

FREE 81 MINUTES OF EXPERT PHP VIDEOS

designer™

Expert tutorials, techniques and inspiration

GSAP MASTERCLASS

Get pro web animation skills

BETTER MOBILE
WITH AMP
& PWA

MUST-KNOW
GITHUB
REPOSITORIES

DRAW WITH JAVASCRIPT

Turn data into charts and
diagrams for the browser

MUST-VIEW
YOUTUBE
CHANNELS

HOW TO USE
CHROME
DEVTOOLS

DESIGN DEVELOP CREATE

TOP GOOGLE TOOLS
TO BUILD BETTER
APPS AND SITES

ESSENTIAL
RESOURCES

MAKE WITH
MATERIAL
DESIGN



- CODE UX PATTERNS
- ADD GLITCH TEXT EFFECTS
- MULTI-LANGUAGE ANGULAR
- IMAGE PREVIEWS ON HOVER



CODE A CSS IMAGE CHANGER

Use CSS Grid and vanilla JavaScript
to reveal new images

STORM

FASTER EVERYTHING

Introducing **STORM**, a hosting platform that helps your agency and websites run smoother. 82% of our customers have seen a 25% improvement in site speeds.

OVER 130

5★

GOOGLE REVIEWS



ARE YOU A FREELANCER?

Looking for an alternative control panel? STORM will streamline your workflow and allow you to spend more time on web development, and less time on server management.

We are offering a **14-DAY FREE TRIAL** of our revolutionary control panel STORM, with full access to explore just how much we can simplify how you manage your hosting. Don't just take it from us, take it from our clients.

*Terms and conditions apply, please visit the website for more information

Call, email or visit us at

0203 126 6787

sales@nimbushosting.co.uk

nim.host/clientstories

 **Nimbus Hosting**

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

Be friends with Google



earch is what Google is best known for, but it has so many more strings to its bow. Chrome is one, now being the most popular browser in the world with around 60 percent market share. But there is still so much more on offer for developers and designers. In our latest lead feature we take a closer look at Chrome DevTools and how they can help you tweak and create better performing code, and find out how you can use Material Design to build smart, uniform design patterns. We take a peek at how PWAs and AMPs are good news for mobile, but also work well with desktop. Plus, there is a comprehensive list of the best YouTube channels

that you need to watch, essential resources and a selection of GitHub repositories that enable you to get closer to the code that Google are creating.

If web animation is your thing, then you know that GSAP – aka Greensock – is an industry-standard tool for dynamic designs. Find out how to create tweens, use timelines for complex animations and introduce control over animations. Plus, discover what callbacks are, how to integrate with Three.js to create even more impressive animations and discover a collection of plugins to add more power and poise to the animation platform.

Elsewhere we have a collection of in-depth tutorials to help create the elements you want in your designs. Start by learning how to add contemporary glitch effects to text and images with just the power of CSS.

Chrome lets you edit most attributes on the fly. Double-click any of the black texts to transform the label into an editor

Highlight



We're always looking to the future. Innovation is a fundamental facet of how our business was first shaped and it always remains at the core of what we do

Web Designer finds out what's going on at Manchester-based J B Cole UK. 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
63 Designer resources

Video Tuition - 81 minutes of expert PHP video guides from Killersites ([shop.killervideostore.com](#))

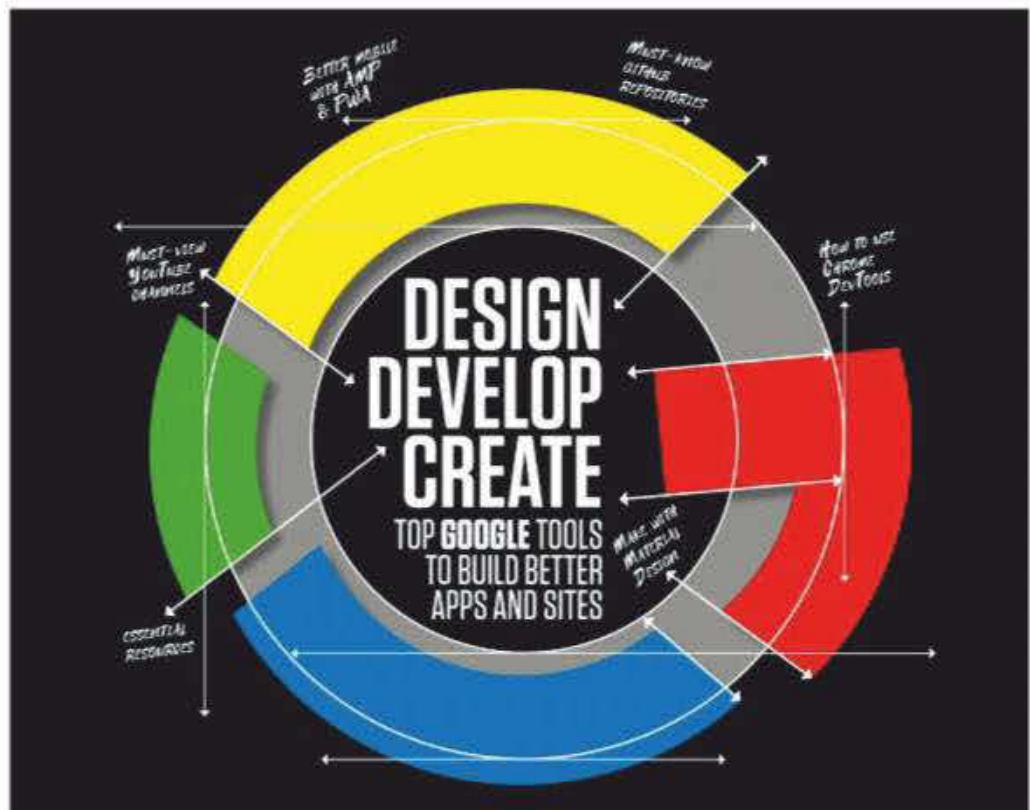
Assets - 40 Instagram Lightroom presets and Sky Replacer Photoshop Actions from Sparklestock ([sparklestock.com](#))
- Tutorial files and assets



[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...



Tam Hanna

Tam is a seasoned pro developer and works across a host of platforms and programming languages. In this issue he takes a closer look at Google and some of the tools they offer including Chrome DevTools, Material Design and PWAs. Plus, he reveals an essential collection of must-know resources.

Page 46

• The renderer picks up these changes and applies them to the DOM on the fly. This is especially useful when trying to optimise colours or placement issues •

Daniel Crisp



Daniel is a senior frontend developer at a startup in London's Canary Wharf. In this issue he takes a look at Angular's built-in internationalisation tools and shows you how to use them in your projects.

Page 80

Mark Shufflebottom



Mark is a professor of Interaction Design. In this issue Mark is exploring creating glitch animation effects. Both text and image will be given the distressed glitch animation to produce an interesting visual aesthetic with CSS.

Page 58

Richard Mattka



Richard is an award-winning interactive director, designer and developer specialising in VFX and entertainment projects. In this issue he gives the lowdown on the JavaScript animation library GSAP aka Greensock.

Page 72

Frank Kagumba



Frank is a frontend developer and tech writer based in Nairobi, Kenya. In this tutorial, he takes you through the different attributes of Layer JS library and demonstrate how to create different UX patterns.

Page 68

Mark Billen



Mark is a freelancer writer who has been writing about web design and technology for over 15 years. In this issue he finds out how the creatives at Herdl delivered the perfect site for organic food restaurant Genesis.

Page 30

David Howell



David is a journalist with over 20 years' experience in publishing and runs his own business, Nexus Publishing. In this issue he heads to Manchester to talk to the talented crew at J B Cole UK and find out how they operate.

Page 36

Neil Pearce



Neil is a long-time frontend designer and developer who works with HTML, CSS and JavaScript. In this issue show how to create a simple image changer using the power of CSS Grid and some vanilla JavaScript code.

Page 64

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

Follow us!

Facebook: [www.facebook.com/
WebDesignerUK](http://www.facebook.com/WebDesignerUK)

Twitter: [https://twitter.com/
webdesignermag](https://twitter.com/webdesignermag)

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

Editorial

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

Art Editor **Newton Ribeiro**
Editor in Chief **Amy Hennessey**
Senior Art Editor **Will Shum**

Contributors

Tamas Hajas, Mark Billen, Leon Brown, David Howell, Richard Mattka, Mark Shufflebottom, Neil Pearce, Daniel Crisp, Frank Kagumba, Tam Hanna, Ryan Wells, Stephen Mumby, Philip Morris, Rob Mead-Green

Photography

James Sheppard
All copyrights and trademarks are recognised and respected

Advertising

Media packs are available on request
Commercial Director **Clare Dove**
clare.dove@futurenet.com
Advertising Manager **Mike Pyatt**
michael.pyatt@futurenet.com
01225 687538
Account Manager **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
UK orderline & enquiries 0344 848 2852
Overseas order line and enquiries +44 (0) 344 848 2852
Online orders & enquiries www.myfavouritemagazines.co.uk
Head of subscriptions **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 Content 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 Tel: 0203 787 9001

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 paper 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 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

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



Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
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...

webdesigner@futurenet.com • [@WebDesignerMag](https://twitter.com/WebDesignerMag) • www.creativebloq.com

08 Are you ready for global eCommerce?

Web Designer takes a look at what you need to know to be a successful seller.

11 Variable fonts: Times are changing

Senior engineer at Mirum discusses the benefits of variable fonts and how to use them

12 WebKit: The best must-try resources out there

Discover the libraries and frameworks you need

16 Lightbox

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

30 A new beginning

Find out how digital creatives Herdl gave vegan fast food restaurant Genesis a touch of style

36 People Power

Discover where our ProFile protagonists, J B Cole UK, thrive and what makes them tick

44 Back issues

Catch up on any issues of Web Designer that you've missed by downloading a digital edition

46 Design, develop and create

Discover the essential Google tools you need to build better apps and websites

72 Animate with GSAP

Find out how to use the industry standard for web-based animation in your projects

92 Hosting listings

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

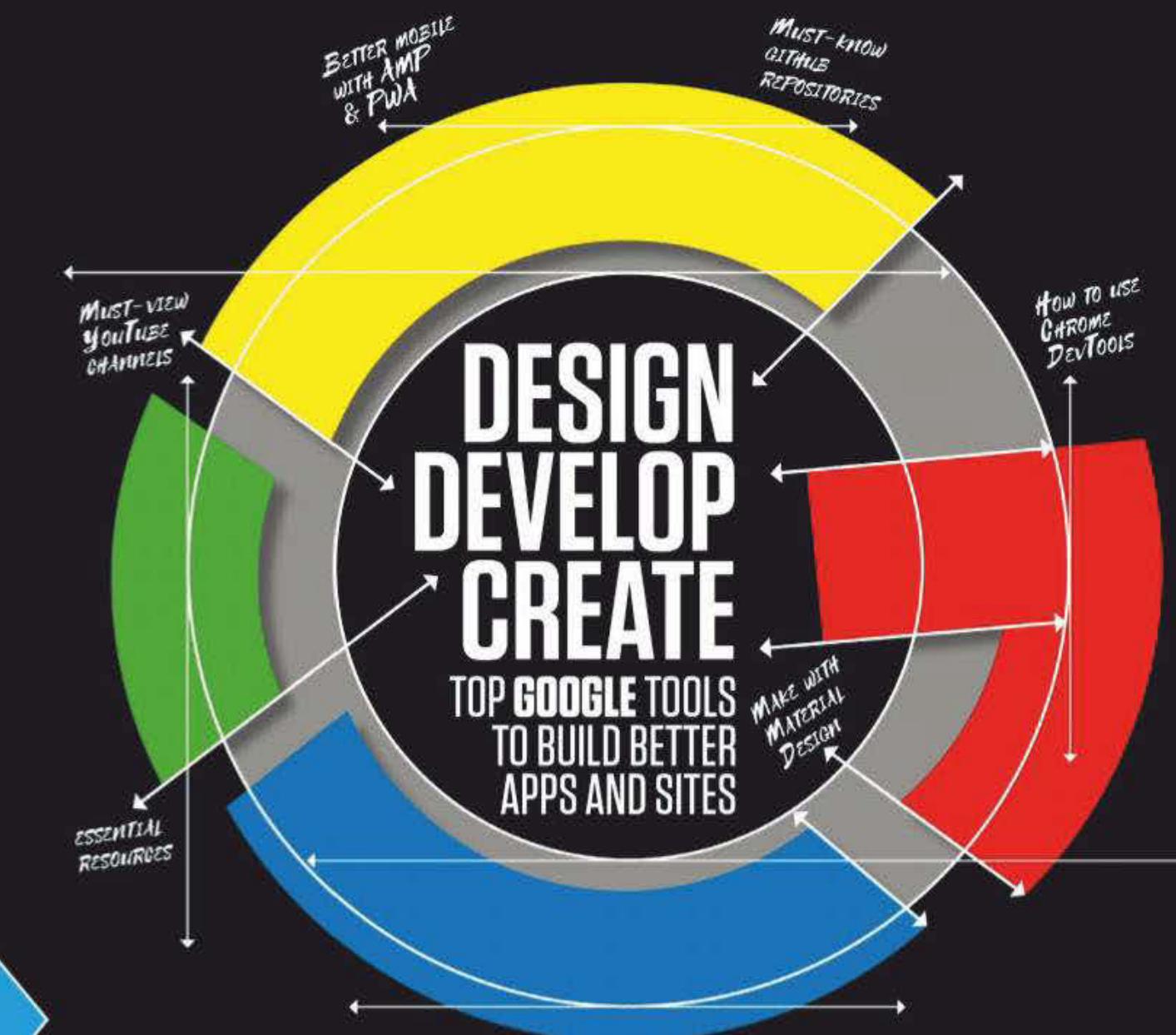
94 Course listings

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

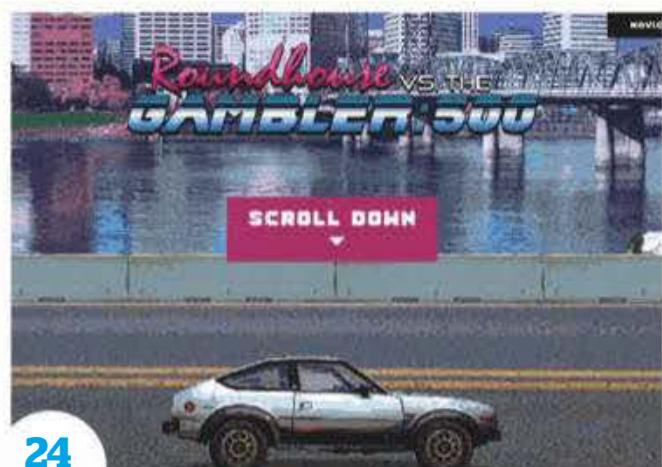
98 Next month

What's in the next issue of Web Designer

Cover focus



46



TUTORIALS

56 Code preview hover effects

Learn how to recreate the impressive hover effect seen on epoch476.com

58 Add glitch with CSS

Create advanced animation effects on text and images using the king of styling

64 Build a simple image changer

Use the power of CSS and vanilla JavaScript to create a simple dynamic image display

68 Create popular UX patterns

Build popular patterns such as sliders and modals with the layerJS library

72 GSAP masterclass

Learn how to get the most out of Greensock, the industry standard for web-based animation

80 Multi-language Angular

Introduce multiple language support to your Angular applications

86 Make data dynamic

Employ the power of the Chart.js library to create dynamic charts and diagrams to make data more engaging



36

FILESILO

96 Get the latest must-have resources and videos

- 81 minutes of expert PHP video guides from Killersites
- 40 Instagram presets and Sky Replacer PS actions from Sparklestock



myfavourite magazines

Visit the **Web Designer** online shop at

myfavouritemagazines.co.uk

for the latest issue, back issues and specials

DIGITAL EDITION

Do you want to get your hands on a digital edition? Head to your preferred app store, download, install and purchase the issue of choice

Google Play – bit.ly/2wetvGp
iTunes – apple.co/2igtBYq



Subscribe today and save

<https://bit.ly/2sGwB3h>

Header

The tools, trends and news to inspire your web projects

Are you ready for global ecommerce?

Getting a product or service out there and ready to buy is critical to success. **Web Designer** looks at what you need to know

Selling online is more important than ever to any business who wants to be truly successful. Big business typically has big backing, but smaller companies and individuals may not be so lucky. So Ecommerce provides the perfect path to the marketplace. There are a host of providers on the market that allow those who want to start selling online a very low cost entry point. Shopify (www.shopify.co.uk) is a popular provider. For just \$29 a month (at the time of writing) users get an online store with unlimited products and 24/7 support. If you are just starting out and only have a few products to promote Big Cartel (www.bigcartel.com) may be a great starting place. Five products are free and 25 products are just \$9.99 a month. There are plenty of other solutions for all sizes of business. It all very much depends on what you need.

A recent report (Growth 2000 UK

Report 2018) from Internet Retailing - <https://internetretailing.net> - took a closer look at the companies and businesses without the big resources and gave some sage advice. Stand out from the crowd was one excellent suggestion. The report revealed that unsurprisingly more and more brands are finally joining the ecommerce revolution. It used the example of a lesser known-brand Ann's Cottage, which started as a shop within a petrol station at Polzeath, Cornwall back in 1978. The business now has ten shops and its website features the latest updates on the surf at many Cornish beaches - including live web cams, as a point of interest that can entice not always the obvious customer.

As we all know mobile is the preferred platform for those browsing the web and creating an effective will make a difference to sales. The obvious is the user experience, performance and additional services. Ensure that your

pages are as light as possible, the faster a page loads the more likely the user will stay. There is the option to create a dedicated app this much depends on budget. These are simple tech based elements, so also considered not tech services such as free delivery and ideally same-day delivery if at all possible. Customers do not like to wait. What about payment services? Not everyone uses the same payment service and if you want to go international you have to think about what they use in other areas of the world. According to the report global mCommerce is bigger than desktop conversion (+50%) in Asia and approximately 30% higher than desktop in Europe. A point that reinforces how important mobile is and how other markets operate.

This article only covers the tip of the state of Internet retailing. To get a closer look at the report make sure to visit <http://bit.ly/2OdIIMW>.

STAT ATTACK BROWSER SHARE

Google is growing but what about the others

Chrome



A rise of almost exactly 5% over 12 months

Safari



A small rise of just 0.6% since Sept 2017

UC Browser



A decrease of just over 2% since last year

Firefox



A slow decline of almost 1% in 12 months

Opera



Decreased usage by less than 0.6%

Source: <http://gs.statcounter.com> (correct as of September 2018)

CB **CREATIVE BLOQ**
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/2qLxVI4>

DON'T MISS OUT...

REACT SERIES

5 expert tutorials
to help you master
ReactJS.
Starts in issue 282
On sale 11.12.18

SUBSCRIBE
TODAY!

<https://bit.ly/2qLxVI4>

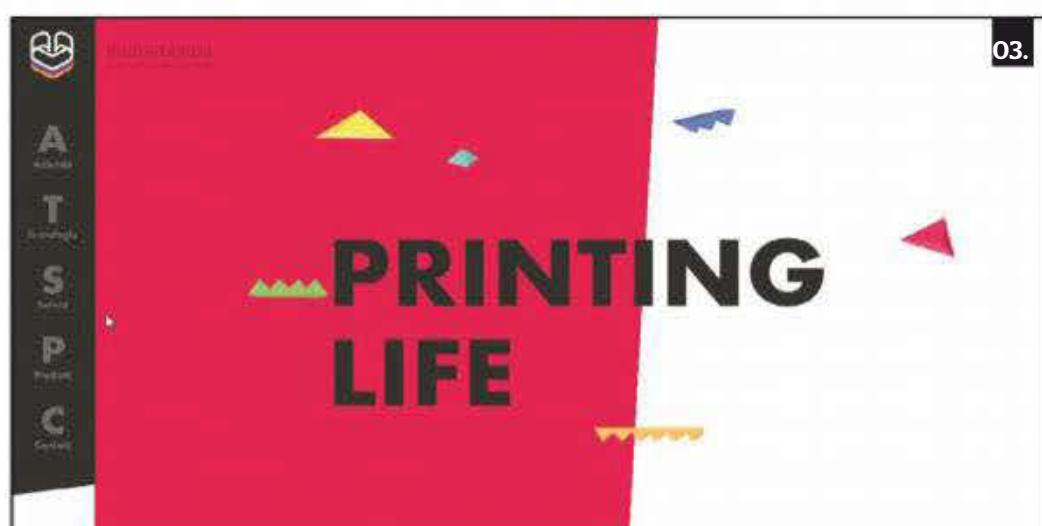
Sites of the month



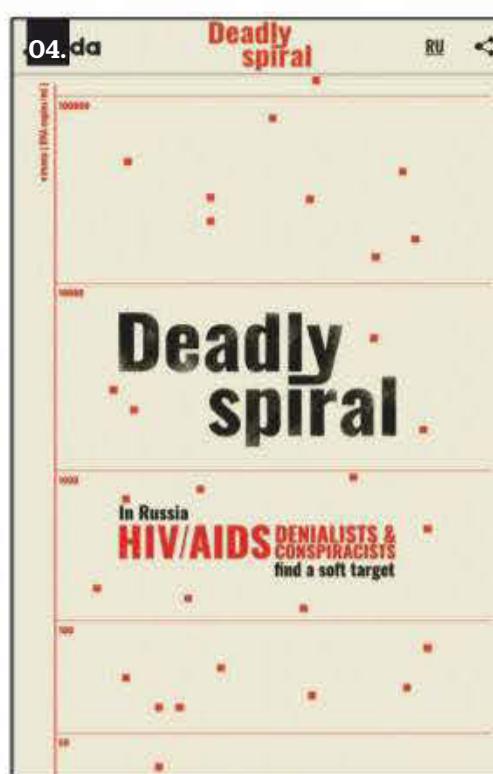
01.



02.



03.



04.da

Deadly
spiral

RU

Deadly
spiral

In Russia
HIV/AIDS DENIALISTS & CONSPIRACISTS
find a soft target

01. Mistretta Coiffure

mistretta.ch/en

Liquid effects and transitions enchant the viewer into staying on the site.

02. EASY TIGER FILMS

easytigerfilms.fr/en

Simple but effective video transitions and animated text effects.

03. Multistampa

multistampa.com

Discover colourful floating 3D objects that follow the cursor.

04. Coda - Deadly spiral

bit.ly/2JiGOF8

Neat animations, impressive illustrations and video enhance the story.

Graphics

Acapulco

bit.ly/2yzlkTV

Photos, Photoshop or both. Beautiful imagery from the talented Paloma Rincón.



Colour picker

Babo Hey

<https://bit.ly/2Rgw3e6>



Typesetter

Quiche Sans

bit.ly/2CFu8f0

A high contrast, sans serif typeface with monoline stroke endings.

ABCabc
0123

WordPress

GenesisExpo

genesisexpo.webgeniuslab.net

Running a conference or festival? This multi-design theme offers style.



Variable fonts: Times are changing

Variables fonts promise an online typographical experience on par with print



Tamás Hajas
Senior Engineer, Mirum
mirumagency.com

“ Every typeface is different because it depends on what features the font designer decided to add ”

While we have had web font support for around a decade, web designers and developers were unable to apply all but the most basic typographic principles to online content. But the times are changing and with them the emergence of the variable font format. What are variable fonts? “A variable font is a single font that acts as many” says **VariableFonts.io** editor John Hudson. But what does it mean in practice? As you know, a font family can have a lot of variations; a font can be thin or bold, narrow or wide, and so on. Regular web fonts store all these variations in separate files. In variable fonts all the variations are stored in a single file. Each character has only one outline and the connector points of this outline have instructions on how they should move to create another style. As a result – most of the time – a variable font file will have a smaller file size than the separate font files together. It will also have more possibilities than a regular web font.

So how should you use variable fonts? Variable fonts are supported by the current versions of all the major desktop browsers, just like the default browsers on mobile platforms. This means you can start using them right now, but it is recommended that you add them as an enhancement. You can add a variable font to your site using the same ol’ ‘@font-face’ declaration you know already:

```
@font-face {  
    font-family: 'My Variable Font';  
    src: url('fonts/my-variable-font.woff2') format('woff2');}  
//Then you can use it in your CSS ruleset:  
.heading {  
    font-family: 'My Variable Font', 'Regular Webfont', system-font;  
    font-variation-settings: 'wght' 812, 'wdth' 72; }
```

There is a strange thing in the above code: ‘font-variation-settings’. This is a so-called low-level property of CSS that’s being used to define variable font settings for the time being. Its value should be built from one or more pairs of a four-character tag and number.

Every typeface is different because it depends on the features the font designer has decided to add. A font can have any number of custom axes (this is the official name of the changeable properties of the font) and there are five registered (or common) axes. We saw ‘wght’ and ‘wdth’ already, which are the weight and the width of the font. There are also ‘ital’, ‘slnt’ and ‘opsz’, which mean italic, slant and optical sizing. You may have noticed already that the values of the weight and the width properties are not the usual round numbers. This is because by using variable fonts we can define any whole number in the property range as a value and they will result in a different font display.

The value of the italic setting can only be ‘0’ or ‘1’ because the italic version of a font is so different from the normal version that an in-between format would not be anything but ugly. (Most of the time the italic version of a variable font will be a separate font file, just as it is with a regular web font). We may have a slant setting, however, which can be used to set an oblique version of the font. It is worth mentioning that this slanting – just like the other settings – won’t mean that the font will be distorted, since it works in the way it was set by the font designer.

Last, but not least, there is optical sizing. This is the possibility that the font will have more or less detail depending on its size – for example, a smaller font will have thicker stems, while a bigger font will have thinner ones. This was the privilege of print design – until now.

Note: You have to use a lowercase tag to set a registered axis and an uppercase tag to set a custom axis. Using the latter, you can set anything provided by the font designer – for example, x-height, roundness, grade, and so on. Oh, and did I mention that any of the registered or the custom axes can be animated? (Beat that, print!). Currently, we have to use the ‘font-variation-settings’ property to style variable fonts, but when the CSS4 Font module is implemented by browsers we will have high-level properties for the registered axes. Some of them may be familiar – like ‘font-style’, ‘font-weight’ or ‘font-stretch’ – but there will be a new one, too, ‘font-optical-sizing’, which will be automatic by default.

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



Angular 7

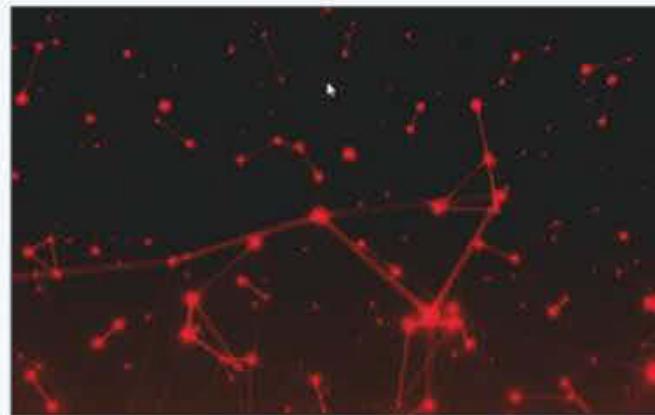
angular.io

The latest incarnation of the popular framework was recently released. So what does it have to offer? The update spans the entire platform including the core

framework, Angular Material and the CLI. It offers CLI prompts, virtual scrolling, drag and drop and more. Check out the Angular Update Guide (update.angular.io).

TOP 5 CODEPENS

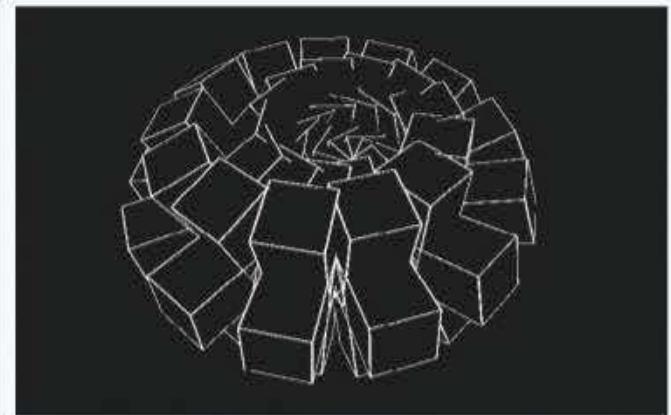
Be inspired by this collection of smart and interesting codebases



my universe

codepen.io/kakaxi0618/pen/JmLpYj

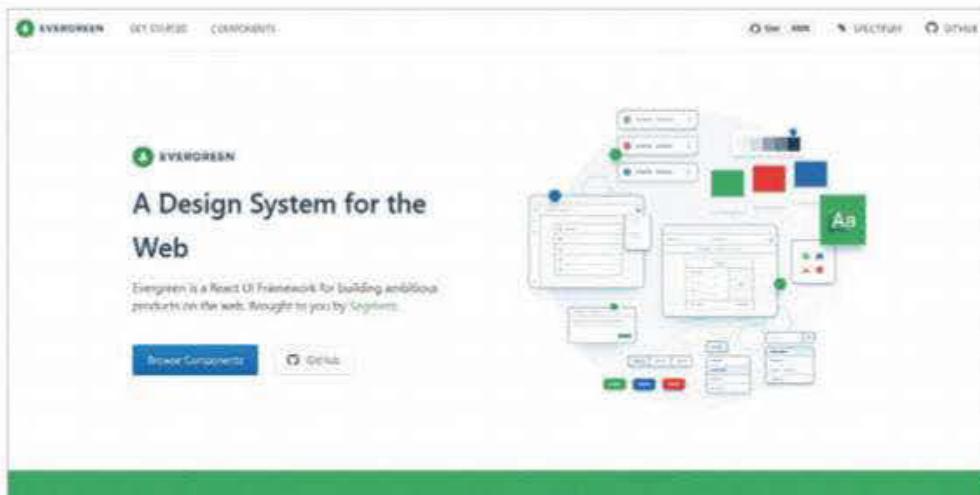
Changing colours and glowing connected stars mesmerise the user. Be sure to check the code to see how it was built.



Untitled

codepen.io/Alca/pen/wPNaoQ

Repeating, hypnotising, expanding animated 3D shapes work with only a few lines of CSS and JavaScript code.



Evergreen

evergreen.segment.com

Evergreen is a React UI framework that contains a set of components that work out of the box. They are built on top of a React UI Primitive for endless composability.

billboard.js

naver.github.io/billboard.js

Want to create a chart quickly and easily? Then this chart generator, which is compatible with D3, could just be what you are looking for. Check it out and get creating.



Eagle.js

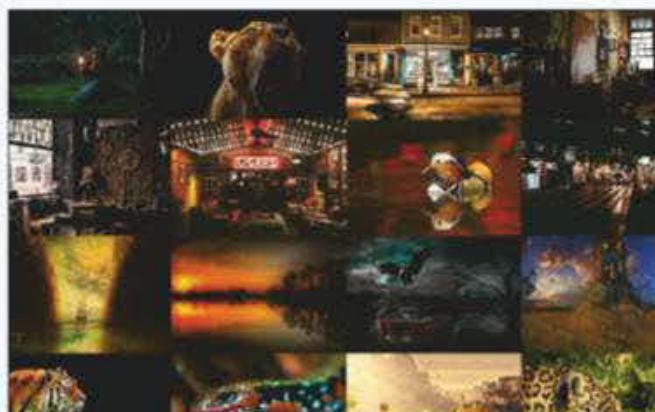
zulko.github.io/eaglejs-demo

Need a web-based slideshow? Then this framework for Vue.js makes it easy to reuse components and offers animations, themes and interactive widgets.

Inferno

infernojs.org

Inferno has been around for a while and is a fast, React-like library for building high-performance user interfaces. Check out the latest version.



Infinite Grid

codepen.io/radixzz/pen/eRJKXy

An infinite grid of images that constantly repeats as the user drags them left, right and up and down.



We're Getting Close

codepen.io/lannymcnie/pen/YJXOVO

An animated canvas sees constantly refreshing bubbles eventually reveal hidden text. Click for colour changes.



Helix CSS Loader

codepen.io/jerrylow/pen/OBBWez

Very much what it says on the tin. An animated helix loader that continually rotates and made with 99 percent CSS.

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



Colorable

colorable.jxnblk.com

When it comes to accessibility and readability, contrast is key. White text on a yellow background is a fine example of poor contrast – hard to read. Colorable is a simple

tools that uses a couple of sliders to change the background colour and font colour whilst giving the overall effect a rating. Get it right and get a triple-A. Get it wrong and get a fail.

TOP 5 WP THEMES

Need a new page design fast? Then check out this collection of themes



Larch

bit.ly/2OS6ALa

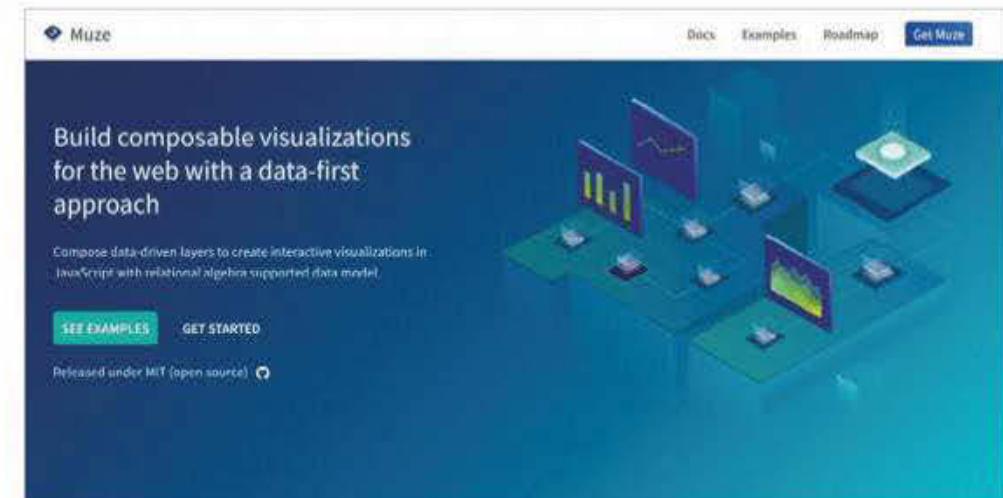
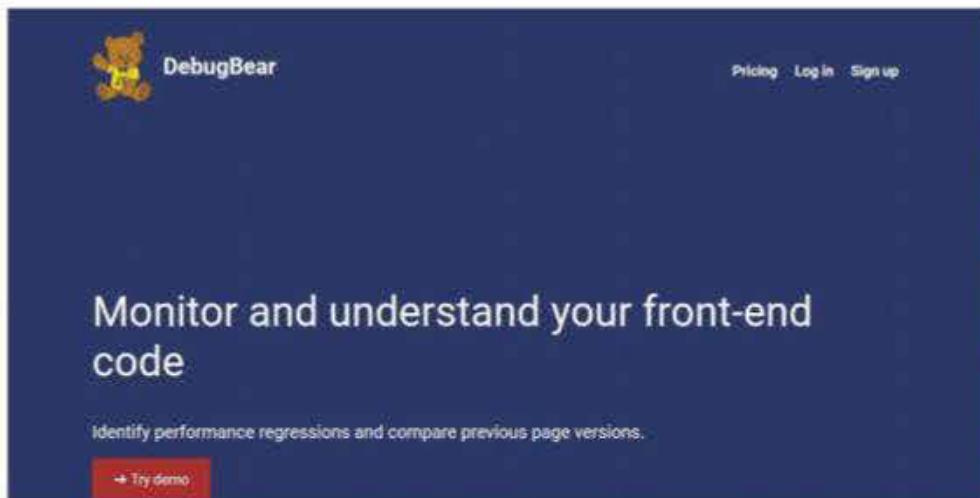
A bold, creative theme with a host of home page designs to enable users to choose their preferred landing page.



Venezia

bit.ly/2SpU7MT

A simple, straightforward single-page theme that offers some neat animations and a twist on image presentation.



DebugBear

debugbear.com

Debug Bear's message is 'Monitor and understand your frontend code'. It analyses and compares against previous versions and monitors performance.

Muze

charts.com/muze

Data is not great to look at. Make it more attractive, visually appealing and engaging with interactive visualisations in JavaScript.



Angular Console

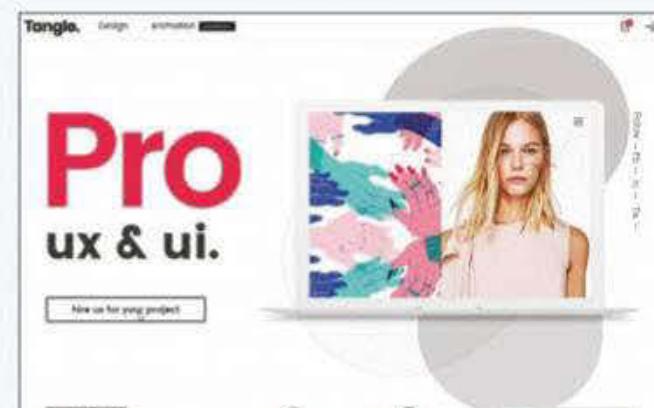
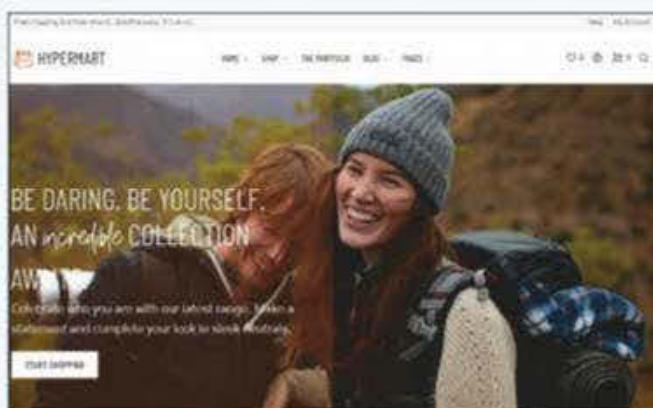
angularconsole.com

Angular Console 'is, first and foremost, a more approachable way to work with what the Angular CLI already provides'. Use Angular? Then give it a go.

Fancy Border Radius

9elements.github.io/fancy-border-radius

Discover what you can do with 'border-radius'. Create a host of unique shapes easily and quickly. Simply copy the code when done.



Hypermart

bit.ly/2PZuM1b

An eCommerce theme, which has been optimised to be fast and efficient. Includes a host of pre-made demos.

Tangle

bit.ly/2PpJUS4

A clean, contemporary theme with six different home page designs that can be combined to create more unique designs.

Faulkner

bit.ly/2yC8lg8

A responsive theme aimed at SaaS, startups and small tech agencies. Pick from over 200+ content blocks to create a design.

