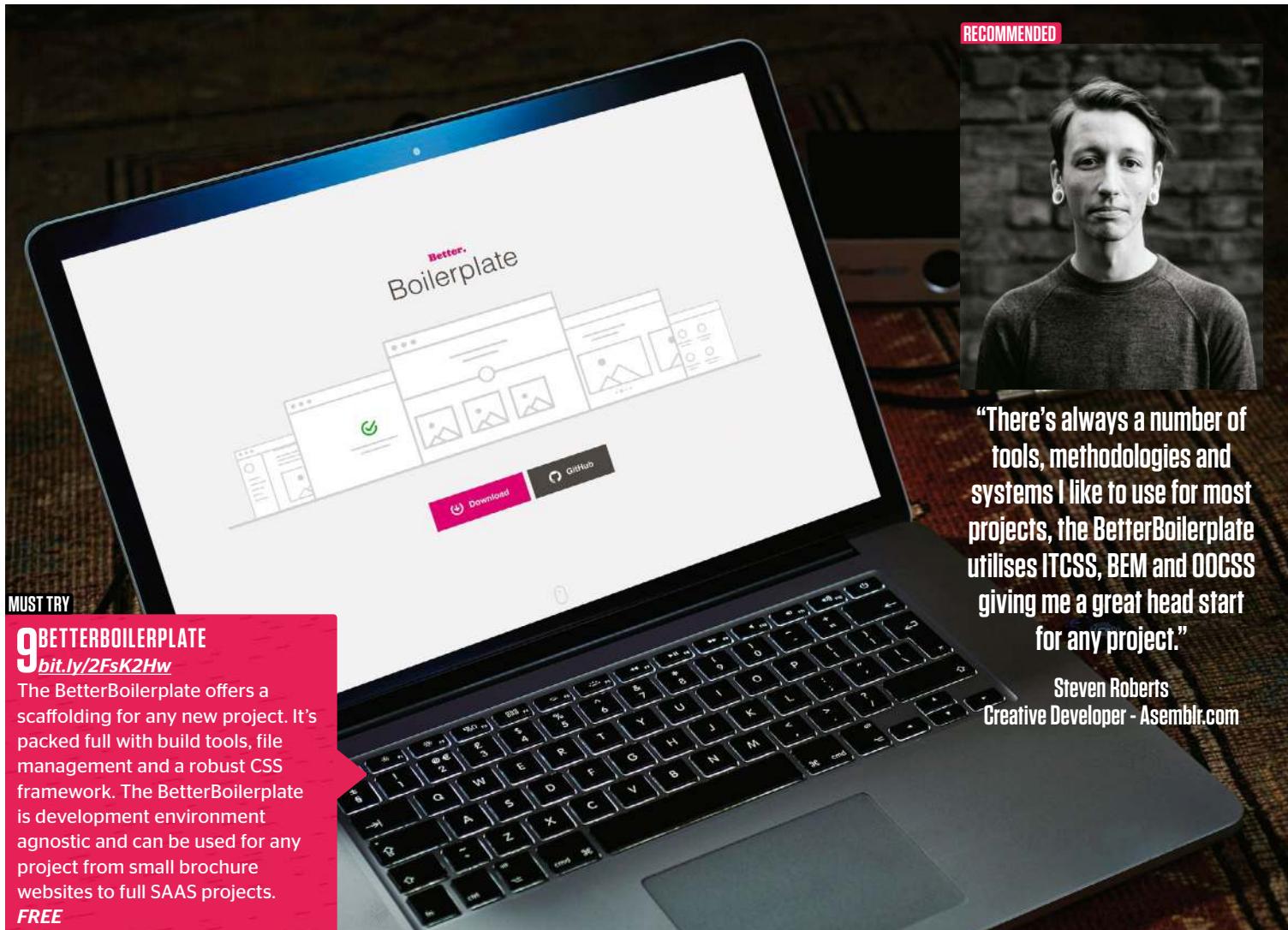
**FREE****8 WEBPACK**[webpack.js.org](http://webpack.js.org)

Webpack is a JavaScript module bundler that enables you to keep your JavaScript files small and focused. In its simplest form Webpack will then bundle them all into one minified JS file which can be included in your website or application.

**FREE**

**"MARVEL, MUCH LIKE FIGMA, PROVIDES AN ALL IN ONE EXPERIENCE FOR DESIGN, PROTOTYPING AND COLLABORATION. THE DESIGN UI IS SIMPLE AND BEAUTIFUL, WHILE OFFERING ALL THE TOOLS YOU NEED"**

# FRAMEWORKS



MUST TRY

## 9 BETTERBOILERPLATE

[bit.ly/2FsK2Hw](http://bit.ly/2FsK2Hw)

The BetterBoilerplate offers a scaffolding for any new project. It's packed full with build tools, file management and a robust CSS framework. The BetterBoilerplate is development environment agnostic and can be used for any project from small brochure websites to full SAAS projects.

**FREE**

## 10 VUE.JS

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

Vue is a progressive JavaScript framework. It can be easily integrated into an already existing app and used in one small part if needed. Equally, it can be used to build full single page apps. The framework is approachable and the basics can be easily grasped.

**FREE**

## 11 KNOCKOUTJS

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

KnockoutJS is a small and lightweight JavaScript framework – weighing in at just 59kB. It enables developers to create simple data binding for interactions with automatic UI refreshing. The KnockoutJS website also includes a really nice

interactive tutorial, which can help you get started.

**FREE**

## 12 FOUNDATION

[foundation.zurb.com](http://foundation.zurb.com)

Foundation is a responsive frontend framework created by Zurb. The framework is broken down into two different frameworks; Foundation for Sites – which rivals Bootstrap – and Foundation for Emails, which offers developers a much nicer way to develop email templates.

**FREE**

## 13 PREACT

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

Preact is another JavaScript framework, which touts itself as the "Fast 3kB alternative to React

with the same modern API." Its small size makes it quick to download, parse and execute your application. Geared towards performance and with a handy "Switching to Preact (from React)" tutorial on the Preact website.

**FREE**

## 14 SVELTE

[svelte.technology](http://svelte.technology)

Svelte is a component-based JavaScript framework, much like React, Angular and Vue but with a difference. Svelte compiles your code into vanilla JavaScript at build time – not at runtime like the others – which means no framework abstractions or lengthy load times, generating fast and performant apps.

**FREE**

RECOMMENDED



"There's always a number of tools, methodologies and systems I like to use for most projects, the BetterBoilerplate utilises ITCSS, BEM and OOCSS giving me a great head start for any project."

Steven Roberts  
Creative Developer - Asemblr.com

BIG HITTER

## 15 REACT

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

React is a JavaScript framework for building interactive user interfaces. The framework is developed and used by Facebook in all of its apps and websites. React aims to provide speed, simplicity and scalability for any project by providing a component based library.

**FREE**



# COLOURS & FONTS

## 16 TYPORAMA

[apperto.com/typorama](http://apperto.com/typorama)

Typorama is an iOS app that enables anyone – with design skills or not – to create awesome graphics. Packed full of image editing features and beautiful typography options, it's like Instagram on steroids. The app is frequently updated and has gathered a loyal following since its launch in 2015.

**FREE – WITH IN APP PURCHASES**

## 17 SPARKLINE TYPEFACE

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

The Sparkline Typeface enables you to create sparkline graphs using just a font, no need for complicated JavaScript libraries and configurations. While it can be limited, it's also a huge timesaver.

**FREE**

## 18 ADOBE CAPTURE

[adobe.ly/2r8VeEH](http://adobe.ly/2r8VeEH)

Adobe Capture is a smartphone app – available for Android and iOS – that enables you to take photos of anything inspirational, and then generate colour palettes, vector shapes and repeating patterns, all of which you can upload to Adobe's Creative Cloud.

**FREE**

## 19 FONT BASE

[fontbase.se](http://fontbase.se)

Font Base is a font management tool which works across all

**MUST TRY**

## 20 COOLORS

[coolors.co](http://coolors.co)

Coolors is a great little web app for when you need inspiration; fire up the app and hit the Spacebar to generate a random colour pallet. Once you find a colour you like you can 'lock' that colour and hit the Spacebar again to generate more colours.

**FREE**

platforms, offering a beautiful and intuitive UI for searching and sampling fonts. Font Base also enables fonts to be shared among teams so you'll never need to hassle your designer to send you the fonts ever again!

**FREE WITH PAID PLANS**

## 21 TYPEKIT

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

Typekit is Adobe's answer to Google Fonts. The website includes hundreds if not thousands of fonts, most of which can be used on the web. While Typekit isn't free it does have a much wider selection of fonts than Google Fonts does.

**FREE WITH PAID PLANS**

**Gothic A1**

**Designer:** Hwang IAC Co Principal Design

**About:** Gothic A1 is a Korean and Latin font that is a versatile sans-serif typeface with multiple weights and optimized spatial distribution.

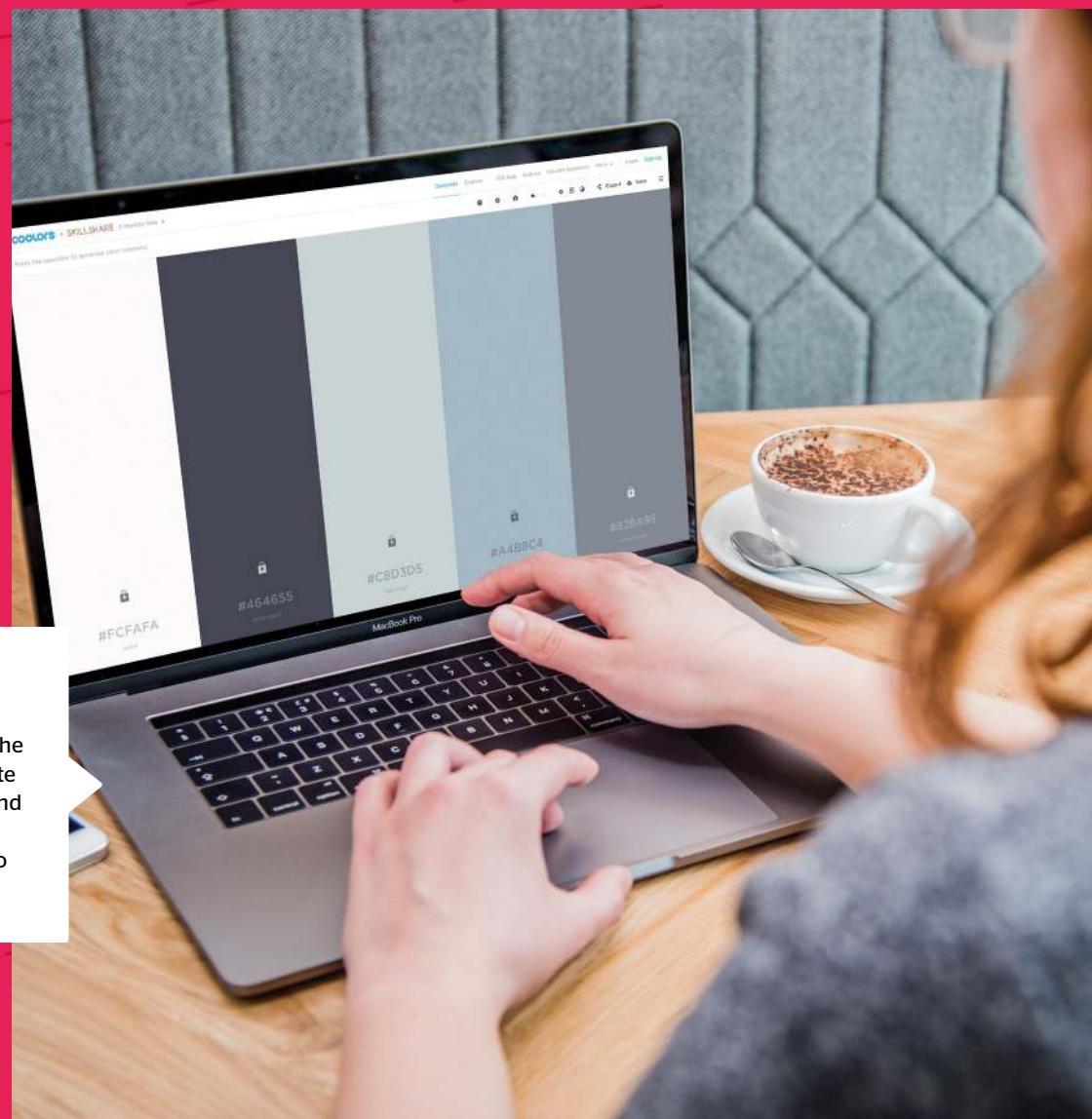
**Usage:** Gothic A1 is a Korean and Latin font that is a versatile sans-serif typeface with multiple weights and optimized spatial distribution.

## 22 GOOGLE FONTS

[fonts.google.com](http://fonts.google.com)

Google Fonts offers a large selection of free web fonts which can be used on any project. The interface is clean and minimal but enables you to easily see and test new fonts. It also offers a number of different ways to include the fonts in your project.

**FREE**



# WEB ANIMATION

## 23 KUTE.JS

[bit.ly/2rc0Znf](http://bit.ly/2rc0Znf)

KUTE.js is a JavaScript animation framework, built with fast code execution and memory efficiency in mind. With everything from two and three dimensional transforms to SVG manipulation KUTE.js is a strong contender for creating simple and lightweight animations.

**FREE**

## 24 ANIME.JS

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

Anime.js is a JavaScript animation library packed full of features. The library is performant and produces beautiful, seamless animations.

There's detailed documentation on the website and a variety of demos produced by the developer Julian Garnier on his Code Pen.

**FREE**

## 25 BLOTTER

[blotter.js.org](http://blotter.js.org)

Blotter is a JavaScript API for drawing unconventional text effects. Powered by three.js and underscore.js it is capable of producing some weird and wonderful animated text effects. The library offers five different material effects all of which can be customised.

**FREE**

## 26 ANIMATE.CSS

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

Animate.css is a small library of CSS animations, which can be used to add subtle (or not so subtle in some cases) animations to elements in your page. All you need to do is include the Animate.css code and then use the classes provided to animate elements in your application.

**FREE**

## 27 THREE.JS

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

three.js is a 3D animation library built for the web. three.js works by creating a three-dimensional

scene in which objects are rendered, a camera is then placed inside this scene. Some of the best animated websites on the web utilise three.js!

**FREE**

**MUST TRY**

## 28 SPIRIT

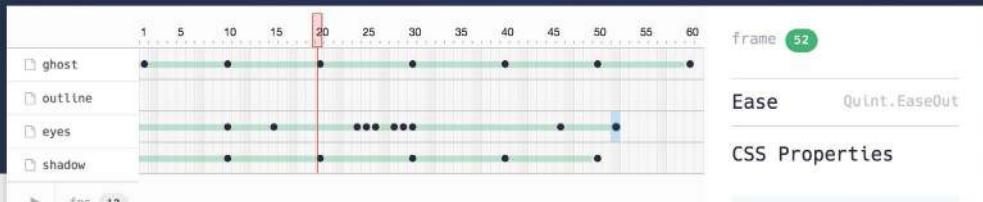
[spiritapp.io](http://spiritapp.io)

Spirit is a web animation timeline GUI currently in beta. The interface is intuitive and simple to use enabling you to make changes or additions and see the results in the browser in real time. Spirit could be a real game changer for web animation!

**FREE**

## SPIRIT

Easily animate objects  
on the web. For Designers &  
Developers.



**BIG HITTER**

## 29 GSAP

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

GSAP is a powerhouse when it comes to web animation. Starting life as a Flash animation plugin, it has since evolved to be one of the best web animation libraries available, enabling complex animations and time-based functions to be easily authored.

**FREE WITH PAID ADD-ONS**

# METHODOLOGIES & TESTING

## 30 SMACSS

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

SMACSS is a scalable and modular architecture for CSS. There is nothing to download or install here, just a whole host of ideas for structuring projects both large and small. The website includes a free eBook for you to download, read and study.

**FREE**

### MUST TRY

## 31 PINGDOM

[tools.pingdom.com](http://tools.pingdom.com)

Pingdom offers free performance testing. The speed test will give you an overall grade as well as a breakdown of everything your users download when visiting your website and handy tips for fixing any performance issues. Pingdom also offer uptime and performance monitoring in their paid accounts.

**FREE TEST – PAID ACCOUNTS**

## 32 GOOGLE PAGE SPEED TEST

[bit.ly/18oQghi](http://bit.ly/18oQghi)

Google Page Speed Test is an online tool used to analyse a website and give a score based on performance. You start by entering the URL of your website and running the test, Google will then suggest ways in which the site's performance can be improved.

**FREE**

## 33 OOCSS

[github.com/stubbornella/oocss/wiki](http://github.com/stubbornella/oocss/wiki)

OOCSS stands for Object

### RECOMMENDED



**“After deployment you must fine tune your site. Tools such as Pingdom and Google Lighthouse can measure the impact of not performing optimisations, such as file minification, compression, and image compression.”**

Jon Constantine  
CTO - Asemblr.com



FULL PAGE TEST

DNS HEALTH

LOG IN

SIGN UP

## Pingdom Website Speed Test

Enter a URL to test the load time of that page, analyze it and find bottlenecks.

URL

www.example.com

Test from

New York City, New York, USA

START TEST

Sign up for **free** to test your site every minute



SIGN UP FOR FREE

### State Colors

The following colors are used in the bars in the waterfall chart to indicate the different stages of a request.

- DNS Web browser is looking up DNS information
- SSL Web browser is performing a SSL handshake
- Connect Web browser is connecting to the server
- Send Web browser is sending data to the server
- Wait Web browser is waiting for data from the server
- Receive Web browser is receiving data from the server

### Content Types

### About Pingdom

Pingdom offers cost-effective and reliable uptime and performance monitoring for your websites. We use more than 70 global polling locations to test and verify our customers' sites 24/7, all year long. With Pingdom you can monitor your website's uptime, performance, and interactions for a better end-user-experience. Your customers will thank you.

**Nobody Likes a Slow Website** – We built this Website Speed Test to help you analyze the load speed of your websites and learn how to make them faster. It lets you identify what about a web page is fast, slow, too big, what best practices you're not following, and so on. We have tried to make it useful both to experts and novices alike.

In short, we wanted it to be a easy-to-use tool to help webmasters and web developers everywhere optimize the performance of their websites.

## 34 ITCSS

[www.creativebloq.com/web-design/manage-large-css-projects-itcss-101517528](http://www.creativebloq.com/web-design/manage-large-css-projects-itcss-101517528)

ITCSS or Inverted Triangle CSS is a methodology for CSS organisation. ITCSS advocates splitting your CSS into multiple layers: settings, tools, generic, elements, objects, components and trumps. These layers of styles get more specific per each layer avoiding a lot of problems with specificity.

**FREE**

## 35 ATOMIC DESIGN

[atomicdesign.bradfrost.com](http://atomicdesign.bradfrost.com)

Atomic Design is a methodology for creating design systems. It breaks design into five different levels: Atoms, Molecules, Organisms, Templates and Pages. Each level can and should include elements from the previous layers to build up designs.

**FREE – BOOKS AVAILABLE**

### BIG HITTER

## 36 BEM

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

BEM is a CSS methodology, which is widely used throughout the industry. The simplicity of its naming convention helps to tackle specificity problems in CSS – enabling elements to be scoped to a block name using a double underscore and modified using a double dash.

**FREE**



BEM – Boris Giltburg is a methodology that helps you to create reusable components and reuse styling in front-end development.

# IDEs & CODE PLAYGROUNDS

**BIG HITTER**



**37 SUBLIME TEXT**  
[sublimetext.com](http://sublimetext.com)

Sublime Text has been a popular text editor among developers for years now. It's packed full of useful features from multiple cursors to powerful find and replace tools, and also includes syntax highlighting. Sublime Text also has a rich ecosystem of plugins for almost anything!

**PAID WITH FREE TRIAL**

**MUST TRY**

**38 CODEPEN**  
[codepen.io](http://codepen.io)

CodePen is the go-to code playground for most frontend developers. The UI is clean and has support for most of the popular frameworks in HTML, CSS and JavaScript. The 'Picked Pens' section on the homepage is a great place to find inspiration and see some of the amazing community creations.

**FREE WITH PAID ACCOUNTS**

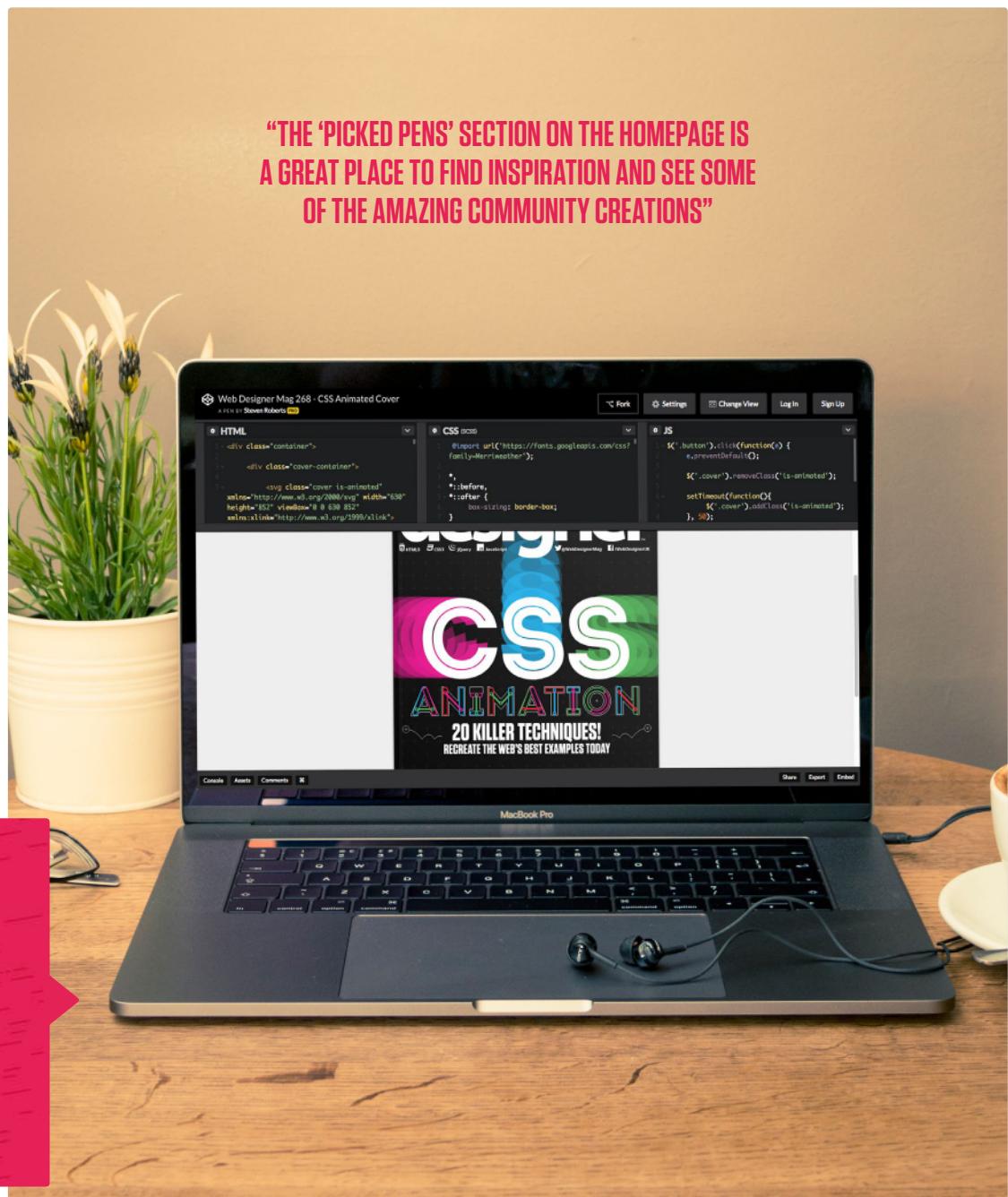
**39 ATOM**  
[atom.io](http://atom.io)

Atom is billed as "A hackable text editor for the 21st Century" and is built by the team at GitHub. There are a vast array of themes and packages to extend Atom, so it's definitely worth checking it out if you haven't done so already.

**FREE**

**40 VISUAL STUDIO CODE**  
[bit.ly/1KvI3PL](http://bit.ly/1KvI3PL)

Visual Studio Code is a free text



editor from Microsoft. It has all of the features you'd expect and some nice little extras, such as its built-in-git and command line features. Visual Studio Code is available for Windows, Mac and Linux, so is worth trying out.

**FREE**

## 41 BRACKETS

[brackets.io](http://brackets.io)

Brackets is a free text editor from Adobe, which has been built using JavaScript. The editor is packed

with useful features for developers. The inline editor feature can locate styles from clicking an element in a HTML file to make quick changes.

**FREE**

## 42 JSFIDDLE

[jsfiddle.net](http://jsfiddle.net)

JSFiddle started life as an online editor for MooTools and was called MooShell, before being renamed with support for all JavaScript frameworks in 2010. The platform

offers a fantastic playground for testing JavaScript.

**FREE**

## 43 NOTEPAD++

[bit.ly/1giSNHN](http://bit.ly/1giSNHN)

Notepad++ is only supported on Windows, but it can be used as a replacement for the built-in notepad. Supporting several key programming languages, it is minimal but functional. Its clean UI ensures your code is the focus.

**FREE**

# EXTENSIONS AND TOOLS

## 44 VUE.JS DEVTOOLS

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

Vue.js has optional devtools, which enable you to inspect component states, track events and access all of the data held in the virtual DOM. This is extremely useful when working with Vue.js and is a much nicer alternative to console logging all of these things.

**FREE**

### MUST TRY

## 45 SVGOMG

[bit.ly/2re4sj2](http://bit.ly/2re4sj2)

SVGOMG is an online tool for optimising SVG. Under the hood SVGOMG uses the incredibly powerful command-line tool SVGO to perform optimisations, but SVGOMG does so while visually showing you any changes the optimisation has made to an image.

**FREE**

## 46 PASTEBIN

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

Pastebin is great when you just need to share a snippet of code, you go to the website, paste the code, copy the link and send it. You can add optional settings such as syntax highlighting and choose how long your URL will last.

**FREE**

## 47 SITE PALETTE

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

Site Palette is a Chrome extension that enables you to grab the colour

scheme from any website, which you can then turn into a number of useful elements, such as Sketch Templates and Adobe Swatches, or you can print or download them.

**FREE**

## 48 WIREFRAMIFY

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

Wireframify is a Chrome extension that enables you to turn any website into a wireframe. It works by applying a custom stylesheet onto the website, which removes all images and modifies all of the

colours and borders, leaving you with what looks like a wireframe.

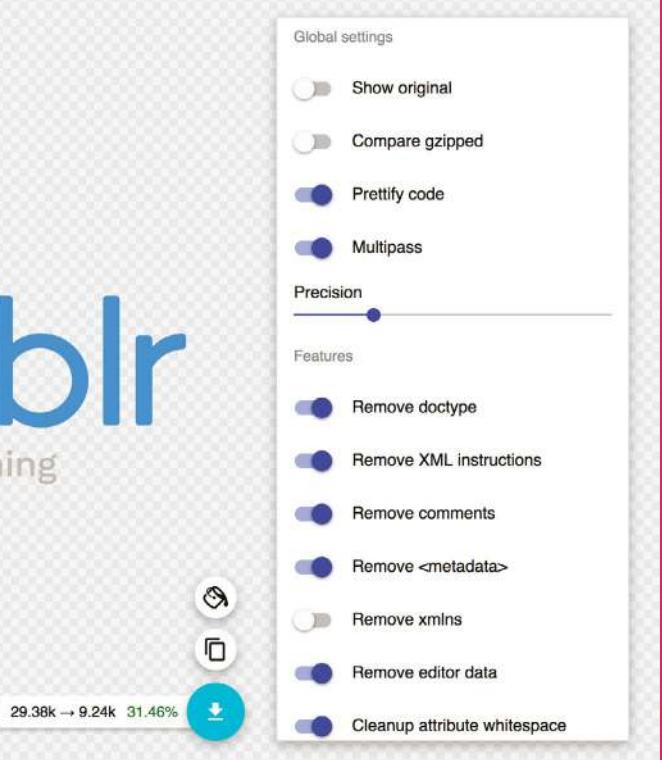
**FREE**

## 49 LOREM IPSUM GENERATOR

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

Both designers and developers rely on lorem ipsum text when producing mockups and testing type. lorem ipsum.io has a host of plugins for text editors and programming languages. The generator's page has some comical alternatives too.

**FREE**



chrispederick.com • Web Developer • User Agent Switcher • MicroBlog • Blog • About

**Web Developer**

Download Features Help Issues To-do Source

Sponsor United three offer. Get 10 free Adobe Stock Images. Join us today!

Enter search query Search

Donation

The Web Developer extension adds various web developer tools to a browser. The extension is available for Chrome, Firefox and Opera, and will run on any platform that these browsers support including Windows, macOS and Linux.

Download for Chrome Version 0.8 Download for Firefox Version 0.8

### BIG HITTER

## 50 WEB DEVELOPER TOOLBAR

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

The Web Developer toolbar has been around for almost as long as web design has, but it's still really useful for web development today, offering a number of really handy tools, from deleting cookies to displaying an image's alt text.

**FREE**

# web workshop

# Create ripple effects and transitions

Inspired by [themustafacelik.com](http://themustafacelik.com)

## Contextual information

The background is a project that Mustafa has worked on, as well as the title, there is contextual information on the role he undertook for that project.

## Ripple effect

Every time the user moves their mouse, the image in the background ripples and distorts in real time, instantly drawing attention to the movement.

## Direct to the work

If you would rather have a simpler interface then the work is on display in a simple list style from this menu.



## On screen navigation

To the left and right of the interface are quick links to 'about' and 'contact' which are commonly sought-after information.

## Change the slide

The up and down arrows at the bottom of the interface allow the image slide to change and the ripple effect is used again to create a transition between the images.



<comment>

What our experts think of the site

### Be unique

There are so many interesting effects that can be added to pages to make elements stand out, it's important to pick the right effects that work together to make your site unique rather than just a copy of other sites. That is what Mustafa has done here with the displacement ripples.

**Mark Shufflebottom, Professor of interaction design**

## Technique

### 1. Creating ripples

In order to create ripple effects like Mustafa Celik, PixiJS will be used since this provides simple displacement effects. Here the JavaScript code sets up the variables needed and loads the images to create the effect. Once the images are loaded the 'setup' function is called.

```
var app = new PIXI.Application(window.innerWidth, window.innerHeight);
document.body.appendChild(app.view);
app.stage.interactive = true;
var posX, displacementSprite, displacementFilter, bg, vx;
var container = new PIXI.Container();
app.stage.addChild(container);
PIXI.loader.add("img/ripple.png").add("img/bg.jpg").load(setup);
```

### 2. Creating the displacement

In the 'setup' function the displacement sprite is created that will create the ripple effect and this is added to a displacement filter. It's then set to move its anchor point to the centre of the image and positioned on the screen.

```
function setup() {
  posX = app.renderer.width / 2;
  displacementSprite = new PIXI.Sprite(PIXI.loader.resources["img/ripple.png"].texture);
  displacementFilter = new PIXI.filters.DisplacementFilter(displacementSprite);
  displacementSprite.anchor.set(0.5);
  displacementSprite.x = app.renderer.width / 2;
  displacementSprite.y = app.renderer.height / 2;
  vx = displacementSprite.x;
```

### 3. Finish the setup

To finish off the 'setup' function, the displacement filter scale is set and the background positioned. Notice the scale is '0' for the displacement, that's because it will be set to a height as soon as the mouse moves.

```
app.stage.addChild(displacementSprite);
container.filters = [displacementFilter];
displacementFilter.scale.x = 0;
displacementFilter.scale.y = 0;
bg = new PIXI.Sprite(PIXI.loader.resources["img/bg.jpg"].texture);
bg.width = app.renderer.width;
bg.height = app.renderer.height;
container.addChild(bg);
app.stage.on('mousemove', onPointerMove).
```

```
on('touchmove', onPointerMove);
loop();
}
```

### 4. Get the mouse

The next code just grabs the position of the mouse on the x-axis whenever the mouse moves. This will be used to trigger the amount of ripple displacement effect when the user moves their mouse. More movement will make the ripple bigger.

```
function onPointerMove(eventData) {
  posX = eventData.data.global.x;
}
```

### 5. Making it move

The 'loop' function continually updates the screen. A velocity for the x-axis is worked out using the position of the mouse and the ripple. This is then mapped onto the filter to give a value between 0 and 120.

```
function loop() {
  requestAnimationFrame(loop);
  vx += (posX - displacementSprite.x) * 0.045;
  displacementSprite.x = vx;
  var disp = Math.floor(posX - displacementSprite.x);
  if (disp < 0) disp = -disp;
  var fs = map(disp, 0, 500, 0, 120);
  disp = map(disp, 0, 500, 0.1, 0.6);
```

### 6. Finishing the code

At the end of the 'loop' function the sprite is scaled to the amount of displacement and filter scaled to the amount of depth it should have. Finally, the map function is declared that maps value ranges to new values.

```
displacementSprite.scale.x = disp;
displacementFilter.scale.x = fs;
}
map = function(n, start1, stop1, start2, stop2) {
  var newval = (n - start1) / (stop1 - start1)
  * (stop2 - start2) + start2;
  return newval;
};
```

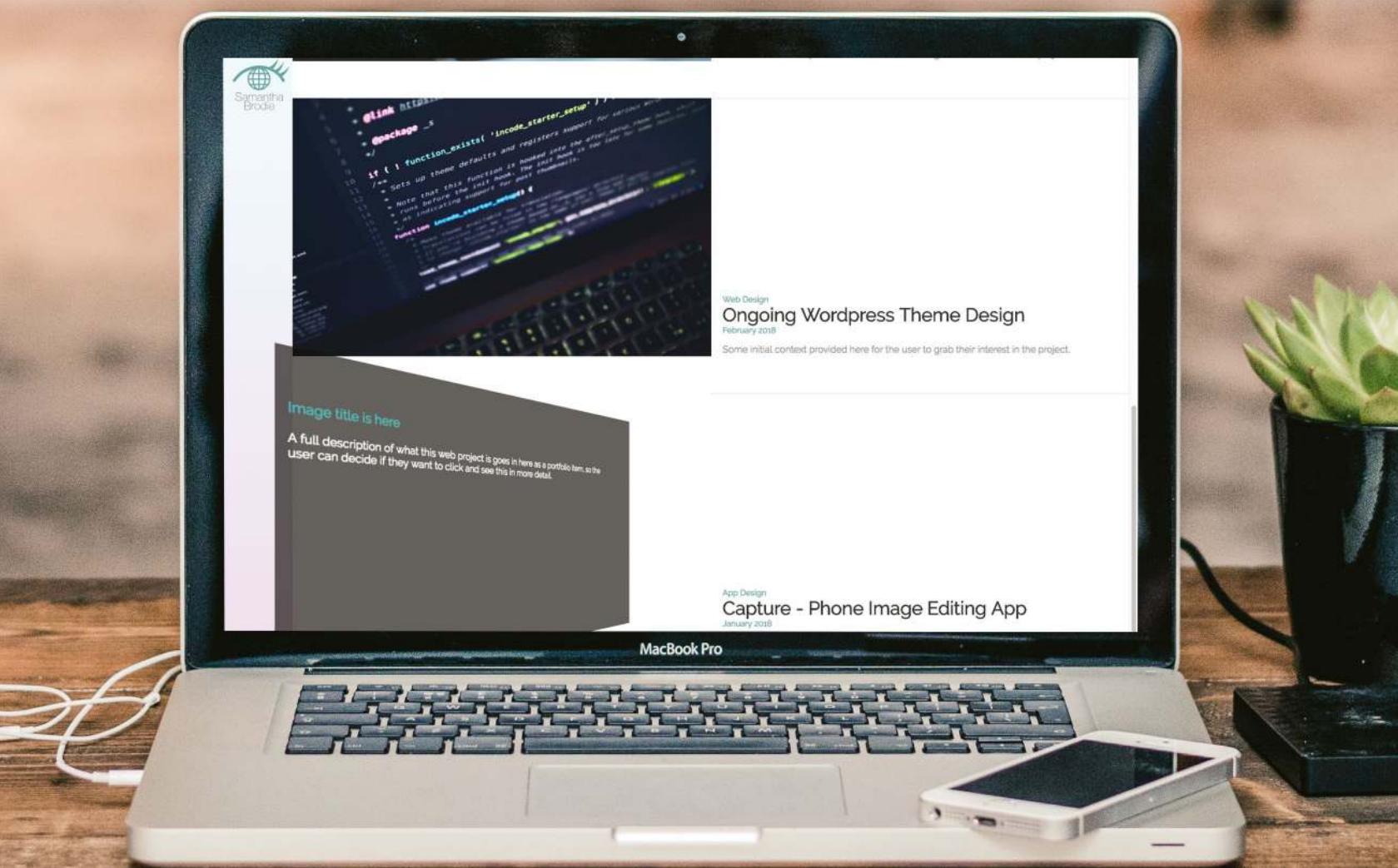


### EXPERT ADVICE

**Keeping it on display**  
So often many interfaces are hidden off-screen, but sometimes you can put everything you need on the interface in front of the user. Mustafa includes everything needed to navigate on the screen, but with an off-screen menu as an alternative browsing solution.

# Add interactive directional images

Use the Direction-Reveal library to respond to user movement and animate images in different directions with a selection of styles



 If you have your own portfolio site, you have probably spent more than your fair share of time trying to bring images of your work to life, making them interactive in some way that gets the attention of the user so that they want to explore the content further.

Direction-Reveal is a small CSS and JavaScript library that will really help out any kind of portfolio that requires image and description formatting. In this day and age nearly every site has some sort of interaction on a page before you actually click through to the content. The way Direction-Reveal brings a novel approach to this problem is by grabbing the direction that the user's mouse moves over the image. It uses this to trigger an animation, from the side that the user entered the image, so that the reveal of the content comes over the top. Again, as the user leaves that image, the content will animate away again, but this time using the direction that the user left the image. What you end up with is animations that don't follow the usual 'appear and disappear' rules. At first the user probably won't notice, but it soon becomes apparent as the user moves in and out of a list of images. This effect makes for a great way to grab the user's attention without being too over the top!

## 1. Starting off

Open the 'start' project folder in your code editor. Here you will see the index.html file and associated folders, css, images and js. Now open 'index.html'. You will see that some basic content has already been created. Before the closing body tag add the code here for the first direction-reveal element.

```
<div class="grid">
  <div class="space">
    <div class="direction-reveal direction-reveal-swing">
      <a href="#" class="direction-reveal__card">
        
        <div class="direction-reveal__overlay direction-reveal__anim--in">
          <h3 class="direction-reveal__title">Image title is here</h3>
          <p class="direction-reveal__text">A full description goes here</p>
        </div>
      </a>
```

## 2. Content to reveal

The last content was made up of an image and the reveal that will 'swing in' when the mouse is moved over the image. The content here shows a little context, but not too much as the main text description will show when the mouse moves over. This will be placed next to the image.

```
</div>
</div>
<div class="box">
  <div class="infoWrap">
    <h2>Web Design<br>
      <span>Wind - Augmented Reality Interface</span><br>
      April 2018</h2>
```

```
<p>Some initial context provided here for the user to grab their interest in the project.</p>
</div>
</div>
</div>
```

## 3. Add the style sheet

It's time to link up the style sheet now. In the head section add the link to the page-design.css' file. Open that file to see some initial design styling. This makes a start but more needs to be added to set up the grid and other elements of the design for the reveal elements.

```
<link rel="stylesheet" href="css/page-design.css">
```

## 4. Setting up the grid

The CSS styling here should be added before the media query as this is for the tablet and mobile design, just to keep everything full width in the browser. Later in Step 9, the media query will have the design set for a much larger screen, but this is taking a mobile-first approach.

```
.grid {
  display: flex;
  flex-wrap: wrap;
  flex-direction: row;
}

.grid > div {
  display: flex;
  flex-basis: 100%;
```

## 5. Full width images

The images in the grid are going to be set so they always fill the full width of the container that is holding them, regardless of what size that container is. The text on the direction reveal is set so that there is 50px of space on the right-hand side of the display.

```
.img-fluid {
  max-width: 100%;
  height: auto;
  vertical-align: middle;
  border-style: none;
}

.direction-reveal__text {
  padding: 0 50px 0 0;}
```

## 6. Title and overlay

The title of the revealed text is given some padding to the top so that it isn't too close to the top of the image. The whole overlay is set to include the border and padding in its sizing. This isn't so important here, but will be on the larger desktop styling.

```
.direction-reveal__title {
  padding: 50px 0 0 0;
  color: #56c1c1;
}

.direction-reveal__overlay {
  box-sizing: border-box;
}
```

## 7. Keeping it in check

At the bottom of each row will be a little space so that the images are not too close together. The info panel to the side of the image is in the flex display, so it will fill the size of the image. The text will be placed at the bottom of the display, due to the 'auto' setting for 'margin-top'.

```
.space {
  margin-bottom: 60px; }

.infoWrap {
  padding: 0 0 50px 20px;
  margin-top: auto;
  color: #6ca7a7; }
```

## 8. Text styles

The text inside the infoWrap is set now so that the heading displays at the right size and the span section for that is much larger with the right colour. Finally the paragraph section is set to display in a mid-grey colour, and that means all the text in that area is correctly styled.

```
.infoWrap h2 {
  font-size: 14px; }

.infoWrap h2 span {
  font-size: 32px;
  color: #333; }

.infoWrap p {
  color: #999; }
```



Samantha Brodie  
Bespoke Front End Designer

Hello! I help my clients create awesome digital experiences to enable meaningful connections with their users.

HTML, CSS & JavaScript served fresh with a pinch of dedication

Web Design Wind - Augmented Reality Interface

**Top** With the CSS in place at this point the design works fully on mobile and tablet-sized screens

## Mobile first

This just refers to the order of your CSS in which you target the mobile design first in your CSS then build up other screen sizes using media queries rather than starting with desktop first.

## Tutorials

Add interactive directional images

### 9. Moving into the media query

Inside the media query add the following section of code. This ensures that the grid moves out of the way of the side panel and each block is set to be a 50% column in the grid so that it flows correctly through the design for all the images and text that will be added. Save the style sheet now.

```
.grid {  
    margin-left: 120px;  
}  
  
.grid > div {  
    flex-basis: 50%;  
    flex-direction: column;  
}
```

### 10. Adding the JavaScript

Move back to the 'index.html' page and add the following code here for the JavaScript to link it up to the page. Save the 'index.html' page and switch over to the 'main.js' ready to add the JavaScript code in there that will control the way it works.

```
<script src="js/main.js"></script>
```

### 11. Swings and roundabouts

Scroll down in the JavaScript file until you find the comment to show where the code goes. The default setting for direction-reveal is the swing setting. It is set here to work on any class that has 'direction-reveal-swing' attached to it. This moves the new content in as a swing movement over the top of the image.

```
var directionRevealSwing = (0, _  
directionReveal2.default){  
    selector: '.direction-reveal-swing'  
};
```

## Making it work

The easiest way to think of Direction-Reveal working, is just to target the right CSS class in the JavaScript and it triggers the animation from there.



#### Top

Here the first direction reveal is shown moving in from the left because that's where the mouse has entered the image from

#### Right

On the second image, with the mouse moving from the bottom of the screen onto the image, you can see the direction-reveal here is sliding over the top from the bottom of the image

### 12. Next section

Save the 'main.js' file now and test the whole thing in the browser to see the content working and swinging in and out depending on which side the mouse enters or leaves the image. Now back in the 'index.html' add this content before the closing 'grid' div tag.

```
<div class="space">  
    <div class="direction-reveal direction-reveal-slide">  
        <a href="#" class="direction-reveal__card">  
              
            <div class="direction-reveal__overlay direction-reveal__anim--in">  
                <h3 class="direction-reveal__title">Image title is here</h3>  
                <p class="direction-reveal__text">A full description here.</p>  
            </div>  
        </a>  
    </div>
```

### 13. Sliding into place

The content in the previous step and this step is revealed using the 'direction-reveal-slide' class. This will slide the overlay over the top of the image. The image will remain in place and the overlay will just move. At the end of the tutorial there's a way to also push the image as well.

```
</div>  
    <div class="box">  
        <div class="infoWrap">  
            <h2>App Design<br>  
            <span>Pearson Airport Watch App</span><br>  
            March 2018</h2>  
            <p>Some initial context provided here for the user to grab their interest in the project.</p>  
        </div>  
    </div>
```

### 14. Slide reveal

Over in the 'main.js' add this under what you added

previously. This shows some further settings; the main points to look at are the type of animation and that this is also touch enabled for mobile devices. The touch threshold is how long it should be pressed. This is a quarter of a second and prevents it accidentally being pressed when scrolling.

```
var directionRevealSlide = (0, _  
directionReveal2.default){  
    selector: '.direction-reveal-slide',  
    itemSelector: '.direction-reveal__card',  
    animationName: 'slide',  
    enableTouch: true,  
    touchThreshold: 250  
});
```