Romain Granai

romaingranai.be

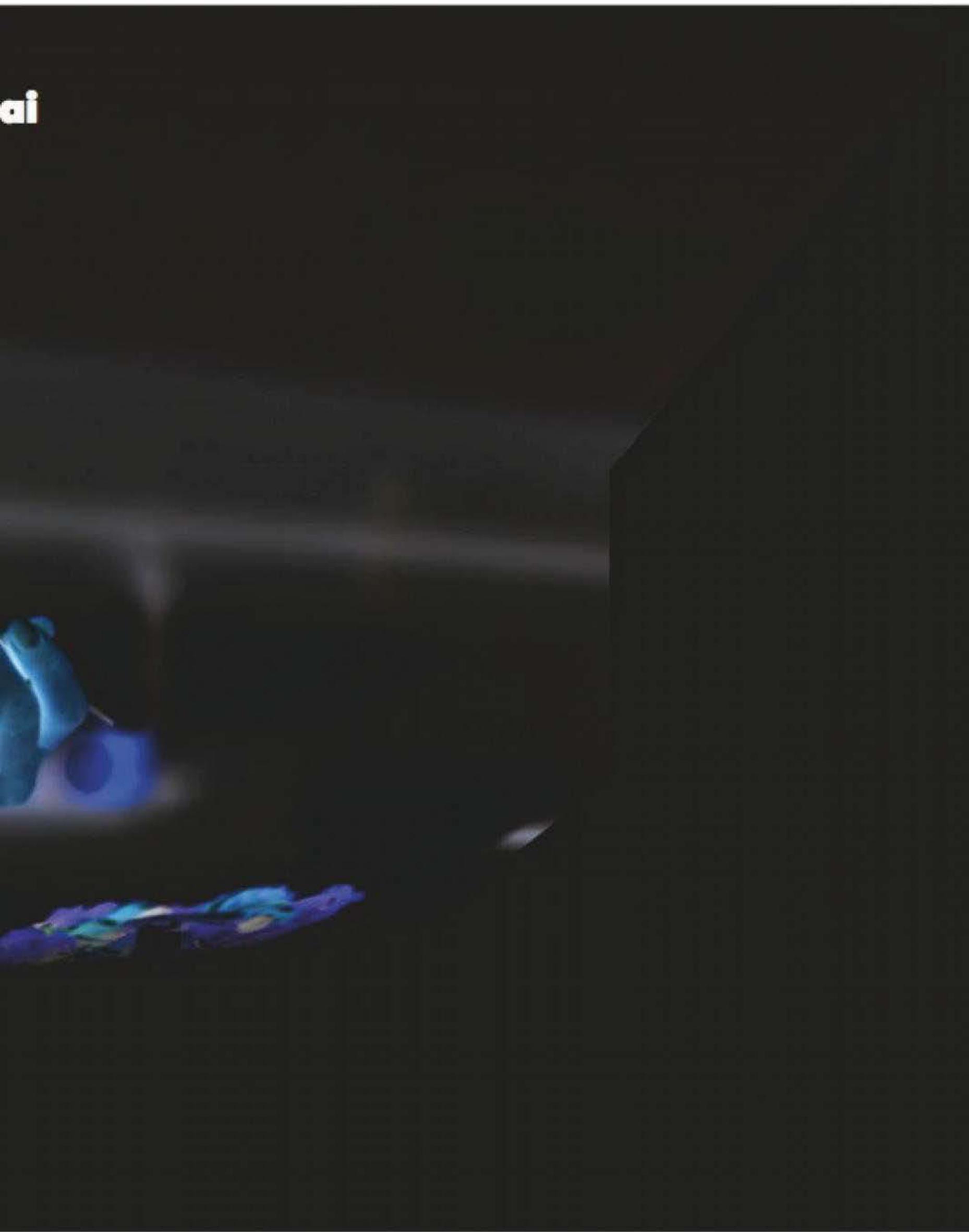
Designer:

Romain Granai (In-house) romaingranai.be

Romain Granai



“Granai’s rather cryptic designer calling card with light and dark modes, twisting and distorting as the visitor scrolls”



Colours



Tools

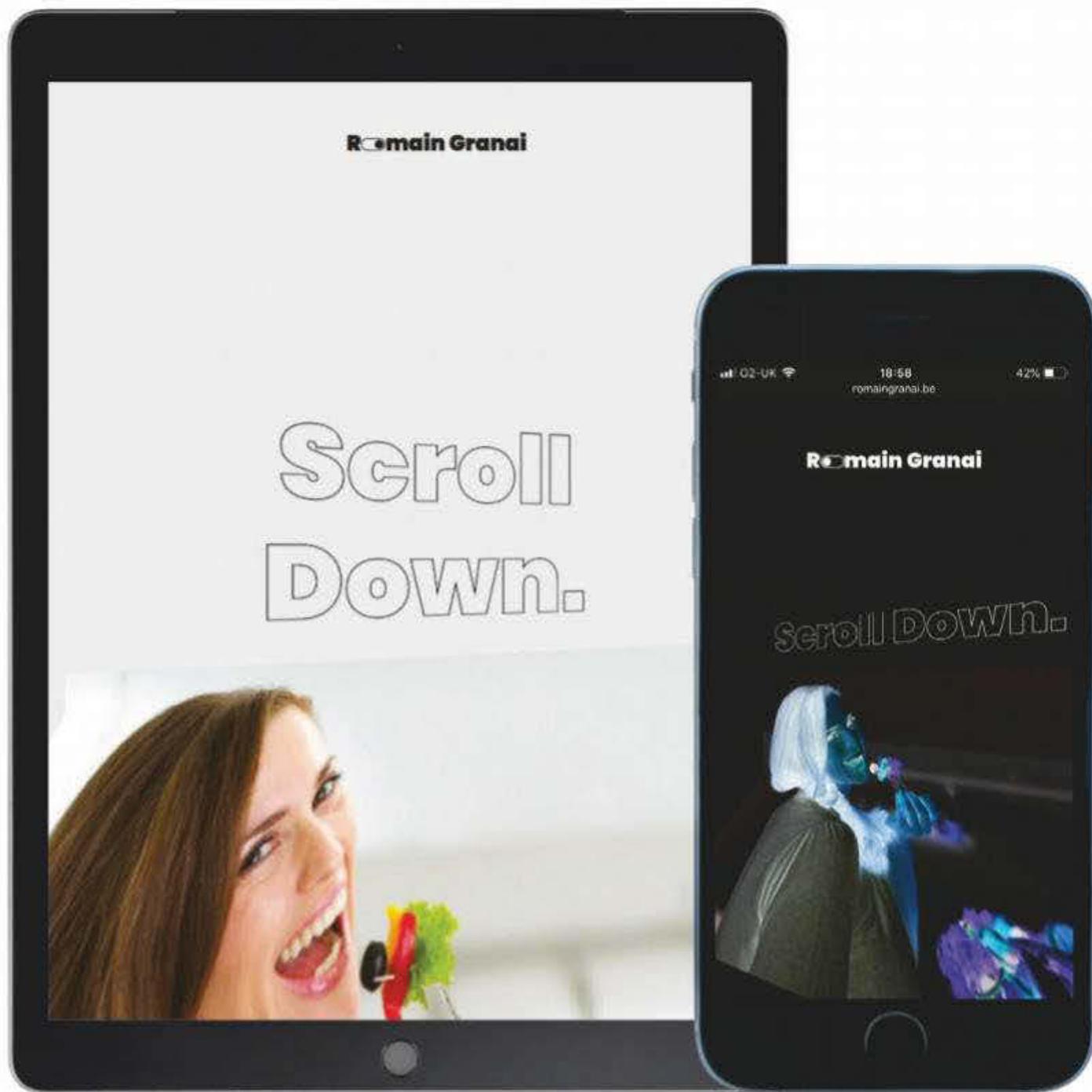
jQuery, GSAP,
PHP, WordPress

Fonts

**abcABC
1234567890**

Poppins font by Jonny Pinhorn and Ninad Kale is available via Typekit and Google Fonts, is found here to style all the typographic elements.

Scroll Down.



Above

Opening with a simple bold instruction, visitors are immediately aware that scrolling down is required to trigger the effect

Far left

The name logo at the site's top features a tiny slider control for toggling between dark and light display modes

Left

On touchscreen displays, the scrolling effect works equally as effectively, promising more to enjoy when experienced on mobiles

Create a colour switcher for better accessibility

Enable visitors to chose a colour scheme that best suits their preference or accessibility requirements

1. Initiate document

The starting point is to initiate the HTML to describing the document structure. This step defines the document container, along with its child <head> section. The <head> section is used to reference the external CSS and JavaScript resources that are defined in later steps. Take note of how the body section is defined in Step 2.

```
<!DOCTYPE html>
<html>
<head>
<title>Colour Switch</title>
<link rel="stylesheet" type="text/css"/>
<script src="code.js"></script>
<script src="code.js"></script>
</head>
*** STEP 2 HERE
</html>
```

2. Body section

The body section is responsible for storing the page content elements. The colour palette technique also relies on the 'data-palette' attribute applied to the body section. This step sets the default value of the 'data-palette' attribute to '0', enabling CSS to present a default colour scheme in later steps.

```
<body data-palette="0">
*** STEP 2 HERE
</body>
```

3. Body content

The page content consists of a heading and a button. Additional content can be added without the need to reference attributes or styles to define their colour. Although no special attributes are required for the heading, a unique ID is applied to the button in order to enable JavaScript to apply functionality.

```
<h1>Colour Me!</h1>
<button id="toggle">Click me</button>
```

4. Initiate JavaScript events

Create a new file called 'code.js'. Page content can't be accessed via JavaScript until the page has completed loading. This problem is solved by defining functionality inside a function that is triggered when the browser window reports its load is complete. This function searches for the element using the 'toggle' ID, then applies a function to trigger for when it is clicked.

```
window.addEventListener("load", function(){
var button = document.querySelector
("#toggle");
button.addEventListener("click", function(){
```

```
});});
```

*** STEP 5 HERE

5. Click processing

The code triggered from Step 4 needs to calculate the colour palette to use. This example keeps the feature simple by toggling the 'data-palette' attribute applied to the body section between '0' and '1'. You could choose other logic to increase the colour counter or to enable the user to specify the palette name/number from an input element.

```
var palette = "0";
if(document.body.getAttribute("data-palette")
== "0"){
    palette = "1";
}
```

6. Palette update

After the palette name/number has been calculated from Step 5, the 'data-palette' attribute assigned to the document's 'body' section needs to be updated with the

new value. The new value can now be used from CSS to define specific presentation rules for each palette name/number generated via the JavaScript code.

```
document.body.setAttribute("data-
palette", palette);
```

7. CSS colours

Create a new file called 'styles.css'. The remaining step is to define the colour combinations for each value that the 'data-palette' attribute can be set to. At its simplest, background and content colours are defined. More advanced rules can be applied by setting unique colours for individual child elements within the 'data-palette' container.

```
[data-palette="0"]{
    background: #fff;
    color: #000;
}
[data-palette="1"]{
    background: #000;
    color: #fff;
}
```

Romain Granai

Scroll

Down.

Lo Pesce

lopesce.com

Designer:
Gummy Industries gummyindustries.com



javascript:void(null)

“Bold and colourful Italian brochure site for promoting a frozen fish brand, fusing hi-res product shots with floating illustrations”



Colours



Tools

jQuery, PHP, SVG

Fonts

**abcABC
1234567890**

The Druk Text Wide Medium font by Bertron Hasebe is used to style the site's header text and button labels.

**abcABC
1234567890**

The Young Serif font by Bastien Sozeau can be found at the foot of the page, styling address and company contact.



Above

The site's design really extends out from the product's packaging, using a complementary blue, yellow, green and red colour scheme

Middle

Page elements appear to float above the bright backdrop, layered to create a 3D effect during scroll within all browsers

Bottom

Click each of the Scopri badge buttons to flip the packet and reveal Italian cooking instructions and ingredients

Code a scrolling foreground background overlay effect

This is a feature that lets content scroll under and over a statically positioned background element

1. HTML document

The first step is to define the HTML document. This consists of the HTML document container, which stores sections for the head and body. While the head section is used to reference external CSS and JavaScript resources, the body section is used in step 2 for the main page content.

```
<!DOCTYPE html>
<html>
<head>
<title>Background Overlay</title>
<link rel="stylesheet" type="text/css"
href="styles.css" />
<script src="code.js"></script>
</head>
<body>
```

2. Body content

The page body content consists of a collection of sections that represent the scrollable foreground and background content. Additionally, the last element in the body will be used as the statically positioned content. Positioning this element as the last item within the page body can help with SEO by keeping main/important content higher in the HTML markup.

```
<section>
<h2>Foreground</h2>
</section>
<section>
<h2>Background</h2>
</section>
<section>
<h2>Foreground</h2>
</section>
<section>
<h2>Background</h2>
</section>
<span>Something</span>
```

3. CSS fixed element

Create a file called 'styles.css'. This step defines the presentation for the element used as the fixed background. The HTML is designed to always have this element placed last in the body, hence the use of the last-child selector. Fixed positioning and z-index are mandatory requirements. Optional attributes include top co-ordinate positioning and font-size.

```
body > *:last-child{
    position: fixed;
    top: 25vh;
    font-size: 25vh;
```

```
z-index: 1;
```

4. Section default

Each section on the page shares a collection of attributes for presenting their size and position. For the sake of visibility, a red border is also applied so that you can see where each section starts and ends. The sections are designed to cover the full space of the page in order to guarantee ability to scroll.

```
section{
    position: absolute;
    display: block;
    width: 100%;
    min-height: 100vh;
    border: 1em solid red;
}
```

5. Section layers

Sections used for foreground and background content require different z-index placement in order to appear over or under the fixed background element. With the fixed element using a z-index of one, the background element is set to use a z-index of 0 while the foreground section uses a z-index of 2. The HTML design allows this to be easily applied using odd and even values with the nth-of-type selector.

```
section:nth-of-type(odd){
    z-index: 2;
}
section:nth-of-type(even){
    z-index: 0;
}
```

6. Presentation extras

For purposes of presenting the effect, different

background colours and margin positioning are applied to the content elements inside the foreground and background sections. This allows you to confirm that the content is being placed correctly for the desired overlay and underlay effect. You could consider how this can be customised to the requirements of your real project.

```
section:nth-of-type(odd) h2{
    background: red;
    margin-top: 50vh;
    font-size: 3em;
}
section:nth-of-type(even) h2{
    background: silver;
    margin-top: 75vh;
    font-size: 3em;
}
```

7. Section positioning

Create a new file called 'code.js'. Each pair of sections used for the foreground and background content require to share the same vertical positioning. This is an unnatural placement for just HTML and CSS, so intervention from JavaScript is required. This step searches for each pair of sections to calculate their shared vertical position.

```
window.addEventListener("load", function(){
    var sections = document.querySelectorAll("body > section");
    for(var i=0; i<sections.length;
    i=i+2){
        sections[i].style.top =
        ((i/2)*100)+"vh";
        sections[i+1].style.top =
        ((i/2)*100)+"vh";
    }
});
```



RH Vs. Gambler 500

gambler500.roundhouseagency.com



Designer:
Roundhouse roundhouseagency.com



“Drive a Gambler 500 rally car through low-res pixelated terrain to discover photos, video clips and even an interactive racing game”

Colours



#CD8772



#7795A3



#BE1A71



#5C6EDF

Tools

GSAP, jQuery,
ScrollMagic, HTML5

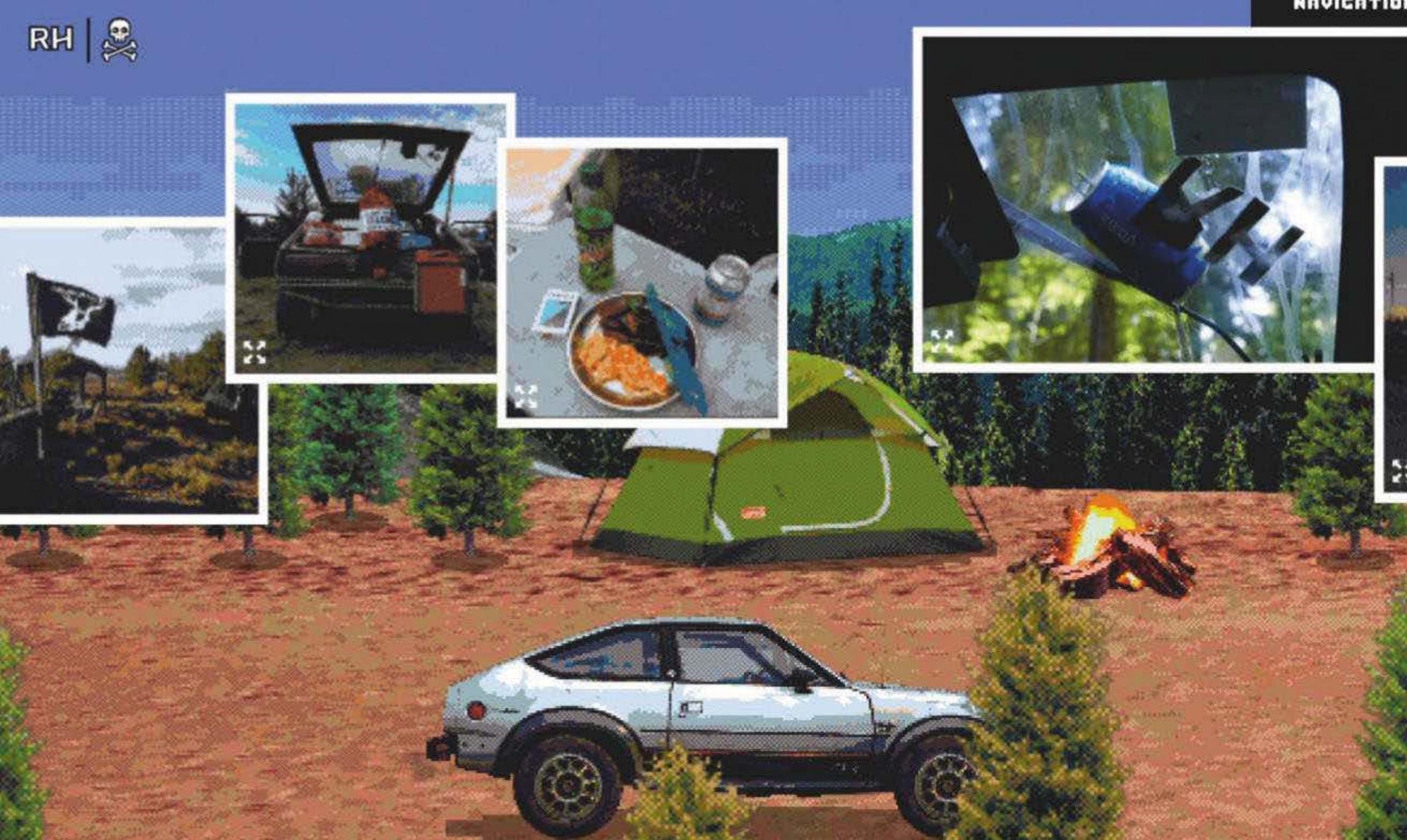
Fonts

abcABC
1234567890

The distinctly old school VCR OSD MONO font by Riciery Leal provides retro blocky styling in keeping with the pixelated aesthetic.

ABCABC
1234567890

Sabo Filled by Philippe Moesch keeps that bitmapped typographic vibe going with a blockier, heavier and bolder font.

**Above**

Visitors scroll vertically as normal, and the site's Gambler 500 car 'drives' along horizontally to reveal the various sections

Middle

Low resolution and deliberately pixelated photo thumbnails come into focus and enlarge when selected for viewing

Bottom

Framed in an old CRT television, an interactive driving game is available to play using arrow keys or by tilting a phone's gyroscopic sensor

Introduce an expanding lightbox for inline images

Allow inline image content to be expanded. Ideal for horizontally designed content

1. Document framework

The first step is to define the HTML page's framework description. This consists of the HTML document container used to store the head and body sections. While the head section is used to load the external HTML and JavaScript resources, the body section will be used in step 2 for storing the page content elements.

```
<!DOCTYPE html>
<html>
<head>
<title>Lightbox</title>
<link rel="stylesheet" type="text/css" href="styles.css" />
<script src="code.js"></script>
</head> <body>
```

2. Body content

The page's content elements consist of images enclosed within a span element. The span elements have a 'lightbox' class that will be used by JavaScript and CSS to control the presentation of the inner image. Each of the images should reference its own unique image using the element's 'src' attribute.

```
<span class="lightbox"></span>
<span class="lightbox"></span>
```

3. JavaScript initiation

Create a new file called 'code.js'. The JavaScript relies on the full webpage being available, hence the use of the 'load' event listener applied to the browser window. Upon completion of loading the webpage, a search will be performed for all elements using the 'lightbox' class. The 'for' loop is used to apply step 4 to reference each element found.

```
window.addEventListener("load", function(){
    var nodes = document.querySelectorAll(".lightbox");
    for(var i=0; i<nodes.length; i++){
        *** STEP 4 HERE
    }
});
```

4. Lightbox toggle

This step applied logic to toggle the 'open' class. Each element found in step 3 has an event listener that executes a function when the item has been clicked on. This function checks to see if the element has the 'open' class applied to it. If so, the 'open' class is removed, otherwise the 'open' class is added.

```
nodes[i].addEventListener("click", function(){
    if(this.classList.contains("open")){
        this.classList.remove("open")
    }else{
        this.classList.add("open")
    }
});
```

```
position: absolute;
color: #fff;
text-shadow: 2px 2px 4px #000;
font-size: 2em;
top: 0;
right: 0;
}
```

5. CSS lightbox

Create a new file called 'styles.css'. This first step of the CSS initiates the presentation for any elements using the 'lightbox' class. Vertical-align is set to 'top' so that size adjustments drop below neighbouring content, while a transition will animate any changes. Relative positioning is also used to help with step 6.

```
.lightbox{
    display: inline-block;
    position: relative;
    transition: all 1s;
    height: 6em;
    vertical-align: top;
}
```

6. Icon definition

The virtual 'after' element is used to present a status icon for the lightbox item. Using the content attribute, an arrow character is presented over the image. Relative positioning defined in step 5 allows the icon to be positioned in relation to the lightbox container.

```
.lightbox::after{
    content: "\2921";
    display: block;
```

7. Inner image

All changes are applied to the container that the lightbox class is applied to. The inside image needs to reflect this, hence its height being set to match 100% of its parent. When the parent's height changes, the inner image will also adapt to change its height to reflect this.

```
.lightbox > img{
    height: 100%;
```

8. Open lightbox

The 'open' class applied by JavaScript needs to trigger a change in the lightbox's height. With the transition set in step 5, the lightbox's height change will animate over a one-second duration. Additionally, this step sets the content of the 'after' virtual element to present a cross character as the status icon.

```
lightbox.open {
    height: 80vh;
}
```

```
.lightbox.open::after{
    content: "\00d7";
```



The perfect gift from just £24.99*

PLUS

**FREE
KITSOUND
RACE
WIRELESS
EARPHONES**

**WORTH
£14.99**

Order hotline 0344 848 2852

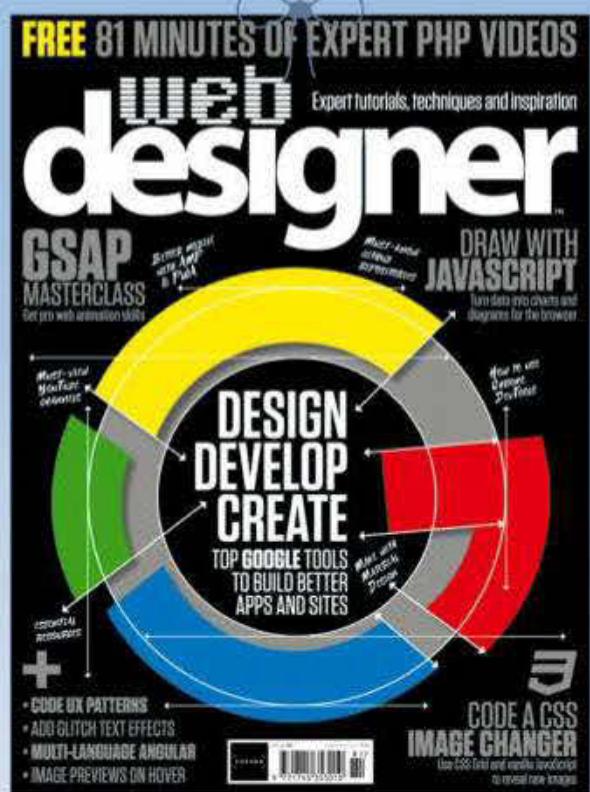
Please quote xmas182 when ordering by phone

Lines open Monday-Friday 8am-7pm and Saturday 10am to 2pm (GMT)



myfavouritemagazines
Official Magazine Subscription Store

**SUBSCRIBE
FROM JUST £24.99***
every 6 months*



Reasons to subscribe

- Saving up to 41% against the shop price
- Never miss an issue
- Free delivery to your door
- The gift that keeps delivering



Upgrade to include digital for only £5

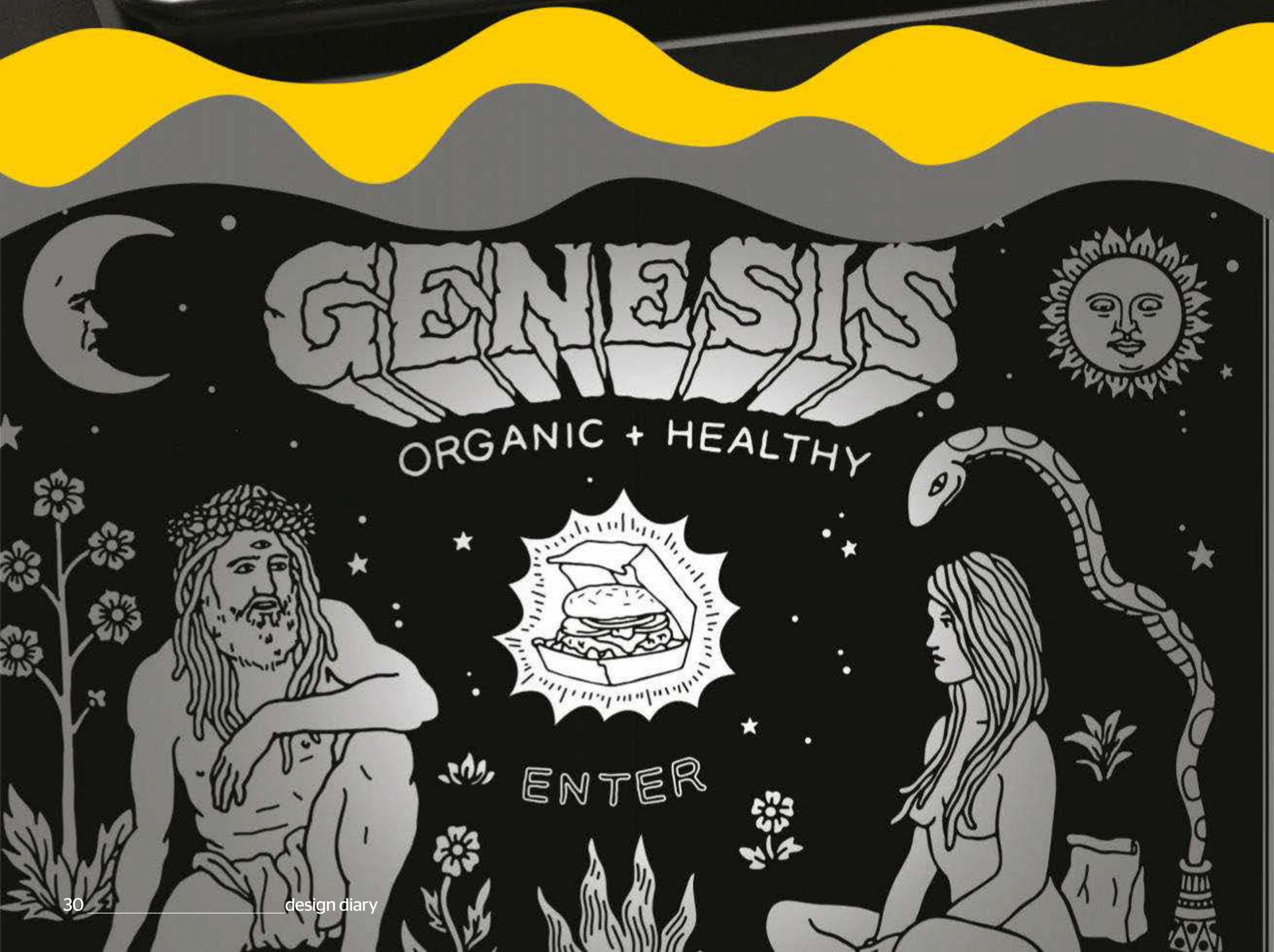
- Increase your saving by 63%

**SEE THE
FULL RANGE AND
ORDER ONLINE AT**

www.myfavouritemagazines.co.uk/pin/xmas182

Call UK 0344 848 2852 or overseas +44(0) 344 848 2852 & quote xmas182

***Terms and conditions:** Savings calculated against the full RRP (single issue price x frequency). You can write to us or call us to cancel your subscription within 14 days of purchase. Gift is only available for new UK subscribers. Gift is subject to availability. Please allow up to 60 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. Your subscription is for the minimum term specified and will expire at the end of the current term. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. All gift subscriptions will start with the first issue in January 2019. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available on request. For full term and conditions please visit: bit.ly/magtandc. Offer ends 31st December 2018.



A NEW BEGINNING

WHEN ORGANIC FOOD RESTAURANT GENESIS SOUGHT IMAGINATIVE CREATIVE GUIDANCE FOR ITS NEW ONLINE PRESENCE, COULD AGENCY HERDL VAULT EVERY OBSTACLE AND DELIVER THE GOLD?

Given that so many of Design Diary's featured projects describe projects born out of existing partnerships, you might think 'better the devil you know' is the industry mantra. While it's true that successful creative relationships tend to encourage loyalty, they have to start somewhere. While digital agencies rarely 'face off' in competitive pitches to land new work, it's true that prior achievements can be pivotal. Our featured players this month are poster children for illustrating this, when client Genesis found agency Herdl through the web design awards site, Awwwards. Noticing an interactive project Herdl produced in 2017 for a coffee company called Two Chimps, the Shoreditch-based fast-food vegan restaurant saw something that chimed with the unique approach they craved. Indeed Herdl, who formed in 2013, started with the goal of assembling a team of digital experts capable of tackling any challenge. Since then two have become six, staffing multi-disciplinary services across web design and digital marketing, underpinned by technology, business and creative application expertise. Genesis meanwhile, no strangers to recognition

themselves, serve a 100 per cent plant-based menu with dishes inspired from across the globe, catering for both vegans and meat eaters. Its fully organic and GMO-free menu, makes Genesis one of the only UK restaurants to hold the Soil Association's 'Organic Served Here' Award. "From the very beginning, we knew we weren't going to design a traditional restaurant website, it was the culture and concept behind Genesis that we wanted to bring forward," Designer Sam Day begins. "Our aim was to create intrigue and anticipation, so rather than selling a product or service we wanted to showcase the Genesis ideology."

"WE KNEW WE WEREN'T GOING TO DESIGN A TRADITIONAL RESTAURANT WEBSITE"

Genesis
eatgenesis.com

by
Herd़l
herdl.com
[@herdl](https://twitter.com/herdl)

PROJECT DURATION
3 months

PEOPLE INVOLVED

Sam Day
Designer

James Hobson
Developer

Ged Day
Managing Director

Gareth Morgans
Digital Director

John Arundel
Dev Ops

TECHNICAL CHOICES

What were the tools and tech that powered the project

For the Genesis project, the development team decided to stick with WordPress as its favoured CMS. Largely due to the admin dashboard being intuitive for ongoing site management, the platform is also open source and therefore doesn't tie the client's proprietary system that's licensed and maintained by Herdl. This is then augmented by their own 'tweaked' version of the software, a series of custom plugins, plus the agency's own workflow principles, as Designer Sam Day confirms: "Our development workflow utilises Gulp for enhanced, automated build tasks; Composer for controlling dependencies, including the use of external WordPress plugins; and npm for package management. We rigidly follow DRY and BEM principles in our code and structure theme files accordingly. Our process ensures we create code that's high-performance and easily maintainable."

That's not to say, though, that the Genesis guys had no say at all over the development phases of the project. In fact, when it came to making certain choices over the technologies used on certain sections, the client's own existing stipulations led to work that may continue going forwards.

"The recruitment section of the website uses a system called Recsite for which we supplied the templates and styling. Recsite had been selected by Genesis prior to our involvement and we worked with the Recsite team to create a consistent look and feel across the recruitment portal. In the future, we hope to incorporate an integration with Recsite to create a seamless experience between the two parts."

ASPIRATIONS ALIGNED

Commenting that they had never seen a site quite like Two Chimps, the Genesis guys knew they wanted Herdl to produce something similarly innovative for them. "Genesis shared their initial ideas and brand document with us and we instantly sensed this could be a really exciting project. We picked out certain illustration styles and design routes that we could confidently translate to a digital format and developed a creative direction that would utilise these distinctive elements." A few emails and Skype chats later and both were on board, eager to find out how these initial ideas could be refined. Very quickly a connection between both organisations emerged, with aspirations for the project in tune enough morally and creatively to let Genesis surrender complete control over the design. Tasking Herdl with telling their story most effectively from a digital perspective, the client was simultaneously conversing with artists and designers to

overall creative direction, specific aesthetics and frontend functionality. "We prepared a digital mood board for the design meeting and broke down various examples into two specific sections, UI and UX. The meeting itself was full of energy and we encourage clients to give as much input as possible, drawing on their knowledge of their target audience and wider market. In our mood board for Genesis, the user interface section centred around selected visuals and inspiration that would help us develop the overall look and feel of the website. We included GIFs and mockups to demonstrate effects and animations as well as other materials such as gig posters, illustrations, graffiti art, album covers and tattoos to pick out design cues that we felt aligned with the client's brand values." The user experience talks then discussed specific elements, including taking bookings, careers, opening times, about the restaurant, menus and data collection. Here the team were able to harness assets from previous

"THE USER INTERFACE SECTION CENTRED AROUND SELECTED VISUALS AND INSPIRATION"

produce branding and packaging elements. "So there was already a stimulating atmosphere around the project and various assets beginning to take shape. We had access to commissioned artwork from some very talented illustrators and graphic designers, while we needed to develop a consistent look and feel, it was important to establish our own interpretation that would deliver a unique digital experience." This trust extended to leading the project management with Herdl adhering to the same tried-and-tested process they apply to all projects. Tasks would be broken into a five-phase roadmap spanning research and planning, design, development, testing and aftercare, with a crucial design meeting scheduled quickly after project kick-off.

SETTING A DIRECTION

In this meeting, the ideas and references that will be used throughout the project are brought together and presented to the client. While known as a 'design' meeting, a range of topics get aired here with the aim of nailing down the

project work to quickly develop some early working examples, while sourcing other areas of design inspiration from everyday life as opposed to digital. "Digital marketing is a core service at Herdl, so our aim is never to just make a pretty website, it always has a job to do. With everything understood we were able to confirm a design direction and the functionality that would be included. It definitely helped that the guys at Genesis share a similar taste in design to ourselves and trusted us to lead the design process."

PRIORITISING CONTENT

At this point, Herdl had a much stronger idea of the approach they wanted to take, viewing animation as a central component to explore. Storyboarding began on an opening animation sequence, based on a quote – 'a light unto nations' – lifted from the Genesis brand document. Little more than a very rough illustration of the main elements and the sequencing of their appearance on the page, it would plot a literal interpretation of this

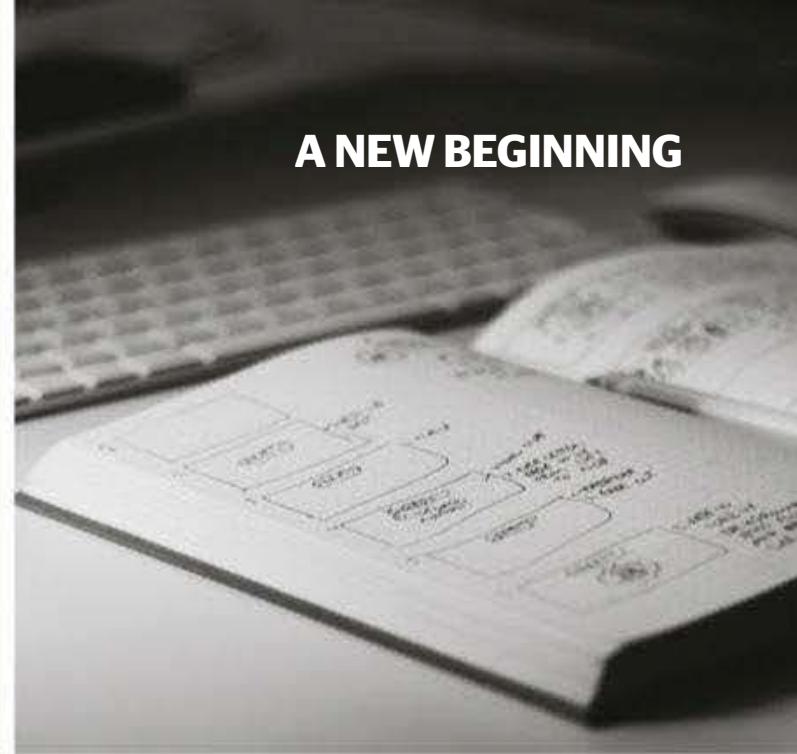
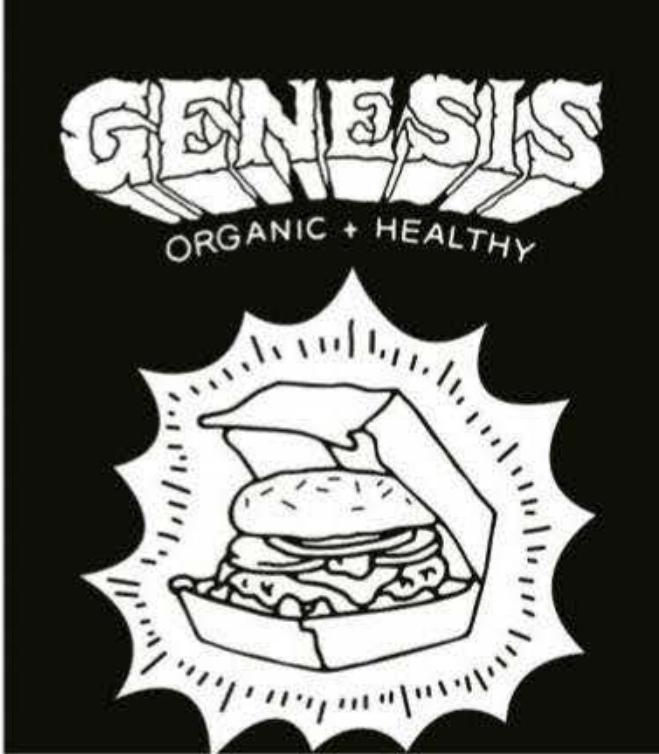


Illustration is a key part of the Genesis brand



HERDL-ING HANDOVER

Every site needs to be reviewed by the client. But, each agency will do it a different way

Upon completion of development, Herdl made the site available on its 'staging server' for review by the client. It is at this point that any final content is added and the site updated prior to actual launch. Once this has been completed and reviewed it's on to the next stage of the process.