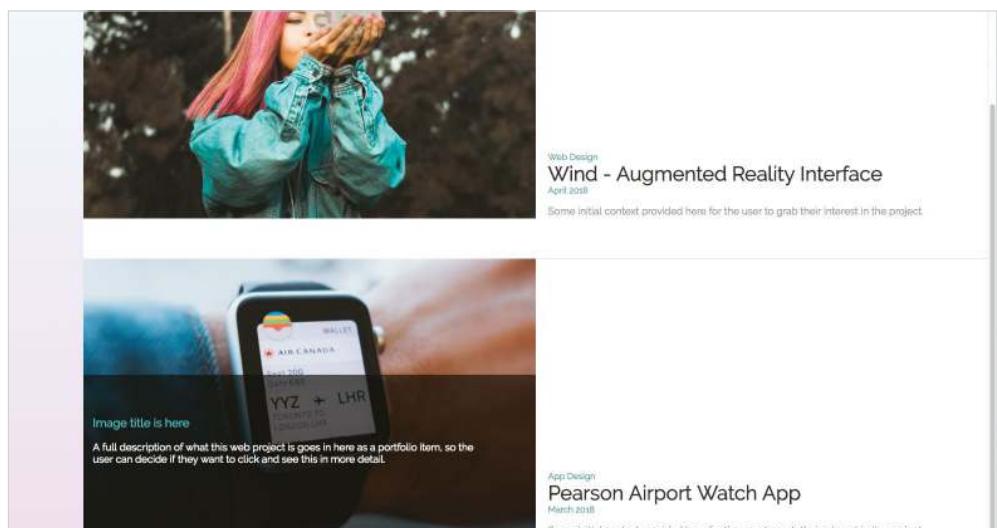
### 15. Testing the slide

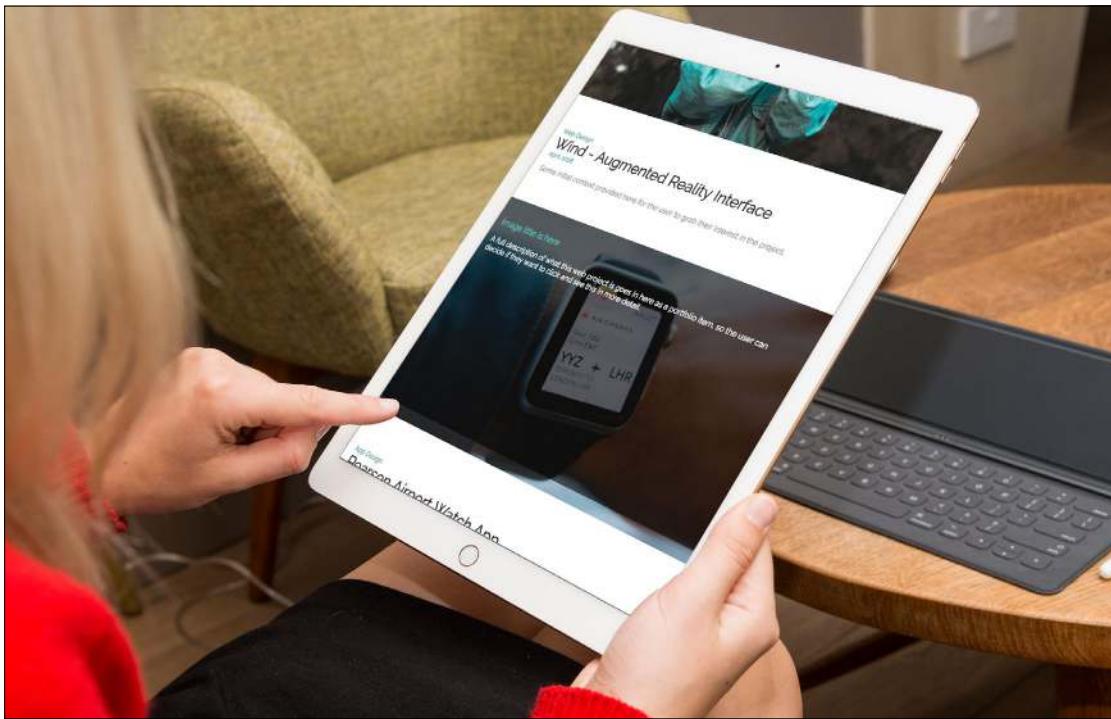
Save 'main.js' and refresh your browser to see the next section added, this time with the slide working from the direction of movement. Back in 'index.html' add this code before the closing tag of the 'grid' div. This places the next section onto the screen and this will use a 'rotate' animation.

```
<div class="space">  
    <div class="direction-reveal direction-reveal-rotate">  
        <a href="#" class="direction-reveal__card">  
              
            <div class="direction-reveal__overlay direction-reveal__anim--in">  
                <h3 class="direction-reveal__title">Image title is here</h3>  
                <p class="direction-reveal__text">A full description here.</p>  
            </div>  
        </a>  
    </div>
```

### 16. Complete the content

Adding in this section of code completes that area on the screen, so save 'index.html' as this row is now finished. The image, image overlay and side panel of text are all





## Making it work on mobile

You may be wondering exactly how this works on a mobile or tablet device, given that there is no mouse to rollover the image. In step 14 the 'enableTouch' option is set to true; do this on all images that you want to work on mobile. The way it works is that to enable scrolling when the user flicks, there is a threshold that it will wait for before triggering the animation. The default is 250 milliseconds, so when a long press is detected the animation will happen. That's the first part. If this is a link then the user needs to be taken to a link. Tapping a second time will fire the link and thus take the user to a new page. Because Direction-Reveal has thought about the usability of something like this, it makes it simple to work with.

ready for the JavaScript to control the rotation effect on this.

```
</div>
<div class="box">
<div class="infoWrap">
<h2>Web Design<br>
<span>Ongoing Wordpress Theme Design</span><br>
    February 2018</h2>
    <p>Some initial context provided here for the user to grab their interest in the project.</p>
</div>
</div>
```

## 17. Rotating reveals

Back in 'main.js' add this code under the code that was previously added. This takes any div with the class of 'direction-reveal-rotate' and adds the rotate animation for the overlay section onto this. Save this and refresh the browser, and the new content will have a rotation overlay that is triggered from the side that a user's mouse moves over it from.

```
var directionRevealRotate = (0, _directionReveal2.default)({
  selector: '.direction-reveal-rotate',
  animationName: 'rotate'
});
```

## 18. Next reveal

The final reveal section will feature a flip animation which swings the card around 180 degrees to reveal the content over the top. To start with just the overlay will flip around but later the image can also be flipped as well to give a 3D card effect to this animation.

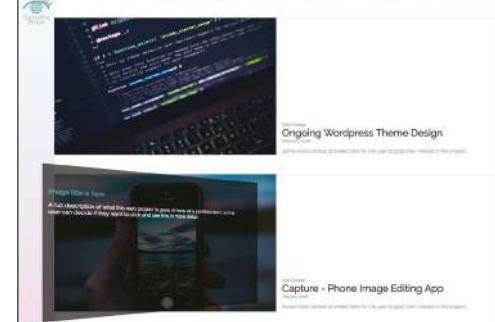
```
<div class="space">
```

```
<div class="direction-reveal direction-reveal-flip">
  <a href="#" class="direction-reveal_card">
    
    <div class="direction-reveal_overlay direction-reveal_anim--in">
      <h3 class="direction-reveal_title">Image title is here</h3>
      <p class="direction-reveal_text">A full description here.</p>
    </div>
  </a>
</div>
```

## 19. Side panel

Now the content for the side panel is created just to complete this section of the design. Save the 'index.html' page now and switch over to the 'main.js' file ready to add in the code that will pick up the selector and trigger the animation here.

```
</div>
<div class="box">
<div class="infoWrap">
<h2>App Design<br>
<span>Capture - Phone Image Editing App</span><br>
    January 2018</h2>
    <p>Some initial context provided here for the user to grab their interest in the project.</p>
</div>
</div>
</div>
```



**Top** The last direction-reveal is a flip effect that spins the overlay into position on top of the image

## 20. Flipping awesome

In the 'main.js' add this code just after the last code that you added. Save the page and refresh your browser. This section will flip over 180 degrees as you move your mouse over and it flips from whichever side you enter the image from. Let's make the image also flip as well.

```
var directionRevealFlip = (0, _directionReveal2.default)({
  selector: '.direction-reveal-flip',
  animationName: 'flip'
});
```

## 21. Final touch

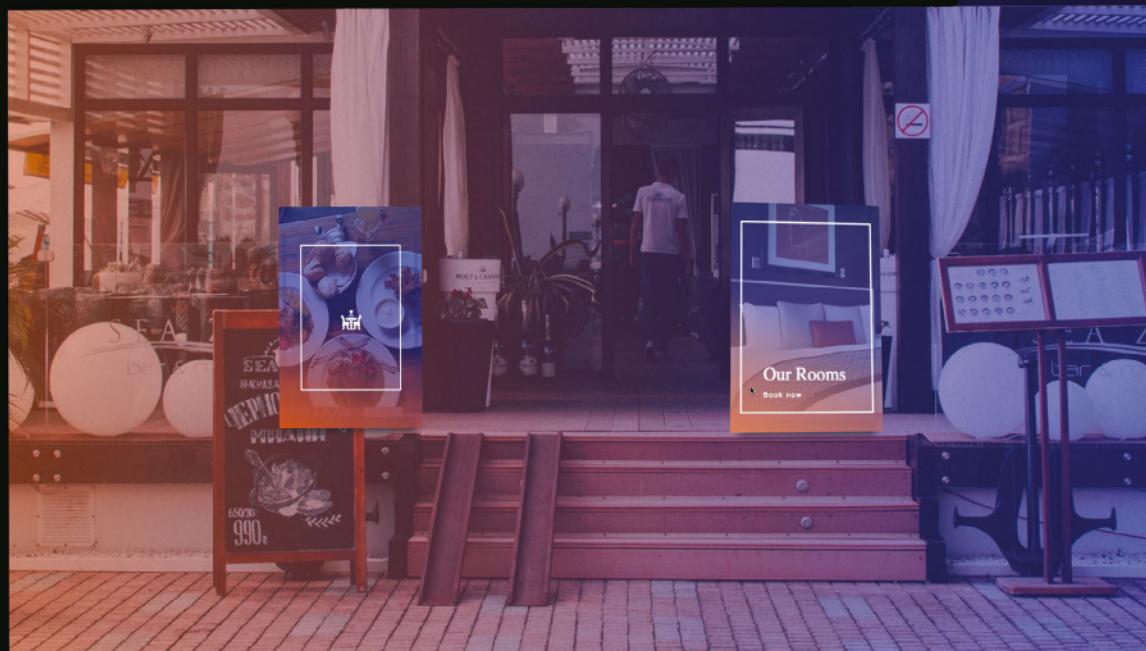
Back in the 'index.html' page, go back to the last image you added in step 18. Add an extra class 'direction-reveal\_anim--out'. Refresh your browser to see the image flip. If you use this class on the other images it will also make the image animate out as the overlay enters, so experiment with this for dramatic results.

```

```

# Build a tilting UI hover effect

Learn how to use CSS, SVG and JavaScript to add a tilting effect to UI elements to add engagement to landing pages





imation applied to user interface elements have now become much more prominent and are no longer the delightful surprises they once were - now, they're expected.

The explosion of prototyping tools in the past few years has given web designers an opportunity to create detailed interactions and animations. These tools also give us an opportunity to learn new ideas and understand the purpose of the animation without getting too heavy. However, in front-end development, DOM elements can now be easily animated using CSS, and together with JavaScript and SVGs, you can create some amazing UI animations.

So in this tutorial we will create a couple of UI elements for a fictitious restaurant and give them a nice hover effect that will include SVGs, CSS animations and a jQuery plugin that creates a tilting and glare effect. And to make things easy for us, we will do this using CodePen.

## 1. Getting set up

Open up CodePen at - [codepen.io](#) - and create a new pen. Within the settings, make sure the CSS section is set to Sass (.scss) and choose 'normalize' as the CSS base. The first bit of HTML we will add will be a section tag, and within that an anchor tag, which has two classes: 'animate' and 'tilt-one'.

```
<section class="content">
  <a href="#" class="animate tilt-one">
    </a>
  </section>
```

## 2. Figure tag and image

Within the anchor tag that we just added in the last step, create a '<figure>' tag and give it two classes: 'tilt-figure' and 'card'. Then choose an image that represents a restaurant and place that in between the figure tag.

```
<figure class="tilt-figure card">
  
</figure>
```

## 3. The image and caption

In the last step we used the '<figure>' tag. The figure tag specifies self-contained content such as photos and is intended to be used in conjunction with the <figcaption> tag and this is ideal for what we need. So within the figure tag and underneath the image link we've already added, we can add in a caption.

```
<div class="tilt-deco tilt-deco--overlay"></div>
<figcaption class="tilt-caption">
  <h3 class="tilt-title">Our Restaurant</h3>
  <p class="tilt-description">Book a Table</p>
</figcaption>
```

## 4. Border lines using SVG

SVG (scalable vector graphics) is now becoming one of the most effective and easiest ways of animating on the web (see the in detail section to learn more). So within the figure tag, we can use SVG to add in a border. We can target the SVG by giving it a class name(s) and then manipulate it using CSS in a later step.

```
<svg class="tilt-deco tilt-deco--lines" viewBox="0 0 300 415">
  <path d="M20.5,20.5h260v375h-260V20.5z" />
</svg>
```

## 5. Second UI element

The first UI element - which we will now call 'card' - will be for booking a table. The second one we will now add is going to be for booking a room. The only difference we have is that we've added a new class called 'data-tilt' which we will use for our jQuery. So make sure you also add that to the first card.

```
<a href="#" class="animate tilt-one">
  <figure class="tilt-figure card" data-tilt>
    
    <div class="tilt-deco tilt-deco--overlay"></div>
```

```
<figcaption class="tilt-caption">
  <h3 class="tilt-title">Our Rooms</h3>
  <p class="tilt-description">Book now</p>
</figcaption>
<svg class="tilt-deco tilt-deco--lines" viewBox="0 0 300 415">
  <path d="M20.5,20.5h260v375h-260V20.5z" />
</svg>
</a>
```

## 6. Sass variables

As we're using Sass we can specify a few default variables that we will use for background colours. Then underneath our variables we will make sure that all elements have the 'box-sizing' property by using the '\*' selector.

```
$lightBurgundy: rgba(214, 100, 40, 0.6);
$lightPurple: rgba(16, 11, 192, 0.5);
*, *::after, *::before {
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}
```

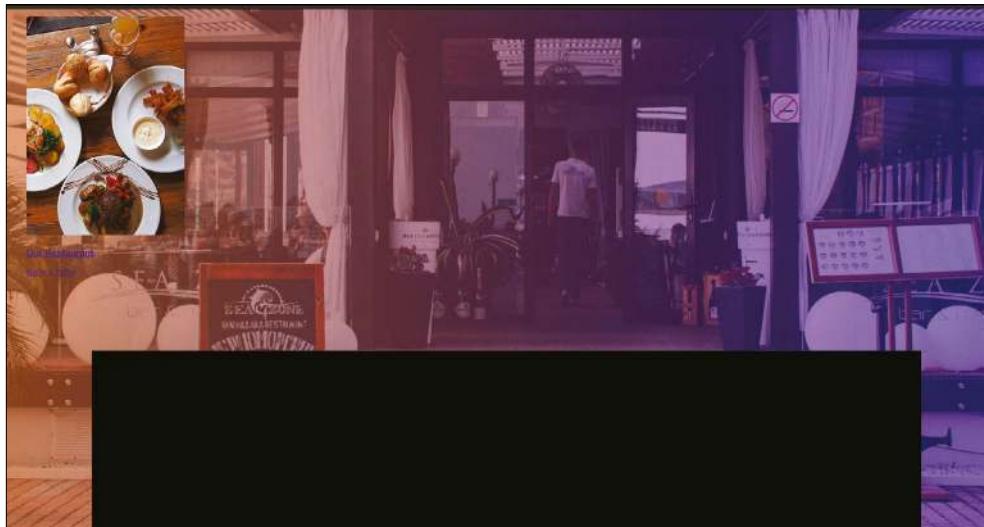
## 7. Background colours and image

By adding a background image that fills the whole page, this gives us a nice look and feel. To help with that even further, we will add a linear gradient overlay using the Sass variables we specified earlier. We do need to make sure though that our image has a height set to 100vh (viewport height). This will allow our image to stretch according to the viewport.

```
body {
  font-family: 'Roboto', Arial, sans-serif;
  color: #463f51;
  background: linear-gradient(to right,
$lightBurgundy, $lightPurple),
```

## Project images

When using CodePen for your projects and you need to use some images, it's often a good idea to use a free image hosting service, such as <https://image.ibb.co>



### Left

Using Sass variables, we've given our background image a nice linear gradient overlay

### Top

Using Flexbox on the content section we can use several Flexbox properties to align and centre our cards

## Tutorials

### Build a tilting UI hover effect

```
url('https://image.ibb.co/cHB5gx/architecture_bar_business_791810.jpg')  
center no-repeat;  
background-size: cover;  
height: 100vh;
```

## 8. Colour transition

All the content within our cards are wrapped within an anchor tag, and on hover this content will appear from the bottom of each card and animate up. To give this a better effect we will first set the default text colour to our light burgundy variable and then use a colour transition to change it to white for its final state.

```
a {  
outline: none;  
color: $lightBurgundy;  
text-decoration: none;  
transition: color 0.2s;  
}  
a:hover, a:focus {  
color: #fff;  
}
```

## 9. Using Flexbox

To help align and centre our cards, we can use Flexbox on the content section. This will allow us to use a number of Flexbox properties that will align, centre and space out our cards evenly.

```
.content {  
display: flex;  
flex-wrap: wrap;  
align-items: center;  
justify-content: space-evenly;  
padding: 0 0 4em;
```

## Figma for your icons

If you're good at designing and would like to try and design your own vector icons, then we would recommend trying out Figma <https://www.figma.com> - which is ideal for this type of work.



### Top

To create a white border we first remove the fill of our black rectangle and then give this a white stroke with a width of 3.5 pixels

### Right

The tilt jQuery plugin has been added and if you hover over the edges of a card you will get to see how far these tilt

```
height: 750px;  
min-height: 100vh;  
position: relative;  
}
```

## 10. Card sizing

We now need to set the size of the cards. Firstly though, we will clear any margins, and then set the width and height of both the figure tag and the image to 100%. Then lastly we need to make sure these are displayed as a block level element.

```
.tilt-figure,  
.tilt-image {  
margin: 0;  
width: 100%;  
height: 100%;  
display: block;  
}
```

## 11. Drop shadow

Creating a drop shadow using CSS is pretty straightforward these days, however we're going to do this a little different and create a hidden element which will go behind the main card and then put a shadow on that. Doing this will give us a bit more control of how the shadow looks.

```
.tilt-figure::before {  
content: '';  
position: absolute;  
width: 90%;  
height: 90%;  
top: 5%;  
left: 5%;  
box-shadow: 0 30px 20px  
rgba(35, 32, 39, 0.6);  
}
```

## 12. Card overlay

The 'tilt-deco' class will be used a couple of times and its purpose is to create a shape that we will place over the main card, and then we will create a linear gradient overlay using rgba colours. We will tone down the

gradient colour in a later step.

```
.tilt-deco {  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 100%;  
overflow: hidden;  
}  
.tilt-deco--overlay {  
background-image: linear-gradient(45deg,  
rgba(226, 60, 99, 0.4), rgba(145, 58,  
252, 0.4),  
rgba(16, 11, 192, 0.4));  
}
```

## 13. The white border

By using the shape we created in the last step, we can now use our SVG path and create a white border that will sit inside our card. Firstly we should make sure the fill is set to 'none' and specify that the stroke is set to white with a width of '3.5' pixels.

```
.tilt-deco--lines {  
fill: none;  
stroke: #fff;  
stroke-width: 3.5px;  
}
```

## 14. Caption, title and description

Our caption, which holds our title and description, will be positioned at the bottom of the card. Then we will give the title and description some default styles. We've given the description a good amount of letter spacing so it appears that we're using two different fonts.

```
.tilt-caption {  
position: absolute;  
bottom: 0;  
width: 100%;  
padding: 4em;  
}  
.tilt-title {  
margin: 0;
```



## The complete guide to SVG

By Steven Roberts 18 days ago Web design

Discover why you should be using Scalable Vector Graphics and how to design and implement them.



### More about SVG

Poorly supported and largely ignored for most of the 2000s, in 2017 all modern web browsers can render SVG – and most vector drawing programs export it. SVG is the one graphic format that most closely responds to current web development demands of scalability, responsiveness, interactivity and performance.

SVG is an XML language and file format, which allows you to code two-dimensional graphics that scale and can be manipulated via CSS or JavaScript.

All SVG graphics are plotted on a coordinate system of at least an x and y axis. When authoring SVG, we give the browser instructions on where to plot points on the coordinate system and connect them. By connecting the plotted points we can create shapes, lines or paths.

The `<svg>` tag allows you to add in custom classes, which then gives you more control using CSS or JavaScript. Also within the tag you're given a attribute called 'viewBox'. The value of the viewBox is a list of four numbers that represent the 'x' and 'y' axis as well as the width and height, all in that order. `<svg class="yourClass" viewBox="0 0 300 415"> <path d="M20.5,20.5h260v375h-260V20.5z" /></svg>`

For more SVG info go to <https://bit.ly/2ryLgf>

```
font-weight: normal;
font-size: 2.2em;
font-family: 'Roboto', serif;
line-height: 1;
}

.tilt-description {
margin: 1em 0 0 0;
font-size: 0.85em;
letter-spacing: 0.15em;
}
```

### 15. Extra overlay colours

As I mentioned in a previous step, we are going to tone down the first overlay by adding another. We could use the Sass variables again, but we wanted this to have a darker overlay at the bottom so the text blends better for when we animate the text up on hover.

```
.tilt-one .tilt-deco--overlay {
background-image: linear-gradient(20deg,
rgb(214, 100, 40), rgba(46, 39, 173, 0.58),
rgba(53, 74, 165, 0.6));
}
```

### 16. Animations

This is where the fun begins. Now we can begin to animate the title, the description and the white SVG border. As we said in the last step, you can now see the title and description appear from the bottom colour and it gives us a nice effect.

```
.tilt-one .tilt-deco--lines {
transform: scale3d(0.8,0.8,1);
transition: transform 0.4s;
}

.tilt-one:hover .tilt-deco--lines {
transform: scale3d(1,1,1);
}

.tilt-one .tilt-title,
.tilt-one .tilt-description {
```

```
transform: translate3d(0,80px,0);
opacity: 0;
transition: transform 0.4s, opacity 0.4s;
}

.tilt-one:hover .tilt-description {
transition-delay: 0.1s;
}

.tilt-one:hover .tilt-title,
.tilt-one:hover .tilt-description {
transform: translate3d(0,0,0);
opacity: 1; }
```

### 17. Optional icons

This is an optional step and will involve creating a more detailed SVG icon(s). There's plenty of online resources that allows you to download icons in SVG format, so try and find a couple that represents a restaurant table and a bed.

```
.svg-icon {
transform: translate3d(130px, -200px, 0);
opacity: 1;
transition: transform 0.4s, opacity 0.4s;
}
```

### 18. Animate the icons

In the previous step we set the opacity of our icons to be visible. And then we positioned it in the centre. In this step we will now add in some animation that will move the icon up and as it does so, it will gradually fade out.

```
.tilt-one:hover .svg-icon {
transform: translate3d(130px, -250px, 0);
opacity: 0;
transition: transform 0.4s, opacity 0.4s; }
```

### 19. Adding the tilt jQuery

To create the tilting effect we are going to use a jQuery plugin which you can get from its github repo: <https://github.com/gijsroge/tilt.js>. Do make sure you include

the CDN link into CodePen (within the JS settings) and then in the JS section add in the jQuery shown.

```
(function( $ ) {
"use strict";
$(".card").tilt({
maxTilt: 10,
perspective: 1400,
easing: "cubic-bezier(.03,.98,.52,.99)",
speed: 1200,
scale: 1.05
});
```

### 20. Adding some glare

There's quite a bit we can do with this handy little plugin, so go ahead and play around with the values. However, one nice effect we can use is the glare effect. We don't want too much of a glare so set it to around '0.3' and you'll see how nice and subtle this now looks.

```
(function( $ ) {
"use strict";
$(".card").tilt({
maxTilt: 10,
perspective: 1400,
easing: "cubic-bezier(.03,.98,.52,.99)",
speed: 1200,
glare: true,
maxGlare: 0.3,
scale: 1.05
});
```

### 21. Final thoughts

UI animation and interactions are certainly a nice skill to have. You can't develop a website, web app or mobile app without using some sort of animation. But try not to over do it as this could have a negative effect on the user experience.

# Animate text using WebGL techniques

In this tutorial, you will learn how to animate text on our webpage using Three.js and advanced WebGL techniques

The monitor screen shows a teal background with white text that reads "WE WIND". Above the text is a 3D rendered effect of many small, thin, white lines radiating outwards, resembling wind or a fan. At the bottom of the screen, there is a small white text overlay that says "click and drag to control the animation". The monitor is positioned on a desk next to a keyboard and a mouse. To the left of the monitor, there is a red circular button with the text "Subscribe to Web Designer" and "Go to page 34 to learn more". Inside the red circle, there are two small magazine covers for "Web Designer" magazine.



The beauty with front-end development is that, as a developer, you have access to numerous technologies that you can use to display interactive content on the web. From the popular CSS and Canvas to SVG and plugin-based alternatives such as Silverlight and Flash, the options are endless. However, the evaluation of some of the top ranked interactive websites (<https://www.awwwards.com/websites/web-interactive>) reveals a significant trend: the wide scale adoption of WebGL techniques in displaying content in landing pages. The evaluation revealed that top ranked websites not only adopted 3D graphics on the web platform, they also introduced a rich interactive experience, a feat that was previously possible through using plugins. As such, the inspiration in developing the tutorial arose from the desire to create simple yet impactful interactive content using WebGL techniques.

In developing the tutorial, we use Three.js, an open source, lightweight JavaScript library for creating 3D graphics on the web. Essentially, the library acts as an abstracted layer on top of WebGL, making it much easier to use. Three.js enables you to create advanced graphics by writing only a few lines of code. In this tutorial, we will learn how to use the library to create an interactive text animation on your website.

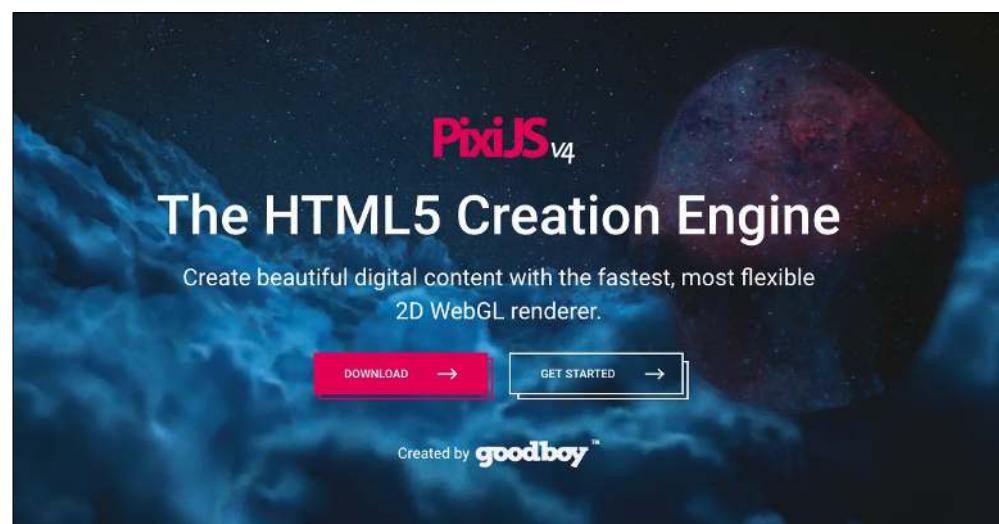
## 1. Getting started

Begin by creating a folder, 'text\_reveal', on your desktop to store the tutorial files. Create two additional folders within it: CSS to store the styling files, and JS to store JavaScript files. HTML files will be stored in the root folder.

## 2. Creating the HTML page structure

Open your code editor and create an index.html document to contain mark up for the main webpage. Begin by creating the basic structure and give a suitable title to the page.

```
<!DOCTYPE html>
<html lang="en">
```



```
<head>
<meta charset="utf-8">
<title>Three JS Text Animation </title>
</head>
<body> </body>
</html>
```

## 3. Adding content to the HTML structure

Essentially, we would like to display the animation on the main area of the screen and have a lower section that gives instructions to users regarding their actions on the page. Create two divs in the body section: three-container where the animation will be displayed and instructions where the user instructions are displayed. Render the page in your browser.

```
<body>
<div id="three-container"></div>
<div id="instructions">
    click and drag to control the animation
</div>
</body>
```

## 4. Linking the CSS file

Currently, the page has no styling to it and it appears very basic. We are interested in styling the two divs in order to have them placed at the centre of the page. Open your code editor and create a 'styles.css' file inside the CSS folder. Create a link to this file in your HTML document by adding the following code in the head section. Render the page in your browser.

```
<link rel="stylesheet" href="css/styles.css" >
```

## 5. Styling the webpage - styling the body

Begin by setting the margin to 0 for all elements in CSS.

## Three.js alternatives

Despite its popularity, Three.js is not the only JavaScript library making WebGL easier. Other popular libraries include Greensock, Play Canvas and Pixi.js. However, Three.js is recommended given that it is easier to learn.

**click and drag to control the animation**

### Top

This illustrates the structure of the page. Only the instructions are displayed since JavaScript controls the main div

### Right

The cloth animation (<https://bit.ly/2wD34vM>) provides the style applied in this tutorial. Shown here without the cloth or ball aspects.

```
ion / cloth
ion / keyframes / json
ion / scene
ion / skinning / blending
ion / skinning / morph

/ array
/ cinematic
/ logarithmicdepthbuffer
ng
ng / advanced
ng / intersection

/ texture
s / anaglyph
s / parallaxbarrier
s / peppersghost
s / stereo
uffer / texture
ries
ries / parametric
ry / colors
ry / colors / blender
ry / colors / lookuptable
ry / convex
ry / cube
ry / dynamic
ry / extrude / shapes
ry / extrude / shapes2
ry / extrude / selines
```



## Tutorials

### Animate text using WebGL techniques

This ensures the design remains consistent throughout the page. Next, add an overflow with the hidden parameter. This cuts off any content that breaks out of its bounds. At this point, you should not notice any change.

```
*{  
    margin:0;  
    overflow: hidden;  
}
```

#### 6. Styling the divs

We only style the instructions div as JavaScript controls much of the actions that occur in the main div displaying the animation. Add the following styling code in the CSS file.

```
#instructions {  
    position: absolute;  
    color: #666;  
    bottom: 0;  
    padding-bottom: 6px;  
    font-family: sans-serif;  
    width: 100%;  
    text-align: center;  
    pointer-events: none;  
}
```

The code simply directs the div to move relative to the main div as the page is scrolled.

#### 7. Setting up JavaScript environment

JavaScript's security feature, same-origin policy, restricts the execution of any externally hosted files in your JavaScript code. This offers a challenge since Three.js requires to load geometry, textures and other files. Two alternatives exist to solve the challenge. We can opt to set

## Learning from examples

One of the best ways to learn Three.js is through reviewing numerous applications of the code. Head over to <https://threejs.org/examples> where there are tons of examples of how Three.js is used.

up a local server using WAMP or XAMPP applications, or host the files on free servers.

The second alternative offers a much easier solution as you can reuse the files already hosted thereby saving you time. Simply copy the following script tags below the instructions div.

```
<script src='http://cdnjs.cloudflare.com/ajax/libs/three.js/r75/three.min.js'></script>  
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/175711/bas.js'></script>  
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/175711/TextGeometry.js'></script>  
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/175711/FontUtils.js'></script>  
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/175711/pnltri.min.js'></script>  
<script src='https://s3-us-west-2.amazonaws.com/s.cdpn.io/175711/droid_sans_bold.typeface.js'></script>  
<script src='http://cdnjs.cloudflare.com/ajax/libs/gsap/1.18.0/TweenMax.min.js'></script>
```

Each JavaScript file plays a different role as is explained later in the tutorial. You should not observe any change in the webpage.

#### 8. Creating the text animation

To create the text animation, begin by creating a new file, 'textanimate.js', in your JS folder. Since we would like the animation to run when the window loads, capture this using the init function as below.

```
window.onload = init;  
function init()  
{  
    var root = new THREERoot({  
        createCameraControls:false,  
        antialias: true,  
        fov:60});
```

The code creates a new variable root which is assigned three parameters using the THREERoot function. The first informs that the camera controls are not turned on while

the second enables antialiasing in order to smooth out any jagged lines. The last sets the field of view to 60 in order to view the animation comfortably.

#### 9. Setting up the scene

Next, set up the scene where the text animation will take place. Change the background colour and set up the camera position. Copy the code below.

```
root.renderer.setClearColor(0x3AAFA9);  
root.renderer.setPixelRatio(window.devicePixelRatio || 1);  
root.camera.position.set(0, 0, 400);
```

#### 10. Create the text animation variable

Create a text animation variable and assign the function createTextAnimation(). The function controls the nature of the animation and will be defined later. Next, identify the exact position (based on the y axis) where the animation will play at. For instance, in this case, the position is set at y = -10. Afterwards, add the variable to the scene.

```
var textAnimation = createTextAnimation();  
textAnimation.position.y = -10;  
root.scene.add(textAnimation);
```

#### 11. Set up the timeline animation aspects

The TweenMax.min.js file referenced earlier offers useful animation features. Among these is the TimelineMax function which helps set the repeat, delay and yoyo aspects. By yoyo, this implies the animation moving back and forth in a yoyo-like manner. Setting yoyo to false turns off this feature. A repeat of -1 makes it loop. Simply copy the code below.

```
var tl = new TimelineMax ({  
    repeat: -1,  
    repeatDelay:0.25,  
    yoyo: true  
});  
tl.fromTo (textAnimation, 4, //4  
{animationProgress:0.0},
```

##### Top

The image illustrates the animation of the text in a blowing wind-like simulation. Only some of the characters can be seen as the animation progresses

##### Right

The image illustrates the animation of the text once the wind has blown it into particles. None of the characters are recognisable since they are animated



**Learning through example**

The website, <https://threejs.org> presents numerous real world applications detailing how the library is used in creating different visual effects. Additionally, a link further below in the site, (<https://threejs.org/examples>), provides access to diverse examples that you can learn from. However, it may appear daunting at first before you get to know your way around. To help you get started, click the very first example [https://threejs.org/examples/#webgl\\_animation\\_cloth](https://threejs.org/examples/#webgl_animation_cloth). This is the animation that you will see.

```
{animationProgress:1.0, ease:Power1,
easeInOut},
0 );
createTweenScrubber(tl); }
```

**12. Create the text to be animated**

Next, create the text to be animated using the `generateTextGeometry` function. Type in the text you would like to be revealed, its size, height and font features. Note that earlier, 'droid\_sans\_bold.typeface.js' and 'TextGeometry.js' files were loaded. These control the font and geometry aspects.

```
function createTextAnimation() {
  var geometry = generateTextGeometry('GONE
WITH THE WIND', {
    size:14,
    height:0,
    font:'droid sans',
    weight:'bold',
    style:'normal',
    anchor:{x:0.5, y:0.5, z:0.0}
  });
  THREE.BAS.Utils.separateFaces(geometry);
  return new TextAnimation(geometry);
}
```

**13. Code the animation characteristics**

Next, we define how the animation should behave. The aim would be to simulate blowing by the wind from the left-hand side and once particles have disintegrated, recreate the text from the right-hand side.

```
function generateTextGeometry(text, params)
{
  var geometry = new THREE.
TextGeometry(text, params);
  geometry.computeBoundingBox();
  geometry.userData = {};
  geometry.userData.size = {
```

```
width: geometry.boundingBox.max.x -
geometry.boundingBox.min.x,
height: geometry.boundingBox.max.y -
geometry.boundingBox.min.y,
depth: geometry.boundingBox.max.z -
geometry.boundingBox.min.z
};
var anchorX = geometry.userData.size.width *
-params.anchor.x;
var anchorY = geometry.userData.size.
height * -params.anchor.y;
var anchorZ = geometry.userData.size.depth *
-params.anchor.z;
var matrix = new THREE.Matrix4();
makeTranslation(anchorX, anchorY, anchorZ);
geometry.applyMatrix(matrix);
return geometry; }
```

To investigate the code providing the functionality, click view source at the bottom-right corner. This should take you to its respective github page from where you can

view the code. Copy the code to your code editor and customise it to your preference.

**14. Adding important classes**

Now that we have added the required functions for the animation, we need to append different classes required to initiate the WebGL constructs.

**15. Adding important 'utils'**

Additionally, add the code (from FileSilo) to help control the interactive aspects of the animation. The code provides interactivity in response to mouse events such as mouse up, mouse down and movement of the mouse.

**16. Initiating the animation**

At this point, all the code necessary to initiate the application is set. To initiate the text animation, add the following line of code beneath the last script tag entered.

**I** <script src = "js/textanimate.js"></script>  
The final outcome should appear as below.



# **AWESOME AUDIO**

---

**Why and how to add  
the power of sound  
to your projects**



**"Sound, even more than visuals, is capable of making our mind believe we are in another world. It communicates on a deep level, directly tapping into our brains."**

**Richard Mattka**  
Award-winning interactive director, designer and developer

# THE IMPORTANCE OF SOUND

Put the power of sound to work in your sites and applications

Sound is a critical element when making experiences engaging and compelling. If you need proof, try playing your favourite game or watching your favourite movie with the volume off.

Ambient city noises, distant sirens, and low rumbling hums, drop you into the scene, even without the visuals. Sounds create the mood and set the pace of an interactive experience. Music enhances the suspense, ramps up the intensity, and ultimately triggers an emotional response, much more powerfully than visuals alone.

Sound is capable of transporting us to another world as it communicates on a deep level, directly tapping into the brain. Psychology studies have found music and sound connect on a instinctive level in the brain, into our most primal selves.

Immersive sites, such as world building models, in-world web experiences and games are greatly enhanced with great sound effects and musical scores.

Sites and applications with engaging animations, dynamic visuals or 3D, depend on solid sound design. But, even the simplest sites can benefit from

user feedback and meaningful signals through audio.

In your projects, sounds can form part of the reward system in completing tasks. For example, sound effects that trigger with completing a puzzle make the act much more satisfying. A pleasant chime as a bar fills up, or as the user achieves a specific goal – these all work together to give short-term rewards, while the user works towards a larger goal.

Sound can provide feedback, to complement visual feedback. Buttons make noises when clicked, a clicking metronome can count you in before recording. Sounds can help reinforce that something was done as a further way to acknowledge the user's actions. A way to say 'yes' your interaction was received. In some cases, sounds can call out something in a visually busy landscape. In time critical applications, they can more rapidly communicate something faster than visuals might alone.

User interfaces (UIs), since the earliest days of the command-line,

have used sound effects in the form of simple audible beeps, to warn of errors. And today, apps such as Twitter or Facebook use well-placed sounds for notifications and alerts. Skype uses audio to facilitate communications. As we move into technologies away from screens, speech and gesture interfaces are increasingly dependent on audio input and feedback.

Assistant software and tools such as Alexa and Siri give audio feedback fuelled by powerful artificial intelligence. Audio cues and speech recognition can also be used to create a better experience for those with disabilities or learning difficulties.

With ever increasing support, bandwidth and networking power, now is the time to put the power of sounds and music to work in your sites and applications, through solid sound design.

# BREAKING DOWN SOUND

Audio and sound come in many forms. Understanding what they are and how they work is the key to success



## Types of sound

### MUSIC/SCORE

Imagine your favourite movie without the iconic score. Jaws without the ominous two-chord warning? Star Wars without the incredible score, driving the action from the first iconic frame? Great music has its own life; it's essential to the success of the production.

### AMBIENT SOUND

In film and game design this is also called asynchronous sound. It is not directly synced with other actions, but forms a mood bed or atmospheric sound-scape. Crickets chirping, a crackling fire or sounds of the city at night. Ambient loops that set a tone for the experience. Of all the types of sound, this has

Consider each separately, but plan for combining

the strongest ability to subconsciously transport the user into the world of the experience.

### SOUND FX

Every interaction, achievement unlocked and warning can be enhanced with sound effects. They can reinforce the tone of the experience, and communicate greater detail quickly to a user. A clear warning tone might signal a low battery, before the user notices the flashing bar. A gentle bell or satisfying click when pressing a button can encourage and make an action more enjoyable. The nice pop effect when you pull down on Twitter to refresh results is a great example.

### VOICE

Voiceovers can humanise an experience, guide a user with instructions, and serve the functional roll of communicating important information in the absence of visuals. For example, a navigational app for driving or an instructional tool. Pitch recognition for tuning an instrument or voice recognition to receive commands are just a couple of possibilities. Voice assistance through apps such as Alexa or Siri are rapidly becoming part of every expense, from using our phone, operating our vehicles or setting the mood with some music at home. Voice recognition in sites and apps is growing and will continue to grow.

**Above**  
The 'Oblivion' music video is comprised of real-time shader animations, synced to audio data for precisely timed effects, like travelling down this fret-board tunnel

## The psychology of sound

Sounds are able to create powerful reactions on deep, instinctive levels. A baby's cry can set a new parent into action without conscious thought, a sharp warning hiss of a snake can trigger adrenaline, and hearing that favourite song on the radio might bring a smile before you realise it.

Scientists and thinkers have been pondering the relationship

between sound and emotion for thousands of years. Modern day neuroscience has uncovered amazing insights into the relationship between our ears, minds and emotions.

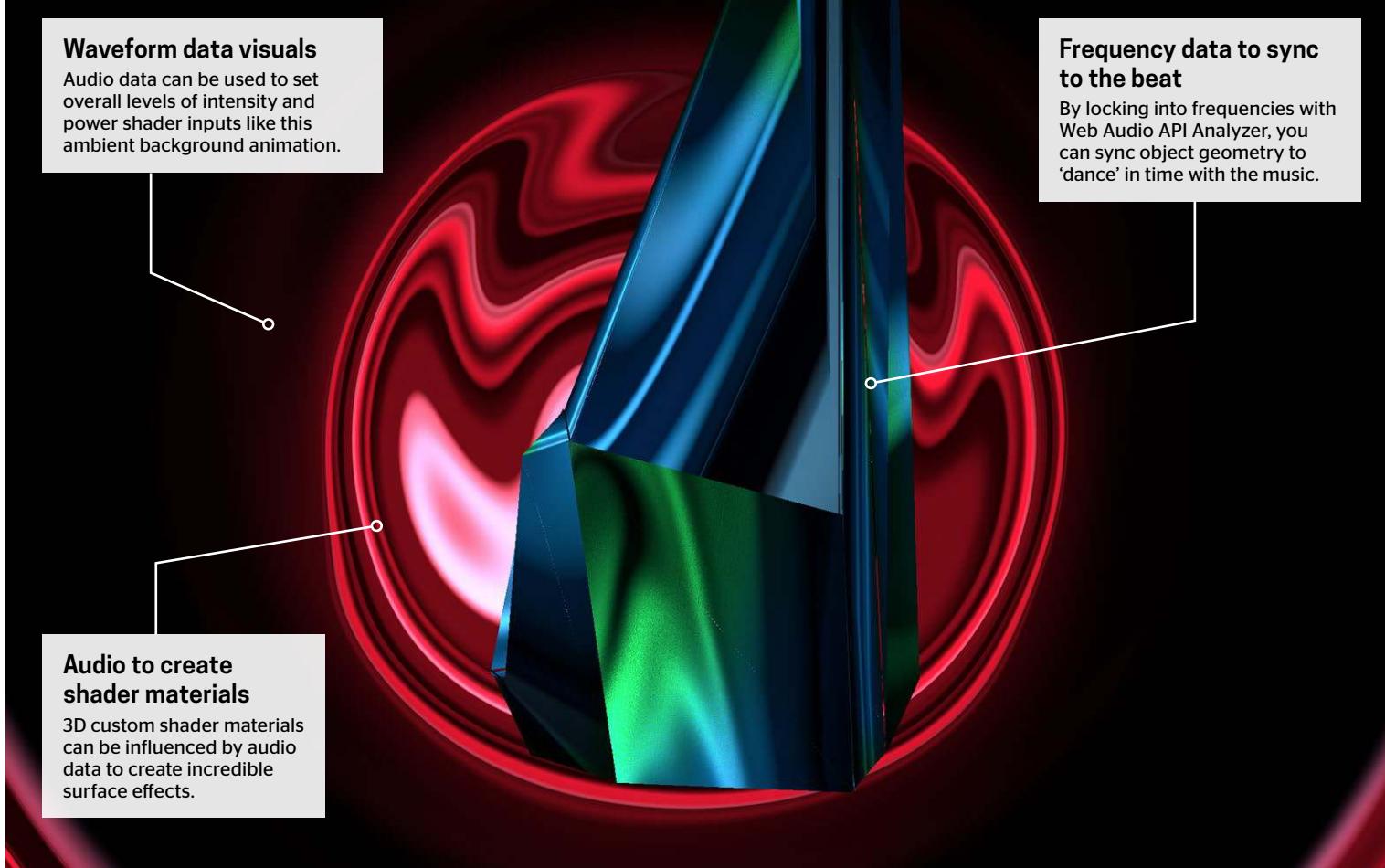
Deep within the brain, in the Amygdala, we process memories and emotional reaction, while the Hippocampus controls behaviour and helps form memories. Within these

most primal sectors of the mind, responses are triggered when we hear sound. Breathing, heart rate, brainwaves and hormone secretions are continuously affected by incoming audio. Sounds are associated with experiences and the feelings connected to them, both good and bad.

How someone feels at any given moment, is very often

affected directly by what they hear. People have the ability to remember hundreds, if not thousands of songs and voices. Sounds are remembered more easily because of the impression they make on your mind, and they also are much harder to ignore. Hearing someone say your name across a crowded room, but seeming to ignore everything else, is just one example.

What you hear can trigger a host of different reactions



## Visualisations and effects

### Power stunning visuals with audio data

Web Audio API's AnalyserNode lets you extract time, frequency, waveform and other data from your audio. By using features like `getByteFrequencyData` and setting the min and max decimal ranges you can zero in on specific aspects of audio data.

Beyond music beds, effects and great music, we use sound to also drive visuals. Moving beyond complimenting or enhancing what the user sees, the audio data can

actually drive the animations. Simple effects that use the overall level (volume) of a music track can make your background pulse in time with a beat. Swells in the musical score can be used to change the opacity of an image or shift its colour. By tapping into the audio data through the Web Audio API we delve into frequency and waveform data as well. You can visualise the sound in an infinite number of variations.

“

**Simple effects that use the overall volume of a music track can make your background pulse in time with a beat**

**Above**  
This audio-driven animation was featured at Digital Design Days in Milan for the 2018 conference, as part of the DevX Experiment showcase (<http://devx.ddd.it/en>)

Film and game-makers have capitalised on the psychological power of sound for years. They know how sound can evoke emotions. The right music or ambient sounds can instantly connect an audience to a flood of associated feelings. Everyone has that favourite game theme song or movie score. As soon as they hear it, they recall how they 'felt' when they played it or saw it the first time. Perhaps it's the iconic

Super Mario Bros theme or the soundtrack from Azeroth in World of Warcraft.

The right sound design, the right music, can not only set the mood, but emotionally connect to people in a way visuals alone simply cannot.

#### SOUND AND EMOTIONS

#### Brain stem reflex

When loud or dissonant sounds are heard, they signal a potentially

important and urgent event, causing us to react on an instinctive level. Notifications, beeps and sirens are examples.

#### Learned response or conditioning

When we have heard a sound repeatedly in a certain situation, it can often lead to an association between that sound and situation. Hearing it again, can instantly elicit emotions or feelings.

#### Emotional contagion

Perceiving emotions expressed by a piece of music. It doesn't have to sound sad. Instead we recognise it as conveying 'sadness'. Sound is so fundamental, that it often transcends languages. People from diverse cultural backgrounds often agree on whether a piece of music sounds happy or sad. This has led some to term music the 'language of emotions'.

*Interactive installation, 'Sound In Performance', focussed on balancing sound loop quantity and interactions to trigger them*

## THE CHALLENGES OF SOUND

Why aren't sounds used more often on websites and apps?

### 1. Download size and streaming limits

We can overcome this a few ways. First, optimising sounds through good compression, that balances quality, size and bit rate. MP3 format is going to give a good balance of these factors, and audio tools will give you options for levels of compression. Stream larger files instead like music and ambient tracks. Load FX for instant playback.