The site gets deployed to the production server if the agency is also providing its hosting service. "For every site we produce, there's a limited bug-fixing period post-launch to ensure that any issues are dealt with as quickly and seamlessly as possible," explains Designer Sam Day. While we have a rigorous testing and feedback process in place, occasionally bugs do slip through the net and we offer our clients prioritised support if this ever happens." Such support also extends to training the client on how to use the website's Content Management System, which is bespoke to each project due to the typically heavy customisation of the WordPress admin area. "We remove access to plugins, WP core updates, theme files and any other clutter so the client is comfortable knowing they only have access to the parts they require and safe in the knowledge they can't break anything. But seeing as the web is not a static environment, compatibility and security mean that sites do require ongoing management and maintenance, so we do that too. What's more, our process means that our sites are easily extended so, as a clients' needs change over time, their website can be continually developed to support their requirements."

statement, designing an opening scene that 'lit up' and revealed a setting with obvious biblical connotations. "Using this element as our focal point, we started to break out our ideas into the most important questions that users arriving on the website would have. We had identified two core user groups in our planning – potential customers and employees – and so began to develop sections that would facilitate their needs." The team would then decide on the key information that needed to be communicated, its logical order and also how content should be prioritised, and how it would be displayed to provide an optimal user experience. Paper-based wireframing kept things structural and conceptual, with less attention paid to aesthetic choices like typography and illustrations until frontend work began. "We started with the colour scheme, which we opted to keep black and white. Our vision involved a lot of animation work so we refrained from bringing in colour, which we felt might distract from the experience we had in mind. The black-and-white aesthetic gave the site a distinct style that we wanted to feel recognisable and unique."

ORGANIC APPROACH

This 'unique' look would also permeate into page layout, favouring a more fluid, organic method for positioning the illustrations and content accordingly. "In doing so, we abandoned more traditional web design rules and worked without a grid system, arranging elements to create an effective balance between the illustrations and the content. It was also important that the site would be responsive and display correctly on all devices, so we devised a method that would utilise percentage-based positioning and a process of graceful degradation to maintain the free-flowing

design." Several variations of the one-page design were created before settling quickly on the final choice, laying out the components in a way that would provide the spacing needed for the larger illustrations. Remaining space would then be filled at random with the smaller graphics of flowers and plants in an attempt to maintain a natural, organic feel. Although the site is fully content managed, Herdl built minimum and maximum word counts into the content editor to retain the fluid layout, while also selecting a typeface harmonious to the illustrations. "By far the most significant design task was creating the animations, which were all produced using After Effects. Animations were made without the help of any shortcuts or plugins, as we wanted to retain the hand-drawn style of the illustrations and animate them in a way that best suited this characteristic. We also had difficulties animating the snake, which originally was longer and coiled up. This was redrawn so the animation would be achievable within the timeframe." A basic functional specification was then produced to guide development on how each interactive element would animate or react. Using InVision to add comments describing how each element should behave, this is the technique that Herdl uses on every project to smooth the transition from one department to another.

OPTIMAL PERFORMANCE

During development, the big challenge soon became about incorporating so many CPU-intensive animations on one page. Here the team relied on Airbnb's JS plugin, Lottie, to power the animation due to its ability to export out from After Effects to a JSON file. However this 'shortcut' would have performance implications, with Lottie draining memory while



proving tough to speed up. "We ended up speaking with the guys at Airbnb and dropping the version of Lottie back to an earlier, leaner point in order to get the website performance to an acceptable level." WordPress would, meanwhile, remain Herdl's CMS of choice for its intuitive client admin and open-source architecture, with the team utilising its own special 'flavour' based on the Sage theme developed by Roots, plus several external plugins that assist with backend development. "We find Advanced Custom Fields (ACF) to be a brilliant resource that enables us to heavily customise and streamline the admin experience. Editable content areas are managed through ACF, with added controls for things like character limitations, mobile and desktop menu uploads. Additionally, Gravity Forms powers the Bookings and Contact form and integrates with Mail Chimp for email sign-up."

FURTHER RECOGNITION

With digital marketing very much at their core, Herdl applies that expertise to every website they build. In the context of the Genesis project, the team were therefore expecting a large amount of traffic as a result of PR activities surrounding the launch of the restaurant and also the unveiling of the website itself. It was crucial then that the site made a memorable

first impression on launch to maximise the value of press and media coverage. "Website performance was something we gave particular consideration to, including how we would facilitate high levels of concurrent users at any one time. We spent a considerable amount of time during testing to ensure that the website would render optimally across multiple devices, utilising responsive and server-side techniques (RESS) to tweak layouts and elements. Other frontend optimisations were implemented including caching and optimised file delivery to provide the best experience possible."

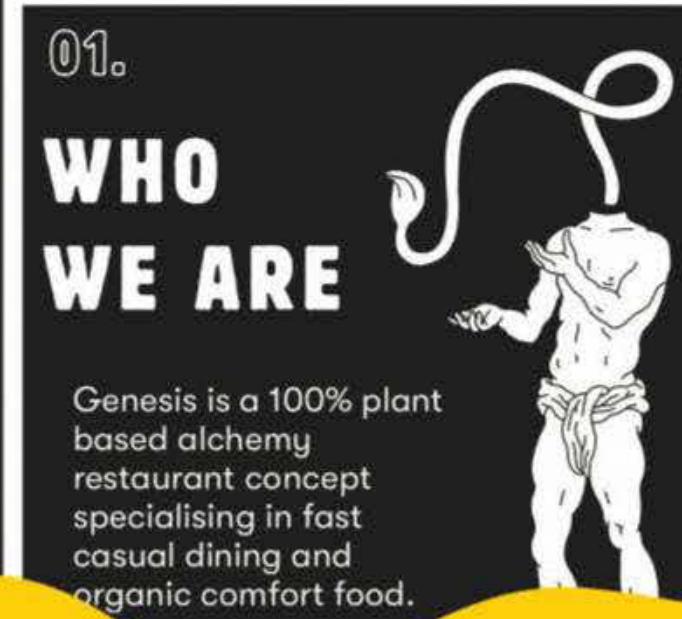
This, coupled with Herdl's emphasis on industry-leading hosting, has gone on to see the finished live website receive an amazing response. Generating a huge amount of exposure through web design awards and inspiration sites since launch, it's rather fitting that the recognition for Herdl's work that attracted Genesis has come full circle. "As an agency, we've been delighted to see the site receive such high recognition as it's a piece of work that we're all really proud of and we're excited to see how the website will develop further," Day concludes. "Most importantly to us in judging the success of any project, however, is the client feedback and thankfully the Genesis guys seem to absolutely love the site and couldn't be happier."

SITE HIGHLIGHT

Designer Sam Day reflects on an aspect of the site that constitutes a highlight and why it might be seen as a defining characteristic

"The Genesis brand message – 'a light unto nations' – was a central component in our design. This theme is present throughout many of the components we created, including the biblical connotations found in illustrations, elements such as the sun and moon, and also the use of revealing animation effects. However, more literally was our use of a flashlight mouse effect, which has been widely commented on. This effect is not only an interpretation of the brand statement but has helped to emphasise elements that interact on mouse movement."

"IT WAS CRUCIAL THAT THE SITE MADE A MEMORABLE IMPRESSION"





Members of JB Cole UK deep in concentration.
You could hear a pin drop

PEOPLE POWER

The event horizon between people and technology is where J B Cole UK thrive. Their innate understanding of how human behaviour can be shaped by technology enables them to enhance the user experience and shape new digital environments



JB Cole UK was founded in 2010 by brothers Josh and Ollie Bolland. Created out of a love for both the creative and technical processes, this agency melds cutting-edge technologies with unique user experiences.

Over the years, the Bolland brothers have honed their skills in user experience and system architecture, building their expertise to develop the agency's current focus on growing and scaling their business.

Today, the brothers are supported by an array of technical, functional and creative minds that enables the company to continue to run with the same vision and mission it had from day one: improve and transform their clients' businesses through the right technology and innovative thinking.

Ollie is CTO and a technical architect with over 12 years' development experience, eight of which has seen him co-running J B Cole UK. He has led complex technology projects for household names including the NHS, TfL, ITV and the National Trust, working on over 200 web projects during his career.

His expertise stretches across system architecture, full-stack development and various frameworks and languages, as well as an end-to-end experience of developing digital products and apps for high traffic websites. He is also a graduate from Trinity Hall, Cambridge and the former Chairman of the Old Fullerton Association for Watford Grammar School for Boys.

Josh is their CEO. The brother with a more entrepreneurial focus, his goal has been to grow the business rather than pursuing further academia after the sixth form. Initially starting with a DSLR camera and opening a photo studio at 18, over the last eight years at the helm of J B Cole UK, Josh has worked alongside Ollie on hundreds of digital technology projects and honed his craft as a UX professional and business leader.

In 2016 and 2017 consecutively, Josh was awarded recognition in the BIMA100 for achievements in the digital industry. He now sits on the BIMA North West Committee and provides knowledge and experience to help shape the future digital sector in

Who J B Cole UK

What Website Design, UX Consulting, Website Development, Django Development, WordPress Development, Technical Direction

Where John Swift Building, 19 Mason St, Manchester, MCR M4 5FT

Web jbcole.co.uk

Key Clients

Hamerville Media Group

Seconique

Intelligent Conversation

Quest Solutions

The Worshipful Company of Skinners'

Transport for London

tfl.gov.uk/

THE CHALLENGE

TfL tasked us with a UX and tech brief to create a bespoke journey planning touchscreen kiosk that enabled customers to work out routes from their current location to another location through different transport methods such as the trains, buses and tram networks.

THE SOLUTION

We used existing internal APIs and scraped data from the website while creating an experience that felt seamless

and was robust enough to cater for a large number of potential users and any security issues that brought.

Working closely with the internal design team, we tested the design and UX hypothesis internally to create the simplest and most effective user experience for tech-inexperienced users.

We came up with an organic user experience, wireframes, architecture, design and build within a four-week period from start to finish.

The screenshot shows the 'PLAN A JOURNEY' screen of the kiosk. At the top, there's a Twitter feed with the message: 'The Victoria Tube Station Upgrade project is one step closer to completion with the opening of 3 new'. Below the feed, there's a search bar with 'From' and 'To' fields, a date selector, and a 'Plan my journey' button. To the right of the search bar is a 'Travel by' section with checkboxes for various modes of transport: Bus, National Rail, London Overground, River Bus, Tube, DLR, TfL Rail, Tram, and Pedestrian. A note at the bottom of this section says: 'The new Bus Safety Standard will make London's buses the safest in the world and help us remove deaths on or by'. At the top right of the screen, it says '14.45:59' and 'Return to Main Screen'.

The screenshot shows the 'Plan a journey' screen. It features a large blue button with the text 'Plan a journey' and a hand cursor icon. On the right side, there's a vertical list of London Underground lines with their current service status: Central (Good Service), Bakerloo (Special Service), Circle (Suspended), DLR (Part Suspended), District (Planned Closure), H'smith & City (Part Closure), Jubilee (Severe Delays), London Overground (Reduced Service), Metropolitan (Bus Service), Northern (Minor Delays), Piccadilly (Good Service), TfL Rail (Good Service), Victoria (Good Service), and Waterloo & City (Good Service). Below this, there's a BBC News tweet: 'The Victoria Tube Station Upgrade project is one step closer to completion with the opening of 3 new'. At the bottom right, it says '14:45:59'.

The screenshot shows the 'Continue your journey' screen. It features a large blue button with the text 'Continue your journey' and a hand cursor icon. On the left, there's a smaller screenshot of the 'PLAN A JOURNEY' screen. At the bottom left, there's a note: 'The journey planner was designed to be both effective and easy to use.'



“We’re always looking to the future. Innovation is a fundamental facet of how our business was first shaped and it always remains at the core of what we do. We always want to innovate ourselves and ensure we improve our clients’ businesses through our innovation and technology strategies”

Josh Bolland
CEO and Co-Founder



Alice, Amy, Jonny and Ollie hard at work. We particularly like the rustic desk they are all using

Manchester, as well as hosting a number of leading industry events.

At first glance, naming their agency seems to be pretty straightforward. However, as Josh explains, there is more to the story than meets the eye: "Funny story! Originally, I founded the company as a photography and moving image studio in North West London back in 2008. At the time, Josh was trading under the name 'Josh Cole' because, well, that's his name. Unfortunately for him, it soon became apparent there was another, more established – and published – photographer in London who went, and still goes by that name. The photographer was somewhat of an inspiration to Josh, so he decided it would make the most sense to change the company name whilst still fresh in the game, and not upset the apple cart.

"After many hours and several conversations around creating new names, as well as consulting family and friends, it was decided to use a variation of Josh Cole Bolland and to still focus on the 'Cole', reducing his initials to J B Cole.

"Over the years, we have dabbled with the idea of changing the name, created new trading styles and tried moving to something more 'agency.' However, the name has always just stuck. Clients know it, people remember it and it's a name that separates us out from the competition by being just a little bit more personable."

All agencies use their websites as the first point of contact for prospective clients. As experts in web design, J B Cole UK have always strived to ensure their site stands out. "As with any business, our website is an extremely important brochure tool," says Ollie. "For us, it's a direct reflection of the work we do. This means that our current clients, and those we

haven't met yet, need to feel assured that we understand what makes a great looking and, above all, usable website.

"We put as much time as possible into the content and upkeep of our site. We refine the experience for the user based on our analytics. And we are always promoting work we are proud of and continue to share knowledge, thinking and experience on relevant subjects."

Locating the work they want to be involved with has meant using every channel that is available today. As Josh outlined, being established for nearly a decade has meant a presence in the marketplace: "We have good working relationships with all of our clients,

"We are constantly refining our skills, updating our web presence and always striving to be the best we can be"

so we often get repeat work and referrals. We also have a strong strategic agency partnerships approach that stretches across the country that contributes a percentage of our revenue each year.

"We are constantly refining our skills, updating our web presence and always striving to be the best we can be. This means when it comes to pitching, we are confident that we can give both new and existing clients the best results. No two projects are the same and so we treat them all individually."

Choosing which businesses to work with needs careful consideration. A project clearly has to have a commercial component, but also enable J B Cole UK to extend and expand their

skills. Josh explains their approach: "We have a specific type of client we aim to work with and a qualification process to rule out those who we don't. We try and be as selective as possible when it comes to pulling ourselves out of opportunities that don't quite match. However, the reality of any growing business of our size is that sometimes projects are required to stabilise cashflow and aren't easy to turn away from.

"Additionally, the larger the project value, the lengthier the sales or pitch process. This means it's absolutely essential to have the right balance and combination of recurring revenues and smaller projects to ensure adequate cash flow management. We don't take on ongoing marketing projects as that's never been our area of specialty. We have a large pool of strategic partners who we can always tap into for this type of support and referrals are passed both ways."

Able to push the boundaries of digital experiences, J B Cole UK understands how each of their clients' needs to transform. "Working with long-term client Hamerville Media Group, we're proud to be a part of their digital and transformational change," says Ollie. "A traditional print company with a need for change, we've been able to work intrinsically within their organisation, alongside key staff and stakeholders, helping to shape the entire future of the organisation

"This fits every facet of what we're about as a company. Providing strategy, consultancy, production, support, training and an ongoing partnership mentality that has built a solid, trusted relationship over time.

"They understand our processes and our team works regularly in their offices, so we understand their business inside out. This

level of understanding builds trust and a continued growth of the account from within.”

As the work that J B Cole UK carry out for their clients can be highly varied, they have developed a working process that is flexible and able to adapt to the precise needs of their clients. Josh outlines their approach: “Whilst we have several ‘frameworks’ in place for certain project types, these are flexible to meet the unique goals for each project.

“For larger projects using waterfall, we’ll hold an internal discovery session with all members of the team to ascertain our understanding of the brief. We’ll follow this up with a client discovery session to match our understanding to their expectations and settle on an agreed scope of work for the project.

Timeline

2010

J B Cole is founded by Josh and Ollie Bolland to create a new kind of agency to transform user experiences in digital spaces.

Employees: 2

2013

Created Nokia Music – an audio/visual playlist site for a collaboration between Nokia Music and Sundance Film Festival.

Employees: 4

2014

Global Poverty Project – interactive widget for a collaboration between the Global Poverty Project and the Huffington Post, announced at Davos.

Employees: 4

2015

The Worshipful Company of Skinners’ – website and membership platform for heritage Livery Company in London created.

Employees: 4

2016

SLG – bright, colourful and innovative website for creative agency with many sliding features.

Employees: 8

2017

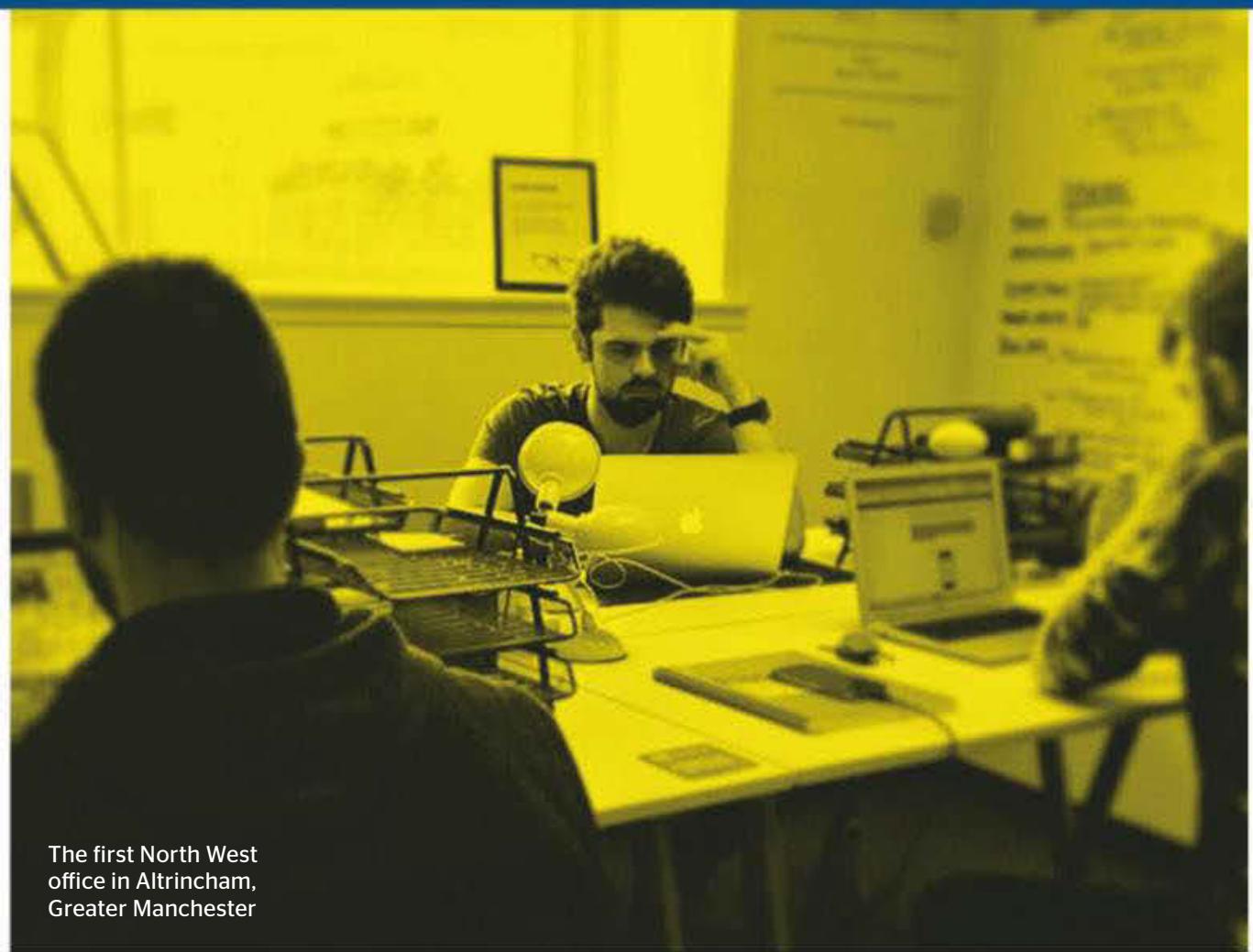
Gigaflex – a cloud-based hosted server management solution for technology agencies.

Employees: 8

2018

Hamerville (KBN) – a visually-led website for Kitchens and Bathrooms News with focus on maintenance and usability. Quest Solutions – An online student accommodation property management system.

Employees: 10



The first North West office in Altrincham, Greater Manchester

“From there, we use this to map out each project phase and assign tasks to the team member most skilled in the specific field. This could be UI/UX design, wireframing, frontend or backend dev, digital strategy or digital and tech support. Whilst not necessarily the longest phase of the project, this is certainly the most resource-intensive phase, often involving the whole team to harvest the information needed to define the project phases.”

A varied client roster means being able to choose and then apply a distinct set of tools. Ollie explains what's in their current toolbox: “We use a variety of tools in the creation of project materials, from Adobe CC apps for design work to coding apps such as Sublime and Coda.

“For general project and team workflow management we use ActiveCollab, which is not without its flaws, but for the time being is working for us. For Agile projects, we work with Jira. We also use Slack as a communication tool, which can be invaluable when remote working but also has the potential to create time vacuums! To combat this, we put preventative measures in place and always try to jump on a quick call or have a brief huddle to discuss and resolve.

“Possibly the most useful tools are those included with our company Google account. From email to calendars to spreadsheets and docs, it can involve the whole team providing the transparency and communication that is key to our business and clients.”

Ollie also explains: “There’s an awful lot you can do with the standard HTML5 set, but it’s

become easier than ever to dump frameworks, plugins and libraries into the frontend to fix minor issues and cause bloat. We’re big fans of the old 10k contest and were always impressed by what could be achieved with some applications nearly one-tenth the size of jQuery. And while we were never fans of the full MEAN stack, the power to produce fully functioning applications with storage using only JavaScript is very impressive – we have had favourable experiences with Angular and React in recent years.

“Of course, as an agency, you’ve often got to trade off the timelines and budgets required for a highly visual site with frontend performance, which can be a challenge. However, with the near-universal adoption of some of the best modern standards, like CSS Grid, Flexbox and SVGs and the better adoption of ES6, it’s becoming easier to write cross-browser and device code that you can be confident just ‘works.’ Further, with the retirement of several legacy (desktop) browsers, many of which caused significant ongoing issues for developers, particularly IE 6-9, we can expect libraries and frameworks to become smaller, with some becoming obsolete.

“In the mobile world, responsive design has had one of the most significant impacts on our industry over recent years and now projects like AMP (Accelerated Mobile Pages) are making mobile performance a priority. Again, there’s always the opportunity for bloat to slow down interfaces, but HTML5 can be highly performant on mobile devices – for most apps, an HTML5 interface can be more than good

Hamerville Media Group

hamerville.co.uk

THE CHALLENGE

Hamerville Media Group is one of the leading trade publishers and exhibition organisers in the UK. For the last 45 years, they have run an audited suite of print magazines, targeting professional tradesmen and merchants, including builders, electricians, plumbers, mechanics, property developers and hairdressers.

The existing print business is mature, and the magazines have a significant circulation, including the largest B2B circulation in the country. However, as advertising budgets

are diverted increasingly to digital media, it became clear that they needed to improve their existing online approach.

Each magazine has a website, as well as supporting social media presence and email lists. They also host 21 exhibitions across the country throughout the year and have supporting systems for these as well. However, the digital approach was fragmented and did not have a single digital vision to promote the businesses online.

THE SOLUTION

From an initial analysis, we produced a roadmap to give the editorial teams the ability to produce the right content for online and the sales teams the confidence to discuss offerings in the new digital market.

The initial roadmap required us to establish the baseline of where the sites currently were. To this end, we cleaned up the existing analytics accounts and ensured there was a measurement plan in place.

Working with the editorial teams, we implemented Google Tag Manager across the sites. We also set up reporting within Google Data Studio to ensure we could easily measure improvements across multiple segments, rather than a single block of traffic. With the baseline established, it became much clearer where improvements were necessary. We are now working in partnership with Hamerville to deliver this roadmap and look forward to working with them to improve their digital strategy over the coming months.

“From an initial analysis, we produced a roadmap to give the editorial teams the ability to produce the right content for online”



“There’s an awful lot you can do with the standard HTML5 set, but it’s become easier than ever to dump frameworks, plugins and libraries into the front-end to fix minor issues and cause bloat. We’re big fans of the old 10k contest and were always impressed by what could be achieved with some applications nearly 1/10 the size of jQuery”

Ollie Bolland
Technical Director and Co-Founder

Gigaflex

gigaflex.co.uk

THE CHALLENGE

The business world is changing. The office structure as we currently know it is becoming obsolete.

Innovative small businesses are pioneering this always-on approach to business connectivity and communications. However, they can only do that if the technology is there, works, and is a slick experience that minimises any fuss. Cloud-infrastructure provider Gigaflex approached J B Cole UK to transform their hosting portal

dashboard to provide the best user experience for their customers.

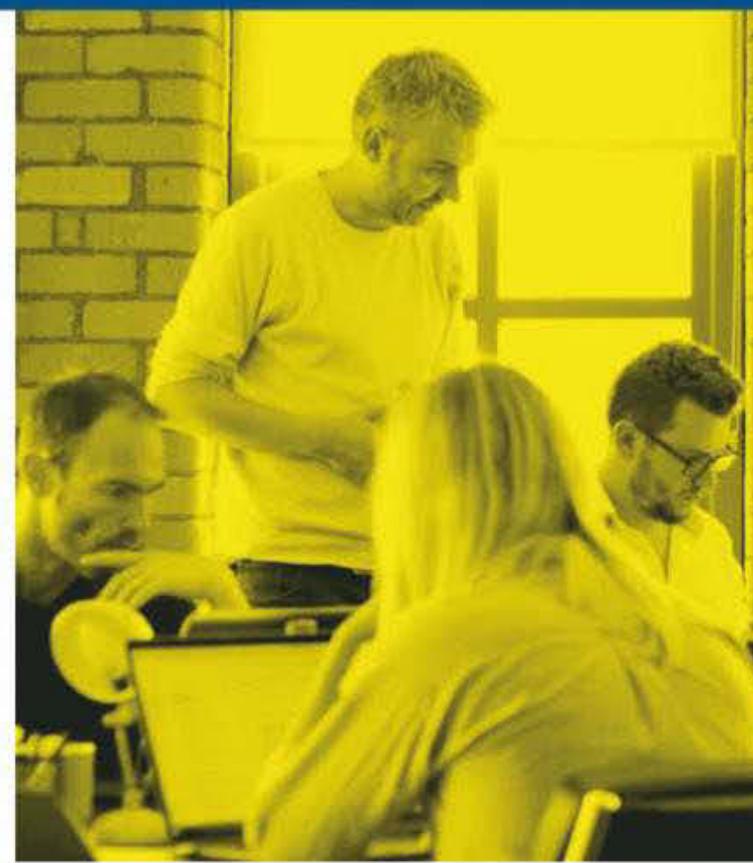
THE SOLUTION

We developed an updated version of the existing dashboard, completely overhauling the customer journey and providing a slick end-to-end user experience. We always begin every transformational solution by evaluating the customer's interaction with the client and the existing technology. By mapping out and analysing each element of the customer cycle, we're able to plot change with simple yet innovative solutions.

This screenshot shows the 'Create a new Instance' form within the Gigaflex Compute interface. It includes fields for 'Instance Name' (with placeholder 'My New Instance'), 'Availability Zone' (set to 'None'), 'Count' (set to 2), and a checkbox for 'Enable auto scaling'. On the right, a sidebar lists configuration options: Details, OS, Instance Type, Network, Network Ports, Security Groups, Key Pair, Disk Partition, Media Data, and Configuration. At the bottom are 'CONTINUE' and 'LAUNCH INSTANCE' buttons.

This screenshot shows the 'Create Image' form within the Gigaflex Compute interface. It includes fields for 'Image Name' (placeholder 'My New Image'), 'Image Source' (URL input field), 'Format' (dropdown), 'Image Requirements' (dropdown), 'Kernel' (dropdown), 'Architecture' (dropdown), 'Image Sharing' (dropdown with 'Public' and 'Private' options), and 'Minimum Disk (GB)' and 'Minimum RAM (MB)' inputs. A 'CREATE IMAGE' button is at the bottom. To the right, a table lists existing images with columns for 'Name' and 'Actions'.

“We developed an updated version of the existing dashboard, completely overhauling the customer journey and providing a slick end-to-end user experience.”

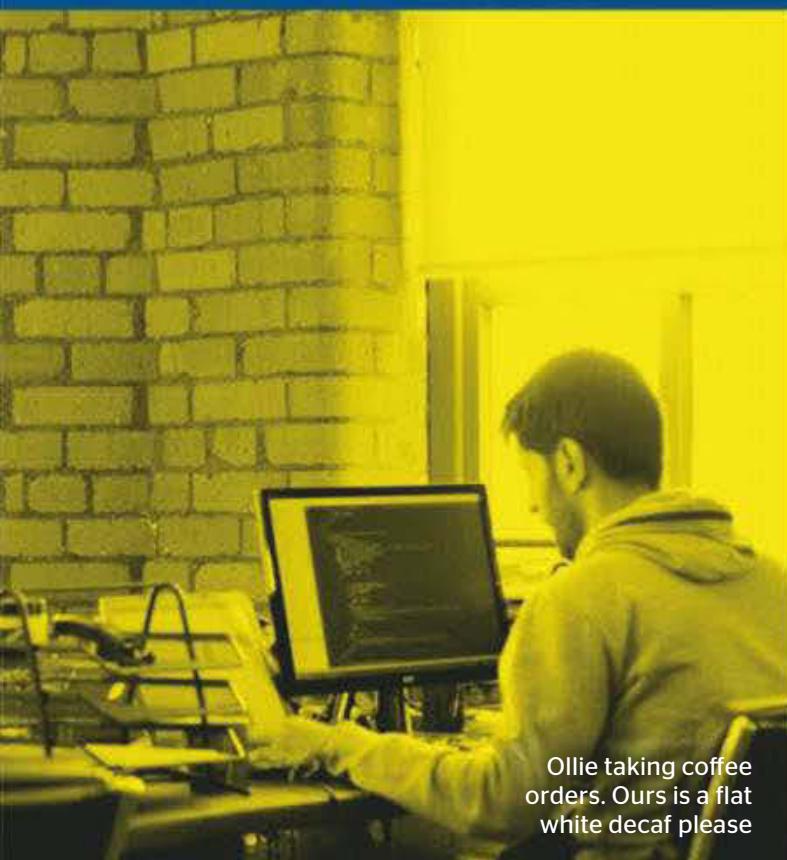


enough and the gains of using one source across platforms can be massive. The future looks very bright for HTML5. Except for HTML emails in Outlook, which will cause developers to pull out their hair for many years to come.”

Josh also comments on the impact that mobile has had on their work: “We treat the mobile layout as equal to the desktop, rather than an afterthought, while responsive design and development allow us the flexibility to manage the ever-increasing variants in between. BrowserStack is a great tool which enables us to view sites on different browsers and operating systems in real time. The only time we wouldn’t employ this approach is if the project was device dependent. For example, in the case of a touchscreen application. In this instance, we would design and build the project according to the size and spec of the launch device.”

As digital denizens, the environment J B Cole UK is working within is constantly evolving. What’s exciting them at the moment? “We’ve started paying a lot more attention to Google Data Studio,” says Ollie. “This is giving us the flexibility for more customisable dashboards from Google Analytics – when you’re trying to show how multiple Segments have changed month-on-month. It’s much easier to see all on one screen, though it can be a little slow if you have too many complex queries.”

“We’re a big fan of Amazon’s Web Services but haven’t really found a great use for Lex yet, though Polly (text-to-speech) and Rekognition (image identification system) have been great for small but key parts of projects. In general, the AWS environment is introducing new services very quickly, both in changes to existing offerings and with completely new products, so it’s a challenge to stay on top of all the things they offer.”



"Hardware-wise, we've spent a lot more time with the Raspberry Pi recently and found the 3 to be an incredibly powerful little unit for the price – particularly when working in schools. We've seen students who have put them to great use, such as auto-registration systems for their classrooms, using the Pi, an NFC Reader and some JS and Python. It's amazing to think what these 15- and 16-year-olds will be building in the future."

Digital design today can't ignore the social media networks. Josh outlines how these spaces are integrated into their projects: "Social media is an important part of any digital campaign to reach the widest possible audience – but high engagement on social media, while possibly making more users aware of your campaign, doesn't always translate to higher site visits or conversions.

"The exciting thing about the future of UX is the fact that users understand more with less"

It's important to be clear about what the target is for any given campaign and to ensure the strategy is designed to drive that approach."

Since J B Cole UK are shaping tomorrow's digital spaces, how does it imagine that these environments will evolve? Ollie explains: "Thousands of companies in the UK are trying to predict the next 12 months. Trying to see anything further than a couple of years ahead is incredibly difficult.

"The changes digital technology has brought to the world to date have been significant – it's affected every industry while also being the starting point for quite a few of our friends' marriages. But we're seeing a shift as the

cutting edge becomes mainstream and moves into the home.

"We've been in schools talking to 14-15 year-olds who regularly use IoT, drones, voice services, VR and 3D printing and have produced some of their own tech on top of these. With driverless cars and trucks around the corner – and the ability to use plug-and-play APIs to access high-powered AI and ML technologies through AWS, Google, IBM and others – guessing what the future holds is a favourite pastime."

How people will interact with technology is often a core focus for J B Cole UK. For Josh, this means understanding the interface between the technology and its users: "UX has always evolved and so has the consumer. Now, more than ever, our user has expectations that never existed before – slick UI, lightning speeds, look and feel, security, etc. They want it all and they expect it – why shouldn't they?"

"In our eyes, the whole package now falls under the UX banner. It's not just one-dimensional and it crosses over into way more than just pixels. How is a user's experience shaped by immersive technologies such as AR and VR? How do we use these tools to begin engaging and enhancing the experience through each facet of the consumer journey? These are the type of questions that will be more and more common when UX is being discussed at the discovery phase."

"The exciting thing about this future of UX is the fact that users understand more with less. You don't have to be as informative (depending on the demographic) as long as you're using the right iconography, shapes and asset positionings and, therefore, you can be less literal. With this what I see UX becoming more and more minimalist and, subsequently, more intuitive over time. It requires fewer stage gates to meet conversion goals."

Josh concludes: "Another aspect – that really crosses deeper into CX – is that users are spending more time interacting with a brand outside of their website than before and therefore, conversion is happening before they even get to your page.

"This means it's about making sure every expected avenue is covered when it comes to a user's experience interacting with a brand. This can be the way they market themselves or communicate through social channels as well as important external factors such as their ratings across third-party sites or Google shopping, for example. Really understanding your users' purchase flows will have more of an impact on everything, and brand ties into the UX more than ever before. It's not just about wireframes."

And what does the future of J B Cole UK look like? "We're always looking to the future,"

says Josh. "Innovation is a fundamental facet of how our business was first shaped and it always remains at the core of what we do. We always want to innovate ourselves and ensure we improve our clients' businesses through our innovation and technology strategies."

With Ollie stating: "Many of our clients are complex and large-scale organisations. This means that technology can often be daunting first and foremost. It can also be hard to change established processes and ways of working. This has led us to launch an extremely exciting strategy product – Technology Tune Up."

"This has seen us reaching out to organisations, in sectors including manufacturing and other traditional industries, to help them understand how they can tune up and enhance their businesses quickly and simply with the right technologies. We're hoping to really enhance these industries and illustrate how game-changing the right digital technologies can really be for forward-thinking businesses."

Leveraging the latest technologies and techniques has ensured J B Cole UK has stayed at the forefront of digital design. It will also ensure that they will be here for a long time.



jbcoll.co.uk

Founders
Josh Bolland
Ollie Bolland

Year Founded
2010

Current Employees
10

Location
Manchester and Hertfordshire

Services

- Digital strategy
- System architecture
- User experience
- Website design and development
- Digital Training

NEVER MISS AN ISSUE!



Issue 280

30 best HTML, CSS and JS APIs and elements, research your keywords, interactive JS charts and What is the Web of Things?

SAVE UP
TO 30%
WITH A DIGITAL
SUBSCRIPTION
[SEE PAGE 28](#)



Issue 279

Pro tips to build animation for the web, 23 ways to make your app shine, work with variable fonts and what's new in Node 10



Issue 274

The new rules and top tools for UX, 5 hot new CSS properties, use the CSS Paint API, add particle effects and discover JSX

GOT AN APPLE DEVICE?

Download Web Designer's digital edition for your iPad, iPhone or iPod touch

<https://apple.co/1hsGYgl>



CATCH UP ON ANY ISSUES YOU'VE MISSED BY DOWNLOADING
OUR DIGITAL EDITIONS <https://bit.ly/2rh73Xk>



Issue 278

What's new in JavaScript, say hello to Google Flutter, build with CSS Grid, work with WebRTC and create patterns with CSS



Issue 277

Hot new CSS properties for today, build interactive 3D, streamline your design workflow with Sketch and code a real-time React app



Issue 276

Build your own augmented reality app, get started with Three.js, power up with Sass and Stylus and generate web components



Issue 275

50 must-try tools for designers and developers, how to add awesome audio, Progressive Web Apps and interactive images



Issue 273

SEO Today, 20 hot new libraries and frameworks, ten pages of ReactJS tutorials, get started with Rust and build a WebGL racing game



Issue 272

Build with Web Components, 8 WordPress security secrets, Web Accessibility, convert visitors to customers and data visualisations



Issue 271

28-page design special. Create the perfect colour palette, top type trends, design and build grid layouts and automate your workflow



Issue 270

Get into creative coding, 16 of the best free fonts, prototype with Facebook Origami and how to power up your portfolio

PREFER TO READ ON ANDROID, PC OR MAC?