### 2. Unexpected intrusion

Give users the option to disable sound and control levels. Let them choose their experience, and they will often choose to listen.

### 3. Mobile limitations

On most mobile devices audio will not play until triggered by a user action. A simple solution is to start the experience with scroll or click event, that you can then bind your

audio star to. Initialise ahead, and wait to fire load and play events the moment you get that first interaction.

### 4. Sound overload

Not every action needs a sound effect. Sounds should serve a purpose. Use music and effects that inform or enhance the experience. Just like a good visual designer, don't be afraid to cut. Too much of a good thing, can still be too much.

“

**Sound design for interactive is challenging because of the unpredictable interactions by users. This requires thoughtful planning to create just the right balance in real-time sound mixes**

# HOW TO ADD SOUND

Between HTML audio and the Web Audio API, it's easy to start adding sound to your projects

## HTML <audio>

The HTML <audio> tag allows you to embed music on websites and apps. Similar to the <video> tag, you can specify a source file, controls and several other options. This allows you to easily add audio to your page with minimal code. You can then target the element via JavaScript to give further control. It supports MP3, WAV, OGG and other formats, with MP3 being universally supported across modern browsers and devices. A simple example of using the <audio> tag:

```
<audio autoplay="autoplay"
controls="controls">
  <source src="music.ogg" />
  <source src="music.mp3" />
</audio>
```

Some handy attributes specific to the element include:

- **Autoplay** - to allow the audio to play once it is ready
- **Controls** - controls for the audio file will be included on the page
- **Loop** - if this one is included, the audio will loop and play again once it has finished
- **Preload** - preload the audio when possible so it's ready for playing

Limitations to the <audio> element include:

- Low limit to number of sounds played simultaneously
- No precise timing control
- No ability to apply real-time effects
- No way to analyse sounds

## Web audio

Understanding the Audio Routing Graph

The Web Audio API is a powerful system for controlling audio on the web. It enables audio sources, adds effects, creates audio visualisations and more.

The Web Audio API manages operations inside an Audio Context. Audio operations are performed with audio nodes, which are linked together to form an Audio Routing Graph.

Multiple sources are supported within a single Audio Context. This modular design is highly flexible allowing the creation of complex audio designs.

### AUDIO ROUTING GRAPH FLOW

Audio nodes are linked into chains and simple webs by their inputs and outputs. They typically start with one or more sources.

Node outputs can be linked to the inputs of others creating chains or webs of audio streams. A common effect is to multiply the

audio by a value to make it louder or quieter using the GainNode.

Once the sound has been effected and is ready for output, it can be linked to the input of a AudioContext.destination, which sends the sound to the speakers. Note: This last connection is only required if you need the audio to be heard.

A typical flow for Web Audio could look something like this:

- » Create audio context
- » Create sources inside the context, such as <audio>, oscillator or streams
- » Create effects nodes, such as reverb, flanger, panner, or compression
- » Choose a destination for the audio, such as speakers
- » Connect the sources to the effects, and the effects to the destination

Below illustrates the Audio Context and the flow of the Audio Graph.

## Where to get sounds

### SITES

#### Freesound

<https://freesound.org>

Freesound is a collaborative database of Creative Commons Licensed sounds. You can browse, download and share sounds.

#### SoundJay

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

Library of free sounds you are allowed to use on your website, free of charge and royalty-free in your projects.

#### Partners In Rhyme

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

Free and royalty-free music and sound effects.

### ROYALTY-FREE

#### Pond 5

<https://bit.ly/2JKstFb>

One of the world's largest collection of royalty-free stock video and creative assets, sounds and music

#### Premium Beat

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

High quality curated royalty-free music library, organised nicely by genres and moods.

#### Audio Jungle

<https://audiojungle.net>

Massive library of royalty-free sounds and music from \$1.

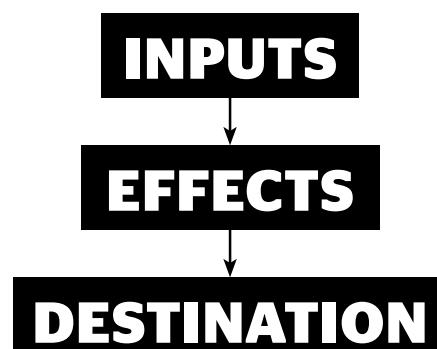
### LICENCED

#### Sound Stripe

<https://soundstripe.com>

Membership model, for unlimited licenses for songs in large, curated music library.

## Audio Context



## The major file formats explained

### MP3

A compressed or 'lossy' format. Audio data is compressed and optimised for size and bit rate. This is the format you want to be using as often as possible.

### WAV

The WAVE container format is an uncompressed audio file. Much larger sizes typically. Files in the WAVE container format typically end with the ".wav" extension.

### FLAC

Uses lossless compression, meaning silence does not take up file space, but the uncompressed audio is retained. Larger file size, (smaller than wav), and not as well supported.

### Ogg Theora Vorbis

Supported in desktop/mobile Gecko (Firefox), Chrome, and Opera. Support can be added to Safari (but not on iOS) with an add-on. The format is not supported in IE in any way.

# Sound libraries

## Howler.JS

<https://howlerjs.com>

A solid, audio library for the modern web. A solid cross-platform solution, for managing multiple sound effects easily, with good documentation to get started quickly. Includes a wide range of file support, use of audio sprites, spatial audio for 3D experiences, automatic caching and has no dependencies.

## Pizzicato.JS

<https://alemangui.github.io/pizzicato>

Another well documented Web Audio JavaScript library. Pizzicato has a simple set of methods for creating cool effects, using sound inputs, and managing sound files. Compressors, stereo panning, tremolo, reverb, flanger and many more effects are included, to start creating cool effects out of the gate.

## SoundJS

<https://createjs.com/soundjs>

SoundJS by Grant Skinner. A JavaScript library that provides a simple API, and powerful features to make working with audio extremely simple. Designed with cross-browser support in mind, it gracefully falls back to HTML audio for wide support. Easily ties in audio file loading to PreloadJS. If you are already using CreateJS tools, this is a solid choice.

# HOW TO USE THE WEB AUDIO API

Discover how to use a powerful modular approach to building sound

While the audio element allows you to include plugin-free audio on your site, it is still limited. For maximum sound control, the Web Audio API allows you to generate sounds, play existing ones, create effects and much more. In this example you'll load and play a sound file using the API.

### 1. INITIALISE THE AUDIO CONTEXT

To start we need to set up our Audio Context, an audio canvas for our sounds. This method ensures maximum cross-browser support and fallback in case the API is not supported.

```
try {
  window.audioCtx = window.
  AudioContext|| window.
  webkitAudioContext;
  audioCtx = new AudioContext();
}
catch(e) {
  alert('Web Audio API is not
  supported in this browser');
}
```

A single audio context supports multiple sound inputs and complex audio graphs, so you only need one for each audio application we create.

### 2. CONNECT THE AUDIO GRAPH

Any audio node's output can be connected to any other audio node's input by using the connect() function. In this example you will connect a source node's output into a gain node, and connect the gain node's output into the context's destination:

```
// Create the source.
var source = context.
createBufferSource();
// Create the gain node.
var gain = context.createGain();
// Connect source to filter,
filter to destination.
source.connect(gain);
gain.connect(context.
destination);
```

This audio graph is now dynamic, meaning you can change it

whenever you need. You can disconnect audio nodes from the graph by calling node.disconnect(outputNumber). The power of this modular approach allows you to control gain (volume) for all sounds, or ones you wish. You can route sounds through effects or not at all, or in any combination you might need.

### 3. LOADING SOUNDS

To load an audio file into the Web Audio API, we can use an XMLHttpRequest and process the results with context.decodeAudioData. This works asynchronously and doesn't block the main interface thread. Here is what the code would look like:

```
var request = new
XMLHttpRequest();
request.open('GET', url, true);
request.responseType =
'arraybuffer';
request.onload = function() {
  context.decodeAudioData(request.
  response, function(theBuffer) {
    buffer = theBuffer;
  }, onError);
}
request.send();
```

### 4. PLAYING SOUNDS

Audio buffers are only one potential source of audio. You can use direct input from a microphone or line-in device or an <audio> tag among others. Once you've loaded your buffer, you need to create an AudioBufferSourceNode for it, connect the source node into your audio graph, and then call start(0) on the source node. To stop a sound, call stop(0) on the source node.

The code looks like this:

```
function playSound(buffer) {
  var source = context.
  createBufferSource();
  source.buffer = buffer;
  source.connect(context.
  destination);
  source.start(0);
}
request.send(); }
```

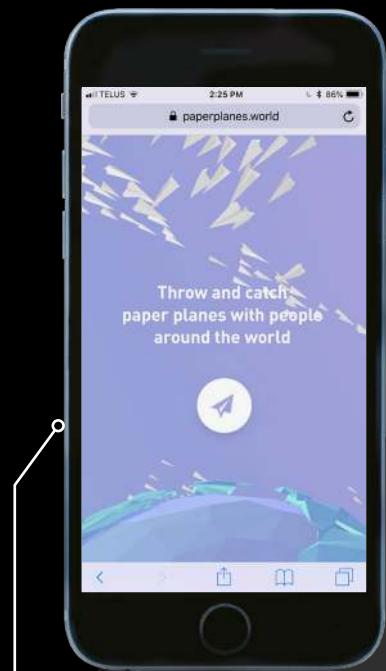
### 5. PUTTING IT ALL TOGETHER

As you can see from the previous code, there's a bit of setup to get sounds playing in the Web Audio API. But, with this modular approach you gain maximum control over audio. Mixing sounds, reading their data via the Analyzer Node and so much more. Here is what a working example to load and play a sound looks like all together. Consider abstracting these steps for managing multiple sounds in larger projects as well.

```
var audioCtx, analyser,
bufferLength, dataArray;
window.addEventListener('load',
initAudio, false);
function initAudio() {
  try {
    window.audioCtx = window.
    AudioContext|| window.
    webkitAudioContext;
    audioCtx = new AudioContext();
    analyser = audioCtx.
    createAnalyser();
  }
  catch(e) {
    alert('Web Audio API is not
    supported in this browser');
  }
  // load the audio file
  source = audioCtx.
  createBufferSource();
  var request = new
  XMLHttpRequest();
  request.open('GET', 'assets/
  beat3.mp3', true);
  request.responseType =
  'arraybuffer';
  request.onload = function() {
    var audioData = request.
    response;
    audioCtx.
    decodeAudioData(audioData,
    function(buffer) {
      source.buffer = buffer;
      source.connect(analyser);
      analyser.connect(audioCtx.
      destination);
      source.loop = true;
      source.start(0);
    },function(e){"Error with
    decoding audio data" + e.err});
  }
  request.send(); }
```

# AUDIO IN ACTION

There are so many amazing examples to choose from, but here are just a few to inspire you!



## Paper Planes

<https://paperplanes.world>

Catch and throw Paper Planes with people around the world, using the power of the web sockets and connections.



## Discover your Patronus

<https://my.pottermore.com/patronus>

This experience transports users into a Harry Potter-inspired 3D environment, rich with sound and beautiful FX, eventually discovering their Patronus.

## Get more info on Web Audio API

### MDN Web Docs - Web Audio API

<https://mzl.la/1ioq44h>

An in-depth look into the API with rich documentation and examples. Every aspect of the API is well covered.

### W3C - Web Audio

<https://bit.ly/2HM83z9>

A repository containing the latest editor's drafts of the W3C Web Audio API. This is the source where the standards are presented.

### Introduction to Web Audio API

<https://bit.ly/2jlgRgR>

A good introduction to using the API to create sounds by Greg Hovanesyan. Create a music-specific application using the oscillator audio source.

### Web Audio Weekly

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

A collection of news, stories and demos all about the Web Audio API. Covers a wide range of topics and examples to keep you learning.



## The War of 1996

[www.warof1996.com](http://www.warof1996.com)

An incredible use of the custom sound FX as President Whitmore's powerful speech combines with music from Independence Day for a unique audio experience.



# BUILD PROGRESSIVE WEB APPS WITH WORKBOX

Google's Workbox makes it easy to build web apps which work offline

For many years, the gulf between web apps and desktop or native mobile apps has been shrinking. These days, there are relatively few features available for native build that aren't also offered via a web API. These days the web supports multi-touch events and push notifications, using device hardware such as the camera and much more.

Progressive Web Apps, or PWAs, is a term (coined in 2015 by Google) which is used to describe web apps that take advantage of newer web technologies to provide an equivalent user experience to native apps. This means not only offering the sort of functionality that you'd typically expect from native apps, but also matching their security, performance and availability. While the concept has been around for a while, PWAs remain in their infancy, and it's still vastly more common to use either native build or hybrid apps (embedding web content in a native container). Nonetheless, many people feel that PWAs represent the future of how many types of app will be delivered, and there are a growing number of frameworks, libraries and tools available to help develop them.

## Getting apps to work offline

One of the biggest barriers to providing a true native-like experience from a web app is dependency on the network. It's no good for the user if the app only works with an internet connection. Native apps can be loaded up even when the device is completely offline, and don't fail during periods of intermittent connection (think about when you're on the train...).

Fortunately, the Service Worker API solves this.

A service worker is a piece of JavaScript code downloaded with a page that is stored by the browser and runs in the background – even when the page isn't being viewed. It allows you to intercept network requests (such as fetching a URL) and determine how to

handle them. When combined with the Cache API, this enables you to decide when and how to serve up cached content if a network connection is not available. As a result, it is now possible to develop web apps which, once downloaded and cached, can be used entirely offline.

Because you define the behaviour for each and every request, you can also begin to implement more complex caching strategies. For example, you might decide to serve from the cache immediately but also refresh the cache from the network in the background, so that next time the same page is requested, the data is more up to date.

## 3 reasons to use service workers

- 1
- 2
- 3

### Offline support

The headline feature of service workers is that they allow you to build pages that work offline and with intermittent network connection, by serving up content from a browser cache in the manner of your choosing.

### Speed

Nobody likes waiting for a page to load. Loading content from a cache is nearly instantaneous, and because you can control caching at a granular level, service workers allow you to take advantage of this without risking showing stale data.

### Native features

Because they run in the background, service workers let you do things you take for granted in native apps but couldn't previously do on the web. Push notifications and background synchronisation are two examples of this.



## Introducing Workbox

By now, service workers probably sound great. But with the level of control and flexibility they introduce, also comes a great deal of complexity. Caching is a notoriously hard problem to solve, and if you start writing 'vanilla' service workers (which we've covered in the past), you'll quickly find that you end up with pages and pages of code handling

different scenarios of network request. Workbox is a library from Google which simplifies the code you need to write to implement caching strategies. Where you might previously have required dozens of lines of code, Workbox can handle in a single line. It also provides easier ways to handle background sync, and improved debugging.

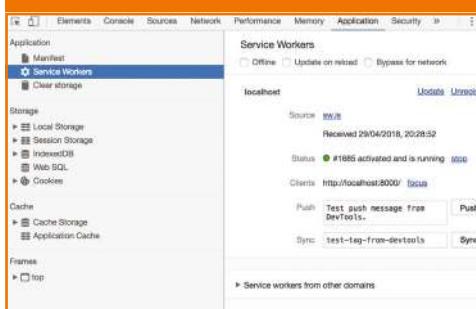
Service workers are currently supported by Firefox, Chrome and Safari, but not Edge or IE, so make sure you're using a browser that works with them



## Debugging

The behaviour of service workers can be confusing to start with because they keep running in the background, so refreshing the page might not always do what you expect. As a result, as you start out we'd suggest including frequent console logging in your code to help follow what's happening. Additionally, Chrome Developer Tools can provide detailed insight into the service workers on a page. In Developer Tools, navigate to the Application tab and you'll see a Service Workers tab. It's particularly worth noting that you can manually update and unregister service workers on this page, which is helpful when developing to ensure you pick up the latest changes. Additionally, the Clear Storage tab will enable you to purge cached resources, which can also come in very useful.

Once you have Workbox up and running and handling your caching, you'll also find that it provides its own very rich console messages which are certainly worth keeping an eye on.



## Create your first service worker

Workbox makes it easier than you think

### 1. Set up an HTTP server

We'll begin by installing and running an HTTP server for our site. The easiest way to do this is using npm, with which we can run:

```
npm install -g http-server
```

We can then create a new project director and run:

```
http-server -p 8000 -c-1
```

The -c-1 parameter disables HTTP caching to make it easier to see our service worker in all its glory.

### 2. Create a basic site

To test our service worker, we'll create a simple static website. If we use Bootstrap (loaded from a CDN), we can throw together something reasonable-looking fairly swiftly. Let's assume that we embed all of our content in index.html for now, and include a script on the site called site.js. Our HTTP server will automatically serve index.html when we run it, which we can access in the browser at:

<http://localhost:8000/>

### 3. Registering a service worker

To get started with Workbox, we need to register a service worker. This tells the browser to install a specific JavaScript file when the page is loaded. Let's call our service worker file sw.js. In this case, we need to add some code to site.js to register it:

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', function() {
    navigator.serviceWorker.register('/sw.js').then(registration => {
      console.log("Successfully registered service worker.");
    }, function(error) {
      console.log("Error registering service worker: " + error);
    });
  });
}
```

On a remote server, you need to serve your site over HTTPS to use service workers for security reasons. We won't worry about that for our local server, but keep it in mind

### 4. Creating the service worker script

The service worker script itself is where the power of Workbox starts to come into play. Let's create sw.js. The first thing we need to do is use the service worker API's importScripts function to import Workbox, which we can do from a CDN:

```
importScripts('https://storage.googleapis.com/workbox-cdn/releases/3.1.0/workbox-sw.js');
```

It's worth noting that this is the 'manual' way to create a service worker script. There is also a Workbox CLI which you can use to generate a service worker automatically from configuration, and can be integrated into your build pipeline.

## 5. Did it load correctly?

We can test to see if Workbox has been imported correctly:

```
if (workbox) {
  console.log("Workbox loaded successfully.");
} else {
  console.log("Workbox could not be loaded.");
}
```

importScripts is synchronous so we don't need to worry about promises for this to work.

## 6. Configuring your first caching behaviour

Now that Workbox is loading with your page, let's use it to control our content caching. Let's say we want to default to loading our JavaScript and CSS content from the cache, and hit the network only if a resource couldn't be found. We can do this with a straightforward routing function.

```
workbox.routing.registerRoute(
  new RegExp('.*\.(?:js|css)'),
  workbox.strategies.cacheFirst()
);
```

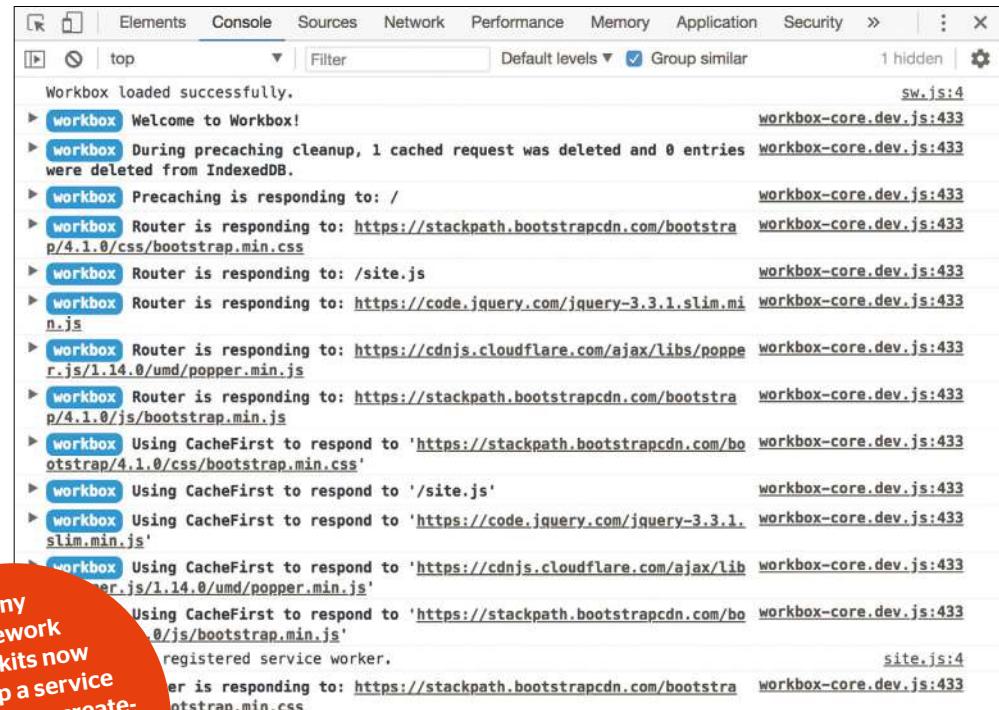
Here we're using a regular expression to match the request types we want to handle, and using Workbox's built-in cacheFirst function to specify our desired behaviour.

## 7. Comparing to 'vanilla' service workers

At this point, it's useful to contrast this code with the approach we'd need to take to do the same thing if we weren't using Workbox. We'd have to manually populate the cache (which we won't go into here), then respond to HTTP fetch events by checking the cache to see whether a matching resource is available, serving it up if it is, or falling back to the server if it isn't. Sound complicated? The fetch handling on its own looks like this:

```
self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request).then(response => {
      if (response) {
        console.log("Retrieving resource from cache: " + event.request.url);
        return response;
      }
      console.log("Retrieving resource from server: " + event.request.url);
      return fetch(event.request);
    })
  );
});
```

Many framework starter kits now bootstrap a service worker for you. create-react-app, for example, will set one up that you can use if you want



## 8. Testing it out

Fire up Chrome and take a look at the Console. You should see a few things happening. The first time you load your page,

Workbox will cache all JavaScript and CSS assets requested, including those you're loading from CDN. When you refresh, you'll see it responding to these requests with our cacheFirst behaviour.

## 9. Precaching for offline viewing

Okay, so we're retrieving some assets from our cache, but thus far we've only handled static resource caching for our CSS and JavaScript. But really, we want our page to work in full when we're offline. To achieve this, we want to use precaching, which downloads files before the service worker is installed. This is again very straightforward with Workbox:

```
workbox.precaching.precacheAndRoute([
  '/index.html'
]);
```

If you now load the page, then terminate the webserver and refresh it again, you'll find that it still loads correctly offline.

## 10. Alternative caching behaviour

So far we've only looked at precaching and defaulting to the cache. Workbox comes with a number of other built-in caching behaviours which you can use with workbox.strategies. They'll handle most situations you're likely to encounter:

`cacheFirst()` - we've looked at already  
`networkFirst()` - the opposite of cacheFirst, this tries the network first then the cache

`staleWhileRevalidate()` - one of the most popular caching patterns, this responds quickly with a cached response, then uses a network request to update the cache  
`networkOnly()` - ensures you always serve content fresh by only allowing it from the network  
`cacheOnly()` - will serve content only from the cache; you're probably not going to use this much at all.

## 11. Google Analytics

Another very cool feature of Workbox is that it allows you to take advantage of Google Analytics while the user doesn't have a network connection. By default, if you're using Google Analytics to collect data about how your users interact with the app, you'll miss out completely on activities while the user is offline. Workbox allows you to queue failed Google Analytics requests so they can be retried when the user is back online. This can be done with a single line of code:

```
workbox.googleAnalytics.initialize();
```

## 12. Other caching strategies

We've covered the most common ways to handle caching, but there are many other scenarios you might want to cover which Workbox will handle. Google's Workbox pages include some excellent 'recipes' for commonly used caching strategies which you can take a look at. For example, if you want to cache Google Fonts, it has a pre-baked function to do this. Make sure you check out the link: <https://developers.google.com/web/tools/workbox/guides/common-recipes>

## Do I need Workbox?

If you're still asking "would I be better off building my service workers without Workbox?", then there are a few things to think about. Workbox is fairly lightweight and won't make a dent on your page's performance, and is as widely supported as the service worker API itself. Therefore, we'd argue that there's currently not a whole lot of reason not to use Workbox if you're considering introducing service workers to your app. In fact, there's not much reason not to use it full stop - even if some of your users won't be able to take advantage of the service worker, it'll degrade gracefully and benefit those who can. The alternative, to manually code all your caching behaviour, really seems like re-inventing the wheel and will drain time you could be spending on other things. For serious production apps, it's also worth getting familiar with the Workbox CLI. This can be integrated into your build process, and as well as generating the service worker itself from the configuration you provide, it can also do things like automatically generate a manifest of files to cache, which saves you manually updating a list as you develop your app.

## What's new in Workbox 3?

The stable release of Workbox 3.0 came in March (we're now on 3.1), and as you'd expect from a major release, brought both numerous changes and new features.

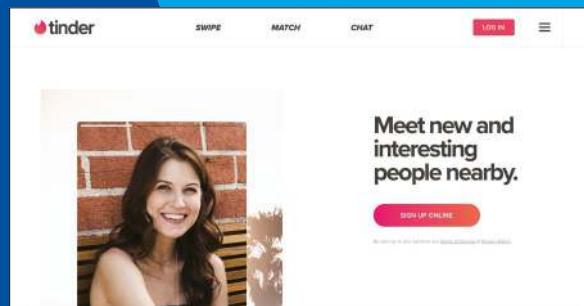
The CDN which we used earlier is introduced with Workbox 3.0, making it easier to get up and running with Workbox and reducing the resource consumption from your own hosting.

Debug logging is also greatly improved, with logging enabled by default when Workbox is used from localhost as you probably will when you start developing with it.

Alongside this, the payload of Workbox has been optimised, and it's now set up to avoid downloading features which you're not using.

As a result, there are a number of breaking changes compared to version 2. Chances are, this probably won't be a huge issue since Workbox is relatively new and you're probably not using it in production. Nonetheless, Google provide a comprehensive guide on migrating from v2 to v3 on their developer pages.

# 5 places service workers are in use today



### Tinder

<https://tinder.com>

Tinder (don't pretend you don't know what it is) now offers a web alternative to its mobile app. And guess what... it's a mainstream app built with Workbox.



### Pinterest

[www.pinterest.co.uk](http://www.pinterest.co.uk)

Pinterest is a great example of using service workers for complex caching behaviour on a site with extensive functionality. You can inspect the service worker in Chrome to see how it's implemented.



### Unreal Engine

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

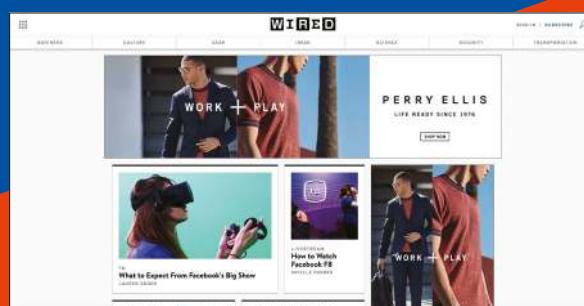
If you thought Pinterest's service worker was complex, check out Epic Games' Unreal Engine website, which has some impressive visuals and interaction. It uses sw-toolbox - a predecessor to Workbox.



### Google

[www.google.co.uk](http://www.google.co.uk)

If you're questioning the ubiquity of service workers, look no further than Google itself, which will install one to cache its static assets when you visit its homepage.



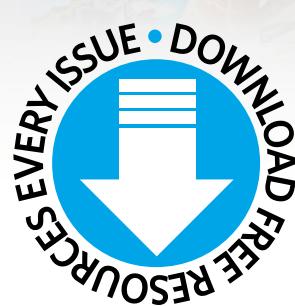
### Wired

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

Wired is another one which uses Workbox, and is a great place to see the Workbox CLI generating a manifest of files to cache as part of the build pipeline.

# CREATE THE IMPOSSIBLE

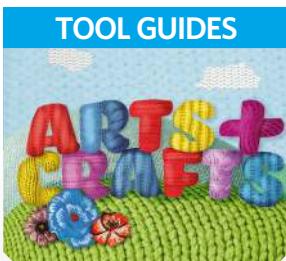
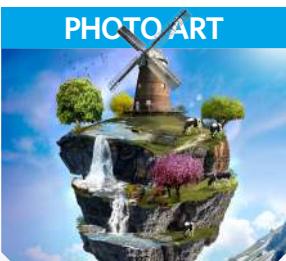
[www.photoshopcreative.co.uk](http://www.photoshopcreative.co.uk)



# Photoshop creative

ON SALE NOW

• Striking imagery • Step-by-step guides • Essential tutorials



Available  
from all good  
newsagents and  
supermarkets

## BUY YOUR ISSUE TODAY

Print edition available at [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)

Digital edition available for iOS and Android



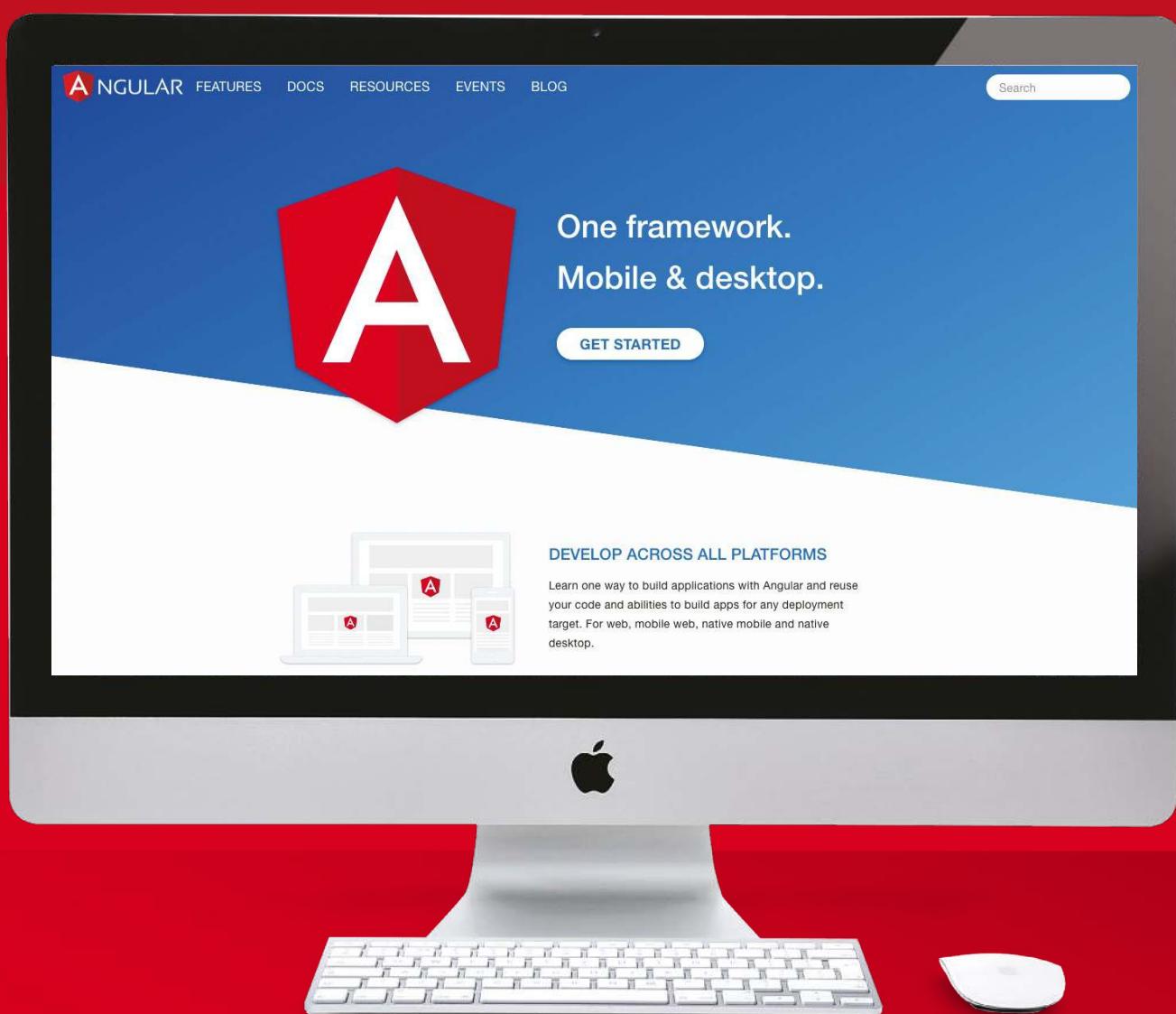
[facebook.com/PhotoshopCreative](https://facebook.com/PhotoshopCreative)



[twitter.com/PshopCreative](https://twitter.com/PshopCreative)

# State management with Angular

Learn how you can use the power of RxJS to manage your application's state with Redux



 **DOWNLOAD TUTORIAL FILES**  
<https://bit.ly/2KzBdZl>



In this tutorial we are going to show you how you can use the power of RxJS to manage your application's state with Redux.

In particular we'll look at how you can lazy-load portions of your store and handle asynchronous actions with epics. To demonstrate this we'll create an app which hooks up with Wikipedia's API. You'll need a working knowledge of Redux and its concepts.

There is a repo to accompany this tutorial which contains the step-by-step code: <https://github.com/danielcrisp/reactive-state-management>

## 1. Create the app

We're going to use Angular's excellent CLI to help us quickly build out an app. Use the following commands to install it, create an app and start a dev server:

```
npm install -g @angular/cli
ng new demo --routing
cd demo
ng serve --open
```

Next we need to install Redux and 'redux-observable', which is RxJS-based middleware for Redux. Note that at the time of writing 'redux-observable' doesn't have support for Redux v4 so you need ensure you only install v3 of Redux:

```
npm install --save redux@^3 redux-
observable
```

'redux-observable' was created by the team at Netflix.

There is a great video about it here:

[youtube.com/watch?v=AslnCYG8whg](https://www.youtube.com/watch?v=AslnCYG8whg)

And finally we need '@angular-redux', which bridges the gap between Angular and Redux. Install it like this to get the Angular 5+ compatible version:

```
npm install --save @angular-redux/store@^7
```

## 2. Wiring it up

The next step is to wire everything up with Angular. Create a folder inside 'src/app' called '@store'. I like to use '@' to denote core modules that aren't lazy-loaded because they are listed at the top in the directory. Inside the folder add two files:

```
store.module.ts
import { NgModule } from '@angular/core';
import { DevToolsExtension, NgRedux,
NgReduxModule } from '@angular-redux/store';
import rootReducer from './store.reducers';
@NgModule({
  imports: [
    NgReduxModule
  ],
  providers: []
})
export class StoreModule {
  constructor(ngrx: NgRedux<any>, devTools: DevToolsExtension) {
    const middleware = [];
    const enhancers = [];
    if (devTools.isEnabled()) {
      enhancers.push(devTools.enhancer());
    }
    ngrx.configureStore(
      rootReducer(), {}
    );
  }
}
```

```
middleware,
enhancers
);
}

store.reducers.ts
import { combineReducers } from 'redux';
const initialState = {
  hello: 'world'
};
export default function rootReducer(injectedReducers: any = {}) {
  return combineReducers({
    // Core reducers
    example: (state: any = initialState) => {
      return {
        ...state
      };
    },
    // Lazy-loaded reducers
    ...injectedReducers
  });
}
```

Note you might need to stop and start the webpack server after creating new files. Also to keep things succinct I'm going to use the <any> type regularly. Now open up 'app.module.ts' and import both

NgReduxModule and your newly created StoreModule:

```
import { NgReduxModule } from '@angular-
redux/store';
import { StoreModule } from './store/store.
module';
```

Add both modules to the imports array in the NgModule config and save to reload the app.

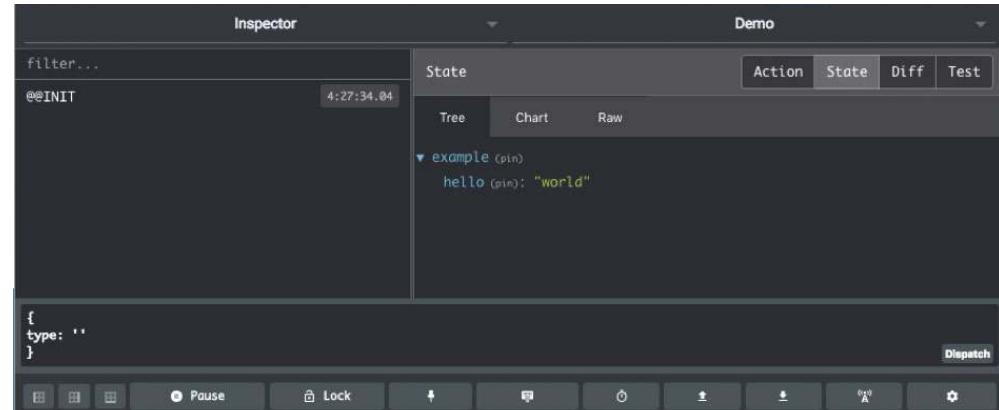
Check your store in DevTools and you should see the defaultValue.

## 3. Creating a lazy-loaded module

In order to demonstrate how to lazy-load portions of the store we'll need to create a lazy-loaded route and module. The working directory is 'src/app'.

```
ng g module search --routing
cd search
mkdir components
cd components
ng g component index
```

This creates a module, with a routing file, and a component to use as the index route. We're going to use this index route to conduct a search of Wikipedia.



## 4. Set up the internal module routing config

Now we need to add the routing config for inside our module. Open up the 'search-routing.module.ts' file that we created with the CLI. You'll see an empty routes array. Import the IndexComponent and add it to the route config, like so:

```
import { IndexComponent } from './
components/index/index.component';
const routes: Routes = [
  {
    path: '',
    component: IndexComponent
  }];

```

## 5. Add another route

Let's also create a non-lazy-loaded module and component (again starting from 'src/app') to provide an alternative route.

Since this is not lazy-loaded it is a good place to store components that are required throughout the app, so we'll call it 'shared':

```
ng g module shared
cd shared
mkdir components
cd components
ng g component root
```

Note: I didn't use the '@' when using the CLI command because it'll try to use it in the output. Instead I just rename the folder manually. We also didn't use the '--routing' flag, because we'll handle routing in the root app routing module. Import this module into 'app.module.ts' and add it to the imports array.

## 6. Add the routing outlet

The next step is to update the 'app.component.html' to declare where the routes should be rendered, and add a simple nav so we can switch between routes. Simply replace all the contents with this:

```
<div class="container">
  <h1>My App</h1>
  <a routerLink="/">Home</a> | <a
routerLink="/search">Search</a>
<hr>
<router-outlet></router-outlet>
</div>
```

Note: It's not necessary, but you can add Bootstrap's CSS if you want it to look a bit tidier.

# Tutorials

## State management with Angular

### 7. Set up the main app routing

Out of the box your app won't have any routing, it'll just load the AppComponent, but because we used the `--routing` flag we do have a routing module. Open up `app-routing.module.ts` and we'll add the root route first.

```
import { HomeComponent } from './shared/components/root/root.component';
const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  }];
}
```

Check the app in the browser and hopefully you will see a 'root works' message in the page. Now let's add the search module with lazy-loading:

```
const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  },
  {
    path: 'search',
    loadChildren: './search/search.module#SearchModule'
  };
]
```

This tells Angular which module to load when the user hits the 'search' path. Try it in the app (I needed to restart the server at this point) and open the Network panel of DevTools – you'll see the 'search.module.chunk.js' loaded when you click the link in the nav and an 'index works' message in the page. Pretty cool!

### 8. Using some RxJS magic

Before we can modify our store structure we need to add a mechanism for lazy-loaded modules to add their own Reducers and Epics to the main store.

Return to the '@store' directory and create a service with the CLI:

```
ng g service store --flat true --spec false
Add two BehaviorSubjects to the service, one called 'epics$' and one called 'reducers$'. These will allow other modules to interact with the main store.
```

```
import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs/BehaviorSubject';
@Injectable()
export class StoreService {
  epics$: BehaviorSubject<any>;
  reducers$: BehaviorSubject<any>;
  constructor () {
    this.epics$ = new BehaviorSubject(null);
    this.reducers$ = new BehaviorSubject(null);
  }
}
```

Import the newly created service into the 'store.module.ts' and add it to the providers array.

### 9. Integrate with main store

Next we need to hook up this up to the store itself. In

'store.module.ts', inject the StoreService in the constructor. We're also going to need some extra imports:

```
import { ActionsObservable,
createEpicMiddleware } from 'redux-observable';
import { filter, mergeMap } from 'rxjs/operators';
Now create a rootEpic and add it to the middleware array. This monitors the 'epics$' subject we created, ignores falsey values and merges each Epic into the chain.
```

```
const rootEpic = (action$:
ActionsObservable<any>, store: NgRedux<any>) => {
  return storeService.epics$.pipe(
    filter(epic => !!epic),
    mergeMap(epic => {
      return epic(action$, store);
    })
  );
  const middleware = [createEpicMiddleware
    (rootEpic)];
}
We also need to do something similar for the Reducers. Do this after the 'configureStore' block:
```

```
const injectedReducers = {};
storeService.reducers$.pipe(
  filter(reducer => !!reducer)
).subscribe((reducer) => {
  injectedReducers[reducer.name] = reducer.reducer;
  ngRedux.replaceReducer(rootReducer
  (injectedReducers));
})
```

### 10. Create the search actions

Let's see if our mechanism works by creating our search store. We're going to need Actions, Reducers and an Epic. This is when Redux starts to feel verbose!

To do this, first create 'search.actions.ts' in a 'search/store' directory:

```
import { Injectable } from '@angular/core';
import { dispatch } from '@angular-redux/store';
@Injectable()
export class SearchActionsService {
  static readonly FETCH_PAGES = 'FETCH_PAGES';
  static readonly PAGES_LOADING = 'PAGES_LOADING';
  static readonly PAGES_SUCCESS = 'PAGES_SUCCESS';
  static readonly PAGES_ERROR = 'PAGES_ERROR';
  @dispatch()
  fetchPages = (payload: {
    q: string
  }) => ({
    type: SearchActionsService.FETCH_PAGES,
    payload
  })
}
```

This defines the four Actions and the structure of the payload required by the 'fetchPages' action - the only one exposed for other components in the app to call.

### 11. Search reducers

Next up, the Reducers, which will modify the store for each Action. Create 'search.reducers.ts':

```
import { SearchActionsService } from './search.actions';
const INITIAL_STATE = {
  q: null,
  pages: [],
  _loading: false,
  _error: false };
export default function searchReducer (state =
INITIAL_STATE, action: any) {
  switch (action.type) {
    case SearchActionsService.FETCH_PAGES: {
      return {
        ...INITIAL_STATE, // start with clean state
        q: action.payload.q
      }; }
    case SearchActionsService.PAGES_LOADING: {
      return {
        ...state, // copy the previous state
        pages: [],
        _loading: true
      }; }
    case SearchActionsService.PAGES_SUCCESS: {
      const { response } = action.payload;
      const titles = response[1];
      const descriptions = response[2];
      const urls = response[3];
      return {
        ...state, // copy the previous state
        pages: titles.map((title, index) => {
          return {
            title,
            description: descriptions[index],
            url: urls[index]
          };
        }),
        _loading: false
      }; }
    case SearchActionsService.PAGES_ERROR: {
      return {
        ...INITIAL_STATE, // reset state
        _error: true
      }; }
  }
  return state; }
```

### 12. Search epics

The Epic is the piece of the puzzle that handles side-effects. It can listen for particular Actions and return another stream of Actions.

This is particularly useful for async things, such as making an Ajax request, which start with an Action (e.g. 'FETCH\_PAGES') and at some point in the future will receive the data and dispatch another Action (e.g. 'PAGES\_SUCCESS').

Let's create a simple Epic that listens for the 'FETCH\_PAGES' Action, has access to the Action's payload, but doesn't do anything yet. Do this in 'search.epics.ts':

```
import { Injectable } from '@angular/core';
import { NgRedux } from '@angular-redux/
```

## Redux DevTools

Installing this extension will enable you to inspect your store in DevTools. You can get it from here: [extension.remotedev.io](https://extension.remotedev.io)

The screenshot shows the official Redux website's 'Introduction' page. The left sidebar contains a navigation menu with sections like 'Introduction', 'Motivation', 'Core Concepts', 'Three Principles', 'Prior Art', 'Learning Resources', 'Ecosystem', 'Examples', 'Basics', 'Advanced', 'Recipes', 'FAQ', 'Troubleshooting', 'Glossary', 'API Reference', 'Change Log', 'Patrons', and 'Feedback'. The main content area has a title 'Introduction' with a subtitle 'I don't understand what I'm doing'. It lists 'Motivation', 'Core Concepts', 'Three Principles', 'Prior Art', 'Ecosystem', and 'Examples'. Below this is a 'Read Me' button and a 'Was this page helpful?' section with three rating icons. At the bottom, there's a footer with links to GitHub, Discord, and a 'Need Help?' section.

## Do we really need Redux?

It's worth noting that there has been a lot of hype around Redux, however it is definitely not a necessity – even in large applications. Sure, it can be very helpful, but it does come with some significant overhead. Firstly, it has quite a steep learning curve, so consider if that is suitable for your team and your project. Secondly it can be very verbose – it can feel like you have to write quite a bit of code to do even the simplest thing. And, finally, you can easily introduce hard-to-spot bugs which have been caused by impure reducers.

The most important thing is to carefully consider the management of your state, especially with single-page applications. You can absolutely do this without Redux, but it does offer great tooling, plenty of tutorials and a standardised way of doing things. Your call.

```
store';
import { ActionsObservable } from 'redux-observable';
import { switchMap } from 'rxjs/operators';
import { empty } from 'rxjs/observable/empty';
import { SearchActionsService } from './search.actions';
@Injectable()
export class SearchEpicsService {
  onFetchPages = (action$:
  ActionsObservable<any>, store: NgRedux<any>) => {
    return action$.ofType(SearchActionsService.FETCH_PAGES).
    pipe(
      switchMap(({ payload
      }) => {
        console.log(payload);
        return empty(); // Return an empty stream. We'll replace this later
      })
    ); } }
```

## 13. Hook up to main store

The last step is to inject the search module's Reducers and Epics into the main store when the module is activated. For this let's create 'search-store.module.ts' alongside 'search-routing.module.ts':

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { combineReducers } from 'redux-observable';
import { StoreService } from '../store/store.service';
import { SearchActionsService } from './store/search.actions';
import { SearchEpicsService } from './store/search.epics';
import s
@NgModule({
  imports: [CommonModule],
  providers: [
    SearchActionsService,
    SearchEpicsService
  ] })
```

```
export class SearchStoreModule {
  constructor (storeService: StoreService,
  searchEpics: SearchEpicsService) {
    storeService.reducers$.next({
      name: 'search', // this will be the
      object name
      reducer: searchReducer
    });
    storeService.epics$.next(combineEpics(
      searchEpics.onFetchPages,
      // ... add each epic here
    )); } }
```

Finally we need to import this module into 'search.module.ts' and add it to the imports array.

Now to check that it is working OK open up the Redux pane in DevTools and refresh the home page. Click on State and you should see the original 'example: { hello: "world" }' state.

Now (using the nav) go to the Search route. Check the state again in DevTools and you should now see the 'search' object. Ta-dah! This is your Search module's store.

## 14. Import some modules

We need to import the FormModule into 'search.module.ts'. While we're here, let's also import the HttpClientModule so we can make a request to the API. In a real app we'd probably import these in the root 'app.module.ts' so they can be used throughout.

```
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
```

## 15. Add a form

We want to be able to fetch some data from Wikipedia, so let's add a simple form. Crack open the search 'index.component.html' file and update the contents to this:

```
<form (ngSubmit)="onSubmit()"#searchForm="ngForm">
  <input type="search" name="query"
  [(ngModel)]="query" required>
```

The screenshot shows the Redux DevTools Inspector. The left panel is titled 'Inspector' and has a 'filter...' input field with '@@INIT' selected. The right panel is titled 'Demo' and shows the 'State' tab with a tree view. The tree structure includes 'example' (with 'hello' set to 'world') and 'search' (with 'q' set to null, 'pages' set to an empty array, 'loading' set to false, and 'error' set to false). At the bottom, there's a 'Dispatch' button.

# Tutorials

## State management with Angular

```
<button type="submit"
[disabled]="!searchForm.form.valid">Search</button> </form>
```

### 16. Add the results list

Time to add some simple HTML for displaying our search results. Add this directly below the form:

```
<hr>
<div *ngIf="q$ | async">
  <div class="alert alert-info">
    *ngIf="loading$ | async; else
    resultsBlock>Loading...</div>
  <ng-template #resultsBlock>
    <h1>Results for {{ q$ | async }}</h1>
    <div *ngIf="(pages$ | async).length; else
    noResultsBlock">
      <div *ngFor="let page of pages$ | async">
        <h3><a [attr.href]="'page.url'">{{ page.
        title }}</a></h3>
        {{ page.description }}
      <hr>
    </div>
    <ng-template #noResultsBlock>
      <em>No results</em>
    </ng-template>
    </ng-template>
  </div>
<div class="alert alert-danger">
  *ngIf="error$ | async">An error has
  occurred</div>
```

We're using the AsyncPipe so that Angular manages our subscriptions for us.

### 17. Connect to the store

Now for the fun stuff (finally). We need to hook up our IndexComponent with the store so that it can add the search query and receive the results. Thanks to RxJS this is very simple. This is what you'll need:

```
import { Component } from '@angular/core';
import { Observable } from 'rxjs/Observable';
import { select } from '@angular-redux/store';
import { SearchActionsService } from '../../../../../store/search.actions';
@Component({
  selector: 'app-index',
  templateUrl: './index.component.html',
  styleUrls: ['./index.component.css']
})
export class IndexComponent {
  // Selects properties from the store and
  creates Observables
  @select(['search', 'q']) q$:
  Observable<string>;
  @select(['search', 'pages']) pages$:
  Observable<any[]>;
  @select(['search', '_loading']) loading$:
  Observable<boolean>;
  @select(['search', '_error']) error$:
  Observable<boolean>;
```

```
// The value of the input
query: string;
constructor (private searchActions:
SearchActionsService) { }
onSubmit () {
  // Dispatches a FETCH_PAGES action with
  the value of the input
  this.searchActions.fetchPages({
    q: this.query
  });
}
```

The `@select` lines select a property from the store and create an Observable, meaning you can bind directly to this in your template using the AsyncPipe. Changes to the value in the store will automatically be reflected in your UI. When the form is submitted we dispatch an Action to the store, containing the value of the input in the search field. This is processed by our Reducer and added to the store. Our Epic is also listening so you should see the payload logged to the console.

### 18. Update the Epic

We need to update our Epic so that it loads the data and adds it to the store. In order to do this it will dispatch Actions, as we did with the form above. First off we need to import some more things into 'search.epics.ts':

```
import { HttpClient, HttpParams,
HttpErrorResponse } from '@angular/common/
http';
import { catchError, concat, switchMap,
takeUntil } from 'rxjs/operators';
import { of } from 'rxjs/observable/of';
Inject HttpClient in the constructor as 'http' so we can use
it in our Epic. Also on the first line inside 'onFetchPages'
– before the return – add this line:
const cancel$ = action$.
ofType(SearchActionsService.FETCH_PAGES);
```

### 19. Setting up the request flow

Let's replace the 'console.log' and 'empty' Observable with our real request. First, we define the url and then set up the necessary query params:

```
const url = 'https://en.wikipedia.org/w/api.
php';
// Set up the query params we need
const params = new HttpParams({
  fromObject: {
    // Required by Wikipedia
    action: 'opensearch',
    format: 'json',
    formatversion: '2',
    namespace: '0',
    limit: '10',
    origin: '*',
    // Our search term
    search: payload.q  }
});
```

Now we use Angular's HttpClient, which wraps the request in an Observable, and emit either PAGES\_SUCCESS (with the response as the payload) or PAGES\_ERROR, depending on the result of the request:

```
// Makes the request and switches to
```

```
success / error Action
const request$ = this.http.get(url, {
  params
}).pipe(
  // Handles success
  switchMap((response: any) => {
    // Emit PAGES_SUCCESS with the response
    return of({
      type: SearchActionsService.PAGES_SUCCESS,
      payload: {
        response
      }
    });
  },
  // Handles failure
  catchError((response: HttpErrorResponse) => {
    // Emit PAGES_ERROR
    // In reality we would inspect the actual
    error response and even
    // retry the request automatically
    return of({
      type: SearchActionsService.PAGES_ERROR
    });
  }),
  // Cancels any inflight request if a new
  search is made
  takeUntil(cancel$)
);
return request$;
```

Note the tiny 'takeUntil' line. This will cancel any inflight requests if another FETCH\_PAGES Action is dispatched, i.e. the user has made a new search.

### 20. Wrapping up

At this point the search will work, but it doesn't display the loading message. So we need to dispatch the PAGES\_LOADING Action first and then run the request. We can do that by concatenating the two Actions like this (replace the existing return request\$ line):

```
const loading$ = of({
  type: SearchActionsService.PAGES_LOADING
});

// Emit PAGES_LOADING first, then run
request
return loading$.pipe(
  concat(request$)
);
```

Give it a go in the browser. Try searching and you should see your results listed in the page and see the store updating in DevTools.

### 21. In conclusion

The real power comes when managing complexity. Defining the stream of Actions – and data – in the Epics helps reason about the flow and makes it inherently reusable across components.

Processing and shaping the data in the Reducers also feels very neat and means you only store what you really need, in a format that suits your use-case. And, as you saw, this keeps the components very clean meaning they can focus on the UI.

Special offer for readers in *North America*



# 6 issues FREE

When you subscribe\*

**FREE**  
resource  
downloads  
every issue

30 PAGE EXPERTS

**web designer** FREE 107 MINUTES OF JS VIDEOS

**DESIGN FASTER**

**AWESOME AUDIO**

Add the power of sound to your projects

**CODE QUICKER**

> 50 must-try tools for today's designers and developers

- > Frameworks
- > Prototyping
- > Browser extensions
- > IDEs > Code playgrounds
- > Testing > Animation
- > Colour > Fonts

+  
INTERACTIVE IMAGES  
ANGULAR STATE MANAGEMENT  
ES MODULES IN JAVASCRIPT  
UI TILT EFFECT WITH SVG

HTML CSS JS JavaScript

PROGRESSIVE WEB APPS  
Get offline support with Google's Workbox

**“** *The only magazine  
you need to design  
and develop  
stunning websites* **”**



Order hotline **+44 (0) 344 848 2852**  
Online at **myfavouritemagazines.co.uk/WEDPQ17**

**\*Terms and conditions** This is a US subscription offer. 6 free issues refers to the USA newsstand price of \$14.49 for 13 issues at \$188.37, compared with \$103 for a subscription. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. 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 upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: [bit.ly/magtandc](http://bit.ly/magtandc) Offer ends July 31 2018.

**Expires  
31 July  
2018**

# Understand how ES modules work

ECMAScript 6 brings a new native modularisation API intended to see off existing JavaScript modularisation technologies



 Both object-oriented and traditional programming languages provided developers with some way of ‘access control’ – by breaking down libraries and other elements into modules of their own, outside meddling with internal data structures can be avoided. Furthermore, the act of declaring an interface between a program and its environment is beneficial for all parties involved. In addition to the clear separation of concerns, developers tend to improve code quality and concept coherency while working out the details of a module access API.

JavaScript did not provide a modularisation API of its own. This has led to the appearance of various modularisation systems like CommonJS, which will be discussed tangentially in this article. Some developers also resorted to server-side or workstation-based transpilation, where groups of files were bundled together before delivery.

While all of these workarounds are fine and well – when used properly, JavaScript, today, is a first-class language and getting a native way to handle modularisation is much more convenient. ECMAScript 2015 – the standard is also known as ES6 – provides exactly that with its modularisation API.

### 1. Getting started

Getting started with ECMAScript requires the presence of a worker file. Enlist the structure shown accompanying this step to get started – it contains a ‘work area’ and loads a module.

```
<html>
  <body>
    <script type="module" src="./mainmodule.js"></script>
    <script>puber();</script>
  </body>
</html>
```

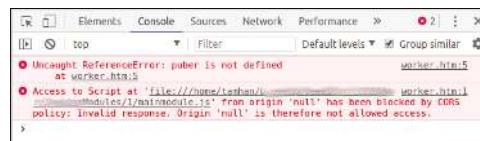
### 2. Alternative load

Using a dedicated `<script>` tag for loading the modules is not to everyone’s taste. The module import syntax normally intended for inter-module imports can, however, also be applied to script tags hosted inside of webpages.

```
<script type="module">
  import { puber } from './tree.js';
  ...
</script>
```

### 3. First test

Given that our test harness requires the presence of a function called ‘lives\_in\_module’, we can – in theory – simply declare it in ‘mainmodule.js’. Sadly, running the program yields the results shown in the figure accompanying this step.



### 4. Break out

As mentioned in the introduction, one of the most significant benefits of modularisation is the capability to compartmentalise programs cleanly. Due to that, we need to export the functions needed by adding the export statement below. Then, proceed to running the website once again.

```
export function puber(){
  console.log("puber() says Hello!");
}
```

### 5. Fix the CORS issues

Like most other dynamic content, ECMAScript is also subject to various methods of origin confirmation. When working on an unixoid system, Python provides a nice workaround – simply enter the command shown accompanying this step to spin up a little web server.

```
sudo python -m SimpleHTTPServer 80
```

```
laptop@laptop:~/Desktop$ sudo python -m SimpleHTTPServer 80
[sudo] password for laptop:
Serving HTTP on 0.0.0.0 port 80...
[07/08/2018 18:54:31] "GET /worker.htm HTTP/1.1" 200 -
[07/08/2018 18:54:31] "GET /mainmodule.js HTTP/1.1" 200 -
[07/08/2018 18:54:31] "GET /mainmodule.htm HTTP/1.1" 200 -
[07/08/2018 18:54:31] "GET /favicon.ico HTTP/1.1" 404 -
[07/08/2018 18:54:31] "GET /worker.htm HTTP/1.1" 200 -
[07/08/2018 18:54:31] "GET /mainmodule.htm HTTP/1.1" 200 -
[07/08/2018 18:54:46] "GET /worker.htm HTTP/1.1" 200 -
[07/08/2018 18:54:46] "GET /mainmodule.js HTTP/1.1" 200 -
[07/08/2018 18:54:46] "GET /mainmodule.htm HTTP/1.1" 200 -
```

### 6. Imports, redux

Inserting a random `<script>` tag to load a module is not enough to get our program working. If two modules would expose a function with the same name, a namespace collision would occur according to the scheme shown in the image below.

### 7. A question of need is a question of taste!

Including elements directly and into a namespace of its own has its merits. However, namespace contention is proportional to project size. Thus, large projects are well-advised to avoid direct includes to prevent problems as the amount of library code increases.

### 8. Limited include

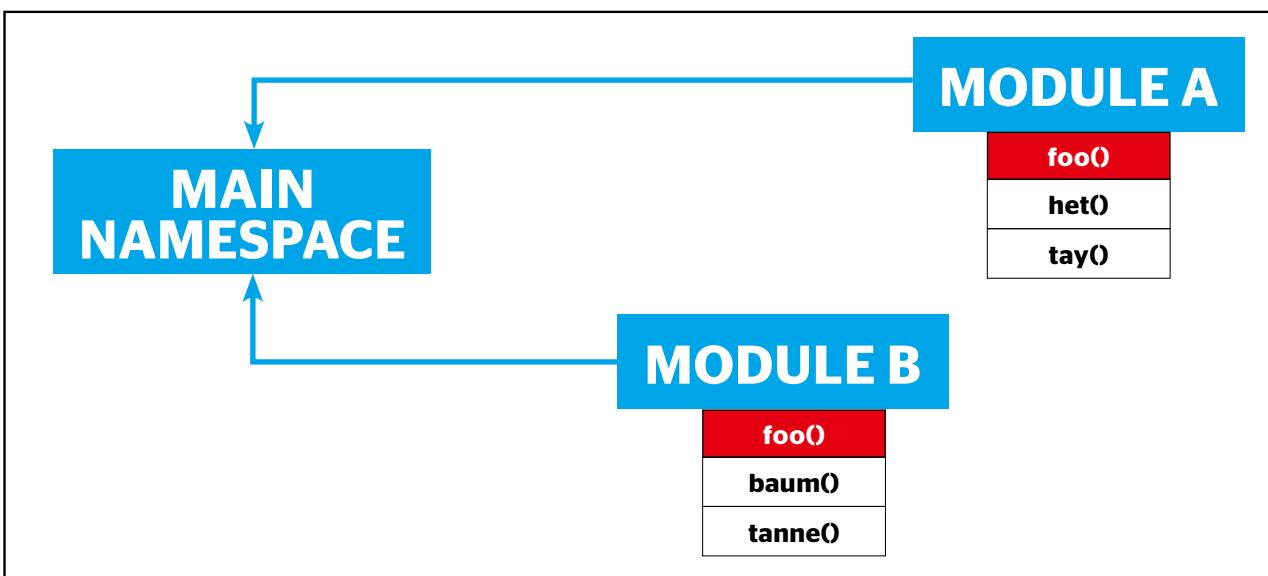
As a first ‘revision’, let us load the module into the ‘TamsModule’ namespace. This requires no changes in the module file – instead, modify the worker harness’s code as shown in the listing accompanying this step.

```
<body>
  <script type="module" src=".mainmodule.js"></script>
  <script type="module">
    import * as TamsModule from "./mainmodule.js";
    TamsModule.puber();
  </script>
```

## What about Node.js?

Developers who use Node-based modules typically use the CommonJS ([www.commonjs.org](http://www.commonjs.org)) syntax. However, the Node modules working group is working on adding ES module support to Node.js.

**Left**  
If modules would be loaded globally, the red function tiles would collide



# Developer tutorials

Understand how ES modules work

```
</script>  
</body>
```

## 9. General include

The second option loads the module part into the global namespace. This does not preclude you from loading additional parts into other namespaces – keep in mind that each module file gets parsed when it gets addressed the first time. This caching lets you save HTTP latency, thereby increasing application performance.

```
<body>  
  <script type="module" src="./mainmodule.  
js"></script>  
  <script type="module">  
    import { puber } from "./mainmodule.js";  
    puber();  
  </script>  
</body>
```

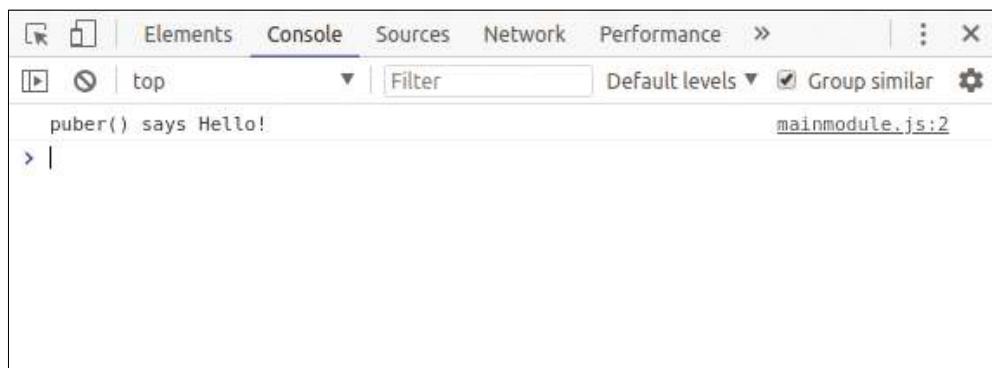
## 10. Even more includes

Opening the Modules feature documentation at <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import> reveals almost a dozen different include syntaxes, all of which are demonstrated in action.

```
import defaultExport from "module-name";  
import * as name from "module-name";  
import { export } from "module-name";  
import { export as alias } from "module-  
name";  
import { export1 , export2 } from "module-  
name";  
import { export1 , export2 as alias2 ,  
[...] } from "module-name";  
import defaultExport, { export [ , [...] ]
```

## ./ is life

Simply passing in the file name of the module in question is not allowed in most browsers. Instead, use `./` to inform the loader that the file lives in the current folder.



```
} from "module-name";  
import defaultExport, * as name from  
"module-name";  
import "module-name";
```

## 11. Check encapsulation

In the next step, let us verify that our module's content really is private. For this, we create a second, non-exported function which gets called from the test harness. When invoked via `puber()`, all is fine – invoking `foo()` directly will lead to a permission denied error like the one shown in the screenshot below.

```
export function puber(){  
  console.log("puber() says Hello!");  
  foo();  
}  
function foo(){  
  console.log("foo() also says Hello!");  
}
```



## 12. Create a singleton

While the `export` statement can also be applied to a class, in some cases, creating a 'common' singleton in the module itself can be convenient. For this, let us start out with another test harness which emits a message during loading and furthermore adds some state variables.

```
var myName="";  
var myWeight=0;  
console.log("Singleton global load done!");
```

## 13. Add accessory functions

While we could simply put an `export` in front of the member variables, this allows users to wreak havoc inside them. Instead, we will implement a set of functions which marshal the values of `myName` and `myWeight` into and out of the module. Incidentally, the length of the getter functions for local variables is a common eyesore – look at <https://stackoverflow.com/questions/4475177/how-to-export-getter-in-es6-module-system> for a

### Right

Mozilla recently added support for modules. Make sure to check out the first-class documentation

Jump to: [Syntax](#) [Description](#) [Examples](#) [Specifications](#) [Browser compatibility](#) [See also](#)

[Web technology for developers](#) > [JavaScript](#) >  
[JavaScript reference](#) >  
[Statements and declarations](#) > [Import](#)

The `import` statement is used to import bindings which are exported by another module.

This feature is only implemented natively in Safari, Chrome, and Edge at this time. It is implemented in many transpilers, such as the [Traceur Compiler](#), [Babel](#), and [Rollup](#).

### Related Topics

#### [JavaScript](#)

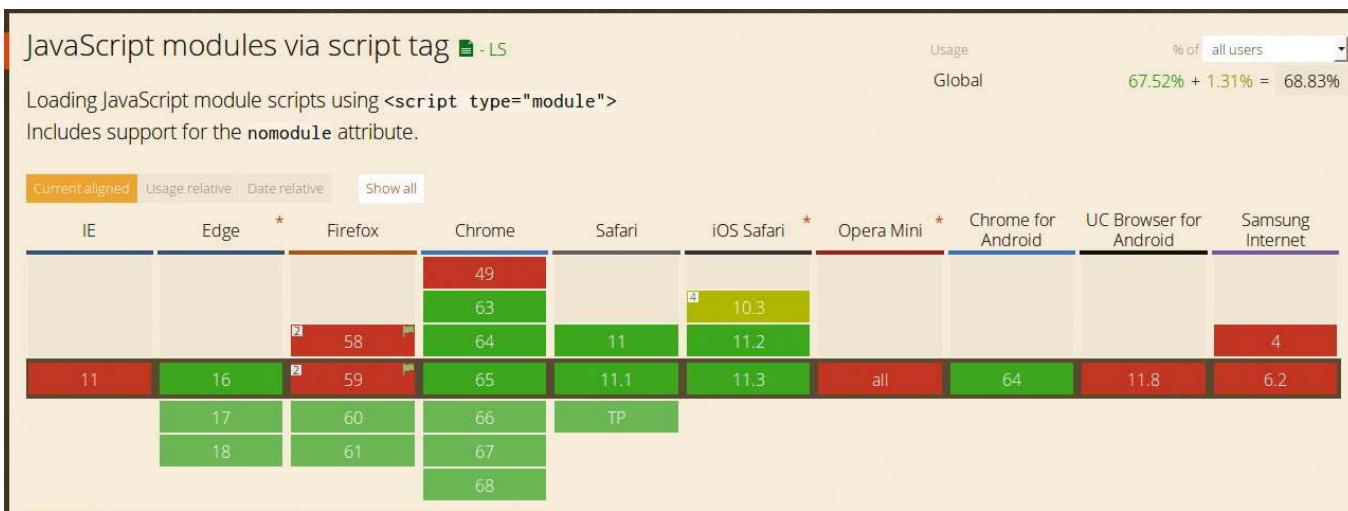
##### Tutorials:

- ▶ Complete beginners
- ▶ JavaScript Guide
- ▶ Intermediate
- ▶ Advanced

##### References:

- ▶ [Built-in objects](#)

## Syntax



## A question of support

Traditionally, new JavaScript technologies need quite a bit of time before reaching wide-scale market acceptance. As quite some time has gone by since ECMAScript modules first saw the day, they have achieved somewhat wide-ranging industry

support. The latest versions of Edge, Chrome and Safari all support modules out of the box, and Mozilla has finally joined the party with the recent release of Firefox 60. This ensures that browser support is now applicable to all the major vendors.

Furthermore a set of 'adapter systems' are available. They let developers wrap ECMAScript modules into a more 'compatible' way, allowing the code to be reused on older platforms. Check out <https://caniuse.com> for latest updates.

discussion of the topic.

```
export function getName(){
  return myName;
}

export function getWeight(){
  return myWeight;
}

export function setName(what){
  myName=what;
}

export function setWeight(what){
  myWeight=what; }
```

## 14. Spin it

Our test code loads the module twice; once into the 'TamsModule' namespace, and once into the namespace 'TamsModule2'. The changes made in the 'TamsModule'

instance will also propagate into 'TamsModule2' – this can easily be verified by looking at the output of the program in the console. Incidentally, this behaviour is caused by the above-mentioned caching system, which loads the module but once for each page.

```
import * as TamsModule from "./mainmodule.js";
console.log(TamsModule.getName());
TamsModule.setName("Its a name");
import * as TamsModule2 from "./mainmodule.js";
console.log(TamsModule2.getName());
```

## 15. Export classes quickly

Should you create a module which contains one class, setting a default export is a particularly neat way to simplify access. In particular, the two code snippets

accompanying this step demonstrate both the use of the 'export default' attribute and inclusion of the class into a worker or a test harness. The file name worker.js, incidentally, is not a typo – the ECMAScript module system also supports JavaScript-on-JavaScript includes.

```
----- TamsClass.js -----
export default class { ... };
----- worker.js -----
import TamsClass from 'filename';
let aClass = new TamsClass();
```

## 16. Eliminating latency

With that out of the way, let us return to our singleton. It contains some global code, which gets run during the initialisation of the program. At this point, an interesting question remains – how does the JavaScript runtime handle delays caused by network latencies?