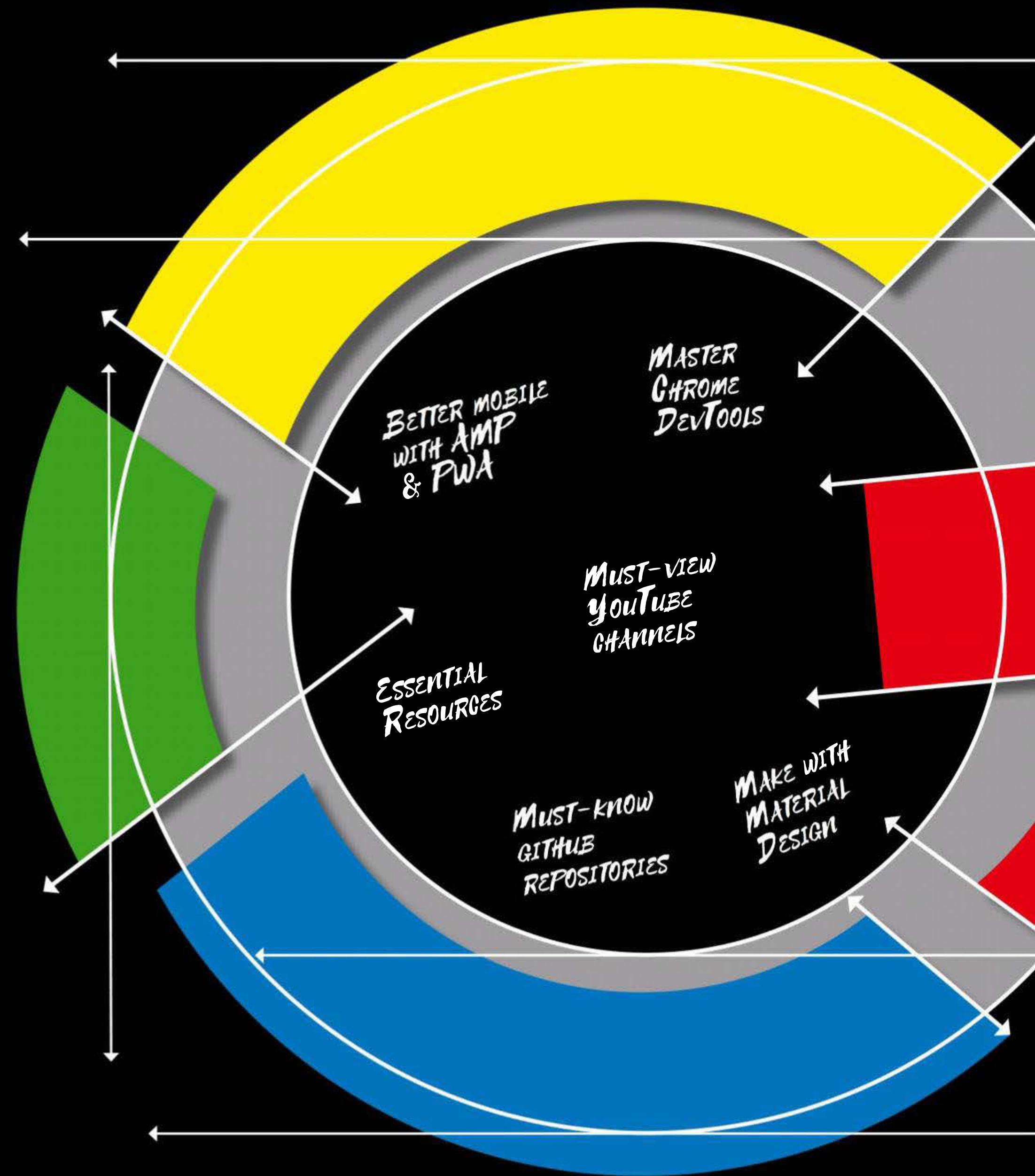
Web Designer magazine is also available on Google Play and Zinio

<https://tinyurl.com/yalm3wul>



<https://bit.ly/2xPbv4p>





DESIGN DEVELOP CREATE

DESIGN DEVELOP CREATE WITH GOOGLE

BIG G PROVIDES DEVELOPERS A PLETHORA OF TOOLS INTENDED TO MAKE WEB DESIGN EASY. THIS ARTICLE SHOWS YOU SOME OF THE BEST OUT THERE



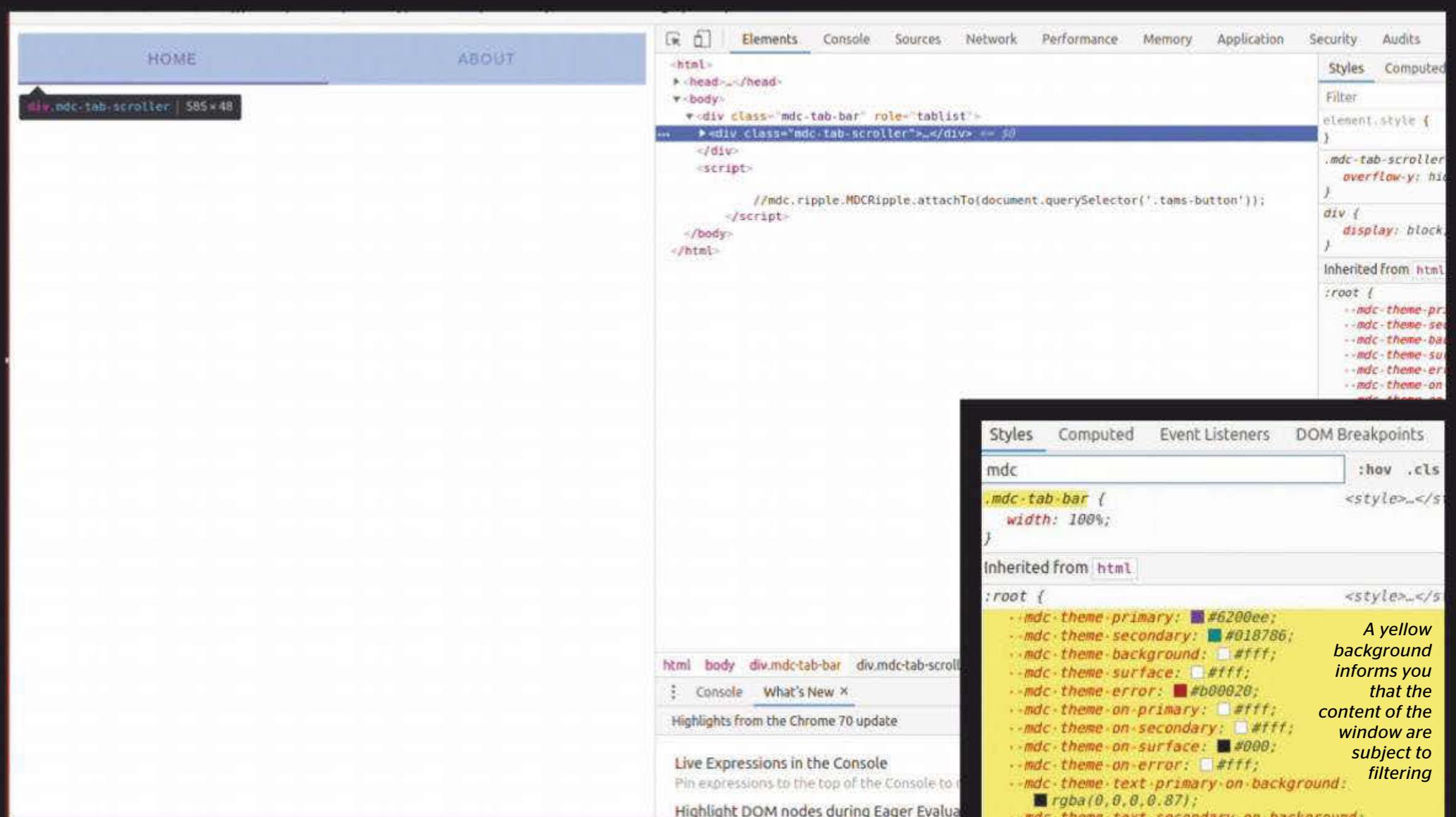
“Combining debugger and rendering feature in one application speeds up the debugging workflow. Making small changes “on the fly” ensures that our price information looks great on all displays **”**

Sam Cowley, Founder
oemsecrets.com



VIEW AND CHANGE HTML, CSS & JS

As long as web sites are unminified, their content can be viewed by anyone. Chrome's developer tools simplify the action



First of all, check the version of Chrome – the following steps work on version 70.0.3538.67 running on a 64-bit Ubuntu workstation. Next, open the hamburger menu and click More Tools > Developer Tools. The browser responds by opening a pane on the left-hand side of your screen. It should be resized in the first step – by default, the browser doesn't assign enough screen real estate. When done, the screen looks similar to the one shown in the figure.

By default, Developer Tools starts up with the Console tab loaded. It contains the output emitted by invocations of the console-logging function. Rendering errors also show up there – if you find red lines, something is amiss in the markup or the code of your page. A click on the little 'URL' on the right-hand side brings you to the line, which caused the message to show up.

INSPECT THE MARK-UP

If your interest focuses on markup, use the Elements tab shown in the figure. It shows the markup the browser currently renders. You can expand and compact it in a fashion similar to a tree view – hide unneeded elements to prevent sensory overload.

The *raison d'être* of the tab is the display of the CSS structure. Widgets 'collect' CSS from various sources, which are displayed neatly and next to one another. In the case of the example shown in the figure, we see that styles were provided from both the 'div' tag and the 'main' markup of the page.

Chrome lets you edit most attributes on the fly. Double-click any of the black texts to transform the label into an editor, then enter the newly desired value and press the Return key. The renderer picks up these changes and applies them to the DOM on the fly. This is especially useful when trying to optimise colours or placement issues; having to reload a page to preview changes becomes tiresome.

Finally, take a look at the text box on top of the pane. It enables you to filter the markup items shown: for example, enter 'mdc' to limit the view to all styles inherited from the Material Design library used later in this article. When a filter is active, a yellow background shows up in a fashion similar to the one shown in the figure.

PERFORM A DETAILED ANALYSIS

In modern web browsers, the DOM is not limited to 'static' presentation issues. It

A yellow background informs you that the content of the window are subject to filtering

```
mdc
  mdc-tab-bar {
    width: 100%;
}
Inherited from html
:root {
  -mdc-theme-primary: #6200ee;
  -mdc-theme-secondary: #018786;
  -mdc-theme-background: #fff;
  -mdc-theme-surface: #fff;
  -mdc-theme-error: #b00020;
  -mdc-theme-on-primary: #fff;
  -mdc-theme-on-secondary: #fff;
  -mdc-theme-on-surface: #000;
  -mdc-theme-on-error: #fff;
  -mdc-theme-text-primary-on-background: rgba(0, 0, 0, 0.87);
  -mdc-theme-text-secondary-on-background: rgba(0, 0, 0, 0.54);
  -mdc-theme-text-hint-on-background: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-disabled-on-background: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-icon-on-background: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-primary-on-light: rgba(0, 0, 0, 0.87);
  -mdc-theme-text-secondary-on-light: rgba(0, 0, 0, 0.54);
  -mdc-theme-text-hint-on-light: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-disabled-on-light: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-icon-on-light: rgba(0, 0, 0, 0.38);
  -mdc-theme-text-primary-on-dark: #fff;
  -mdc-theme-text-secondary-on-dark: rgba(255, 255, 255, 0.7);
  -mdc-theme-text-hint-on-dark: rgba(255, 255, 255, 0.5);
  -mdc-theme-text-disabled-on-dark: rgba(255, 255, 255, 0.5);
```

also governs event flow and placement. Switch to the Computed tab to reveal a list of 'dimensional' properties. This is incredibly useful when arranging widgets on the screen in an exact fashion.

Moving to the Event Listeners tab provides an overview of event flows. Use this feature to quickly weed out problems related to user interaction: if an event does not trigger, start out by checking the event handler connections.

Finally, the Properties window lets you take a look at the attributes stored inside the individual DOM nodes. If you've ever spent a lot of time hunting down DOM-manipulating code, the value is clear.

USING DEBUGGER

Coding errors are among the most common problems faced by programmers. Handling Chrome's debugger well saves time and effort

In the good old days, developers debugged by emitting messages to the command line. The introduction of dedicated debuggers accelerated the error-finding process.

Debugger work centres around breakpoints. They act as an entry point into the debugging session – if code hits them, its execution pauses. The easiest way to add a breakpoint involves the Sources tab, where you click the margin to add a blue rhomboid.

When the relevant line is hit, a yellow insert pops up over the rendered view. Furthermore, the debugger window populates with various bits of information about the current 'context'. Moving your mouse over any variable opens a pop-up window similar to the one shown in the figure. If the element in question is an object, a tree view will appear instead. It enables you to drill down into the individual variables.

Global variables show up in the Scope node. Click it and be prepared to wait for a second or two. Its population cannot be accomplished instantaneously due to the number of elements, using the Filter textbox at the top of the screen is highly recommended for usability.

As populating the entire state tree is slow, some fields get shown as (...) instead. Double-click any one of these attributes to load its

value on the fly. Finally, use the call tree to find out where you are. It lists the methods called to arrive at the execution position.

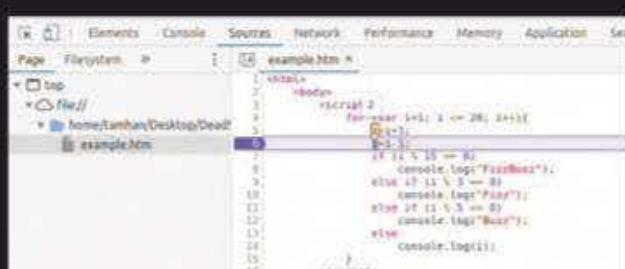
ADVANCED EXECUTION

Placing breakpoints all over the place is inefficient. Analyse the flow of a variable by clicking the three step-over buttons next to the blue Play button. They enable you to run a single line or return from a function while keeping the debugger active during the process. This is helpful when hunting down value changes as an algorithm does its work.

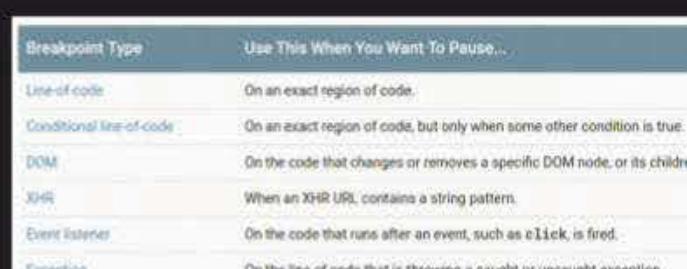
Another neat trick involves the use of conditional breakpoints. Chrome supports half a dozen of them, the table accompanying this boxout describes them.

Setting these is done outside of the debugger. In the case of the DOM tree, for example, a node must be selected in the Element view. You can then specify that a debugging session must launch whenever the content of this node changes.

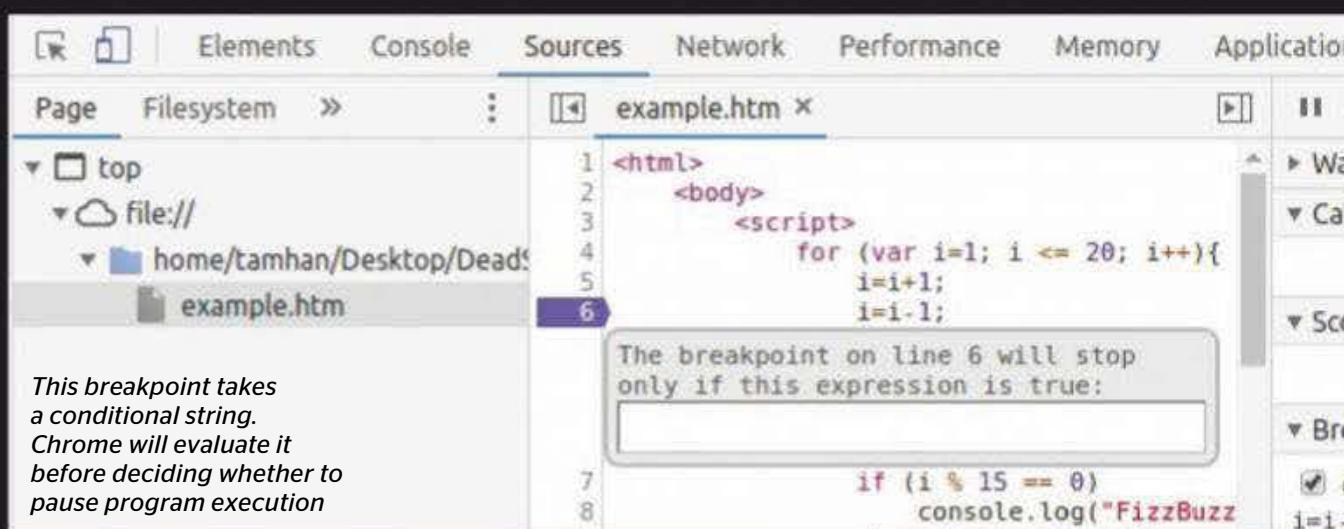
Just like with modern oscilloscopes, providing an overview of all trigger modes would require a book of its own. Let us, thus, end this little trip with a reference to developers.google.com/web/tools/chrome-devtools/javascript/breakpoints – it provides an excellent overview of the possibilities.



Letting your mouse pointer rest over a variable reveals its contents in a small pop-up window



Chrome's debugger provides 'thinking' breakpoints, which trigger only if specific conditions are met



RESOURCES

SUNDY DEBUGGING TIPS

raygun.com/javascript-debugging-tips
Finding errors in code is both an art and a science. The folks at **RayGun.com** have compiled a set of methods, tools and approaches intended to help you find hard-to-track issues in JavaScript applications. Create scrollable elements which have defined regions that should snap into view. Those larger than the viewport are handled automatically.

DEBUGGING JAVASCRIPT CHEAT CARD

dzone.com/refcardz/debugging-javascript?chapter=1
Like most other scientific endeavours, a few 'best practices' have arisen over time. The reference card from **dzone.com** might have an extremely annoying layout, but it does provide an overview of interesting aspects.

CHROME DEVELOPER TOOLS REFERENCE

developers.google.com/web/tools/chrome-devtools/
Chrome's debugger contains dozens of additional features that we don't have room to cover in this tutorial. Seasoned web developers are advised to take a look at Google's official documentation – some of the functionality tends to be a real timesaver.

WEBSTORM

jetbrains.com/webstorm/
Sometimes, Chrome's developer tools simply don't offer what you might be looking for. If that is the case, you can give a dedicated product like WebStorm a chance. Its debuggers tend to analyse the entire project structure, leading to even more advanced analytical capabilities.

WORKING WITH DEVICE MODE

An old adage states that mobile content should be tested on a real device. Recent developments make desktop emulators look more appealing...



While complete emulation of mobile devices on Chrome is impossible, the product does make your life easier. First of all, click the little smartphone pictogram shown in the top-left corner of the Developer Tools screen. Chrome will adapt the view in a fashion similar to the one shown in the figure.

The combo boxes at the top of the screen enable you to pick various commonly used devices ranging from the old Samsung Galaxy S5 to more recent products such as the iPhone X. Bear in mind that this feature does not switch the actual rendering engine. The browser does its magic purely by adjusting viewport coordinates. This limitation also applies to screen rotation, which you enable via the Rotate pictogram in the toolbar.

Mobile apps usually use sensor data. Chrome can emulate these – open the Developer Tools hamburger menu. Next, select More Tools > Sensors. The Console window takes up an additional pane with sensor settings. They provide access to geolocation, accelerometer and advanced touch input.

For completeness sake, let us reiterate that desktop tests can't replace 'acceptance runs' on a handset. Mobile applications live and die by their haptics, which are completely different from those found on a PC.

As of this writing, Google has not 'unified' the various aspects of device mode. The various functions, which are described in detail at developers.google.com/web/tools/chrome-devtools/device-mode/, might get a new user interface in the future – stay tuned for small changes in this regard.

WHAT IS CONSOLE

Simply sending out logging information via console.log reeks of the 1990s. Why not use some advanced features instead?

One of the defining features of Android is LogCat – developers can emit logging messages with various levels of severity. Chrome also supports this feature. Simply use one of the following methods:

```
console.debug() == console.log()
console.info()
console.warn()
console.error()
```

Chrome's console is not limited to displaying information. You can also interact with the content of your website in a fashion not dissimilar to PowerShell or BASH. When working on code, be careful to look at the combo box at the top of the screen.

It lets you select where JavaScript will run – many types of content live in their own iframe. They cannot be reached by code entered into the top execution context. Once the right execution area is selected, changing the content of elements is as easy as entering a single command:

```
document.getElementById('changeMyText').textContent = 'Hello'
```

Chrome's console tends to flood once projects get complex. This is addressed by collating similar messages. Click a button five times, and emit the same message each time – it will show up once. Disable this by opening Settings > General. Then, check the Show Timestamps option. Alternatively, introduce tags and use the Filter feature.

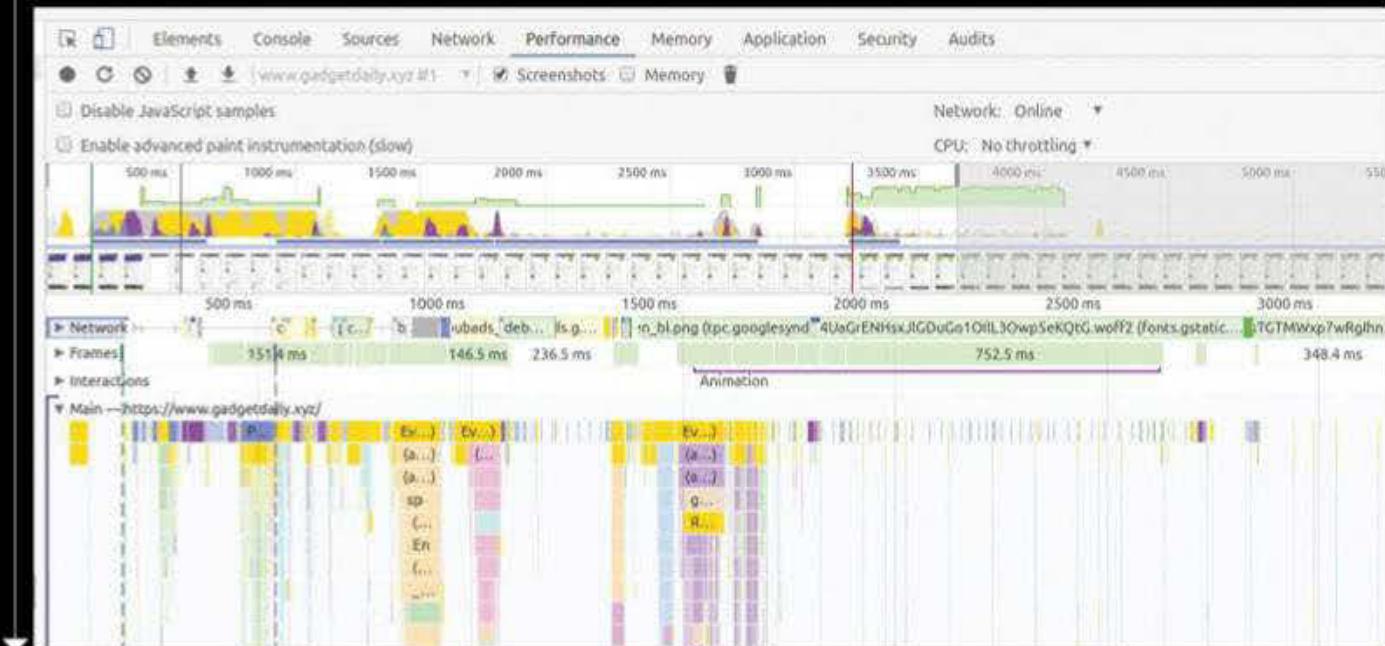
ADVANCED ANALYSIS

Chrome comes with a complete set of advanced analysis features for tracking down performance problems

Finding coding errors is but part of the game. As applications become more complex, memory, performance and network consumption problems arise.

Chrome assists you in addressing these problems. The browser provides analysers not dissimilar to profilers found in embedded operating systems. For example, the memory analyser provides an overview of which part of your code allocates memory. This information enables targeted optimisations; 'hacking away' at code tends to be unproductive.

Opening the Performance tab lets you slow down both Network and CPU – this way, your workstation simulates slower computers. Combine this with the time-consumption analysis shown in the figure accompanying this step to get a 360-degree view of application performance. Due to space constraints, we cannot cover this topic completely. However, the overview at developers.google.com/web/tools/chrome-devtools/memory-problems/memory-101, provides an excellent introduction to advanced debugging functionalities.

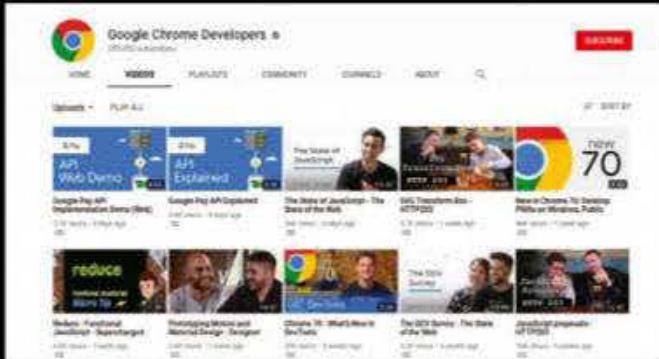


ESSENTIAL TOOLS & RESOURCES

Watching videos, studying tutorials and checking code repositories makes it easier to learn. Here are a few good Google-related ones

YOUTUBE

Watching videos makes Chrome easier to learn. YouTube and Google combine to offer some great video content



GOOGLE CHROME DEVELOPERS

bit.ly/28PpwDs

Google's Chrome team regularly publishes videos on various aspects related to the Chrome ecosystem. They furthermore explain interesting, helpful and/or oddball aspects of web development.

NEW IN CHROME

bit.ly/2D4Yw3x

This regularly updated playlist spotlights new features of Chrome's renderer and the various accompanying web APIs. The concise format makes them ideally suited for a morning run.

A11YCASTS

bit.ly/2EKn3wu

Accessibility is not only 'moral', but also helpful – accessible pages tend to work better as bandwidth constraints kick in. This series of tutorials provides a 360-degree view of the topic.

ANDROID DEVELOPER CHANNEL

youtube.com/androiddevelopers

Google's Android and Web Technology teams cross-pollinate. In many cases, keeping an eye on the state of Android provides a preview of what to expect from the web development team.

MIKE LOCKE

bit.ly/1SMCVcu

Google loves well-designed pages. Mike Locke's channel looks 'behind the scenes' – he doesn't focus on code, but on dishing out valuable career and UX design advice. These include 'Why Most Designers Fail at Freelance and Advice on How to Approach it' and much more.

GITHUB

Sometimes, looking at finished code is the fastest way to learn. Here are a few repositories with valuable content



GOOGLE CHROME

github.com/googlechrome

Google's Chrome team frequently open-sources all kinds of content. Visit the hub URL to receive an overview of all content hosted inside of the repository.

MATERIAL COMPONENTS

bit.ly/2yu1hYc

Should you feel like looking behind the CSS magic of the Material Design, simply take a look at the source code provided this GitHub repo.

AIRHORN SAMPLE

bit.ly/2sIPDLN

AirHorner is a simple and extremely annoying website. It derives its value as an example from the wide range of technologies used in its implementation.



Material Design has evolved into a project of its own. Visit the Material Design hub to find out more about the various implementations of the Design pattern



OTHER ESSENTIAL TOOLS AND RESOURCES

Sometimes, neither videos nor source code rock your boat. In that case, we also have a few 'classic' resources for you



17 COOL PLUGINS

bit.ly/2qbrdDe

Chrome can do a lot, but its functionality can be expanded. This site lists 17 plugins, which can turn out to be real timesavers.

GOOGLE WEBMASTER TOOLS

bit.ly/2CELJDO

While not the newest kid on the block, Google's Webmaster Tools still provide a set of reliable indicators, enabling you to fine-tune your website's performance.

GOOGLE SEARCH CONSOLE

bit.ly/2yxRxvW

Like its above-mentioned predecessor, the more modern search console offers info from Google's search algorithms.

GOOGLE DESIGN

design.google

Google's girth enables the company to dedicate significant resources to research. Get more information at this Design hub.

MATERIAL DESIGN HUB

material.io/

Material Design has since evolved into a project of its own. Visit the Material Design hub to find out more about the various implementations of the Design pattern.

GETTING STARTED WITH MATERIAL DESIGN

Google introduced Material Design to give a more consistent look and feel. Discover the basics to start building

Google's Material Design was initially seen as another tiresome redesign of the GUI stack. But, as time went by, users started to embrace the typography-centric design approach. Technologies like Progressive Web Apps required the presence of a JavaScript version. Given that JavaScript GUI stacks are pretty common, Google did not face a difficult task. The favourite way to include Material Design components involves npm. It's not simple, though, which is why we're starting out with a simple example:

```
<html>
<head>
<link rel="stylesheet" href="https://unpkg.com/
material-components-web@latest/dist/material-
components-web.min.css">
<script src="https://unpkg.com/material-
components-web@latest/dist/material-components-
web.min.js"></script>
</head>
</html>
```

In the next step, test framework functionality by adding a button to the body of the page:

```
<body>
<button class="tams-button mdc-button">Button</
button>
<script>
mdc.ripple.MDCRipple.attachTo(document.
querySelector('.tams-button'));
</script>
</body>
</html>
```

Material Design components behave like most other JavaScript GUI stacks. Developers add widgets to the page, CSS classes ensure the decoration gets deployed. Finally, a JavaScript invocation enforces the attachment of event handlers.

GO FULL FRAME

Experienced developers only need one look at 'mdc.ripple.MDCRipple' to see the source of the problem. The JavaScript and CSS files we included come out of a complex project. The sanest way involves loading the entire framework to your machine. First of all, obtain the toolchain and deploy it into a Node project:

```
tamhan@tamhan-thinkpad:~/materialsplace$ npm
init
tamhan@tamhan-thinkpad:~/materialsplace$ npm
install --save-dev webpack@3 webpack-dev-
server@2 css-loader sass-loader node-sass
extract-loader file-loader
tamhan@tamhan-thinkpad:~/materialsplace$ npm
```

```
install --save-dev autoprefixer postcss-loader
tamhan@tamhan-thinkpad:~/materialsplace$ npm i
--save-dev babel-core@6 babel-loader@7
babel-preset-es2015 babel-plugin-transform-
object-assign
```

The actual deployment of our modules takes place via WebPack. Open 'Package.json' and add the following markup to set up the workflow:

```
...
"main": "index.js",
"scripts": {
  "start": "webpack-dev-server"
},
...
```

WebPack controls itself via a file called 'webpack.config.js'. Create it in the folder containing 'package.json' – a full installation managing all features looks like this:

```
const autoprefixer = require('autoprefixer');
module.exports = {
  entry: ['./app.scss', './app.js'],
  output: {
    filename: 'bundle.js',
  },
  module: {
    rules: [
      {
        test: /\.scss$/,
        use: [
          {
            loader: 'file-loader',
            options: {
              name: 'bundle.css',
            },
          },
          {
            loader: 'extract-loader',
          },
          {
            loader: 'css-loader',
          },
          {
            loader: 'postcss-loader',
            options: {
              plugins: () => [autoprefixer()],
            },
          },
          {
            loader: 'sass-loader',
            options: {
              includePaths: ['./node_modules'],
            },
          },
        ],
      },
      {
        test: /\.js$/,
        loader: 'babel-loader',
        query: {
          presets: ['es2015'],
          plugins: ['transform-object-assign']
        },
      }
    ],
  }
}
```

WebPack transforms code by chasing it through a sequence of processors, each of which can take one or more parameters. In our case, both CSS

and JavaScript items are to be processed. In theory, the WebPack stack is ready to run at this point – sadly, the compile process will fail due to missing input resources (see figure), like this:

```
tamhan@tamhan-thinkpad:~/materialsplace$ npm
install --save-dev autoprefixer postcss-loader
tamhan@tamhan-thinkpad:~/materialsplace$ webpack-dev-server
Project is running at http://localhost:8080/
Watch content for changes.
Version: webpack 3.12.0
Time: 304ms
chunk Names
 0  multi [webpack]-server/client/main.js 0 (built)
 1  bundle.js 0 (built)
 2  bundle.css 0 (built)
 3  bundle.js.map 0 (built)
 4  bundle.js.map 0 (built)
 5  bundle.js.map 0 (built)
 6  bundle.js.map 0 (built)
 7  bundle.js.map 0 (built)
 8  bundle.js.map 0 (built)
 9  bundle.js.map 0 (built)
10  bundle.js.map 0 (built)
11  bundle.js.map 0 (built)
12  bundle.js.map 0 (built)
13  bundle.js.map 0 (built)
14  bundle.js.map 0 (built)
15  bundle.js.map 0 (built)
16  bundle.js.map 0 (built)
17  bundle.js.map 0 (built)
18  bundle.js.map 0 (built)
19  bundle.js.map 0 (built)
20  bundle.js.map 0 (built)
21  bundle.js.map 0 (built)
22  bundle.js.map 0 (built)
23  bundle.js.map 0 (built)
24  bundle.js.map 0 (built)
25  bundle.js.map 0 (built)
26  bundle.js.map 0 (built)
27  bundle.js.map 0 (built)
28  bundle.js.map 0 (built)
29  bundle.js.map 0 (built)
30  bundle.js.map 0 (built)
31  bundle.js.map 0 (built)
32  bundle.js.map 0 (built)
33  bundle.js.map 0 (built)
34  bundle.js.map 0 (built)
35  bundle.js.map 0 (built)
36  bundle.js.map 0 (built)
37  bundle.js.map 0 (built)
38  bundle.js.map 0 (built)
39  bundle.js.map 0 (built)
40  bundle.js.map 0 (built)
41  bundle.js.map 0 (built)
42  bundle.js.map 0 (built)
43  bundle.js.map 0 (built)
44  bundle.js.map 0 (built)
45  bundle.js.map 0 (built)
46  bundle.js.map 0 (built)
47  bundle.js.map 0 (built)
48  bundle.js.map 0 (built)
49  bundle.js.map 0 (built)
50  bundle.js.map 0 (built)
51  bundle.js.map 0 (built)
52  bundle.js.map 0 (built)
53  bundle.js.map 0 (built)
54  bundle.js.map 0 (built)
55  bundle.js.map 0 (built)
56  bundle.js.map 0 (built)
57  bundle.js.map 0 (built)
58  bundle.js.map 0 (built)
59  bundle.js.map 0 (built)
60  bundle.js.map 0 (built)
61  bundle.js.map 0 (built)
62  bundle.js.map 0 (built)
63  bundle.js.map 0 (built)
64  bundle.js.map 0 (built)
65  bundle.js.map 0 (built)
66  bundle.js.map 0 (built)
67  bundle.js.map 0 (built)
68  bundle.js.map 0 (built)
69  bundle.js.map 0 (built)
70  bundle.js.map 0 (built)
71  bundle.js.map 0 (built)
72  bundle.js.map 0 (built)
73  bundle.js.map 0 (built)
74  bundle.js.map 0 (built)
75  bundle.js.map 0 (built)
76  bundle.js.map 0 (built)
77  bundle.js.map 0 (built)
78  bundle.js.map 0 (built)
79  bundle.js.map 0 (built)
80  bundle.js.map 0 (built)
81  bundle.js.map 0 (built)
82  bundle.js.map 0 (built)
83  bundle.js.map 0 (built)
84  bundle.js.map 0 (built)
85  bundle.js.map 0 (built)
86  bundle.js.map 0 (built)
87  bundle.js.map 0 (built)
88  bundle.js.map 0 (built)
89  bundle.js.map 0 (built)
90  bundle.js.map 0 (built)
91  bundle.js.map 0 (built)
92  bundle.js.map 0 (built)
93  bundle.js.map 0 (built)
94  bundle.js.map 0 (built)
95  bundle.js.map 0 (built)
96  bundle.js.map 0 (built)
97  bundle.js.map 0 (built)
98  bundle.js.map 0 (built)
99  bundle.js.map 0 (built)
100  bundle.js.map 0 (built)
101  bundle.js.map 0 (built)
102  bundle.js.map 0 (built)
103  bundle.js.map 0 (built)
104  bundle.js.map 0 (built)
105  bundle.js.map 0 (built)
106  bundle.js.map 0 (built)
107  bundle.js.map 0 (built)
108  bundle.js.map 0 (built)
109  bundle.js.map 0 (built)
110  bundle.js.map 0 (built)
111  bundle.js.map 0 (built)
112  bundle.js.map 0 (built)
113  bundle.js.map 0 (built)
114  bundle.js.map 0 (built)
115  bundle.js.map 0 (built)
116  bundle.js.map 0 (built)
117  bundle.js.map 0 (built)
118  bundle.js.map 0 (built)
119  bundle.js.map 0 (built)
120  bundle.js.map 0 (built)
121  bundle.js.map 0 (built)
122  bundle.js.map 0 (built)
123  bundle.js.map 0 (built)
124  bundle.js.map 0 (built)
125  bundle.js.map 0 (built)
126  bundle.js.map 0 (built)
127  bundle.js.map 0 (built)
128  bundle.js.map 0 (built)
129  bundle.js.map 0 (built)
130  bundle.js.map 0 (built)
131  bundle.js.map 0 (built)
132  bundle.js.map 0 (built)
133  bundle.js.map 0 (built)
134  bundle.js.map 0 (built)
135  bundle.js.map 0 (built)
136  bundle.js.map 0 (built)
137  bundle.js.map 0 (built)
138  bundle.js.map 0 (built)
139  bundle.js.map 0 (built)
140  bundle.js.map 0 (built)
141  bundle.js.map 0 (built)
142  bundle.js.map 0 (built)
143  bundle.js.map 0 (built)
144  bundle.js.map 0 (built)
145  bundle.js.map 0 (built)
146  bundle.js.map 0 (built)
147  bundle.js.map 0 (built)
148  bundle.js.map 0 (built)
149  bundle.js.map 0 (built)
150  bundle.js.map 0 (built)
151  bundle.js.map 0 (built)
152  bundle.js.map 0 (built)
153  bundle.js.map 0 (built)
154  bundle.js.map 0 (built)
155  bundle.js.map 0 (built)
156  bundle.js.map 0 (built)
157  bundle.js.map 0 (built)
158  bundle.js.map 0 (built)
159  bundle.js.map 0 (built)
160  bundle.js.map 0 (built)
161  bundle.js.map 0 (built)
162  bundle.js.map 0 (built)
163  bundle.js.map 0 (built)
164  bundle.js.map 0 (built)
165  bundle.js.map 0 (built)
166  bundle.js.map 0 (built)
167  bundle.js.map 0 (built)
168  bundle.js.map 0 (built)
169  bundle.js.map 0 (built)
170  bundle.js.map 0 (built)
171  bundle.js.map 0 (built)
172  bundle.js.map 0 (built)
173  bundle.js.map 0 (built)
174  bundle.js.map 0 (built)
175  bundle.js.map 0 (built)
176  bundle.js.map 0 (built)
177  bundle.js.map 0 (built)
178  bundle.js.map 0 (built)
179  bundle.js.map 0 (built)
180  bundle.js.map 0 (built)
181  bundle.js.map 0 (built)
182  bundle.js.map 0 (built)
183  bundle.js.map 0 (built)
184  bundle.js.map 0 (built)
185  bundle.js.map 0 (built)
186  bundle.js.map 0 (built)
187  bundle.js.map 0 (built)
188  bundle.js.map 0 (built)
189  bundle.js.map 0 (built)
190  bundle.js.map 0 (built)
191  bundle.js.map 0 (built)
192  bundle.js.map 0 (built)
193  bundle.js.map 0 (built)
194  bundle.js.map 0 (built)
195  bundle.js.map 0 (built)
196  bundle.js.map 0 (built)
197  bundle.js.map 0 (built)
198  bundle.js.map 0 (built)
199  bundle.js.map 0 (built)
200  bundle.js.map 0 (built)
201  bundle.js.map 0 (built)
202  bundle.js.map 0 (built)
203  bundle.js.map 0 (built)
204  bundle.js.map 0 (built)
205  bundle.js.map 0 (built)
206  bundle.js.map 0 (built)
207  bundle.js.map 0 (built)
208  bundle.js.map 0 (built)
209  bundle.js.map 0 (built)
210  bundle.js.map 0 (built)
211  bundle.js.map 0 (built)
212  bundle.js.map 0 (built)
213  bundle.js.map 0 (built)
214  bundle.js.map 0 (built)
215  bundle.js.map 0 (built)
216  bundle.js.map 0 (built)
217  bundle.js.map 0 (built)
218  bundle.js.map 0 (built)
219  bundle.js.map 0 (built)
220  bundle.js.map 0 (built)
221  bundle.js.map 0 (built)
222  bundle.js.map 0 (built)
223  bundle.js.map 0 (built)
224  bundle.js.map 0 (built)
225  bundle.js.map 0 (built)
226  bundle.js.map 0 (built)
227  bundle.js.map 0 (built)
228  bundle.js.map 0 (built)
229  bundle.js.map 0 (built)
230  bundle.js.map 0 (built)
231  bundle.js.map 0 (built)
232  bundle.js.map 0 (built)
233  bundle.js.map 0 (built)
234  bundle.js.map 0 (built)
235  bundle.js.map 0 (built)
236  bundle.js.map 0 (built)
237  bundle.js.map 0 (built)
238  bundle.js.map 0 (built)
239  bundle.js.map 0 (built)
240  bundle.js.map 0 (built)
241  bundle.js.map 0 (built)
242  bundle.js.map 0 (built)
243  bundle.js.map 0 (built)
244  bundle.js.map 0 (built)
245  bundle.js.map 0 (built)
246  bundle.js.map 0 (built)
247  bundle.js.map 0 (built)
248  bundle.js.map 0 (built)
249  bundle.js.map 0 (built)
250  bundle.js.map 0 (built)
251  bundle.js.map 0 (built)
252  bundle.js.map 0 (built)
253  bundle.js.map 0 (built)
254  bundle.js.map 0 (built)
255  bundle.js.map 0 (built)
256  bundle.js.map 0 (built)
257  bundle.js.map 0 (built)
258  bundle.js.map 0 (built)
259  bundle.js.map 0 (built)
260  bundle.js.map 0 (built)
261  bundle.js.map 0 (built)
262  bundle.js.map 0 (built)
263  bundle.js.map 0 (built)
264  bundle.js.map 0 (built)
265  bundle.js.map 0 (built)
266  bundle.js.map 0 (built)
267  bundle.js.map 0 (built)
268  bundle.js.map 0 (built)
269  bundle.js.map 0 (built)
270  bundle.js.map 0 (built)
271  bundle.js.map 0 (built)
272  bundle.js.map 0 (built)
273  bundle.js.map 0 (built)
274  bundle.js.map 0 (built)
275  bundle.js.map 0 (built)
276  bundle.js.map 0 (built)
277  bundle.js.map 0 (built)
278  bundle.js.map 0 (built)
279  bundle.js.map 0 (built)
280  bundle.js.map 0 (built)
281  bundle.js.map 0 (built)
282  bundle.js.map 0 (built)
283  bundle.js.map 0 (built)
284  bundle.js.map 0 (built)
285  bundle.js.map 0 (built)
286  bundle.js.map 0 (built)
287  bundle.js.map 0 (built)
288  bundle.js.map 0 (built)
289  bundle.js.map 0 (built)
290  bundle.js.map 0 (built)
291  bundle.js.map 0 (built)
292  bundle.js.map 0 (built)
293  bundle.js.map 0 (built)
294  bundle.js.map 0 (built)
295  bundle.js.map 0 (built)
296  bundle.js.map 0 (built)
297  bundle.js.map 0 (built)
298  bundle.js.map 0 (built)
299  bundle.js.map 0 (built)
300  bundle.js.map 0 (built)
301  bundle.js.map 0 (built)
302  bundle.js.map 0 (built)
303  bundle.js.map 0 (built)
304  bundle.js.map 0 (built)
305  bundle.js.map 0 (built)
306  bundle.js.map 0 (built)
307  bundle.js.map 0 (built)
308  bundle.js.map 0 (built)
309  bundle.js.map 0 (built)
310  bundle.js.map 0 (built)
311  bundle.js.map 0 (built)
312  bundle.js.map 0 (built)
313  bundle.js.map 0 (built)
314  bundle.js.map 0 (built)
315  bundle.js.map 0 (built)
316  bundle.js.map 0 (built)
317  bundle.js.map 0 (built)
318  bundle.js.map 0 (built)
319  bundle.js.map 0 (built)
320  bundle.js.map 0 (built)
321  bundle.js.map 0 (built)
322  bundle.js.map 0 (built)
323  bundle.js.map 0 (built)
324  bundle.js.map 0 (built)
325  bundle.js.map 0 (built)
326  bundle.js.map 0 (built)
327  bundle.js.map 0 (built)
328  bundle.js.map 0 (built)
329  bundle.js.map 0 (built)
330  bundle.js.map 0 (built)
331  bundle.js.map 0 (built)
332  bundle.js.map 0 (built)
333  bundle.js.map 0 (built)
334  bundle.js.map 0 (built)
335  bundle.js.map 0 (built)
336  bundle.js.map 0 (built)
337  bundle.js.map 0 (built)
338  bundle.js.map 0 (built)
339  bundle.js.map 0 (built)
340  bundle.js.map 0 (built)
341  bundle.js.map 0 (built)
342  bundle.js.map 0 (built)
343  bundle.js.map 0 (built)
344  bundle.js.map 0 (built)
345  bundle.js.map 0 (built)
346  bundle.js.map 0 (built)
347  bundle.js.map 0 (built)
348  bundle.js.map 0 (built)
349  bundle.js.map 0 (built)
350  bundle.js.map 0 (built)
351  bundle.js.map 0 (built)
352  bundle.js.map 0 (built)
353  bundle.js.map 0 (built)
354  bundle.js.map 0 (built)
355  bundle.js.map 0 (built)
356  bundle.js.map 0 (built)
357  bundle.js.map 0 (built)
358  bundle.js.map 0 (built)
359  bundle.js.map 0 (built)
360  bundle.js.map 0 (built)
361  bundle.js.map 0 (built)
362  bundle.js.map 0 (built)
363  bundle.js.map 0 (built)
364  bundle.js.map 0 (built)
365  bundle.js.map 0 (built)
366  bundle.js.map 0 (built)
367  bundle.js.map 0 (built)
368  bundle.js.map 0 (built)
369  bundle.js.map 0 (built)
370  bundle.js.map 0 (built)
371  bundle.js.map 0 (built)
372  bundle.js.map 0 (built)
373  bundle.js.map 0 (built)
374  bundle.js.map 0 (built)
375  bundle.js.map 0 (built)
376  bundle.js.map 0 (built)
377  bundle.js.map 0 (built)
378  bundle.js.map 0 (built)
379  bundle.js.map 0 (built)
380  bundle.js.map 0 (built)
381  bundle.js.map 0 (built)
382  bundle.js.map 0 (built)
383  bundle.js.map 0 (built)
384  bundle.js.map 0 (built)
385  bundle.js.map 0 (built)
386  bundle.js.map 0 (built)
387  bundle.js.map 0 (built)
388  bundle.js.map 0 (built)
389  bundle.js.map 0 (built)
390  bundle.js.map 0 (built)
391  bundle.js.map 0 (built)
392  bundle.js.map 0 (built)
393  bundle.js.map 0 (built)
394  bundle.js.map 0 (built)
395  bundle.js.map 0 (built)
396  bundle.js.map 0 (built)
397  bundle.js.map 0 (built)
398  bundle.js.map 0 (built)
399  bundle.js.map 0 (built)
400  bundle.js.map 0 (built)
401  bundle.js.map 0 (built)
402  bundle.js.map 0 (built)
403  bundle.js.map 0 (built)
404  bundle.js.map 0 (built)
405  bundle.js.map 0 (built)
406  bundle.js.map 0 (built)
407  bundle.js.map 0 (built)
408  bundle.js.map 0 (built)
409  bundle.js.map 0 (built)
410  bundle.js.map 0 (built)
411  bundle.js.map 0 (built)
412  bundle.js.map 0 (built)
413  bundle.js.map 0 (built)
414  bundle.js.map 0 (built)
415  bundle.js.map 0 (built)
416  bundle.js.map 0 (built)
417  bundle.js.map 0 (built)
418  bundle.js.map 0 (built)
419  bundle.js.map 0 (built)
420  bundle.js.map 0 (built)
421  bundle.js.map 0 (built)
422  bundle.js.map 0 (built)
423  bundle.js.map 0 (built)
424  bundle.js.map 0 (built)
425  bundle.js.map 0 (built)
426  bundle.js.map 0 (built)
427  bundle.js.map 0 (built)
428  bundle.js.map 0 (built)
429  bundle.js.map 0 (built)
430  bundle.js.map 0 (built)
431  bundle.js.map 0 (built)
432  bundle.js.map 0 (built)
433  bundle.js.map 0 (built)
434  bundle.js.map 0 (built)
435  bundle.js.map 0 (built)
436  bundle.js.map 0 (built)
437  bundle.js.map 0 (built)
438  bundle.js.map 0 (built)
439  bundle.js.map 0 (built)
440  bundle.js.map 0 (built)
441  bundle.js.map 0 (built)
442  bundle.js.map 0 (built)
443  bundle.js.map 0 (built)
444  bundle.js.map 0 (built)
445  bundle.js.map 0 (built)
446  bundle.js.map 0 (built)
447  bundle.js.map 0 (built)
448  bundle.js.map 0 (built)
449  bundle.js.map 0 (built)
450  bundle.js.map 0 (built)
451  bundle.js.map 0 (built)
452  bundle.js.map 0 (built)
453  bundle.js.map 0 (built)
454  bundle.js.map 0 (built)
455  bundle.js.map 0 (built)
456  bundle.js.map 0 (built)
457  bundle.js.map 0 (built)
458  bundle.js.map 0 (built)
459  bundle.js.map 0 (built)
460  bundle.js.map 0 (built)
461  bundle.js.map 0 (built)
462  bundle.js.map 0 (built)
463  bundle.js.map 0 (built)
464  bundle.js.map 0 (built)
465  bundle.js.map 0 (built)
466  bundle.js.map 0 (built)
467  bundle.js.map 0 (built)
468  bundle.js.map 0 (built)
469  bundle.js.map 0 (built)
470  bundle.js.map 0 (built)
471  bundle.js.map 0 (built)
472  bundle.js.map 0 (built)
473  bundle.js.map 0 (built)
474  bundle.js.map 0 (built)
475  bundle.js.map 0 (built)
476  bundle.js.map 0 (built)
477  bundle.js.map 0 (built)
478  bundle.js.map 0 (built)
479  bundle.js.map 0 (built)
480  bundle.js.map 0 (built)
481  bundle.js.map 0 (built)
482  bundle.js.map 0 (built)
483  bundle.js.map 0 (built)
484  bundle.js.map 0 (built)
485  bundle.js.map 0 (built)
486  bundle.js.map 0 (built)
487  bundle.js.map 0 (built)
488  bundle.js.map 0 (built)
489  bundle.js.map 0 (built)
490  bundle.js.map 0 (built)
491  bundle.js.map 0 (built)
492  bundle.js.map 0 (built)
493  bundle.js.map 0 (built)
494  bundle.js.map 0 (built)
495  bundle.js.map 0 (built)
496  bundle.js.map 0 (built)
497  bundle.js.map 0 (built)
498  bundle.js.map 0 (built)
499  bundle.js.map 0 (built)
500  bundle.js.map 0 (built)
501  bundle.js.map 0 (built)
502  bundle.js.map 0 (built)
503  bundle.js.map 0 (built)
504  bundle.js.map 0 (built)
505  bundle.js.map 0 (built)
506  bundle.js.map 0 (built)
507  bundle.js.map 0 (built)
508  bundle.js.map 0 (built)
509  bundle.js.map 0 (built)
510  bundle.js.map 0 (built)
511  bundle.js.map 0 (built)
512  bundle.js.map 0 (built)
513  bundle.js.map 0 (built)
514  bundle.js.map 0 (built)
515  bundle
```

```
tamhan@tamhan-thinkpad:~/materialspace$ npm start
```

ADD WIDGETS

WebPack eliminates ‘unneeded’ parts from the compiled output. Adding widgets starts out in ‘app.scss’, which must now look like this:

```
@import "@material/button/mdc-button";
@import "@material/textfield/mdc-text-field";
@import "@material/line-ripple/mdc-line-ripple";
@import "@material/floating-label/mdc-floating-label";
```

In principle, we need one ‘include’ for each component our program plans to load. As both textbox and button come with a few ‘colleagues’, we need a total of four lines worth of ‘includes’. The ripple effect used before is even more complex – we will omit it for now.

Next, touch a file named ‘index.html’. It acts as the main entrypoint and must load the various script files emitted by the WebPack process:

```
<html>
  <head>
    <link rel="stylesheet" href="bundle.css">
  </head>
  <body>
    Hello World
    <script src="bundle.js" async></script>
  </body>
</html>
```

Finally, create a file called ‘app.js’. At this point, re-enter npm run to verify that the configuration still works. WebPack emits an URL similar to `http://localhost:8080`. The packaged results can be downloaded via a server living in npm. Don’t forget, though, to restart the server after changing files, since WebPack does not pick up changes.

CREATE THY WIDGETS

Now that the server is running, return to ‘index.html’ and replace the Hello World message with a bit of markup:

```
<div class="mdc-text-field">
<input type="text" id="txtR" class="mdc-text-field__input">
<label class="mdc-floating-label" for="my-text-field">R</label>
<div class="mdc-line-ripple"></div>
</div><br>
<div class="mdc-text-field">
<input type="text" id="txtU" class="mdc-text-field__input">
<label class="mdc-floating-label" for="my-text-field">U</label>
<div class="mdc-line-ripple"></div>
</div><br>
<div class="mdc-text-field">
```

```
<input type="text" id="txtI" class="mdc-text-field__input">
<label class="mdc-floating-label" for="my-text-field">I</label>
<div class="mdc-line-ripple"></div>
</div><br>
<button class="mdc-button" id="cmdRun">Button</button>
```

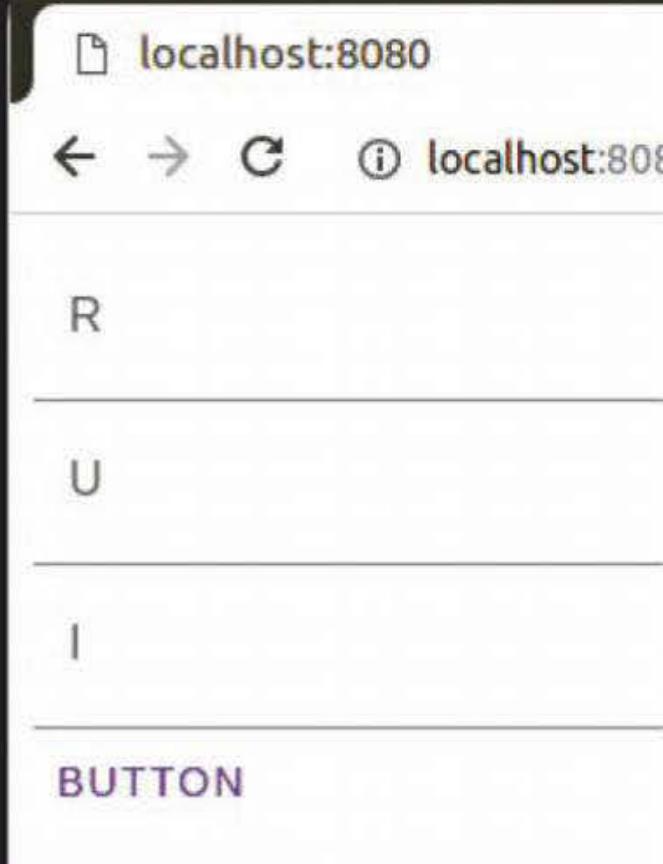
Just like in the case of our button, a text box also consists of a group of attributes. One of them displays a hint with information about what to enter, while the other one supports animation.

However, getting animations to work requires a change in ‘app.js’:

```
import {MDCTextField} from '@material/textfield';
const textField = new MDCTextField(document.querySelector('.mdc-text-field'));
```

This snippet of code is interesting since it targets all three text boxes in one swipe – Google designed the API to sweep over entire CSS selectors in one go.

At this point, our program is ready to run – it will present itself like the figure. Finalising our little resistor calculator requires you to add JavaScript event listeners. After all, MDC components are but fancy HTML5 ones...



FINAL WORD

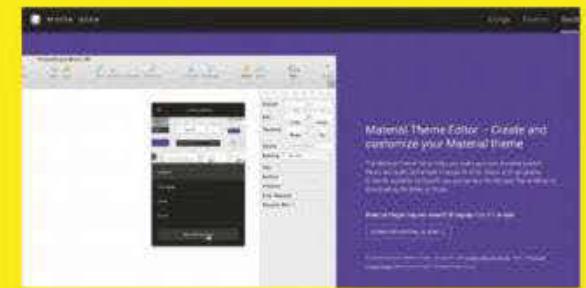
Integrating Material Design widgets into your application definitely takes a bit of work. But, the rewards are substantial and definitely worth the time. This is only the beginning. Be sure to check out material.io/develop for more tutorials, repos and information on what is available, as well as what can be done.

WHAT TOOLS ARE ON OFFER?

MATERIAL DESIGN IS MADE UP OF MANY DIFFERENT PARTS. CHECK OUT A FEW OF THE TOOLS TO MAKE USAGE EASIER

MATERIAL THEME EDITOR

material.io/tools/theme-editor/



Should you find yourself working on a Mac, take Theme Editor for a spin. This small but nifty program creates customised colour palettes, which you can use to change the look and feel of various Material Design-based systems.

ICONS

material.io/tools/icons



Material Design differs from Metro in that it also allows for the use of small symbols. Google provides developers with ample choice – simply visit the URL to pick icons until your heart’s content.

COLOR TOOL

material.io/tools/color



This tool allows users to create, share, and apply color palettes to a UI. Plus, it measures the accessibility level of any colour combination that you decide to put together.

PROGRESSIVE WEB APPS

The web design industry moves in cycles. Progressive Web Apps (PWAs) are big news. Here are a few resources to guide you through them

THE PWA CHECKLIST

LOAD ‘SMOOTHLY’

Native apps were lauded for their swift and nimble response to user input. Use single-page applications and status bars to replicate this experience on your PWA.

PROVIDE METADATA

Users can (and should) add PWAs to their devices' start screens. This requires the presence of a metadata file, whose contents regulate the presentation of content.

ALLOW DEEP LINKS

PWAs differ from native apps due to their more expanded ‘social’ features. One neat aspect involves ‘deep links’, which enable users to share parts of the application with friends and colleagues.

ANALYSE WITH LIGHTHOUSE

Space constraints prevent us from providing a complete list of all PWA to-do items. Lighthouse, found at developers.google.com/web/tools/lighthouse, automatises the checklists for you. It happily runs from the shell, from Node.js or from the Chrome Developer Tools.

WHAT ABOUT AMP?

AMPs – short for Accelerated Mobile Pages – is a completely different technology with a similar set of goals. While PWAs create rich user experiences, an AMP is a snippet of JavaScript-based content intended to show up in search results.

In principle, AMPs live off a subset of HTML, which contains expansions intended to accelerate rendering. For example, an AMP image contains additional properties. They let the rendering process complete before the actual image file is available. JavaScript is run by a dedicated library enforcing multithreading in a fashion not dissimilar to the one seen on

Lighthouse

Lighthouse is an open-source, automated tool for improving the quality of web pages. You can run it against any web page, public or requiring authentication. It has audits for performance, accessibility, progressive web apps, and more.

You can run Lighthouse in Chrome DevTools, from the command line, or as a Node module. You give Lighthouse a URL to audit, it runs a series of audits against the page, and then it generates a report on how well the page did. From there, use the failing audits as indicators on how to improve the page. Each audit has a reference doc explaining why the audit is important, as we



RUN LIGHTHOUSE IN CHROME DEVTOOLS

Check out the video below from Google I/O 2017 to learn more about how to use Lighthouse.

NEEDS HTTPS

Ideally PWAs must be served over HTTPS. This prevents annoying man-in-the-middle attacks, which happen frequently on public WiFi networks

the Metro platform. Finally, an optional CDN caching system by Google ensures that content gets delivered with minimal delay.

Implementing AMPs does bring benefits in mobile search rankings – if your site is SEO-dependant, investing some effort might pay off.

WHAT IS FIREBASE?

While providing storage, initially, is not a problem, fast website growth usually leads to scaling problems. Fortunately, deploying Google's Firebase storage service is a quick way to work around the problem. In addition, Google also provides market-leading authentication,

cloud messaging and real-time data management solutions.

Deploying Google's authentication service is beneficial in that it unburdens end users: instead of having to remember multiple user accounts for each site, a Google or Facebook account can be used in a cross-service fashion. Google can also take care of phone number verification and similar problems for a modest fee. The well-known cloud messaging feature should not be new to Android-experienced developers: the API can also be accessed from the web, thereby providing a degree of feature parity to PWAs. Getting started with Firebase in JavaScript is not difficult. Visit bit.ly/2AsPNpq to find out more.

“ While PWAs are commonly associated with mobile phones, this is a misconception. Many – if not most – of the benefits of the technology also apply when a PWA runs on a desktop computer ”

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



Graphic design

Web design

3D

Digital art

www.creativebloq.com

web workshop

Code preview hover effects

Inspired by epoch476.com

EXPERT ADVICE

Hovering on success

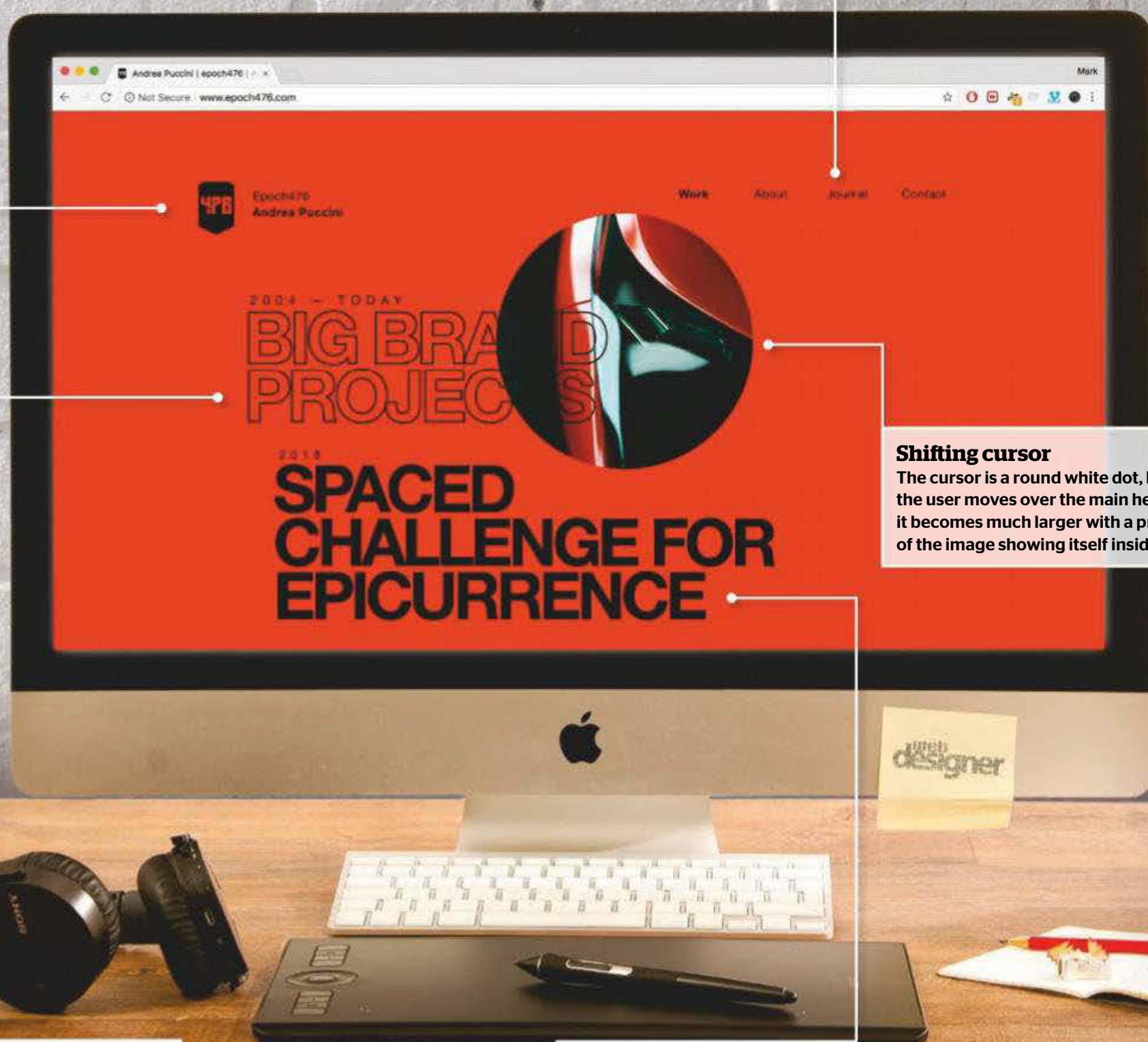
Andrea uses what is normally a simple hover technique to bring meaning to the content. Rather than the hover be just a visual effect it becomes a preview of the content that the user can then choose to view or not. The preview is a great idea, that is original, fluid and very interactive for the viewer.

Site title

The main site title and logo show the author's name of the portfolio and act as a 'home' button for the site.

Main menu

The menu allows access to the rest of the site with another shifting rollover effect that is equally as impressive as the main site content.



Shifting titles

As the user rolls over the headings the text shifts to the left, becomes larger and changes to an outline of the text.

Work headings

The work headings are the main links on the page, these links are what the text looks like when the mouse is not over the title.



<comment>

What our experts think of the site

Understanding user needs

Working on your own portfolio is always difficult. I'm a big fan of Massimo Vignelli; he used to focus on what clients needed not what they wanted. With a portfolio you are your own client, so: what do I need? I need a clear website which looks at the essence of things getting away with appearances

Andrea Puccini, Art Director

Technique

1. DOM elements

To add a similar hover effect to your own web page, you will need a similar structure in the body of your HTML page to what is shown here. The cursor and ball will follow the mouse, while the elements will expand as the mouse moves over them.

```
<div id="cursor" class="cursor">
    <div id="ball" class="ball"></div>
</div>
<div class="grid">
    <div class="container">
        <div class="element"
onmouseover="change('first')"
onmouseout="leave('first')">BIG BRAND
            <br>PROJECTS</div>
        <div class="element"
onmouseover="change('second')"
onmouseout="leave('second')">SPACED
            <br>CHALLENGE</div>
    </div>
</div>
```

2. Expand your horizons

To expand the elements, the following CSS is an easy way to do this. As you can see this has a 0.7-second transition which, when the mouse hovers over, triggers a movement to the left and the text scales up. Moving over and off the element causes it to expand and contract:

```
.element {
    font-size: 4vw;
    margin-bottom: 10%;
    transform: translate3d(0, 0, 0);
    transition: all 0.7s;
}
.element:hover {
    transform: translate3d(-10px, 0, 0)
scale(1.3);
    color: #ff3c00;
    text-shadow: -1px -1px 0 #000, 1px -1px 0
#000, -1px 1px 0 #000, 1px 1px 0 #000;
}
```

3. Mouse follower

The cursor and ball follow the mouse pointer around the screen. The cursor will follow the mouse exactly and the ball will expand and contract when it moves over the element. As such it has a transition, it will also get a different image in the background as well.

```
.cursor {
    position: absolute;
    width: 30px; height: 30px;
```

```
    transform-origin: 50% 50%;
    transform: translate3d(0, 0, 0);
}
.ball {
    width: 30px; height: 30px;
    border-radius: 50%;
    background: #fff;
    cursor: none;
    transition: all 0.5s;
```

4. Mouse expansion

These classes will be added to the 'ball' depending which element they are over. They scale the ball up and position a new image in the background of the ball to give it some context for the element that it's over.

```
.first {
    transform: scale(8);
    background: url(img/one.jpg) no-repeat;
    background-size: cover;
}
.second {
    transform: scale(8);
    background: url(img/two.jpg) no-repeat;
    background-size: cover;
}
```

5. Bringing it together

Now with the right CSS and HTML content in place it's time to add the behaviour through JavaScript. Add some

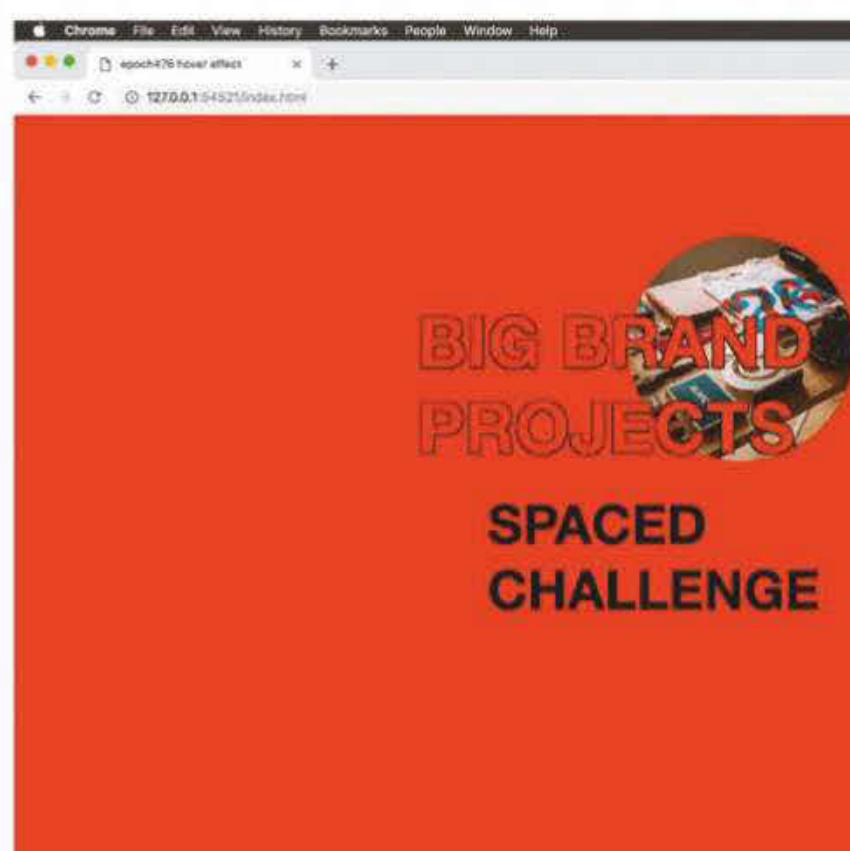
script tags to the end of the page and here the cursor is set to follow the mouse position around the document:

```
var cursor = document.
getElementById("cursor");
var ball = document.getElementById("ball");
var _x, _y;
document.onmousemove = function(e) {
    _x = e.clientX;
    _y = e.clientY;
}
function animate() {
    requestAnimationFrame(animate);
    cursor.style.left = (_x - 15) + 'px';
    cursor.style.top = (_y - 15) + 'px';
}
animate();
```

6. Changing content

As the mouse moves over and out of the elements the right classes need to be added to this. Here you can see that the classes are added when the mouse moves over and then removed from the class list when the mouse leaves. This completes the hover effect of epoch476:

```
function change(elem) {
    if (elem == 'first') {
        ball.classList.add('first');
    }
    if (elem == 'second') {
        ball.classList.add('second');
    }
}
function leave(elem) {
    if (elem == 'first') {
        ball.classList.remove('first');
    }
    if (elem == 'second') {
        ball.classList.remove('second');
    }
}
```



Create glitch text and image effects with CSS

CSS can still hold its own when it comes to advanced animated effects and this tutorial demonstrates its power



The range of special effects and animations that are found on websites often make them stand out and force an immediate impact before the user has had a chance to get into reading the main parts of the site. If you need a lot of wow on your homepage, CSS has tons of effects to offer that don't need the JavaScript coding of other more intense solutions on the canvas or WebGL. In this tutorial, CSS is going to show that quite advanced effects such as glitchy images and titles are all possible with a little effort going into creating keyframes to control some of the action. This is actually pretty similar to working with animation software when keyframes are placed into specific points. The difference with the keyframes in CSS is that they are written as percentages for the timeline of animation in the code. This isn't anywhere near as daunting as it sounds, once you try it, it becomes relatively straightforward to achieve good results using this. There will be a small amount of JavaScript used in the tutorial, just to remove the loading screen once the images load onto the page. The rest of the page will be built up using regular HTML and CSS, with the CSS Grid being used to position elements on the screen.

1. Starting the process

To get started, open the 'start' folder from the project files inside your code editor. Open the 'index.html' page, which contains just a barebones skeleton HTML page. In the head section, the fonts need linking up so that the design displays correctly. Add the style link for the fonts.

```
<link href="https://fonts.googleapis.com/css?family=IM+Fell+English|Playfair+Display:900" rel="stylesheet">
```

2. Linking the CSS

The basic layout in CSS has been started in the 'site.css' file, but all of the important parts relating to the glitch effect are going to be added later. In the head section of the document link up the CSS so that the basic design of the page is in place.

```
<link rel="stylesheet" type="text/css" href="css/site.css" />
```

3. Loading screen

Now move to the body section of the page. This contains all the visible elements of the page that show up in the browser. Here add in a div that will hold the 'preloader screen' until everything on the page has loaded. This will display a logo in the centre of the page.

```
<div id="loader" class="loading">
  <div class="loading-logo"></div>
</div>
```

4. Header bar

Along the top of the screen will be a small header containing an SVG logo for the site on the left with a text heading. Then on the right-hand side of the screen, an inline menu will be in place for easy navigation. The structure of the code here adds those elements onto the page.

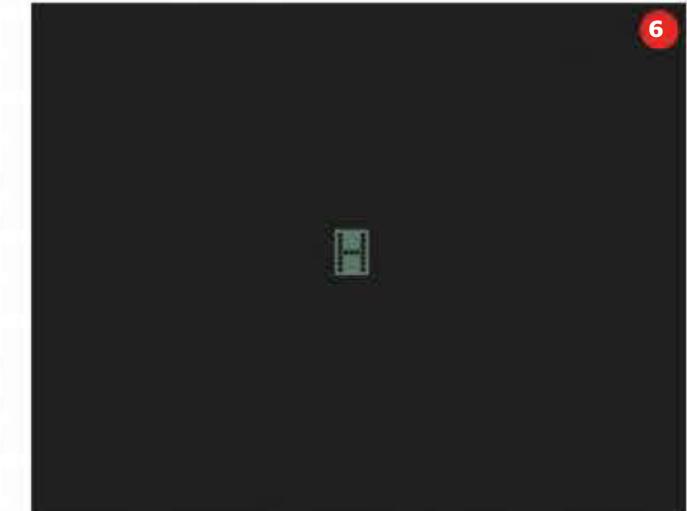


```
<div class="contentFixed">
  <header class="header">
    <h1 class="headerTitle"> HackerCon</h1>
  </header>
  <nav class="menu" id="siteNav">
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="news.html">News</a></li>
      <li><a href="contact.html">Contact</a></li>
      <li><a href="about.html">About</a></li>
    </ul>
  </nav>
</div>
```

5. Adding the images

Now the section that follows contains several versions of the same image in the 'glitchit' class. What this will do is have different parts of these images turned on and off at different times in order to give a glitch effect. Following this is the text that will sit over the top of the images.

```
<div class="content">
  <div class="glitch">
    <div class="glitchit"></div>
    <div class="glitchit"></div>
    <div class="glitchit"></div>
    <div class="glitchit"></div>
    <div class="glitchit"></div>
  </div>
  <div class="contentSection">
    <h2 class="contentTitle">Hacker<span>Con</span></h2>
    <p class="contentText">Add a description</p>
  </div>
</div>
```



6. Starting it off

At the end of the body content the JavaScript tags are placed. All this does is check that the page has loaded and then removes the loading screen, which in turn triggers the glitch effect animation to start by adding a class to the body to affect the relevant glitch sections.

```
<script>
  var loader = document.getElementById('loader');
  window.addEventListener("load", function(event) {
    loader.classList.remove('loading');
    loader.classList.add('loaded');
    document.body.classList.add('imgloaded');
  });
</script>
```

7. Setting up CSS Variables

Save the page now and move over to the 'site.css' file in

Why JavaScript?

If you need to detect some changes in states to the browser, JavaScript can come to the rescue and notify us when content has loaded so that we can add or remove classes in CSS.

Tutorials

Create glitch text and image effects with CSS

9

HackerCon

Cyber security conference, build your hacking skills, develop your network, tips, tricks and keynote speakers to hold off the cyber enemy.

the CSS folder. There is already code here, but right above any other code add in the variables shown below. These CSS Variables will hold colours and sizes that will be used later on in the design..

```
body {  
  --color-text: #fff;  
  --color-bg: #000;  
  --color-link: #555;  
  --color-link-hover: #98fadf;  
  --color-info: #f7cfb9;  
  --glitch-width: 100vw;  
  --glitch-height: 100vh;  
  --gap-horizontal: 10px;  
  --gap-vertical: 5px;  
  --time-anim: 4s;  
  --delay-anim: 2s;
```



With the background image added, the design is fully finished and the glitch effects can run on this content

Blending modes

If you've ever used any form of image manipulation app such as Photoshop, then you'll be right at home with blending modes. These change the way pixels are mixed with content below.

8. Remaining Variables

As you will see these Variables are assigned to the body tag so they can be used by any tag on the page inside the body, which is essentially all the visible page. Here the transparency and blending modes are set up for the different images. There are five images and you can experiment with these settings to get different results.

```
--blend-mode-1: none;  
--blend-mode-2: overlay;  
--blend-mode-3: none;  
--blend-mode-4: none;  
--blend-mode-5: overlay;  
--blend-color-1: transparent;  
--blend-color-2: #7d948e;  
--blend-color-3: transparent;  
--blend-color-4: transparent;  
--blend-color-5: #af4949;
```

9. Filling the screen with images

To keep the code neatly together, scroll down to the comment that marks steps 9 to 13 in the CSS file, adding in this code. Here the glitch code positions the div containing all images to fill the full screen and be positioned absolutely in the top-left of the screen. Note it gets the width and height from the CSS Variables.

```
.glitch {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: var(--glitch-width);  
  height: var(--glitch-height);  
  overflow: hidden;
```

10. Each glitch image

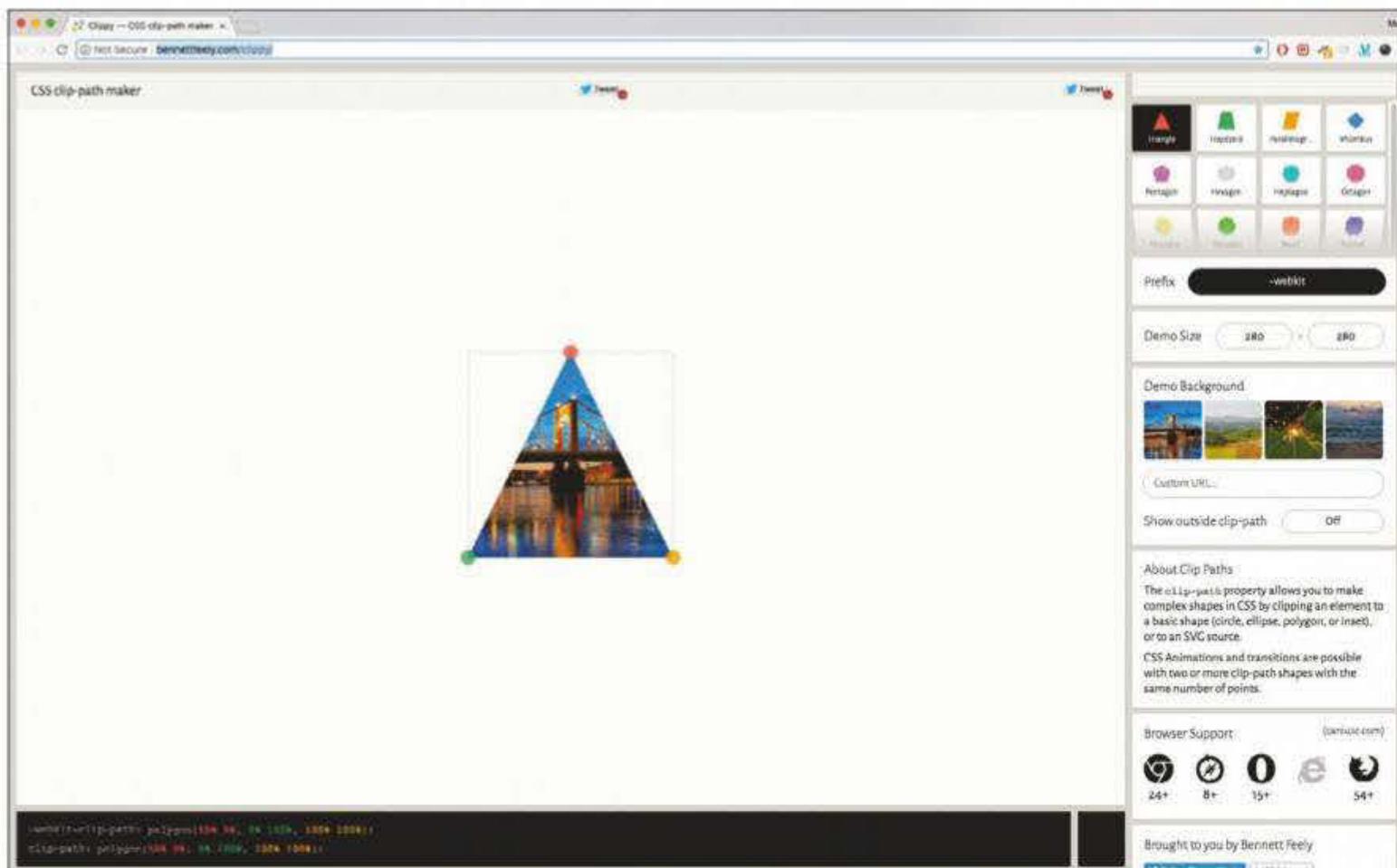
As the glitch effect is made up out of copies of the same image, this code positions each div in the page and makes it slightly larger than the screen. It positions it off the top and left to account for it being bigger, and the image is then placed in the background to fill the image.

```
.glitchit {  
  position: absolute;  
  top: calc(-1 * var(--gap-vertical));  
  left: calc(-1 * var(--gap-horizontal));  
  width: calc(100% + var(--gap-horizontal) *  
  2);  
  height: calc(100% + var(--gap-vertical) *  
  2);  
  background: url(..../img/main.jpg) no-repeat  
  50% 0;  
  background-color: var(--blend-color-1);  
  background-size: cover;  
  transform: translate3d(0, 0, 0);  
  background-blend-mode: var(--blend-  
  mode-1);  
}
```

11. Everything except the first image

The code here selects every image except the first image. This is because the first image stays on the page, while the others turn on and off over the top with the keyframe animation. These top images are hidden until they are needed with the opacity set to zero.

```
.glitchit:nth-child(n+2) {  
  opacity: 0;  
}  
.imgloaded .glitchit:nth-child(n+2) {  
  animation-duration: var(--time-anim);
```



Getting into clipping paths

If the thought of trying to create your own clipping path sounds a little daunting, then you are not alone. Fortunately there is an easy way to create your own clipping paths for CSS with the excellent Clippy online application. This can be found at <http://bennetfeely.com/clippy>. Here you can move handles around to get visual feedback for the kind of clipping path that you are trying to create. What's more, if the image you are trying to clip is already uploaded to your site, then you can enter the chosen URL and make the clipping path directly on it. There's a range of prebuilt shapes or you can make your own custom shape. The best part is that you can simply copy and paste the code right into your CSS so that you are ready to go!

```
animation-delay: var(--delay-anim);
animation-timing-function: linear;
animation-iteration-count: infinite;
}
```

12. Second and third image

The second and third image are set to animate in this code. They are given the respective blend and colour modes so that they show up differently. The biggest difference here is that they are given different keyframe animations to follow to mix up the effects.

```
.imgloaded .glitchit:nth-child(2) {
    background-color: var(--blend-color-2);
    background-blend-mode: var(--blend-
mode-2);
    animation-name: glitch-1;
}
.imgloaded .glitchit:nth-child(3) {
    background-color: var(--blend-color-3);
    background-blend-mode: var(--blend-
mode-3);
    animation-name: glitch-2;
}
```

13. Fourth and fifth image

This time the next two images are set up quite close to the others, but again this time there is different blending modes and animations for these images to show up. The keyframes have not yet been created for some of these animations and that will come next.

```
.imgloaded .glitchit:nth-child(4) {
    background-color: var(--blend-color-4);
    background-blend-mode: var(--blend-
mode-4);
    animation-name: glitch-3;
}
```

```
.imgloaded .glitchit:nth-child(5) {
    background-color: var(--blend-color-5);
    background-blend-mode: var(--blend-
mode-5);
    animation-name: glitch-flash;
}
```

```
horizontal), 0, 0);
clip-path: polygon(0 2%, 100% 2%,
100% 5%, 0 5%);
} 2% {
clip-path: polygon(0 15%, 100% 15%,
100% 15%, 0 15%);
}
```

14. The first set of keyframes

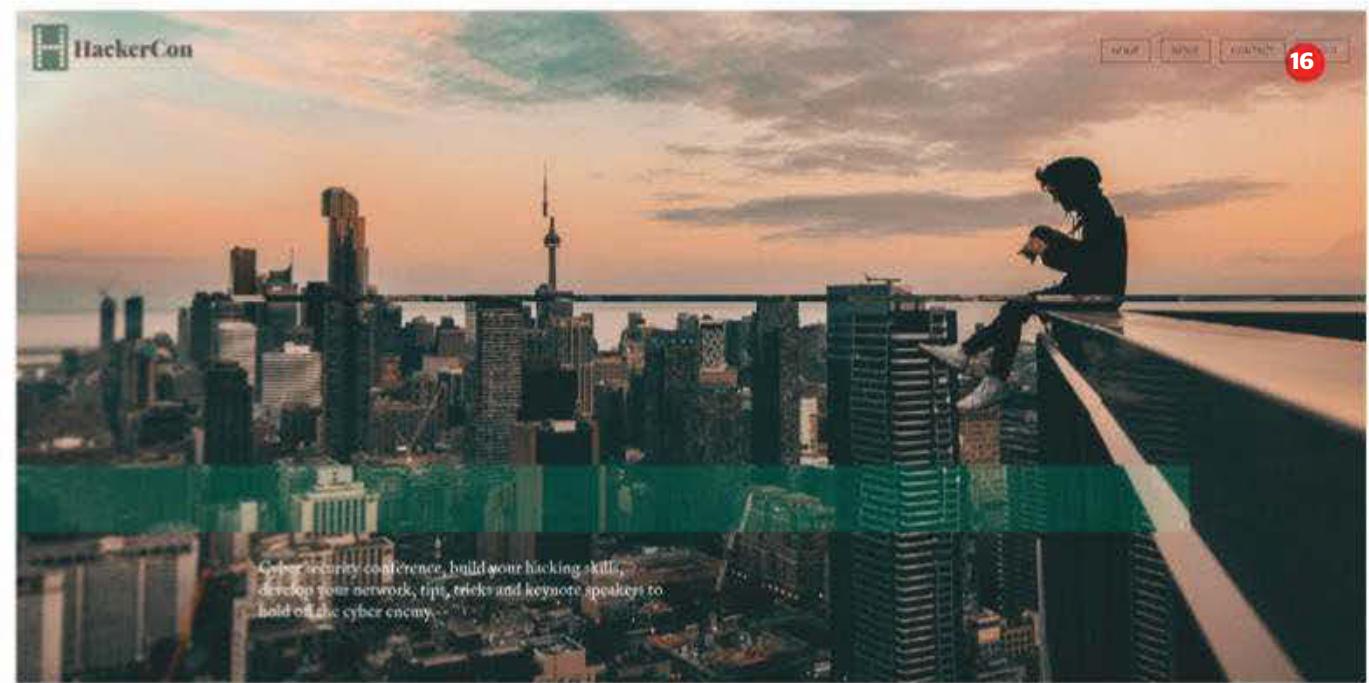
The keyframes work by grabbing different sections of the image and clipping it down so only that will be visible. The opacity is turned on and off at different times so that parts of the image are visible at different times to the other animations and thus creates the glitch effect. The image is moved slightly on the x-axis.

```
@keyframes glitch-1 {
0% {
    opacity: 1;
    transform: translate3d(var(--gap-
```

15. Using the clip-path

The clip path is taking a rectangle so the first two numbers are top-left, then top-right. This is followed by bottom-right and bottom-left. By moving these numbers different parts of the image become visible at different points.

```
4% {
clip-path: polygon(0 10%, 100% 10%,
100% 20%, 0 20%);
} 6% {
```



Tutorials

Create glitch text and image effects with CSS



The text glitch effect can be seen running here, and the text is being clipped so only part of it is visible. This only happens for a brief spell allowing the text to be readable on the screen.



With all elements of the CSS completed, the glitch effect runs every few seconds to give a distressed look to the image

```
clip-path: polygon(0 1%, 100% 1%,  
100% 2%, 0 2%);  
} 8% {  
    clip-path: polygon(0 33%, 100% 33%, 1  
00% 33%, 0 33%);  
} 10% {  
    clip-path: polygon(0 44%, 100% 44%,  
100% 44%, 0 44%);  
}
```

16. Fast movement

By moving the clip path so quickly over a number of the keyframes, the effect builds up and different parts of the image appear to flash around in an erratic fashion. Add that more layers of images are also doing this and the effect works very well at what it does.

```
12% {  
    clip-path: polygon(0 50%, 100% 50%,  
100% 20%, 0 20%);  
} 14% {  
    clip-path: polygon(0 70%, 100% 70%,  
100% 70%, 0 70%);  
} 16% {  
    clip-path: polygon(0 80%, 100% 80%,  
100% 80%, 0 80%);  
} 18% {  
    clip-path: polygon(0 50%, 100% 50%,  
100% 55%, 0 55%);  
} 20% {
```

```
        clip-path: polygon(0 70%, 100% 70%,  
100% 80%, 0 80%);  
    }
```

17. Reaching the end

After 22% the image is turned off until the animation plays back again. This completes the 'glitch-1' effect with 'glitch-2' and 'glitch-3' already being supplied in the code. The next section will glitch the text that is over the top of the image as well.

```
21.9% {  
    opacity: 1;  
    transform: translate3d(var(--gap-  
horizontal), 0, 0);  
} 22%, 100% {  
    opacity: 0;  
    transform: translate3d(0, 0, 0);  
    clip-path: polygon(0 0, 0 0, 0 0, 0  
0);  
}
```

```
100% 44%, 0 44%);  
}
```

19. Clipping crazy

The effect continues in this section of code by rapidly changing the shape of the clipping path. The clip-path also comes with the '-webkit-' prefix but for brevity this has not been shown in any of the code here. Clip path is currently not supported in IE, Edge or Opera Mini, but is in all other browsers, giving it 88% global support.

```
5% {  
    clip-path: polygon(0 50%, 100% 50%,  
100% 20%, 0 20%);  
} 6% {  
    clip-path: polygon(0 70%, 100% 70%,  
100% 70%, 0 70%);  
} 7% {  
    clip-path: polygon(0 80%, 100% 80%,  
100% 80%, 0 80%);  
} 8% {  
    clip-path: polygon(0 50%, 100% 50%,  
100% 55%, 0 55%);  
}
```

20. Flipping back

In the final text animation the text flips back to its original position and waits for the keyframes to come around again. As you can see all of the animation takes place in 10% while it stays dormant for 90% of the time, giving the text the right amount of distress and allowing it to be readable.

```
9% {  
    clip-path: polygon(0 70%, 100% 70%,  
100% 80%, 0 80%);  
} 9.9% {  
    transform: translate3d(calc(-1 *  
var(--gap-horizontal)), 0, 0) scale3d(-1, -1,  
1);  
} 10%, 100% {  
    transform: translate3d(0, 0, 0)  
scale3d(1, 1, 1);  
    clip-path: polygon(0 0, 100% 0, 100%  
100%, 0% 100%);  
}
```

18. Text glitch

This code works extremely similarly to the previous code except that it flips the text upside down and then clips it, to give a jumping effect that is moved dramatically. Following that the clip path is revealing only smaller sections, with the fast movement through the keyframes.

```
@keyframes glitch-text {  
    0% {  
        transform: translate3d(calc(-1 *  
var(--gap-horizontal)), 0, 0) scale3d(-1, -1,  
1);  
        clip-path: polygon(0 20%, 100% 20%,  
100% 21%, 0 21%);  
    } 2% {  
        clip-path: polygon(0 33%, 100% 33%,  
100% 33%, 0 33%);  
    } 4% {  
        clip-path: polygon(0 44%, 100% 44%,  
100% 44%, 0 44%);  
    }
```

21. Quick flash

The final step is that one image is given the glitch-flash animation, and this series of keyframes just place the image on the screen with a 20% opacity for a short period just to give an offset of the original. Save the CSS file now and the effect should start playing once the content loads in the browser.

```
@keyframes glitch-flash {  
    0%, 5% {  
        opacity: 0.2;  
        transform: translate3d(var(--gap-  
horizontal), var(--gap-vertical), 0);  
    } 5.5%, 100% {  
        opacity: 0;  
        transform: translate3d(0, 0, 0);  
    } }
```

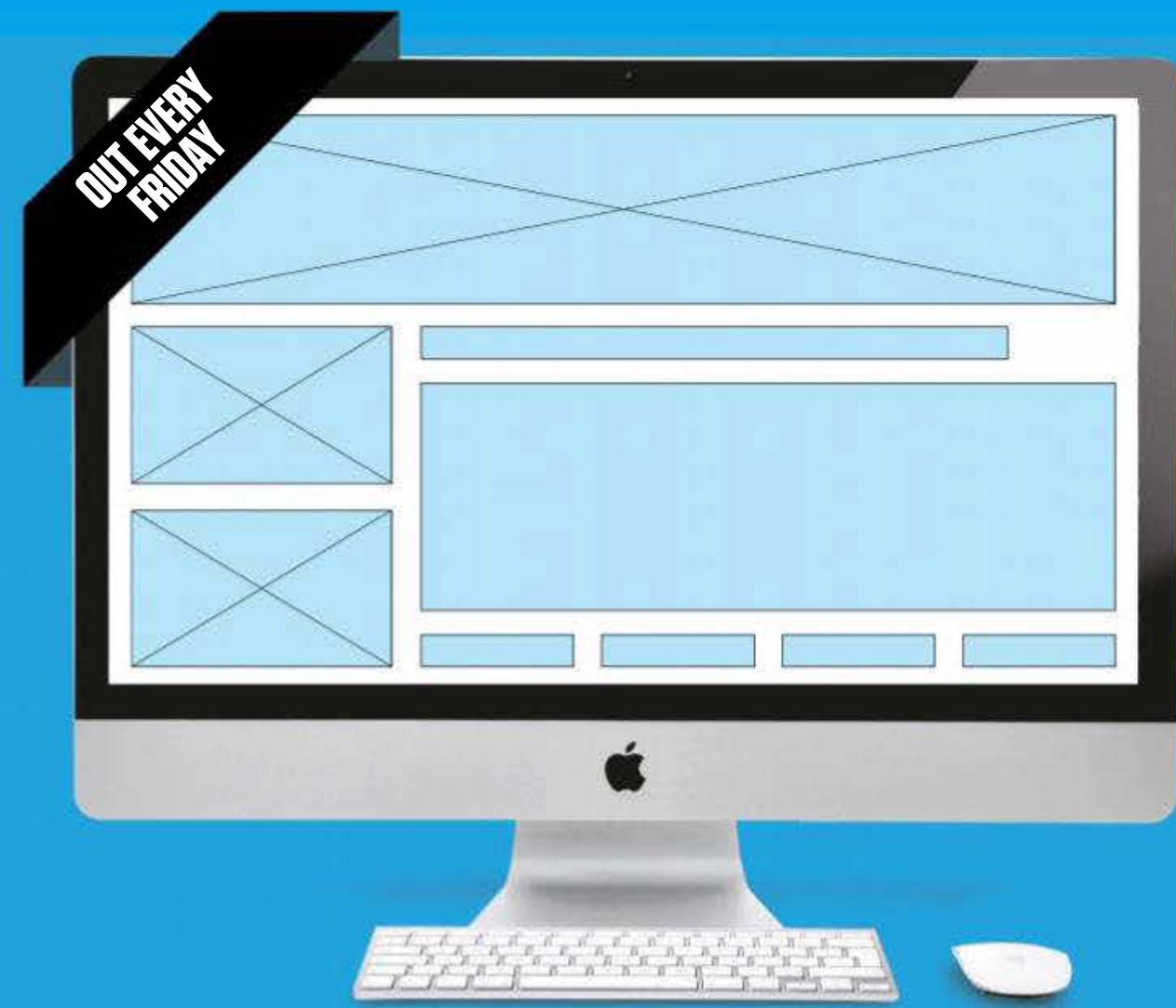
HEAR FROM THE PROS

SIGN UP TO THE

**web
designer**TM

NEWSLETTER TODAY!

OUT EVERY
FRIDAY



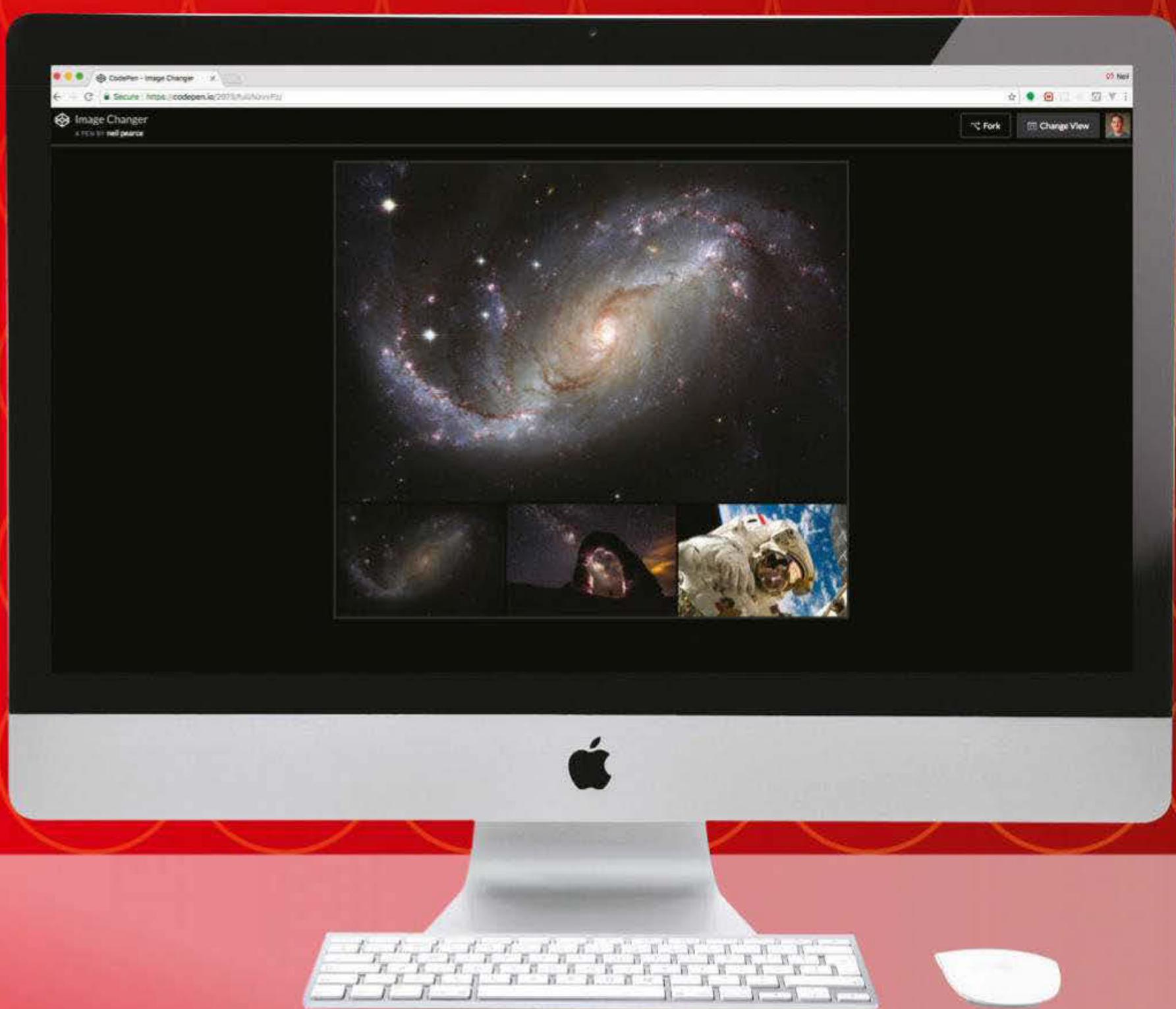
Get weekly news, tips & inspiration

SIGN UP NOW!

bit.ly/2KI5b7Y

Code an image changer with CSS

Use the power of CSS Grid and vanilla JavaScript to create a simple image changer





here are many image sliders or image carousels seen on the internet; some are developed as plugins and many rely heavily on jQuery.

As a front-end developer, you would find it hard to avoid using one when you develop your websites. But that doesn't mean we can't use one or even learn how to develop them, especially if they can be applied as a handy image changer.

Let's say you have an eCommerce website to build and you wanted a simple and easy way of changing images of a specific product. Or perhaps you just want a simple way to shift through your personal images on your portfolio or blog. Well, in this tutorial you will learn how to build this simple image changer using CSS Grid and vanilla JavaScript.

We will be using CodePen throughout as it's easier to set things up, so if you haven't got a free CodePen account yet, quickly go and set one up now. Then open up your <https://codepen.io> account and let's get started!

1. Get set up

To begin with, open up CodePen and create a new pen. In the CSS section, go ahead and change the settings to SCSS as the preprocessor and select normalise as the base reset. Then in the JS section, change the settings to Babel so that we can use the ES6 syntax.

2. Add the HTML

In the HTML section of our pen, we can begin proceedings by adding in a container class. We will be using this as a wrapper. And then in between that, we will be adding our main image. With this in mind, it would make sense to give this a class name of 'main--img'.

```
<div class="container">
  <div class="main--img">
    </div>
  </div>
```

3. Bring in images

The images we will use are going to be stored on a free image hosting and sharing website: <https://image.ibb.co> - it's free to sign up and it's handy for when you want to add images to your pens or to do rapid prototyping. The main image will, of course, be the active image, so make sure you add a class called active to the link. What we will do later on is use JavaScript to target that active image and save it as the current main image.

```
<div class="main--img">
  
</div>
<div class="images">
  
  
  
</div>
```

4. Initial CSS

The HTML section is now finished and we need to begin

adding in the CSS. The first thing we need to do is set the default styles to the body and container. The container will sit centre to the browser window with a maximum width of 900px and we will centre it using margin: auto.

```
body {
  margin: 25px;
  padding: 0;
  background: #000;
}
.container {
  max-width: 900px;
  margin: auto;
  border: 4px solid #333;
```

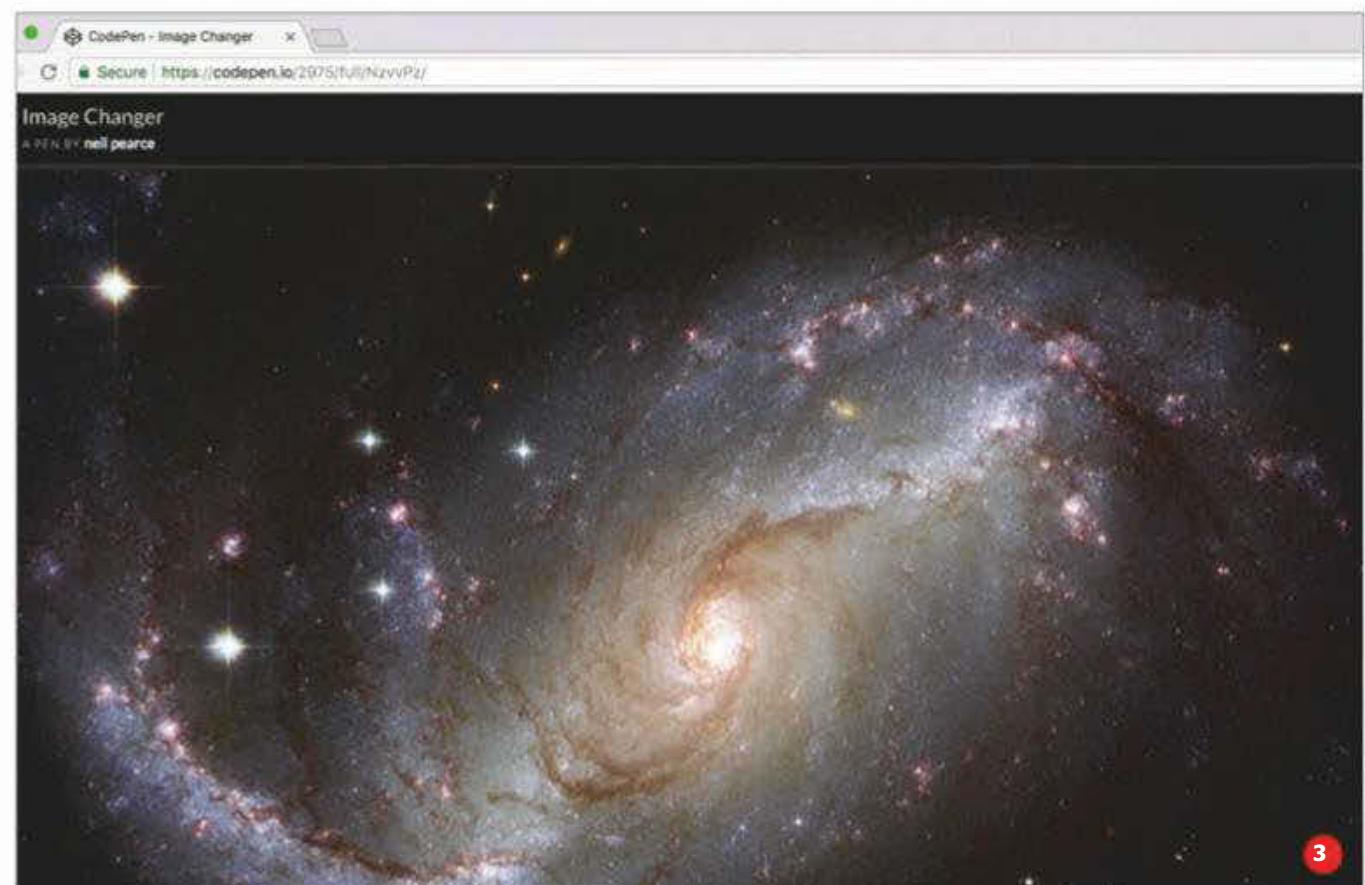
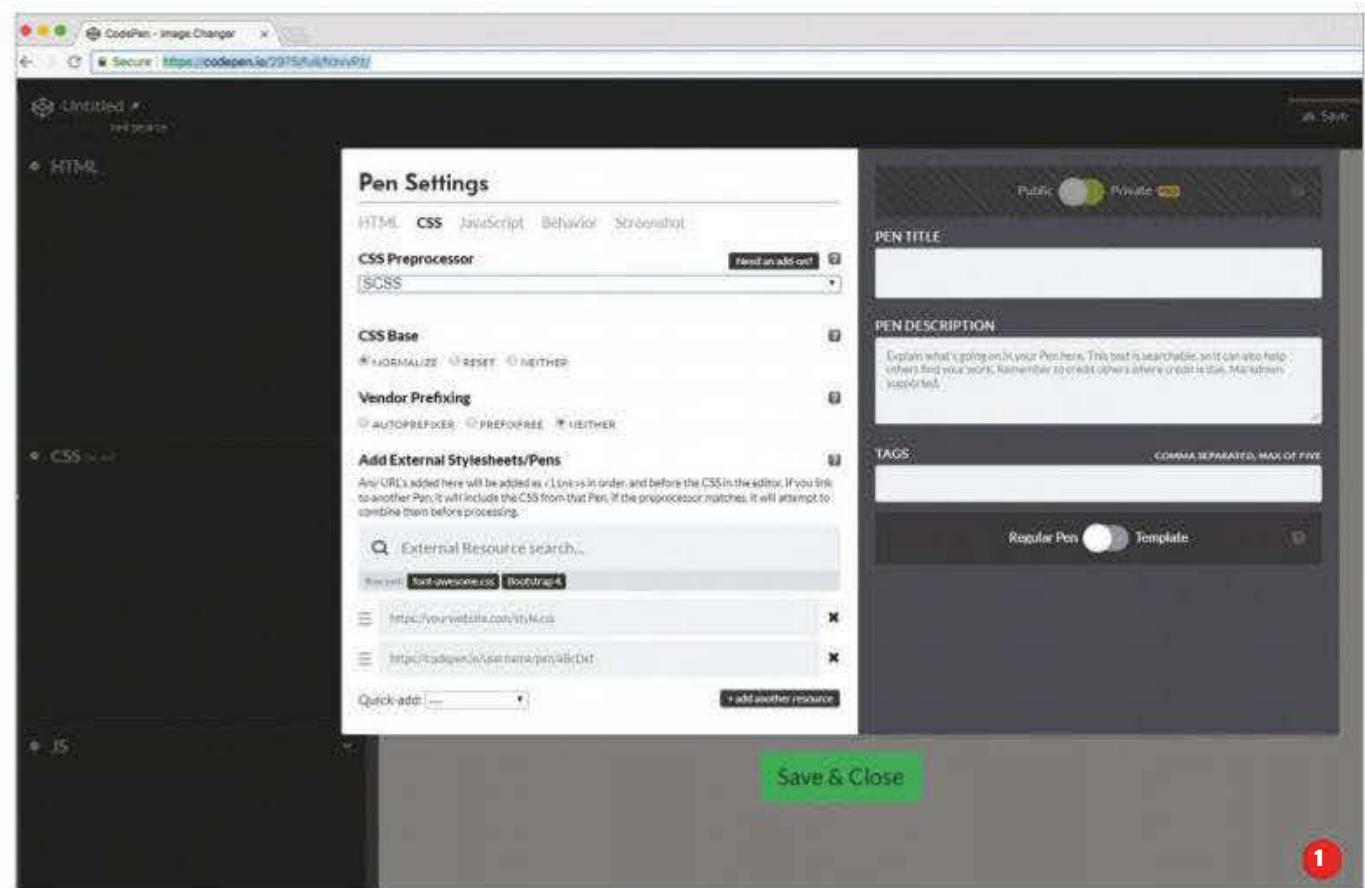
5. Image styles and thumbnails

The first thing we want to do is make sure all our images have a width of 100%. Then because our images won't

link to anywhere, we need to make sure they look clickable so we add a cursor pointer to give us that effect. Then we want to position thumbnail images underneath the main image.

To achieve this we can use CSS Grid. So using the grid template property, we can specify we want to have three columns with each taking up one fraction of the size of its container. So in other words, all three will be the same size. Then we give them a gap of four pixels so that they appear to have a border around them.

```
.main--img img,
.images img {
  width: 100%;
  cursor: pointer;
  transition: .8s;
}
```



Tutorials

Code an image changer with CSS

```
.images {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 4px;  
}
```

6. Keyframe animation

When we click on each thumbnail image and the main image changes, we want some kind of animation to occur, otherwise it'll be a bit jarring and won't offer a good user experience. We can do this quite simply by using CSS keyframes. Firstly, we will add a subtle hover effect by increasing the thumbnail opacity to 80% and adding a transition. Then we'll create a keyframe animation called 'fadeIn' that will be used to bring up the full opacity of the target image.

```
.images img:hover {  
  opacity: 0.8 !important;  
  transition: .8s;  
}  
  
@keyframes fadeIn {  
  to {  
    opacity: 1;  }  
}
```

7. Image fade

The image fade rule will be where the overall animation will trigger. JavaScript will be used to add in this class name (which we will get into in more detail later). But before anything happens, we need to make sure the image opacity is set to zero and then we can trigger the 'fadeIn' animation, which sets the opacity to 1 (100%).

```
.image-fade {  
  opacity: 0;  
  animation: fadeIn 0.6s 1 forwards;  
}
```

8. Constants

Now we head over to the JavaScript section and will begin by adding in some constant variables. First of all we can store the document in a constant called 'doc', which will enable us to shorten our statements whenever we want to target the document. Makes sense, right? Then we will need to store the active image within a constant called 'currentImg'.

And then by using 'querySelectorAll' we can create and store a Node List of all the thumbnail images. And lastly we will store an opacity value that we will use on the inactive images.

```
const doc = document;  
const currentImg = doc.querySelector('.  
active');  
const imgs = doc.querySelectorAll('.images  
img');  
const imgOpacity = .5;
```

9. Set the opacity

Because we created a Node List using 'querySelectorAll', we can use this in a similar way to how we use arrays and target the first image thumbnail [0] and set its style property equal to our 'imgOpacity' constant.

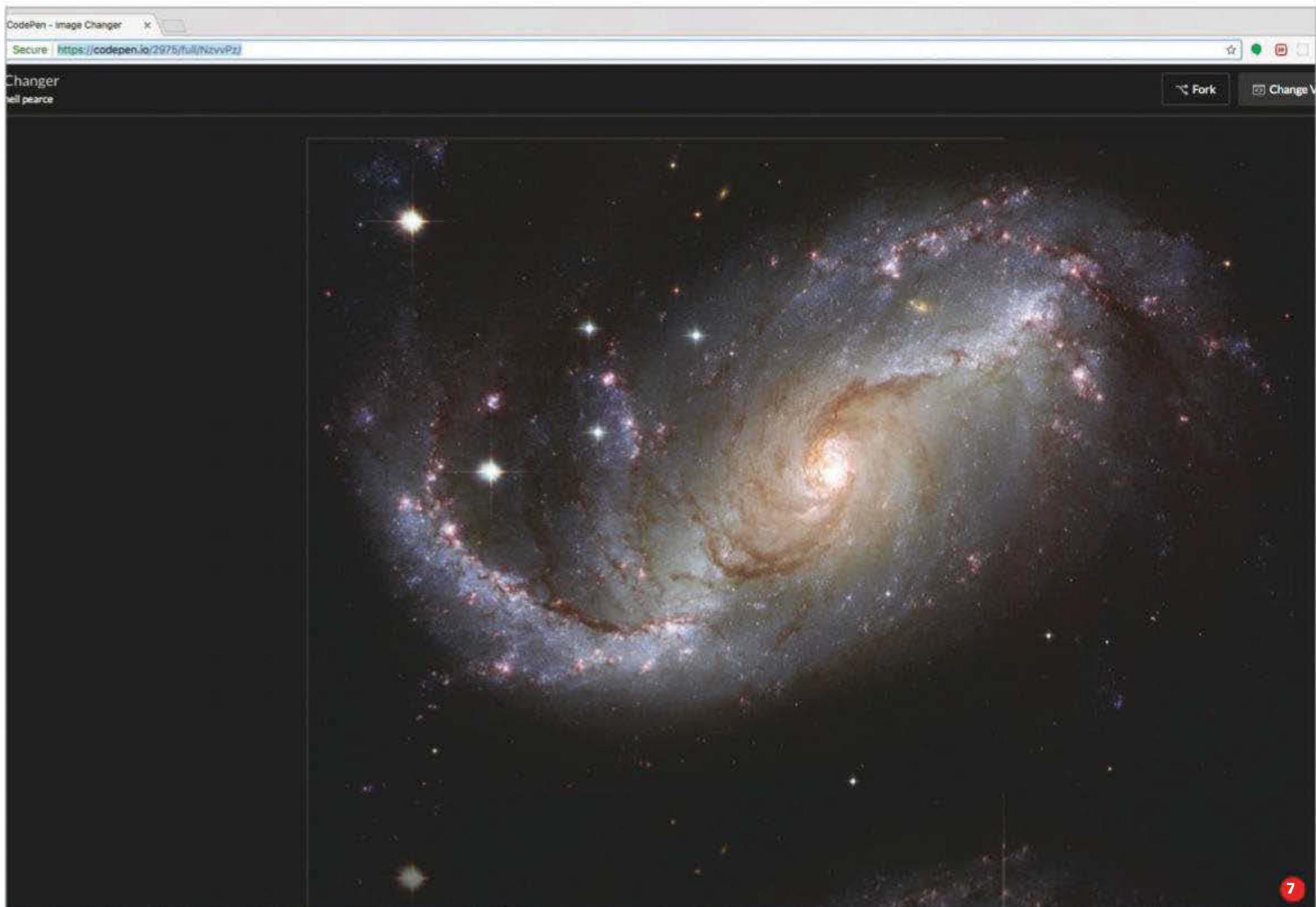
```
// Set the first image opacity
```

```
imgs[0].style.opacity = imgOpacity;
```

10. Loop through the images

Before we do anything else, we first need to create a global variable, which stores the last selected image (more on this in a later step). Then we can loop through each image and add a click event that will fire off a function called 'clickImage', which we will create in the next step.

```
// A pointer to last selected thumbnail image  
let thumbPointer = imgs[0];
```





CSS Grid: the new way to create web layouts

CSS Grid has been with us for a while and with widespread browser support it is now a everyday part of the design process. CSS Grid doesn't have a steep learning curve and there are plenty of resources (including **Web Designer** of course) that offer great pointers on how to use it for creating almost any layout you want. Web Designer 279 (<https://bit.ly/2z06RB7>) has a five-page tutorial on how to create shape-shifting responsive layouts with a fallback for unsupported browsers. To start learning the basics their is Grid Garden (<http://cssgridgarden.com>). This is a simple game that helps users understand the concept of columns and rows. For those needing a quick-start with CSS Grid you could try CSS Tricks starter layouts (<https://css-tricks.com/snippets/css/css-grid-starter-layouts>).

```
// Loop through the images and add a click event
imgs.forEach(img => img.addEventListener("click", clickImage));
```

11. The 'clickImage' function

The 'clickImage' function will be given a parameter called 'e' that will reference any click events that occur. Using the letter e is normal naming convention; however, you can use whatever convention that you wish, as long as it's descriptive to what it does.

We are then going to check and see if the target we clicked on is the same picture; if it is then we will return it and not do anything else. However, if it is not then we will change the main image with the one we just clicked on. As this happens we will add on the class 'image-fade' and fire off our CSS animation. Then the last thing we do is reset the opacity of the last thumbnail so they brighten up again, ready to be clicked on.

```
function clickImage(e)
{
  // In case the same picture is selected again.
  if(e.target === thumbPointer) return;

  // Change current image to src of clicked image
  currentImg.src = e.target.src;

  // Add fade in class
  currentImg.classList.add("image-fade");

  // Change the opacity to opacity var
  e.target.style.opacity = imgOpacity;

  // Reset the opacity of last thumbnail
  thumbPointer.style.opacity = 1;

  // Update thumb Pointer
  thumbPointer = e.target;
}
```

12. Remove the class

The problem we face now is that after the 'image-fade' class has been added, it stays added. So we need a way to remove this class on the current image. To achieve this we can use the 'setTimeout' method and say after half a second, remove the image-fade class.

```
// Remove fade-in class after 0.5 seconds
setTimeout(() => currentImg.classList.remove("image-fade"), 500);
```

13. Conclusion

Throughout this tutorial we have learned how to create a simple but nonetheless handy image changer that could be used anywhere on your websites. We've also been using ES6 syntax, such as const, let and arrow functions. Arrow functions are certainly something that we recommend you learning more about, as they are very useful when it comes to cleaning up your code.

```
• CSS (SCSS)
  .images img {
    width: 100%;
    cursor: pointer;
    transition: .8s;
  }

  .images {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-gap: 4px;
  }

  .images img:hover {
    opacity: 0.8 !important;
    transition: .8s;
  }

  @keyframes fadeIn {
    to {
      opacity: 1;
    }
  }

  .image-fade {
    opacity: 0;
    animation: fadeIn 0.6s 1 forwards;
  }

  • JS (Babel)
  CLICKIMAGE();
  function clickImage(e) {
    // In case the same picture is selected again.
    if(e.target === thumbPointer) return;

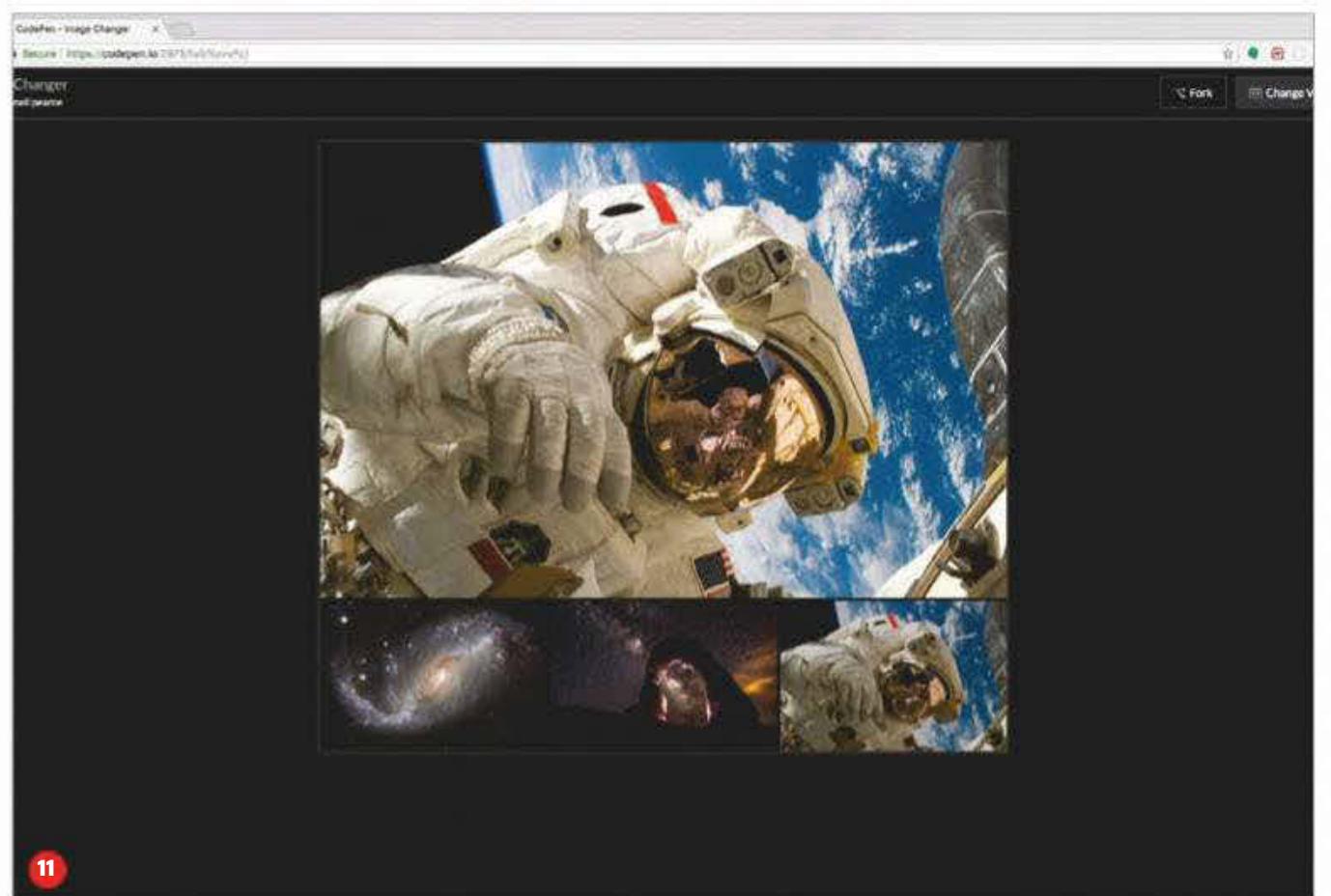
    // Change current image to src of clicked image
    currentImg.src = e.target.src;

    // Add fade in class
    currentImg.classList.add("image-fade");

    // Remove fade-in class after 0.5 seconds
    setTimeout(() => currentImg.classList.remove("image-fade"), 500);

    // Change the opacity
    e.target.style.opacity = imgOpacity;

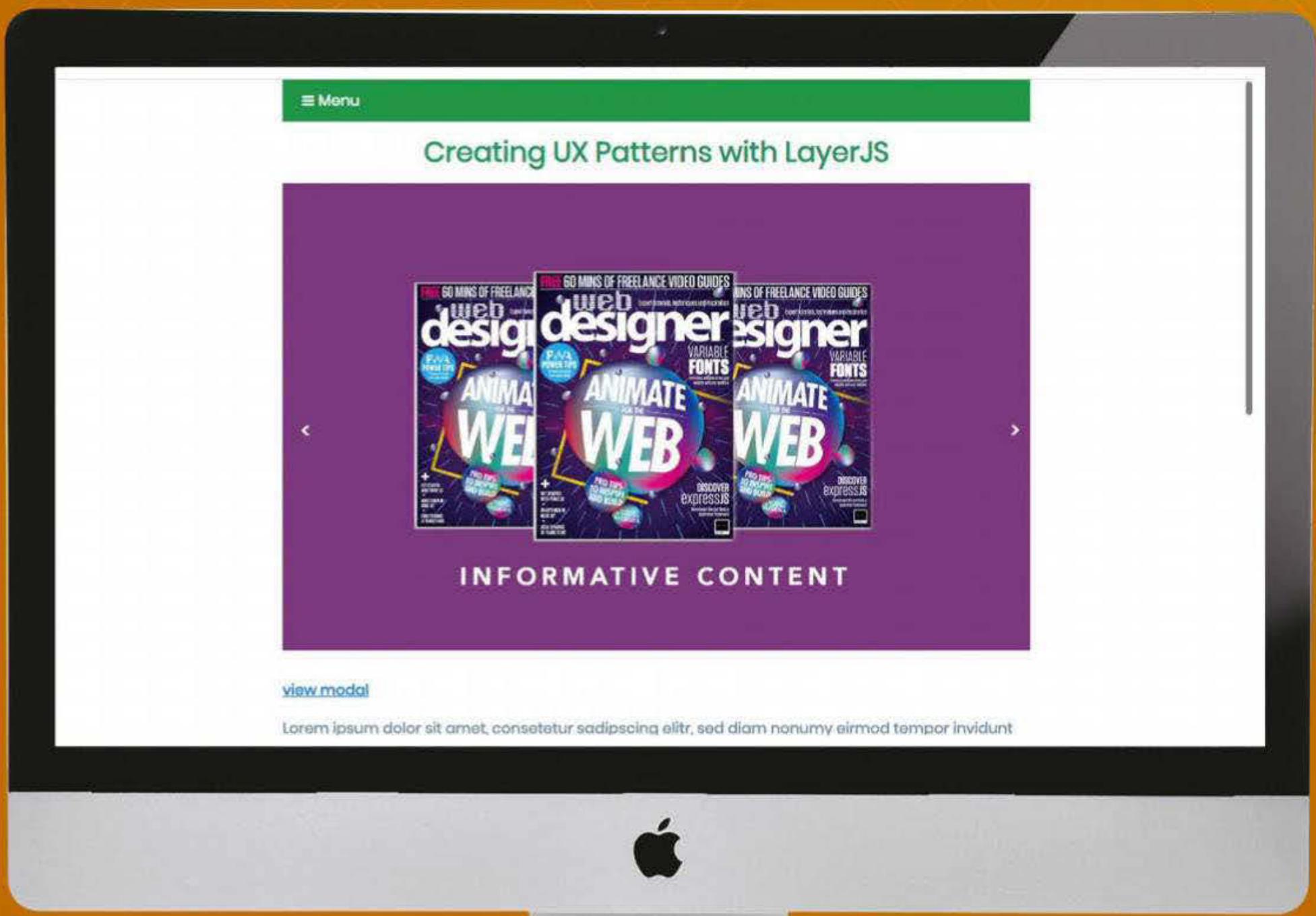
    // Reset the opacity of last thumbnail
    thumbPointer.style.opacity = 1;
  }
```



Developer tutorials

Create UX patterns with layerJS

This tutorial will introduce the basics of using layerJS and demonstrate how to create three UX patterns to help boost brand loyalty and trust



Imagine that every car model came with its own unique layout of the steering wheel, pedals and gear levers. How easy would it be to learn how to drive different cars? You guessed correctly – challenging. Humans are naturally wired to look for patterns or repeating qualities in everything they interact with, as they seek to understand the world around them. This concept is as true for driving cars as it is for using a website. Subsequently, most products used on a daily basis, are designed around time-tested user patterns of behaviour as opposed to reinventing the wheel. UX patterns are described as repeatable solutions to recurring design problems. Using established UX patterns reinforces user expectations, and as a result, boosts brand loyalty and trust. However, a challenge in implementing the different patterns is that, web designers have to learn different UI frameworks, and additionally, figure out how to integrate them without breaking their websites. Enter layerJS, a universal JS library that enables the creation of all UX patterns such as menus, sliders and parallel effects in pure HTML. The beauty with the library is that it is open source, responsive and provides an efficient way to create all UX patterns from one library.

In this tutorial, the basics of the library are discussed and three UX patterns are created: slider, modal and a slide menu.

1. Get started

Begin by creating a folder, 'layerJS', on your desktop to store the tutorial files. Create two additional folders within it: 'CSS' to store the styling files and 'images' to store image files. HTML files will be stored in the root folder (layerJS). Create a file 'styles.css' in the css folder and at least three images of your choice (1400px x 875px). Save these in the images folder as image 1, 2 and 3 respectively. These will be used in a later section of the tutorial.

2. Creating the page structure

Open any 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.

```
<html lang="en">
  <head>
    <title>UX Patterns with LayersJS
  </title>
  </head>
<body> </body>
</html>
```

3. Design the page layout

Before laying out the page, first read the blob section to understand the stage-frame concept that the library is based on. It is important to note that the intent of the tutorial is to create a slide menu, a slider and a modal, defining the following elements within the body of the page. As such, two stages are required, one for the slider and the menu and the other for the modal element. Frames and layers will be added to each of these stages. Notice that 'lj-type' is used to define the element (stage, layer or frame) while 'ids' are added to differentiate the elements. 'lj-name' simply assigns a name.

```
<body>
  <div lj-type="stage" id="sliderstage">
    <div lj-type="layer" lj-name="slider" > </div>
    <div lj-type="layer" id="arrows" > </div>
      <div lj-type="layer" lj-name="menulayer" > </div>
    </div>
    <div lj-type="stage" id="modalstage">
      <div lj-type="layer" lj-name="modal-box" > </div>
        <div lj-type="layer" lj-name="box-layer" > </div>
      </div>
    </div>
  </body>
```

4. Linking the stylesheet

Next, link the 'style.css' file created in step 1 by adding the path to the file in the head section of the page. Styling attributes added to the file will reflect on the main page automatically. Simply copy the code below.

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

5. Adding the layerJS library

The layerJS library can be added to the page in either of two ways. First, the library can be downloaded from GitHub (<https://github.com/layerJS/layerJS>) and saved in the root folder. This alternative is preferred where access to the internet is limited. To add it to the page, simply copy the path to the file.

```
<script type="text/javascript" src="path/to/layerjs.min.js"></script>
```

Similarly, ensure to include the 'layerjs.css' file as well.

```
<script type="text/javascript" src="path/to/layerjs.min.css"></script>
```

Secondly, the library can be directly linked to the CDN server. Simply copy the code below and paste it in the head section.

```
<head>
  <script src="https://cdn.layerjs.org/libs/layerjs/layerjs-0.6.0.min.js"></script>
  <link href="https://cdn.layerjs.org/libs/layerjs/layerjs-0.6.0.css" type="text/css" rel="stylesheet" />
</head>
```

6. Initialising layerJS

Once the library has been added, initialise it in the body section so as to ease the implementation process. It is important to note that the library is initialised only once despite adding its different functionalities in the page. Simply add the code below in the body section.

```
<script>
  layerJS.init();
</script>
```

7. Creating the slider

Creating the image slider is fairly straightforward. A slider needs a single stage, a layer and one frame for each slide. Controls can be placed on a second overlapping layer.



Controls are regular links that link to specific frames or use special links like '#!next' or '#!prev'. Note that the stage and the layer were already defined. Now, frames are created to contain actual slides.

However, it is important to learn several attributes that are used in the frames and layers. 'lj-fit-to' specifies the fitting strategy for the image slides and can be set to width, height, responsive, fixed etc. 'lj-start-position' controls the alignment of the stage borders while 'lj-neighbors.r/l' specifies the name of the frame right or left of the current frame. 'lj-default-frame' specifies the frame that is shown initially. These attributes are used in the next step.

8. Adding slides

In step 3, the layers created were devoid of attributes. However, with the overview of attributes provided in step 7, they are now added. To create slides, simply add the code below within the slider layer.

```
<div lj-type="stage" id="sliderstage">
  <div lj-type="layer" lj-name="slider" lj-default-frame="frame1" lj-native-scroll="false">
    <div lj-type="frame" lj-name="frame1" lj-fit-to="width" lj-no-scrolling="true" lj-start-position="center" lj-neighbors.r="frame2">
      
    </div>
    <div lj-type="frame" lj-name="frame2" lj-fit-to="width" lj-no-scrolling="true" lj-start-position="center" lj-neighbors.l="frame1" lj-neighbors.r="frame3">
      
    </div>
    <div lj-type="frame" lj-name="frame3" lj-fit-to="width" lj-no-scrolling="true" lj-start-position="center" lj-neighbors.l="frame2">
      
    </div>
  </div>
</div>
```

How layerJS works – stage frame concept

layerJS is based on the stage-frame concept where stages represent viewports while frames are different HTML defined fragments of the site where content is defined such as a menu or slider. If more than one frame is added, it becomes possible to navigate content at that page level. Layers are then added to create overlapping elements such as pop-ups. In addition, frames can be sent to different stages which are added onto the page.

Tutorials

Create UX patterns with layerJS



```

</div>
</div>
```

This code simply creates three slide images and sets the slider image as the default frame. Each image is set to fit to the width, which is specified neatly in CSS. The slider neighbours are likewise identified for each slide to aid in navigation.

9. Adding arrow controllers

Next, add the arrow controls that aid in controlling the actual slide images. Simply add the code below to the second layer (id arrows). Ensure to add the attributes.

```
<div lj-type="layer" lj-native-scroll="false"
id="arrows">
<div lj-type="frame" lj-fit-to="width"
lj-start-position="center" lj-no-
scrolling="true">
<a class="btn-left btn" href="#slider.!left?p=
right">< />
<a class="btn-right btn" href="#slider.!right?
p=left">&gt;</a>
</div>
```

This code simply creates a left and right button arrow to facilitate controlling the slider images. Note that at this point, the slides cannot render as CSS styling controls this.

10. Adding the slide images

Next, specify the styling of the slider. However, begin by adding the slide images before adding arrows to control the animation. Simply add the code below carefully in the stylesheet.

Picking up the pace with layerJS

One of the most recommended ways to learn how to use a JS library is interacting with numerous examples. layerJS has an examples page specifically developed to help facilitate such learning (<https://layerjs.org/examples.html>). Simply navigate to the page and review the examples cited. Some of these include the slider, zoomUI, parallax, side menu and a lightbox or modal. Navigate to the GitHub wiki page as well (<https://github.com/layerJS/layerJS/wiki>) in order to learn more on the attributes, layouts and transitions among others.



```
* {
  box-sizing: border-box;
}
[lj-type="stage"] {
  width: 100%;
  height: 100%;
  max-width: 800px;
  margin: auto;
}
```

This code simply sets the stage to a width and height of 100%. However, a maximum width of 800px is specified for the slider images. Now is the time to render the page. Notice that although the images appear, they do overlap.

11. Style the arrow controls

Next, the arrow controls are styled. Simply add the code below to your stylesheet. The code simply specifies the positioning of the different arrows and applies styles to the navigation arrows. You'll quickly notice that the button controls now appear and the slider can be controlled easily.

12. Adding a title to the page

Now that the slider pattern has been added to the page, simply create a suitable title just above the slider layer title.

```
<div lj-type="stage" id="sliderstage">
<h2>Creating UX Patterns with LayerJS</h2>
```

Now, add the following CSS styling to make the title more appealing to your audience.

```
@import url('https://fonts.googleapis.com/css?
family=Poppins');
h2 {
  font-family: 'Poppins', sans-serif;
  font-size: 30px;
  text-decoration: none;
  color: #0d9668;
  text-align: center;
  position: absolute;
  left: 10%;
  right: 10%;
  top: 30px;
}
```

13. Creating the menu structure

Next, the menu's HTML structure is defined. However, two important points are noted. First, since the interest is to have the menu cover the entire width of the current slideshow, it is defined in the same stage as the slider. As

such, only one layer is added. Similarly, two frames are also added within this layer to contain the menu title and the sub-menu items respectively. Simply add the code below in the 'menulayer' div created in step 3.

```
<div lj-type="layer" lj-name="menulayer"
lj-native-scroll="false" >
```

```
<div lj-type="frame" lj-name="hmenu"
class="nav" lj-fit-to="responsive-width"
lj-transition="fade">
```

```
<div class="nav-hor">
<a href="#menu?t=0.5s">&#x2630; Menu</a>
</div>
</div>
```

```
<div lj-type="frame" lj-name="menu" class="nav
menu" lj-fit-to="fixed" lj-neighbors.r="hmenu"
lj-transition="slideOverRight">
```

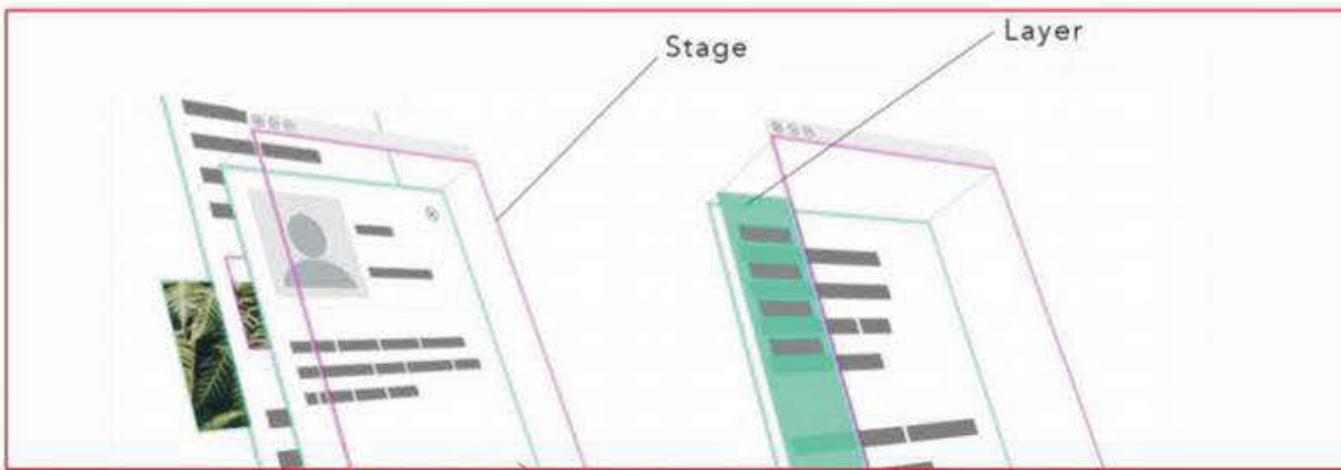
```
<a href="#hmenu?t=0.5s"
class="close">&#x2573;</a>
<ul class="nav-list">
<li class="nav-item">Menu 1 </li>
<li class="nav-item">Menu 2 </li>
<li class="nav-item">Menu 3 </li>
<li class="nav-item">Menu 4 </li>
</ul>
</div>
</div>
</div>
```

The menu should render above the title. Click the menu to observe the transitions specified in the code above.

14. Border background and text decoration

Now that the menu has been placed, style it in CSS to make it more appealing. Currently, the menu renders as a link and the menu items are constrained. We change these by removing text decoration and arranging the menu items more neatly. Add the code below.

```
.menu {
  width: 400px;
  box-shadow: 0px 0px 10px 1px #0d9547; }
.content {
  box-sizing: border-box;
  padding: 60px 20px; }
.close {
  position: absolute;
  right: 0px;
  padding: 5px;
  text-decoration: none;
  cursor: pointer;
  color: black; }
.nav a {
  text-decoration: none;
  color: inherit; }
```



The code simply adds a box shadow to the menu, specifies the layout of menu items as well as the menu status when inactive.

15. Add a background colour

Next, the menu items are styled and a background colour added. The font family used is also specified in addition to the padding of the different menu items.

16. Add a hover effect

Finally, add a hover effect to the menu in order to signify user interaction. Add appropriate background colours and specify padding for the different child menu items.

```
.nav-item: hover
{
    background-color: #eee;
    color: #777;
}

.nav-hor
{
    box-sizing: border-box;
    padding: 10px 20px;
    border-bottom: 1px solid #eee;
    cursor: pointer;
}

.nav-hor: hover
{
    background-color: #58585A;
}
```

17. Creating a modal

The last UX pattern is now created. In creating the modal, we require a stage, two layers and a frame. The first layer created is left without any content while content is placed in the frame.

```
<div lj-type="stage" id="modalstage">
    <div lj-type="layer">

        <div lj-type="frame" lj-fit-to="width">
            <a href="#box-layer.box1?p=fade&t=0.2s">view modal</a>
            <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea </p>
```

```
</div>
</div>
```

The code above simply creates the lightbox content and specifies a link at view modal. When a user clicks the link, the lightbox should appear. However, this functionality is added in the next step.

18. Adding the modal content

In order to enable the lightbox functionality, a new layer is required to be added and the lightbox behaviour directed to the empty layer created in the previous step. Simply add the code below.

```
<div lj-type="layer" lj-default-frame="!none"
lj-name="box-layer" lj-native-scroll="false">
    <div lj-type="frame" lj-name="box1"
lj-fit-to="responsive" class="modal">
        <div class="modal-inner">
            <div>
                <a href="#box-
layer.!none?t=0.2s&p=fade"
class="close">&#x2573;</a>
            </div>
            <h1>A modal</h1>
            <p>Lorem ipsum dolor sit
amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et
dolore magna aliquyam erat </p>
        </div>
    </div>
</div>
```

The code directs the modal to render at the empty frame specified as '!none'. Once closed, the functionality of the lightbox is reset.

19. Styling the lightbox/modal

Finally, style the lightbox in order to render the content.

```
p {
    text-decoration: none;
    color: #728eaa;
    font-family: 'Poppins', sans-serif;
    font-size: 16px;
}

a {
    color: #0a73dd;
    font-family: 'Poppins', sans-serif;
    font-size: 16px;
    text-decoration: underline;
```

The design agnostic features of layerJS

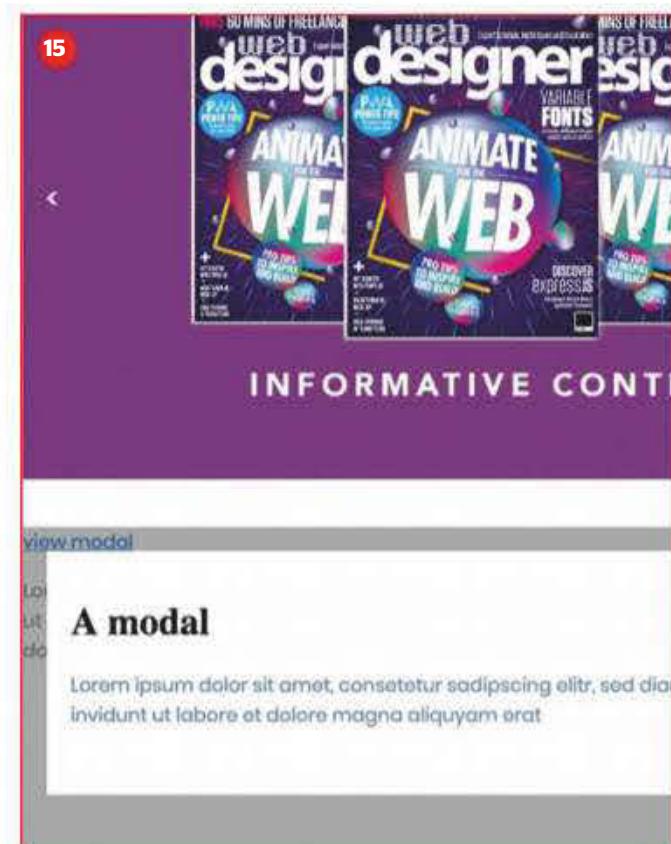
layerJS is design agnostic, that is, it provides the designer with the navigation aspects while leaving the design up to the creative. The stage-frame concept is what makes this possible. Frames are simply DIVs containing the content of your site, and they can be fit into Stages (also DIVs serving as viewports) dynamically. The root Stage is usually the browser window and its Frames represent sub pages or app screens. Frames can be exchanged within Stages using animated transitions like swipes, fades or 3D transitions. Frames can be placed on overlapping layers allowing effects like parallax backgrounds. The library also supports Angular, VueJS, React and jQuery.

```
}
```

```
.close{
    float: right;
    padding: 5px;
    text-decoration: none;
    cursor: pointer;
    color: black;
}
```

```
.modal {
    background-color: rgba(80,80,80,0.5);
    padding: 20px;
    height: 30%;
}
.modal .modal-inner{
    background-color: #fff;
    height: 30%;
    padding: 20px;
}
#modalstage{
    position: relative;
    top: -70px;
}
```

Notice that the code specifies the styling for the paragraphs, links, close button and the modal behaviour.



CREATE WITH THE GREENSOCK ANIMATION PLATFORM

GSAP (Greensock Animation Platform) is an industry standard for web-based animation, used by millions of sites worldwide. Learn how it works, its key features and how to integrate it with your web experiences

Greensock's Animation Platform (GSAP) - <https://greensock.com> - allows you to animate anything you can access with JavaScript, including the DOM, SVG, canvas, CSS and even your own custom objects. It can easily be integrated to work with 2D and 3D animations as well.

Animation is the process of changing property values many times per second, making something appear to move, fade, spin etc.

GSAP takes a starting value, an ending value and then interpolates (tweens) between them a staggering 60 times per second.

GSAP abstracts away many of the underlying browser inconsistencies, allowing you to quickly be cross-browser ready. It also allows you to use objects to manage complex animations, and runs up to 20x faster than jQuery. It has become a standard in the industry, being used in countless award-winning sites.



How something moves creates a feeling, which becomes an experience for your audience. How you want people to feel can be directly connected to how you craft your animations.

RICHARD MATTKA

Creative Director, Developer, Designer
richardmattka.com

KEY FEATURES

1. ANIMATE ANYTHING

GSAP allows you to animate pretty much anything JavaScript or CSS can access. There are no predefined properties you are limited to. Virtually, any numeric property of any kind of object can be 'tweened'. If there are specific values that are tricky to tween, such as colours or very specific object models, chances are good there is a plugin to handle it. Even if there isn't yet, you can create one easily. Animate 2D, 3D, colours, scales, positions, rotations and much more.

2. EXTREMELY FAST

Great animation performance is critical, especially on mobile devices with slower processors. You need slick, smooth motion as objects bounce and fly across the screen. Instant reactions to user interactions make apps feel polished and responsive. Combining optimised choices between JavaScript and CSS animations under the hood, enables the platform to be highly performant, all with normalised functions. GSAP boasts incredible speed (up to 20x faster than jQuery), including automatic GPU-acceleration of transforms.

3. COMPATIBILITY

Whether you need HTML5, SVG, jQuery, Canvas, CSS, new browsers, old browsers, React, Vue, EaselJS, mobile, GSAP is built to work with them all.

In tight production timelines, the last thing you want to waste valuable time on is cross-browser testing and optimisation. GSAP lets you get onto the good stuff, while it handles all the nuances and inconsistencies between browsers and devices for you.

GSAP even accommodates modern browsers, plus IE back to version 6, without requiring special browser prefixes or ugly hacks.

5 THINGS YOU CAN DO WITH GSAP

1. ANIMATE CSS PROPERTIES

GSAP allows you to quickly animate CSS properties, with its normalised behaviours. It can handle colour tweens, SVG animations and optimised performance with caching and other internal tricks. Instead of JavaScript alone, it is often much more efficient to translate positions via CSS.

2. EASY EASING

GSAP includes a variety of easing functions such as back, bounce, elastic, sin, circ and expo. These enable smooth acceleration and deceleration for your tweens. You can 'easeIn', 'easeOut' and 'easeInOut'. You can even create your own custom easing functions as you need to!

3. ANIMATE SVG

GSAP allows you to animate viewBox, width, height, fill, stroke, cx, r, opacity etc, all by default. Easy to use plugins like MorphSVG and DrawSVG can be used for advanced effects, like drawing in images, morphing between them and much more.

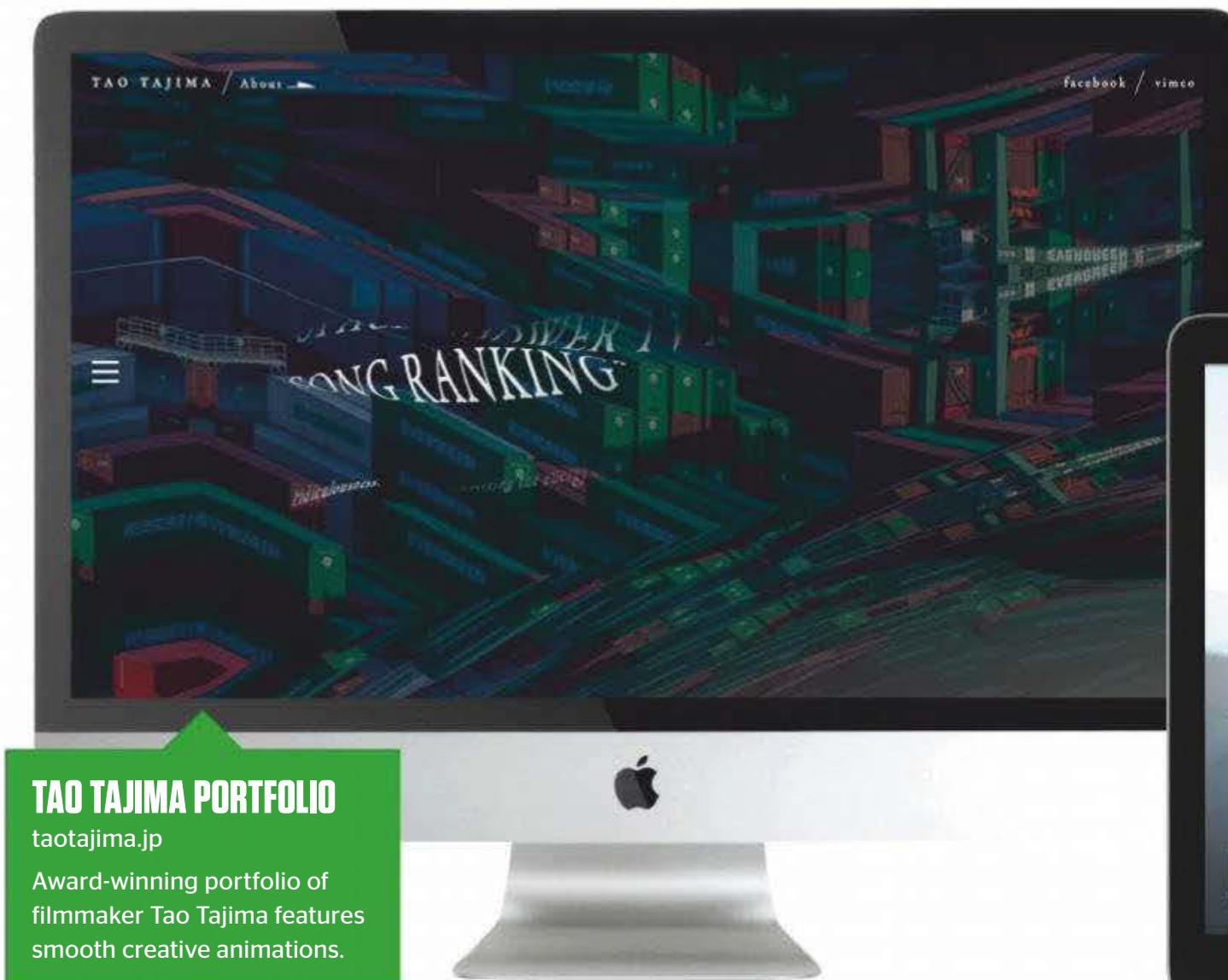
4. STAGGER ANIMATIONS

Using the stagger method, you can apply animations to an array of objects to a common set of destination values. Then you can stagger their start times by a specified amount of time, creating an evenly-spaced sequence with hardly any extra code. You'll use this all the time once you try it out.

5. REPEAT, REVERSE AND YO-YO

GSAP allows you to set animations on repeat, easily reverse their direction at anytime, slow them down, speed them up or even 'yo-yo' them (forward, then reverse). You can apply this to specific tweens or even entire timelines. These are great for ambient animations, loading sequences or loops.

INSPIRATIONAL EXAMPLES OF GSAP IN ACTION



THE NEW MOBILE WORKFORCE

www.thenewmobileworkforce.com

Beautiful design and animations in this Site of the Month, by Citrix, helping Aston Martin Red Bull Racing.



ANIMATING WITH GSAP

Learn the basics of setting up GSAP and creating your first Tweens

The best way to learn GSAP is to see it in action. In the following tutorial you'll learn about the key features of the platform with working examples you can put to use in your projects today!

1. SET UP A BASIC HTML FILE

Get started by setting up a basic HTML file, where you can drop in your JavaScript code. Include any image you like and give it a 'logo' class as seen in the code listing below. You'll use GSAP to animate this element's properties to see how it works.

```
<html lang="en">
<head>
  <style>
    .logo { width:150px; }
  </style>
</head>
<body>
  
<script>
</script>
</body>
</html>
```

2. INCLUDE THE GSAP LIBRARY

Next, you'll need to add GSAP to your project. Between your image and script tags, insert a link to the GSAP library. You can download the ZIP directly here:

<https://bit.ly/2Myg5OC> or grab from GitHub here: <https://github.com/greensock/GreenSock-JS>

Files are also hosted on Cloudflare CDN, so the simplest way is to use the hosted files like this:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/2.0.1/TweenMax.min.js">
</script>
```

3. ANIMATE WITH TWEENS

Animating or 'tweening' is the process of changing a value over time to create an animation. For example, scaling an object, moving it from A to B or rotating it.

To tween a property, the simplest method is to use GSAP's TweenMax.to(). This function needs a target object, a duration and property/value pairs you are animating.

Try out each of these lines of code using the 'logo' as the target object.

```
// tween x position from
current to 400 over 2 seconds
TweenMax.to(".logo",2,{x:400});
```

```
// tween y position from
current to 200 and opacity to
0, over 1 second
TweenMax.to(".logo",1,{y:200,
opacity:0});
```

```
// tween x and y to 100, scale
to 1.5, and rotate 90 deg,
over 2 second
```

```
TweenMax.to(".logo",2,{x:100,
y:100, scale:1.5, rotation:90});
```

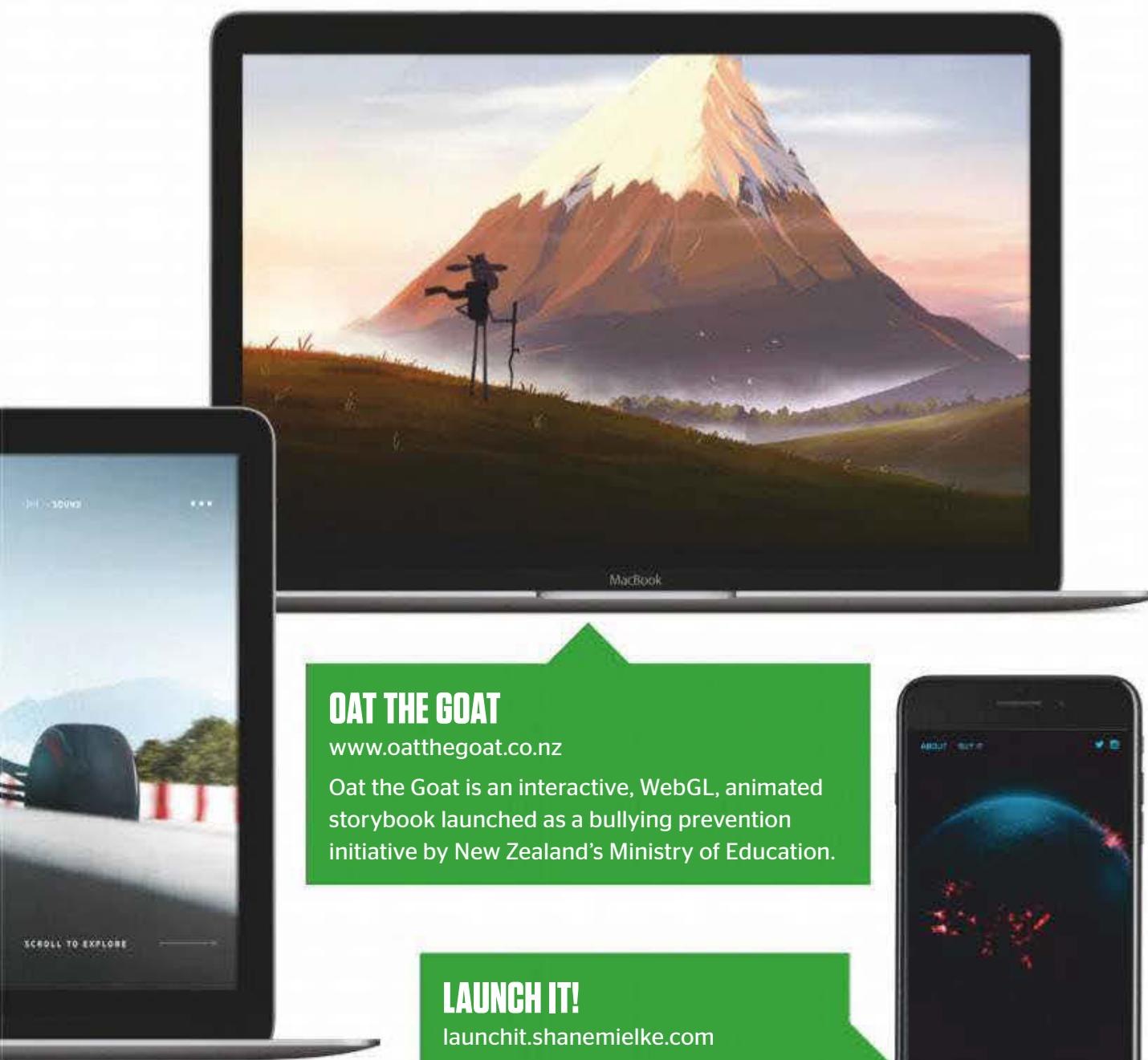
4. TO, FROM AND FROMTO

As you saw, you can tween a property from its current value to a new one using to(), but you can also tween 'from' a value to its current value. If your 'logo', for example, is starting at y position '100' and you want it to end there, you could do this:

```
TweenMax.from(".logo",1,{y:500});
```

You can also define both the start and end values, ignoring current values by using fromTo() like this:

```
TweenMax.fromTo(".logo",2,{x:400},{x:200});
```



OAT THE GOAT

www.oatthegoat.co.nz

Oat the Goat is an interactive, WebGL, animated storybook launched as a bullying prevention initiative by New Zealand's Ministry of Education.

LAUNCH IT!

launchit.shanemielke.com

Shane Mielke's site combines Three.js, beautiful designs and GSAP to promote his latest book *Launch it!*

5. EASING

Easing is the 'style' of animation, as values transition from A to B. Instead of a constant rate of speed, which is called 'linear', you can apply functions to curve the rate of speed. Do they start slowly and gradually speed up? Do they come to an abrupt stop and bounce a little at the end? You can apply an easing function like this:

```
TweenMax.to(".logo",2,{x:100, y:100, rotation:180, ease:Circ.easeIn});
```

6. DELAY A TWEEN

Sometimes you'll want to wait enough time for an event to occur or you want to delay the start of an animation to synchronise with another animation. You can use another of GSAP's 'special properties' called 'delay' to do this. Try out this code to see how you can delay tweens for specific timing:

```
TweenMax.to(".logo",1,{y:100, ease:Bounce.easeOut});  
// delay this tween by 1 sec  
TweenMax.to(".logo",1,{rotation:90,
```

```
ease:Circ.easeOut, delay:1});
```

7. USING TIMELINES

To help manage multiple tweens, GSAP includes a timeline object. You append tweens to the timeline object and can use the 'position' parameter after the 'tween' to time them. You can have tweens run one after another, or have gaps or even overlap them. Add a couple more images to your HTML with classes 'logo2' and 'logo3' respectively. Try out this timeline code:

```
//create a timeline instance  
var tl = new TimelineMax();  
tl.add( TweenMax.to(".logo", 1, {x:50}) );  
  
// note the final "0" to make this one start at 0 sec.  
tl.add( TweenMax.to(".logo2", 1, {y:100} ),"0" );  
//note the ".25" to make this one start at .25 sec  
tl.add( TweenMax.to(".logo3", 1, {rotationY:180, y:50, X:50}),".25" );
```



GSAP's industry-leading performance and intuitive API allows you to focus on the creative aspects of animation, including style, character and storytelling, rather than worrying about the underlying cross-browser code.

What is easing?

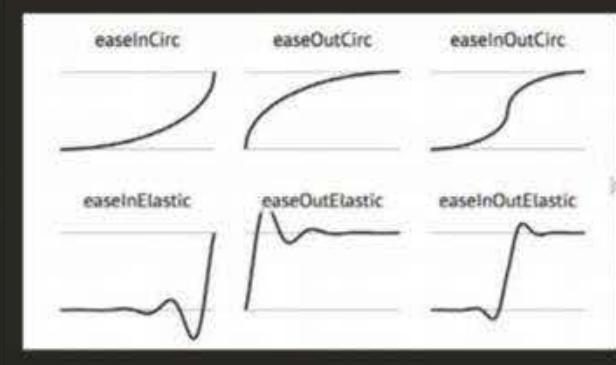
Easing is how you control the rate of change over the animation (tween). But what does that really mean?

Your animation's 'style' of motion gives it character or personality. Is the motion fluid, bouncy or robotic? Is it energetic or sluggish?

How something moves creates a feeling, which becomes an experience for your audience. How you want people to feel can be directly connected to how you 'ease' your animations.

In animation, we call a motion that starts slowly and accelerates a 'slow in'. For motion that starts quickly and slows down, it is called a 'slow out'. In CSS and web-based animations we use the similar terms, 'ease in' and 'ease out'. Sometimes both can be combined, which is called 'ease in out'.

It's this rate of slowing down or speeding up, and how those rates change that is what we call 'easing'. Your easing with GSAP includes all the common functions such as elastic, bounce, circular, quadratic and linear. You can also use any custom easing you choose to create.



THE US OPEN SESSIONS

MUSIC MADE WITH TENNIS DATA

Callbacks – What are they?

Callbacks are functions that are called when an event occurs within the tween or timeline object. You've probably used this lots in your code already, for handling things like asset loading or document-ready events.

In order to integrate animations with your code, it's important to know when events occur. Especially when an animation ends or begins. You can use the 'onComplete' event callback for this. You may also want to know when a tween starts, using 'onStart'.

You may want to sync your tween with another animation, or use the tween and its easing to update some other custom object. Use the 'onUpdate' callback for this.

Here is how you use the callback:

```
TweenMax.to(".logo", 1, {x:300,  
    ease:Bounce.easeOut,  
    onComplete:tweenComplete  
});  
  
function tweenComplete()  
{  
    console.log("tween  
complete");  
}
```

When using callbacks, you can also pass information to the function that handles it. All parameters are passed via 'callback+“Params’'. For example, 'onUpdateParams' would pass along any values to the 'onUpdate' callback handler.



LIVE

WOMEN'S Current
Matches

NEW

JAMES MURPHY

CONTROLLING ANIMATIONS

Introduce methods to finesse how an animation works

So, you can make beautiful tween animations with smooth easing, but how do you control them? Once you've created a tween, you'll need to be able to control it, especially to integrate it with events in your app or website.

Controlling a tween requires a reference to the instance you created. All the methods that create tweens in GSAP return this reference. The 'to()', 'from()', and 'fromTo()' methods, all will do this. Simply assign the tween to a variable and you're all set to control it.

Here's an example:

```
var tween = TweenMax.  
    to("#object", 1, {x:100});
```

Now that you have a reference to the instance stored in the variable, you can use several

common functions to control it. These include:

pause() - Use this to pause an animation, freezing its current progress until you start it again.

resume() - Use this to get the animation moving again after pausing it. This will remember direction as well, in case you used 'reverse()'.

reverse() - Switch direction of the tween, forcing it to go in the opposite direction from its current value(s).

seek() - Allows you to jump to a specific time in the tween. For example, 'seek(1.5)' would jump to 1.5 seconds into the tween.

progress() - Just like 'seek()', except this is a percentage value from 0.0 to 1.0 in the tween. For example, 'progress(0.25)' would

jump to 25% into the tween.

timeScale() - Lets you adjust the duration of the tween at run-time. For example, 'timeScale(.5)' would make it run half-speed, while 'timeScale(2)' would make it run at double speed.

kill() - A very useful control when you need to abruptly kill the tween without it completing.

Combining these controls, with events like clicks and rollovers, will enable you to integrate a wide range of animations easily into your project.



INTEGRATING GSAP WITH THREE.JS

Create beautiful animations fast using Three.js and GSAP

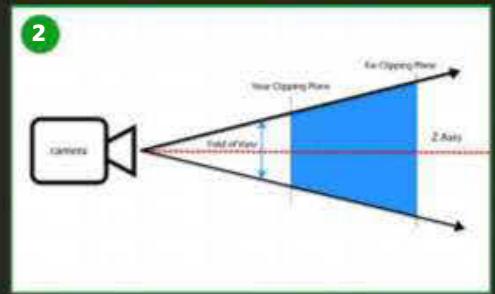
Pairing Three.js and GSAP combines the power of 3D rendering with the easing functions and timeline management of GSAP.

01. SET UP LIBRARIES

Get started by setting up a basic HTML file. You'll be using GSAP to handle animations, which you can download here: <https://github.com/greensock/GreenSock-JS>

You'll also use Three.js to handle 3D rendering. You can get Three.js here: <https://github.com/mrdoob/three.js>

```
<!DOCTYPE html>
<html>
<head>
<title>Three.js and GSAP</title>
<script src="libs/three.min.js"></script>
<script src="libs/TweenMax.min.js"></script>
<style>
html, body { margin: 0;
padding:0; overflow: hidden; }
</style>
</head>
<body>
<script>
// js code goes here
</script>
</body>
</html>
```



02. ADD A SCENE

Next, you'll need to add a Three.js scene to hold your objects, a camera to view them in 3D and a renderer to handle the drawing of the scene to a HTML canvas. You'll also need an objects array to hold references to our objects. Add this code next to set up:

```
var objects=[];
var scene = new THREE.Scene();
var camera = new THREE.PerspectiveCamera( 75, window.innerWidth/window.innerHeight, 0.1, 1000 );
camera.position.z = 500;
var renderer = new THREE.WebGLRenderer({antialias:true});
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
```

03. CREATE SOME OBJECTS

You'll add some simple box objects to the scene to give you something to animate. Feel free to use your own objects if you are familiar with Three.js already. Use Buffer Geometry for optimal performance and a simple Normal material. You'll also pass the objects in a loop to add them quickly. Add this code next:

```
var geometry = new THREE.BoxBufferGeometry(40,40);
var material = new THREE.MeshNormalMaterial();
for ( var i = 0; i < 100; i ++ ) {
```



```
var object = new THREE.Mesh( geometry, material );
// modify position, rotation and scale here
scene.add( object );
objects.push( object );
}
```

04. UPDATE THE POSITION, ROTATION AND SCALE OF YOUR OBJECTS

To add some variety to the scene and to demonstrate just how performant this combination of libraries can be, you will add many objects and vary their rotation, position and scale randomly. Add this code next, where indicated from step 3 previously.

```
object.position.x = Math.random() * 1000 - 500;
object.position.y = Math.random() * 600 - 300;
object.position.z = Math.random() * 800 - 400;
object.rotation.x = Math.random() * 2 * Math.PI;
object.rotation.y = Math.random() * 2 * Math.PI;
object.rotation.z = Math.random() * 2 * Math.PI;
object.scale.x = Math.random() * 2 + 1;
object.scale.y = Math.random() * 2 + 1;
object.scale.z = Math.random() * 2 + 1;
```

05. RENDER THE SCENE

To 'see' your tweens you need two more things. You need a render function to tell Three.js to render the current frame of the scene. You also need to bind the render function to

the GSAP animation loop 'tick'. In Three.js, scale, rotation and position are objects. To access them you need to reference them in the tween function, as seen in the code listed below.

```
// render out threejs scene
var render = function () {
    renderer.render(scene, camera);
};

// use the GSAP animation loop
// to call render
TweenMax.ticker.addEventListener("tick", render);

// create a tween
var tween = TweenMax.to(objects[0].position, 2, {
    x:0, y:0, z:0 } );
```

06. TWEEN ALL THE OBJECTS

Why tween one object when we can tween hundreds of them? In this step, you will apply a random tween to every object in the scene. For fun, we'll make them unique to each object, and add some repeat, yo-yo and easing to make them way cooler. Replace that previous tween, with this final code.

```
// create a tween

for(i=0;i<objects.length;i++){
    TweenMax.to(objects[i], rotation, 2+Math.random()*2,{ 
        x:Math.PI/180*360,
        y:Math.PI/180*360,
        yoyo:true,
        repeat:-1,
        ease:Elastic.easeInOut,
        delay:Math.random()*5
    });
}
```



Complex animations

Beyond one or two tweens, you'll need timelines

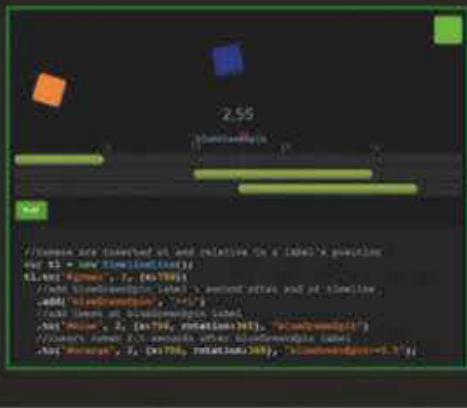
Think of a timeline as a container for all your tweens, where you assign them start times and durations, just like a schedule.

GSAP includes a timeline object to do exactly that. You can append all your tweens to the timeline and then manage it as a larger object. You can adjust the sequence order and timing to get the right overall timing you need.

Here's a quick example:

```
//create a timeline
instancevar tl = new
TimelineMax();
// add a tween to it
tl.add( TweenMax.to(".logo1",
1, {x:50} ) );
// note the final "0" to make
this one start at 0 sec.
tl.add( TweenMax.to(".logo2",
1, {y:100} ),"0" );
//note the ".25" to make
this one start at .25 sec
tl.add( TweenMax.to(".logo3",
1, {rotationY:180, y:50,
x:50} ),".25" );
```

Similar to timelines in apps like After Effects, you can have overlapping animations, gaps between them, and virtually as many as you need. As the timeline's playhead moves, it scrubs across its tween and renders them. You can pause, resume, reverse and control timelines just like animations. You can also add parameters to the timeline to 'repeat', 'yo-yo' and add callbacks for the entire timeline. You can even adjust the entire timeline speed using the 'timeScale' property. Very cool.



The MorphSVG and DrawSVG plugins allow you to create incredible SVG animations, as seen here in these NFL logos by Shane Mielke



POWER UP GSAP WITH PLUGINS

Add more features and functionality to designs

CSSPLUGIN

greensock.com/docs/Plugins/CSSPlugin

Allows you to animate almost any CSS property including 2D and 3D transforms and colours. This plugin automatically checks to see if the object you are animating is a DOM element, then the engine creates that CSS object for you!

TEXTPLUGIN

greensock.com/TextPlugin

The TextPlugin animates the text of any DOM element. It replaces it one character or one word at a time. So when the tween is finished, the DOM element's text has been completely replaced. You can create incredible text effects with this plugin!

PHYSICS2DPLUGIN

greensock.com/Physics2DPlugin

Provides basic physics functionality for animating an object position, based on velocity, angle, gravity, acceleration, 'accelerationAngle', and/or friction. It is incredibly simple to use to create simple physics simulations without the overhead of a full physics engine.

BEZIERPLUGIN

greensock.com/BezierPlugin-JS

This plugin allows you to animate virtually any property along a curved Bezier path. You can define the path as an array of points/values that can be interpreted multiple ways, giving loads of flexibility and dynamics to your motion or animation paths.

MORPHSVGPLUGIN

greensock.com/docs/Plugins/MorphSVGPlugin

Allows you to control tweens that morph SVG paths. This allows you to morph one shape to another. This can be used to make unique and engaging transitions from one object to another, morph buttons or SVG text. The possibilities are endless.

DRAWSVGPLUGIN

greensock.com/drawSVG

This handy plugin allows you to reveal or hide SVG strokes progressively, to create a 'drawing' animation of SVG paths. You can see a great example of this in action in the image above in Shane Mielke's NFL logo animations.



Try out a digital edition
for **FREE** today!

Just search for 'ImagineFX' on these selected platforms...



Add multi-language support to Angular

Providing more than single language support ensures your content can be read by more of the world's online population



 **DOWNLOAD TUTORIAL FILES**
www.filesilo.co.uk/webdesigner

In this tutorial we're going to take you through the process of making your app accessible and user friendly for people around the world.

Only about 20% of the world speaks English so providing other language options can greatly increase your app's reach. We're going to take a look at Angular's built-in internationalisation tools and show you how to correctly use them.

We've created a very simple demo app to demonstrate the process. Clone it from here and then follow the installation instructions:

<https://github.com/danielcrisp/angular-i18n-demo>

Start the app to familiarise yourself with it. It just displays and updates random numbers and values with different contexts, eg currencies, dates etc. We'll cover some of the pipes and features used during the tutorial.

1. An introduction

There are two words that are often used interchangeably when talking about translating an app – internationalisation and localisation – however, they actually mean slightly different things. Internationalisation refers to the process of preparing your app for supporting different languages. In contrast, localisation refers to the process of actually translating your app into your required languages. Essentially internationalisation is something you do once per app, and localisation happens once per locale – at least that's the plan.

These terms might also be familiar in their shortened versions: i18n (where 18 is the number of letters between the first 'i' and the last 'n' of internationalisation) and l10n (where 10 is the number of letters between the 'i' and the 'n' of localisation).

2. What's localisation?

There are over 6,000 languages used around the world today, most of which are only used by very small groups of people. Yet even if we only focus on the top three languages – Mandarin, Spanish and English – there will be significant differences in date formatting, grammatical structure, pluralisation and number formatting.

If we include the fifth most widely used language – Arabic – we encounter another difference; Arabic is a right-to-left (RTL) script which means the UI will also have to be mirrored.

So during localization we have to consider grammar, layout and formatting differences, and of course, we also have to change the text itself. Angular can help with much of this but you'll still need to manually translate the text.

3. Locales

We will need to localise for each locale we need to support. A locale refers to the general set of preferences for the considerations mentioned above that tend to be shared within a region of the world, typically a country. Each locale is represented by a Unicode locale identifier, which specifies the language code and the locale extension.

Angular's default locale is 'en-US', which is the language code 'en' (English) as spoken in the region 'US' (United States of America). An app localised for 'en-US' will be



subtly different from an app localised for 'en-GB' which is English as spoken in Great Britain. For example, in the US dates are (bafflingly) formatted mm/dd/yyyy, whereas here in the UK we use the more sensible dd/mm/yyyy approach. This minor difference can result in a major error in comprehension.

To make things interesting let's localise our demo app for Arabic as spoken in Iraq, aka 'ar-IQ' and English as spoken in the UK, aka 'en-GB'. We'll use English as the default this time.

4. Build configuration

Our demo project was created using Angular CLI, which includes some useful tooling. We're going to use the Ahead-of-Time (AOT) compiler for this project so we need to make some changes to the CLI's configuration file: 'angular.json'. If you want to use Just-in-Time (JIT) you need to configure things slightly differently.

With an AOT build you get a small, faster rendering ready-to-go application which loads without the need for asynchronous requests to fetch things like templates and stylesheets. As a result you must create a build for each locale and serve the appropriate build using the URL or some kind of server-side language detection logic. The simplest approach is to create a directory for each locale, eg www.example.com/en-GB and www.example.com/ar-IQ. The trade off is that you can't switch language on-the-fly, but in reality that is unlikely to be something required by real users.

First of all we need to add a build configuration for our Arabic locale. In the JSON file look for the 'architect.build.configurations' object. Add the following block to define a configuration for the locale:

```
"ar-IQ": {
  "baseHref": "/ar-IQ/",
  "deployUrl": "/ar-IQ/",
  "outputPath": "dist/angular-i18n-demo/ar-IQ",
  "i18nFile": "src/locale/messages.ar-IQ."
}
```

```
xIf",
  "i18nFormat": "xIf",
  "i18nLocale": "ar-IQ"
```

This configuration tells Angular where to output the compiled build and which translations file and format to use. It also sets the locale and tells Angular which directory the app will be deployed to.

We also need to modify the default options in 'architect.build.options' to use the 'en-GB' locale. Set the following properties as shown. Note we're enabling AOT here across the board so it will be used for production and development builds:

```
"outputPath": "dist/angular-i18n-demo/en-GB",
  "i18nLocale": "en-GB",
  "deployUrl": "/en-GB/",
  "baseHref": "/en-GB/",
  "aot": true
```

Angular supports a number of locales. Make sure you use the correct value for the 'i18nLocale' property. You can see the complete list here: <https://github.com/angular/angular/tree/master/packages/common/locales>

Behind the scenes the above configurations simply load and read from one of these locale preference files.

5. Serve configuration

In addition to configuring the build output we also need to set up the configuration for the 'ng serve' command for development. This is more straightforward as we can simply reference the build configuration we just added. In 'angular.json' add the following block to 'architect.serve.configurations':

```
"ar-IQ": {
  "browserTarget": "angular-i18n-demo:build:ar-IQ",
  "servePath": "/ar-IQ/"}
```

Here we are referring the build configuration options using the 'browserTarget' property, and we're also setting

Tutorials

Add multi-language support to Angular

Value at this moment in time</note>' and '<note priority="1" from="meaning">Card title</note>'. There are also sections for 'closingValue' and 'watchers'." data-bbox="46 120 634 445"/>

```
17 messages.ar-IQ.xlf
You, 12 hours ago | 2 authors (you and others)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
3   <file source-language="ar-IO" datatype="plaintext" original="ng2-template">
4
5     <body>
6       <trans-unit id="title" datatype="html">
7         <source>Current value</source>
8         <target> آخذ این</target>
9         <context-group purpose="location">
10           <context context-type="sourcefile">app/app.component.html</context>
11           <context context-type="linenumber">3</context>
12         </context-group>
13         <note priority="1" from="description">Value at this moment in time</note>
14         <note priority="1" from="meaning">Card title</note>
15       </trans-unit>
16       <trans-unit id="closingValue" datatype="html">
17         <source>Yesterday's closing value was<br/><!-- ID=INTERPOLATION --> {{ closingValue | currency:$apos;GBP$apos; }}</source>
18         <target> امس<br/><!-- ID=INTERPOLATION --> {{ closingValue | currency:$apos;GBP$apos; }}</target>
19         <context-group purpose="location">
20           <context context-type="sourcefile">app/app.component.html</context>
21           <context context-type="linenumber">7</context>
22         </context-group>
23         <note priority="1" from="description">Value when the market closed yesterday</note>
24         <note priority="1" from="meaning">Closing value</note>
25       </trans-unit>
26       <trans-unit id="watchers" datatype="html">
27         <source>There<br/><!-- ID=10 --> {{ watchers, plural, =0 (...) +1 (...) other (...) }} > watching right now.</source>
28         <target>
29           <!-- ID=10 --> {{ watchers, plural, =0 (...) +1 (...) other (...) }} > اینجا<br/></target>
30         <context-group purpose="location">
31           <context context-type="sourcefile">app/app.component.html</context>
32           <context context-type="linenumber">8</context>
33         </context-group>
34         <note priority="1" from="description">Number of people watching the value</note>
35         <note priority="1" from="meaning">Watchers</note>
36       </trans-unit>
37       <trans-unit id="2b5f9af1eff0ac5879ff98af75e87a2599228b" datatype="html">
38         <source>{VAR_PLURAL, plural, =0 {is nobody} =1 {is one person} =2 {are two people} other {are<br/><!-- ID=INTERPOLATION --> {{ watchers }}}</source>
39         <target>{VAR_PLURAL, plural, =0 {is nobody} =1 {is one person} =2 {are two people} other {are<br/><!-- ID=INTERPOLATION --> {{ watchers }}}</target>
40         <context-group purpose="location">
41       </context-group>
42     </file>
43   </xliff>
```

the 'servePath'. Before we can either serve or build the Arabic app we need to create the translations file referenced in the 'i18nFile' property above. Angular CLI includes a tool for extracting flagged text into an industry-standard translation source file.

We'll cover these files in more detail later on in the tutorial but for now we just need to export the basic, empty file to allow us to compile.

We'll use the 'ng xi18n' command with the following options. This is the only time we'll include the locale ID in the '--out-file' filename:

```
$ ng xi18n --output-path locale --out-file messages.ar-IQ.xlf --i18n-locale ar-IQ
```

This should create a file in a src/locale directory. From now on we'll always output the file named 'messages.xlf' and manually copy it over the version with the locale ID in the name. The reason for this is to prevent the extraction tool from overwriting any existing translations we've added to the file.

6. Switching configuration

At this point we can now compile the project and see what happens, but we need to tell the 'ng serve' command which configuration to use. First let's take a look at the English version. No changes here because English is the default:

```
$ ng serve
```

As you can see it looks much like the original version, which uses Angular's default locale of 'en-US'. The notable difference is the currency now specifies US\$ instead of just \$. Okay, now let's try the Arabic version. Stop the English version and run:

```
$ ng serve --configuration=ar-IQ
```

As you'd expect there are more obvious differences in this version, in particular the date is now written in Arabic. Angular can do this because the names of some things,

such as months and days, are from a set list and ultimately they relate to a known number. Everything else, however, is still in English.

7. Locale-aware pipes

Take a closer look at the source code of 'app.component.html' and you'll see that we use a number of different pipes. The following Angular pipes are locale-aware, meaning that they adapt their output based on the current locale: 'DatePipe', 'CurrencyPipe', 'DecimalPipe' and 'PercentPipe'.

If you use these pipes carefully Angular will handle a lot of the localisation legwork for you. By carefully we mean use the available predefined options wherever you can. A good example is the US vs UK date formatting we mentioned earlier. If you're in the UK and you want to display a date using the (sensible) day-month-year format, you might be frustrated to find that the predefined "shortDate" option renders as m/d/yy (eg. 10/9/18) and be tempted to hardcode your desired format like this:

```
{{ myDate | date: 'dd/MM/y' }}
```

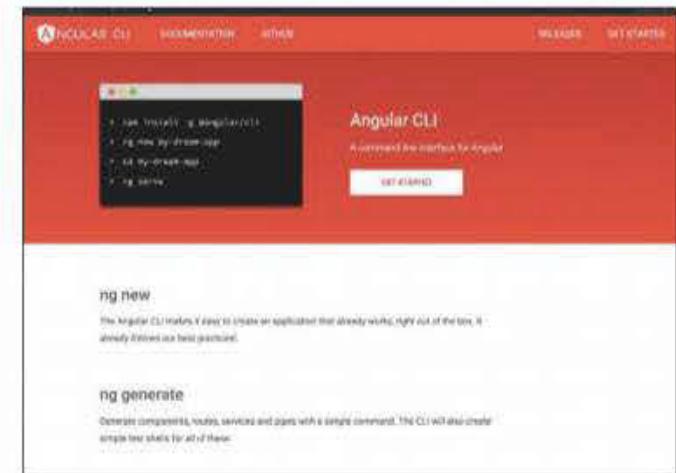
But we now know that we get the m/d/yy format because Angular uses the 'en-US' locale by default. So instead of hardcoding the format we should use the "shortDate" option and localise our app to use 'en-GB'.

```
{{ myDate | date: 'shortDate' }}
```

It takes a tiny bit more effort but then we can add locales to our heart's content and always have a user-friendly date format.

8. Overriding the predefined options

Unfortunately it doesn't seem that there is an easy, built-in way to override a predefined format. For example you can't just decide that you'd prefer the "shortDate" format to be dd/mm/yyyy instead of dd/mm/y as there is no way



to modify the format at runtime. Also you can't add your own predefined options.

For these edge cases you could create a custom date pipe which wraps the Angular 'DatePipe' and handles any custom formats per-locale. Anything it doesn't recognise would be passed on to the built-in 'DatePipe'.

9. CurrencyPipe

Off the shelf the 'CurrencyPipe' will format a number as US Dollars, trim to two decimal places and add groupings as defined in the locale's preferences.

You'll notice that in both our locales the currency is always in US Dollars. It doesn't magically switch to Sterling (GBP) when you use the 'en-GB' locale. The reason for this is that £10 is not the same as \$10, so you must explicitly specify the currency your number refers to.

Let's update 'app.component.html' to use GBP throughout. When specifying the currency code you must use the correct value from the ISO 4217 standard (list available online).

Modify the two currency pipes by adding ':GBP' like so:

```
  {{ value$ | async | currency: 'GBP' }}
```

And you'll start seeing the £ symbol instead of US\$.

Remember, it doesn't do anything clever like automatically convert USD to the equivalent value in GBP if you change the currency - it just changes the symbol it uses.

10. Translation workflow

Okay, so we've got our two locales configured and Angular is helpfully doing some of the work for us out of the box, but the text is all still in English. Angular can't translate this automatically sadly but it can help us with parts of the workflow. This is what has to happen:

- Flag static text in all components for translation
- Export translation file containing this static text
- Modify the translation file and add the relevant translations

• Merge translated translation file back into app

Angular helps us with steps 2 and 4, but as developers we need do step 1 manually. Step 3 would typically be completed by a translation professional or agency, using special software to read and update the translation file.

11. Axis details

To achieve this we have to add a special attribute to every element that contains fixed text to be translated. To be clear if the content arrives from an API then that isn't fixed text and you'd need to localise that in the API. You only

need to add the attribute when the text is written directly in the HTML template in your source code. A key point here is that you should try to keep your TypeScript files locale-agnostic - in other words, avoid putting any text that needs to be translated in the component logic and keep it all in the templates. Otherwise the extraction tool won't be able to extract it. It's good practice anyway to separate your concerns - in life and in code.

Let's open up 'app.component.html' and start with the 'Current value' title. Simply add the 'i18n' attribute to the element that directly contains the text.

```
<div class="meta__title" i18n>
  Current value
</div>
```

It's important to understand that this is just a 'dumb' custom attribute. It isn't an Angular directive that triggers anything at runtime, in fact, the compiler removes it after translation.

Anyway, let's see what happens when we run the extraction tool again to regenerate the translation file. Remember '--out-file' is just 'messages.xlf' now:

```
$ ng xi18n --output-path locale --out-file
messages.xlf --i18n-locale ar-IQ
```

Open up the output XLF file and you should see a new translation unit block that looks something like this with some additional context information:

```
<trans-unit id="face3d45c0f0cd38b726e7798da15
3e2f8d5551" datatype="html">
```

```
  <source>
    Current value
  </source>
```

Great, that means the tool picked up the 'i18n' attribute. That long ID is generated by the tool and will stay the same unless the text changes. If you have multiple instances of exactly the same text they will all get the same ID. Don't edit this ID!

If you prefer, you can specify a custom ID within the 'i18n' attribute. If you do this the ID will remain the same even if the text changes, so you need to be sure you don't have any ID collisions throughout your app. Use the '@@' prefix to set a custom ID. Here the ID will become 'title':

```
<div class="meta__title" i18n="@@title">
  Current value
</div>
```

12. Adding some context

To ensure the translator is able to provide an accurate translation they will often need to know the context that the text is being used in. The 'i18n' attribute allows us to define a description and a meaning to help the translator. The format is as follows:

```
<div i18n="meaning|description@@
customId">Text</div>
```

Let's update our title with a meaning and description:

```
<div class="meta__title" i18n="Card
title|Value at this moment in time@@title">
  Current value
</div>
```

That should give the translator enough context to provide an accurate translation. Regenerate the translation file and you should see these values have been output. It's worth noting that if you don't use a custom ID the

The screenshot shows the homepage of the ngx-translate library. At the top, there's a logo and the text "The internationalization (i18n) library for Angular". Below that is a navigation bar with links to "CORE / LOADERS / EXTRACTOR / EDITORS / DEMO". The main content area has two sections: "What is ngx-translate?" and "How to use ngx-translate?". The "What is ngx-translate?" section describes it as an internationalization library for Angular, letting you define translations for your content in different languages and switch between them easily. It also mentions it's modular and easy to replace parts. The "How to use ngx-translate?" section explains the core component and how to choose a loader. There are also links to documentation and a demo.

Translating at runtime

Angular's built-in 'i18n' approach only works with one language at a time. This is great for performance because it means it doesn't create a binding for every bit of text. The trade-off, however, is that you have to completely reload the app when you switch language and you need a build for each locale. If you do want a solution that allows you to translate at runtime and only create one production build of your app then take a look at 'ngx-translate'. It's been around for years, has a comprehensive API, supports text in templates and components, and can handle any translation file format you like.

<https://github.com/ngx-translate/core>

generated ID takes the meaning and the text into account. So the same text, but with a different meaning, will get a different ID. The description, however, has no impact on the ID.

13. Text with variables

Let's move on to the intro section. The first paragraph contains text and a variable which will be interpolated at runtime. How do we handle this?

Well happily it is quite straightforward. Again we need to add a meaningful 'i18n' attribute to the containing element. Add it directly to the paragraph element:

```
<p i18n="Closing value|Value when the market
closed yesterday@closingValue">
```

Run the extraction tool again and you'll see this new translation unit:

```
<trans-unit id="closingValue"
datatype="html">
  <source>Yesterday's closing value was
<x id="INTERPOLATION" equiv-text="{{ closingValue | currency:'GBP'}}></source>
```

See how the variable interpolation has been detailed in the output. The nice thing about this is it allows the translator to modify the grammatical structure of the sentence if necessary, without breaking the binding. For example, there may be a language where the sentence would be best written: X value was yesterday's closing, ie with the variable at the start.

14. Pluralisation

Moving on to the next paragraph you'll see some intimidating syntax. This is called ICU Message Format and it allows you to specify different chunks of text based on the value of a variable.

You can use this to add the 's' to words in English when the value is zero or not one. For example, if 'seconds' is a variable containing the number of seconds we can use this ICU pluralisation expression:

```
  {{ seconds }} {seconds, plural, one
  {second}, other {seconds}}
```

Which will output:

• 0 seconds

• 1 second

• 2 seconds

etc

It doesn't appear to be documented but you can also use the 'AsyncPipe' inside the pluralisation syntax to work with Observables.

In that example 'one' and 'other' are pluralisation categories. There are a number of categories to choose from, but beware! Not all locales support all the categories, and Angular doesn't tell you if you try to use a category that isn't supported by the current locale. It is easy to end up thinking that you've done something wrong because the 'two' category isn't working in your 'en-GB' locale and instead you are seeing the 'other' text. Inexplicably 'en' (and many other common languages) only support 'one' and 'other', even though 'zero' and 'two' are explicit values.

Check out this file to see what's actually supported:
<https://github.com/angular/angular/blob/master/packages/common/src/i18n/localization.ts>

15. The multiple radial bar charts

We can workaround this limitation by using numbers instead of categories. Just prefix the value with an '=':

```
There {watchers, plural, =0 {is nobody} =1
{is one person} =2 {are two people}
other {are {{ watchers }} people}}
watching right now.
```

This is already set up in the demo app, we just need to add the 'i18n' attribute to the containing paragraph:

```
<p i18n="Watchers|Number of people
watching the value@watchers">
```

Run the extraction tool again to see how this looks. You'll see that this is output slightly differently. It will create two translation units; one for the ICU expression itself and one which interpolates that expression into the original string.

16. Select

If you want to display different text depending on the value of a variable you can use a 'select' ICU expression which is very similar to the 'plural' syntax demonstrated above. In our demo app we monitor the change applied to the value and create an Observable stream called

Tutorials

Add multi-language support to Angular

'trend\$' which outputs 'up', 'down' or 'stable' depending on whether the change is positive, negative or zero.

We then hook up our 'select' ICU expression to output a different string depending on the stream value. Here you can see the 'AsyncPipe' in use:

```
The value {trend$ | async, select, up
{increased} down {decreased} stable
{didn't change}}.
```

This is a somewhat cleaner syntax than using 'ngIf' or 'ngSwitch' to manipulate the DOM, plus it also plays nicely with the extraction tool. Add the 'i18n' attribute to the containing element:

```
<div class="card__info" i18n="Value
trend|Describes the value change trend@trend">
Regenerate the translations file and you'll see the approach is similar to the plural output, with two translation units created. ICU expressions are pretty handy once you get used to them, plus you can nest them to create more complex outputs.
```

17. Adding translations

One more 'i18n' attribute to add:

```
<div class="card__info" i18n="Transactions
count|Number of transactions today@@
transactions">
  Transactions: {{ transactions$ | async |
  number }}</div>
```

Now we've marked up all the text that needs translating we can generate the translation file one last time. Once it is created rename it to 'messages.ar-IQ.xls' and replace the previous incarnation. This is the file we'd be sending to the translation professional, but for the purposes of this tutorial, Google Translate will be standing in!

Open up the XLF file and duplicate every '<source>' element, renaming it '<target>'. Unfortunately it can be quite untidy so it might help to beautify the contents.

To check we've got them all, save the file and start the app with the Arabic locale:

```
$ ng serve --configuration=ar-IQ
```

If you see any messages in the terminal like this that means you've missed one:

```
ERROR in xliff parse errors: Message *id*
misses a translation ("
```

Hopefully you won't have any errors and you'll be able to see the app in the browser. We've not added any actual Arabic yet so it won't look much different.

18. Google Translate

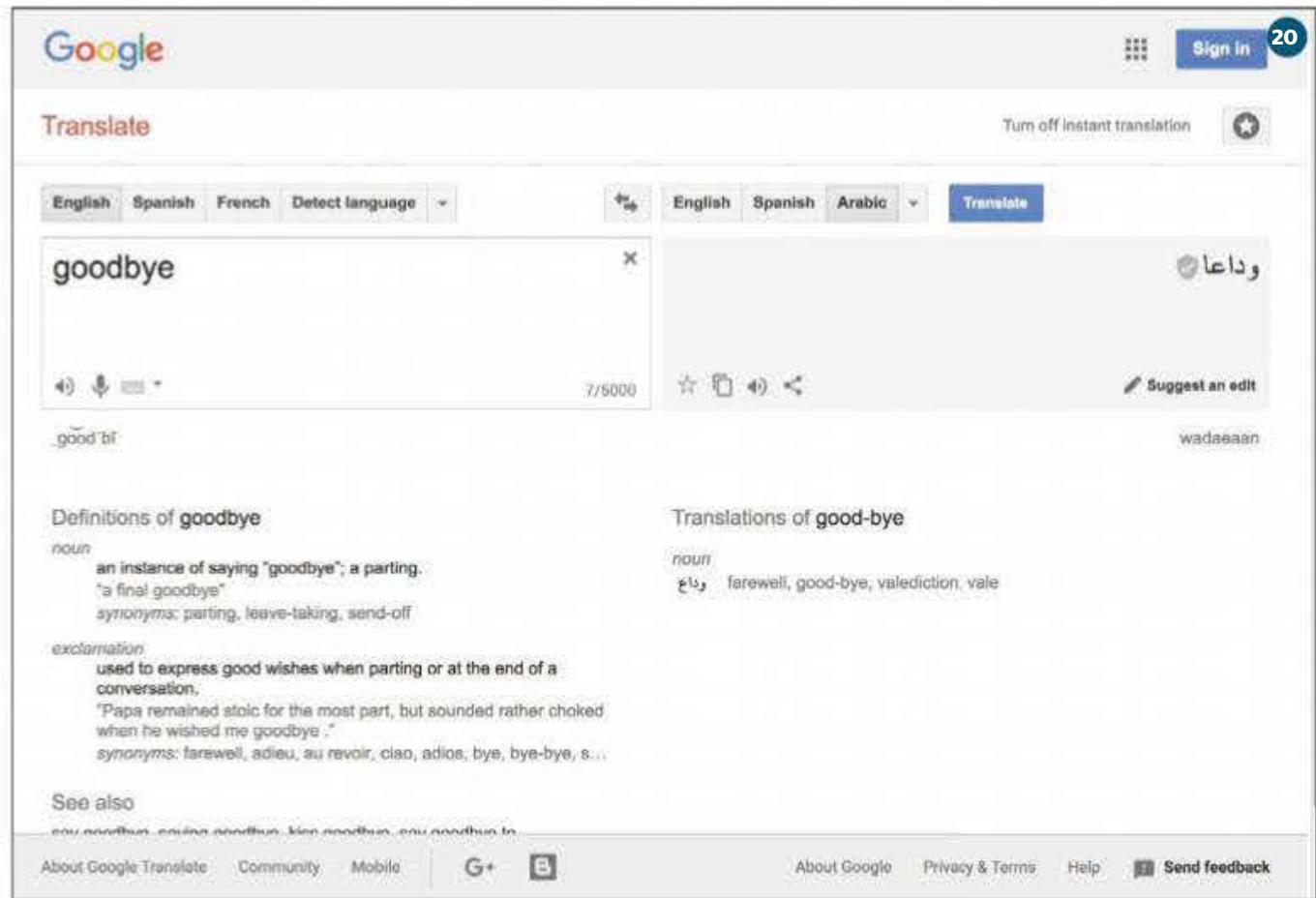
Let's start with something easy - the 'Current value' title. Google Translate tells me it should be (*Arabic text here*) so update the value in the '<target>' element:

```
<source>Current value</source>
<target>Arabic text here</target>
```

So far, so good. Now let's do one with interpolation. Here is "Yesterday's closing value was..." (hopefully!):

```
<target>Arabic text here<x
id="INTERPOLATION" equiv-text="{{ closingValue
| currency:'GBP' }}"/></target>
```

Use a number when you translate so you can see where the interpolation should be. Notice that when you see the translated result in