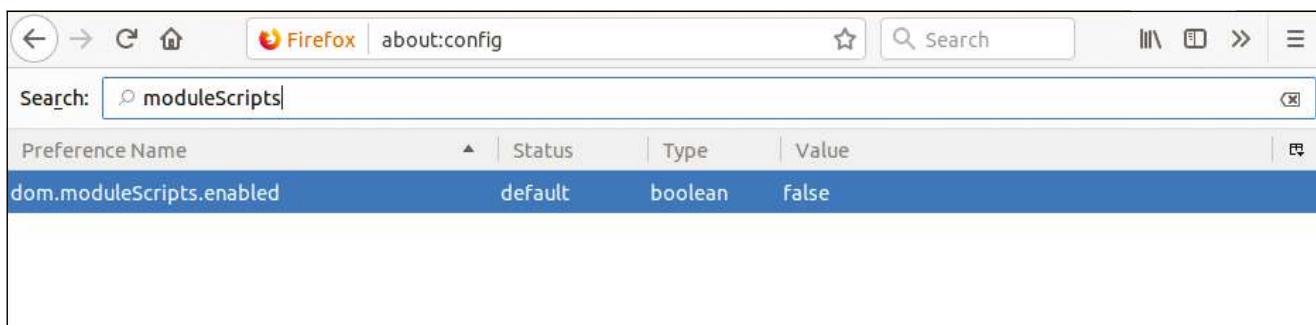
Request	Response
worker.htm:13	Defer Inline Script 1
worker.htm:16	Inline Script 1
worker.htm:19	Defer Inline Script 2
mainmodule.js:23	Singleton global load done!
worker.htm:7	
worker.htm:10	Its a name

## Developer tutorials

Understand how ES modules work

### Right

The advanced rendering settings display can be filtered using the text box at the top of the screen



## 17. Check it out

Some questions are best answered with a little test. Thus, replace the worker code in our testing harness with the one accompanying this step - it contains the singleton operations, and also a few 'local' script blocks. Run it, or take a look at the rendered output in the figure accompanying this step.

```
<script type="module" src=".mainmodule.js"></script>
<script type="module">
import * as TamsModule from "./mainmodule.js";
console.log(TamsModule.getName());
TamsModule.setName("Its a name");
import * as TamsModule2 from "./mainmodule.js";
console.log(TamsModule2.getName());
</script>
<script defer>
  console.log("Defer Inline Script 1");
</script>
<script defer>
  console.log("Inline Script 1");
</script>
<script defer>
  console.log("Defer Inline Script 2");
</script>
```

## 18. Understand the hierarchy

Script tags equipped with the module attribute are assigned a lower execution priority, which - at the time of writing - can not be modified by the developer. Basically, a module script is always equipped with a defer command - treat it like any other valid use of the defer order in a script tag.

## 19. Beat the module

The above-mentioned inclusion of the type="module" string is not born from malevolence on the side of the ECMAScript standardisation. Instead, it uses a peculiarity of HTML - browsers ignore script tags with unknown types. This permits the use of the following fallback structure, which is supported by quite a few browsers. Alternatively, an even better - albeit quite involved - workaround can be found at <https://medium.com/dev-channel/es6-modules-in-chrome-canary-m60-ba588dfb8ab7>.

```
<script type="module" src="has-modules.js">
```

```
</script>
<script nomodule src="has-no-modules.js">
</script>
```

## 20. eval() isn't ideal

While the use of eval() is not to everyone's liking, the number one source of weird program behaviour remains in widespread use. Sadly, code passed to eval() is not allowed to interact with functions or other elements exported from modules. This odd limitation is caused by the underlying loader, and is part of the standard.

## 21. Work with Firefox

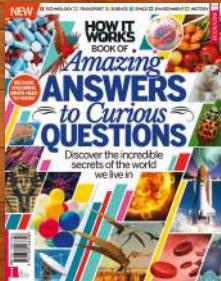
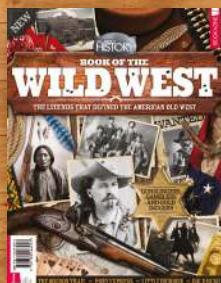
Firefox provides developers with a set of first-class developer tools. If you want to use Mozilla's browser for debugging and testing module-based programs, fret not. Simply open the website <about:config>, and look for the attribute dom.moduleScripts.enabled. Next, set it to true - one restart, and all is golden.

## 22. Find out more

One of the best resources/discussions of the module system can be found as part of Exploring JS. Head to <https://bit.ly/2w2iLft> for a deep dive.

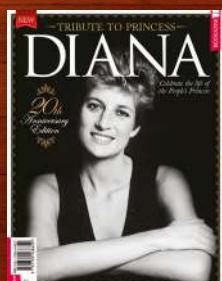
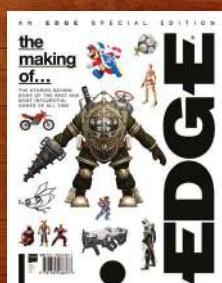
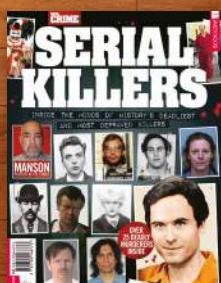
# 16. Modules

- 16.1. Overview
  - 16.1.1. Multiple named exports
  - 16.1.2. Single default export
  - 16.1.3. Browsers: scripts versus modules
- 16.2. Modules in JavaScript
  - 16.2.1. ECMAScript 5 module systems
  - 16.2.2. ECMAScript 6 modules
- 16.3. The basics of ES6 modules
  - 16.3.1. Named exports (several per module)
  - 16.3.2. Default exports (one per module)
  - 16.3.3. Imports and exports must be at the top level
  - 16.3.4. Imports are hoisted
  - 16.3.5. Imports are read-only views on exports
  - 16.3.6. Support for cyclic dependencies
- 16.4. Importing and exporting in detail
  - 16.4.1. Importing styles
  - 16.4.2. Named exporting styles: inline versus clause
  - 16.4.3. Re-exporting
  - 16.4.4. All exporting styles
  - 16.4.5. Having both named exports and a default export in a module
- 16.5. The ECMAScript 6 module loader API
  - 16.5.1. Loaders
  - 16.5.2. Loader method: importing modules
  - 16.5.3. More loader methods
  - 16.5.4. Configuring module loading



## 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**  
Magazines, back issues & bookazines.



## Get your listing in our directory

To advertise here contact *Chris*

[chris.mitchell@futurenet.com](mailto:chris.mitchell@futurenet.com)

+44 (0)1225 687832

# HOSTING LISTINGS



## Featured host: Netcetera

[netcetera.co.uk](http://netcetera.co.uk)  
03330 439780

### About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide.

As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services designed to more effectively manage IT

infrastructures. A state-of-the-art data centre environment enables Netcetera to offer your business enterprise-level colocation and hosted solutions.

Providing an unmatched value for your budget is the driving force behind our customer and managed infrastructure services. From single server to fully customised data centre suites, we focus on the IT solutions you need.

### What we offer

- Managed hosting** - A full range of solutions for a cost-effective, reliable, secure host.
- Cloud hosting** - Linux, Windows, Hybrid and Private Cloud Solutions with support and scalability features.

- Data centre colocation** - Single server through to full racks with FREE setup and a generous bandwidth.
- Dedicated servers** - From QuadCore up to Smart Servers with quick setup and fully customisable.

### 5 tips from the pros

#### 1. Reliability, trust & support

Reliability is a major factor when it comes to choosing a hosting partner. Netcetera guarantees 100 per cent uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

#### 2. Secure and dependable

Netcetera prides itself on offering its clients a secure environment. It is accredited with ISO 27001 for security along with the options of configurable secure rackspace available in various configurations.

#### 4. Value for money

We do not claim to be the cheapest service available, but we do claim to offer excellent value for money. We also provide a price match on a like-for-like basis, as well as a price guarantee for your length of service.

#### 5. Eco-friendly

Netcetera's environmental commitment is backed by use of eco-cooling and hydroelectric power. This makes Netcetera one of the greenest data centres in Europe.



### Testimonials

#### Roy T

"I have always had great service from Netcetera. Their technical support is second to none. My issues have always been resolved very quickly."

#### Suzy B

"We have several servers from Netcetera and their network connectivity is top-notch, with great uptime and speed is never an issue. Tech support is knowledgeable and quick in replying. We would highly recommend Netcetera."

#### Steve B

"We put several racks into Netcetera, basically a complete corporate backend. They could not have been more professional, helpful, responsive or friendly. All the team were an absolute pleasure to deal with, and nothing was too much trouble, so they matched our requirements 100 per cent."

## Supreme hosting



[cwcs.co.uk](http://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.

- Colocation hosting
- VPS
- 100 per cent network uptime

## UK-based hosting



[cyberhostpro.com](http://cyberhostpro.com)  
0845 5279 345  
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



[fasthosts.co.uk](http://fasthosts.co.uk)  
0808 1686 777  
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

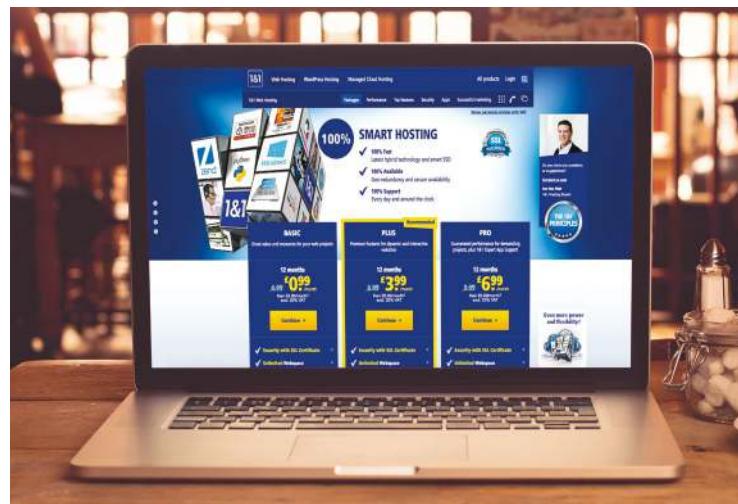


[hetzner.com](http://hetzner.com)  
+49 (0)9831 505-0

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 nationally and internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



## All-inclusive hosting



[1and1.co.uk](http://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



[bargainhost.co.uk](http://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, by email and with a 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](mailto:chris.mitchell@futurenet.com)

+44(0)1225 687832

# COURSE LISTINGS



## Featured: **Northcoders**

[northcoders.com](http://northcoders.com)  
Twitter: @northcoders  
Facebook: Northcoders

### About us

Northcoders is the coding bootcamp for the north, based in the heart of Manchester and built upon northern values of grit, determination and community spirit. No matter what your background, you can fast-track your career and become a web or software developer in 12 weeks at their

full-time bootcamp, or fit their course around your life with their 24-week part-time bootcamp. Their internal career support team will help find you work as a developer, setting up interviews with your choices of Northcoders Hiring Partners across the north of England.

### What we offer

- **Full-time:**  
Fast-track your career in just 12 weeks.
- **Part-time:**  
Fit our curriculum around your life in 24 weeks.

### 5 tips from the pros

#### 1. Get started with coding

The best way to know if coding is for you is to just try it! We recommend the free, online JavaScript track of Codecademy to get you started with the basics.

for you, set aside a few evenings each week to really start making progress! If coding is for you, this should be fun.

#### 4. Be prepared

We'll be with you every step of the way when you apply. Make sure you go through all the materials we recommend and ask for help if you're stuck.

#### 5. Get social

With Northcoders, you're not just on a course, you're part of a community that will stay with you long after you graduate. Make the most of it!

#### 2. Do your research

Make sure you read plenty of student reviews to make sure you're applying somewhere reputable. Read their blog and have a look at their social channels.

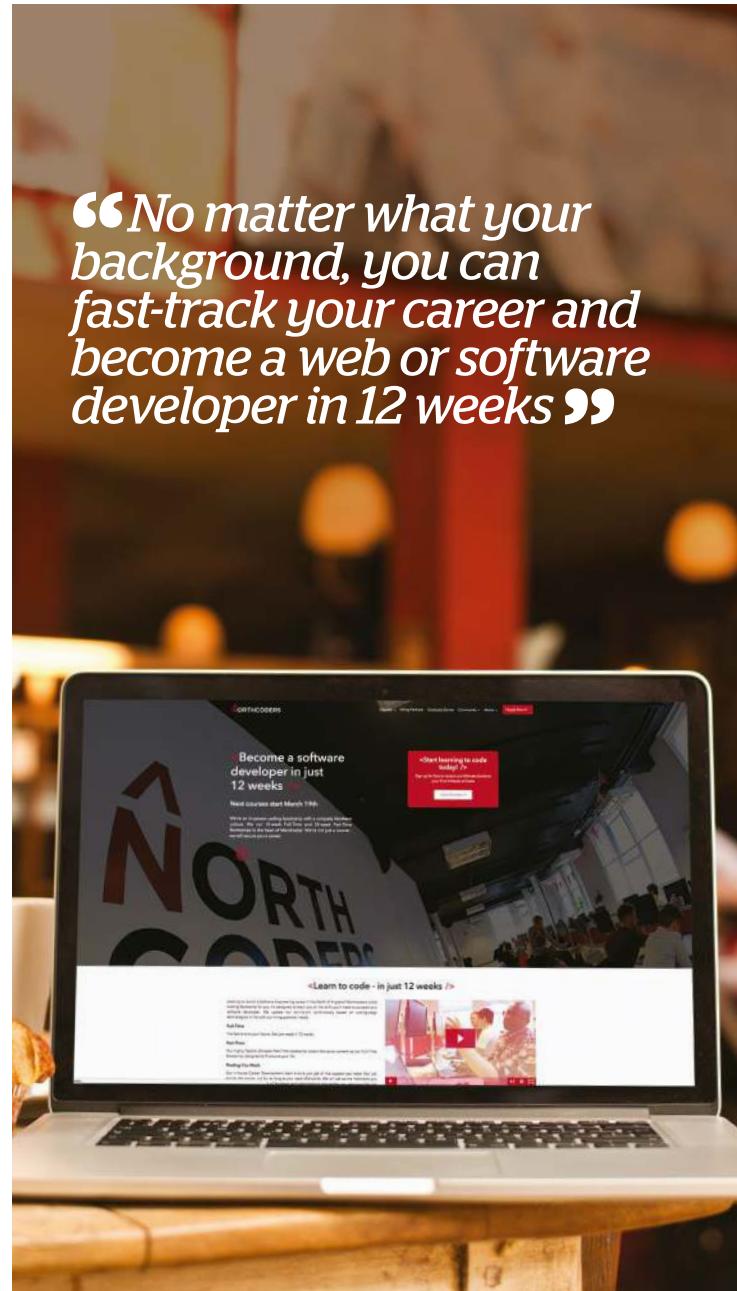
#### 3. Throw yourself in

Once you've decided it's right

**“** Becoming part of this vibrant, caring community was something I hadn't expected before the course, but now I couldn't be without it. To be a Northcoder is to be enlightened, inspired and supported.

**Joanne Imlay**

*Primary school teacher to software developer at Careicon*



**“** Northcoders delivered their part of the bargain in spades. They provided tremendous assistance in turning me into the full product - a well-rounded, capable, future tech employee - and they have the contacts to deliver the opportunities for such people.

**Joe Mulvey**

*Maths teacher to software developer at Auto Trader*

# udemy

## UDEMY

[udemy.com](http://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.



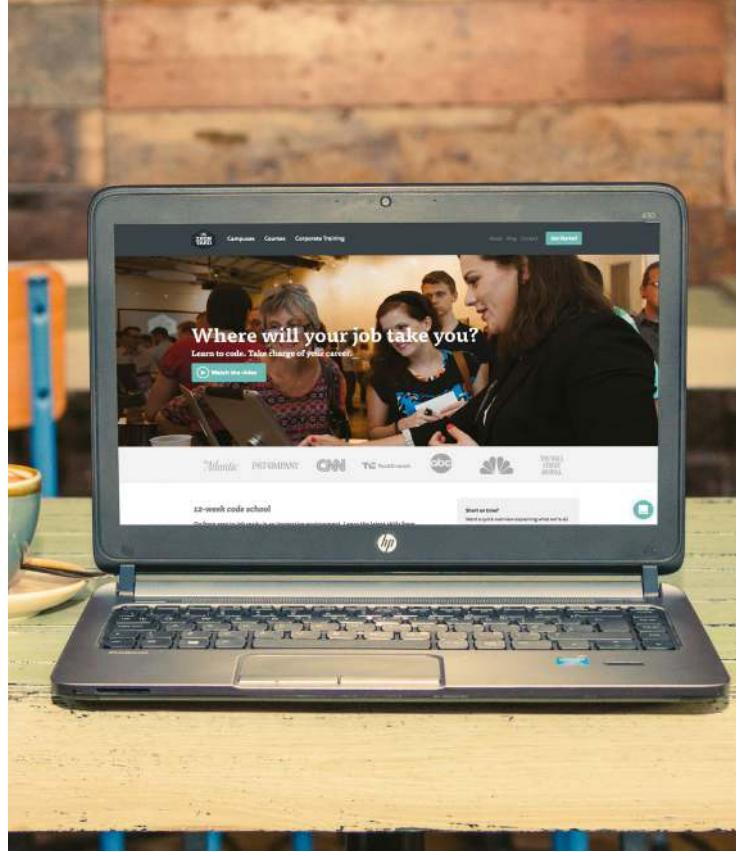
## THE IRON YARD

[theironyard.com](http://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 backend engineering, frontend 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.



## WE GOT CODERS



we got coders.com  
[hello@we got coders.com](mailto:hello@we got coders.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 courses 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



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

## GYMNASIUM



[thegymnasium.com](http://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 107 minutes of Beginners JavaScript video from StudioWeb ([www.studioweb.com](http://www.studioweb.com))
- 6 Gold clothespin-style card mockups and 14 Midnight Photoshop filters from Sparklestock ([www.sparklestock.com](http://www.sparklestock.com))

→ 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



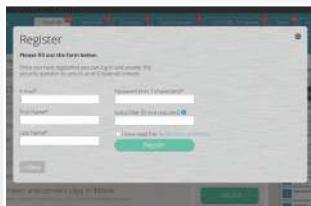


The home of great downloads – exclusive to your favourite magazines from Future!

- 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

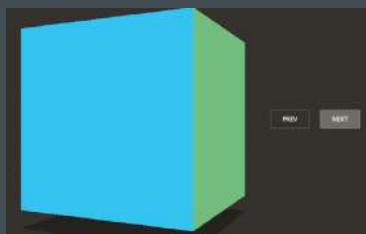
### Over 60 hours of video guides

Let the experts teach you to create and code



### More than 400 tutorials

Get the code you need to get creative



### Over 250 creative assets

Templates, fonts, textures and backgrounds



Head to page 34 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 ten-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**

+  
More added every issue

# NEXT MONTH

---

# AUGMENTED REALITY

---

## BUILD YOUR OWN APP

---

A practical guide to the best frameworks, tools and techniques needed to create your own project

### GET STARTED WITH THREEJS

The first in a five-part series takes you through the basics needed to start building with the library

### CODE A CUSTOM WEB COMPONENT

Learn how to create apps and components based on web standards with Stencil.js

### GOOGLE ANALYTICS 2018 – WHAT'S NEW?

Get an insight into the latest updates and features and how to make the most of them

Visit the **WEB DESIGNER** online shop at



**myfavouritemagazines**

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

for the latest issue, back issues and specials

ALL IN YOUR NEXT  
**WEB DESIGNER**  
Issue 276 on sale  
**Tuesday 26th June 2018**

# moqups



Mockup, wireframe, diagram, prototype  
& collaborate with one flexible solution.

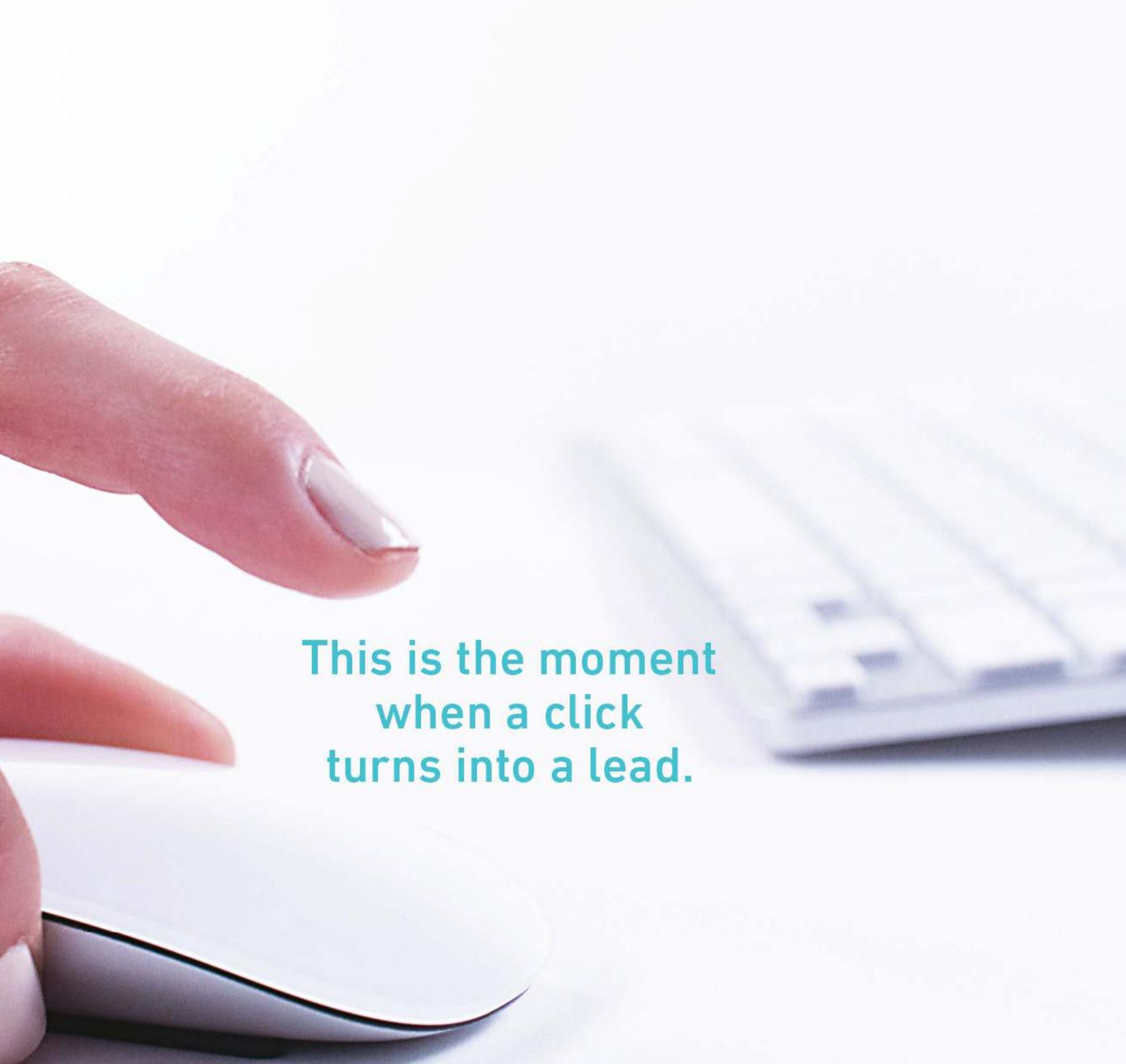
**moqups.com**

OFFER EXPIRES  
MARCH 1<sup>ST</sup> 2018

Save an extra 30% on all new Moqups plans  
Discount code: **NETMAG30**

**SAVE  
AN EXTRA 30%**





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](http://wpengine.co.uk)

**WP** engine\*

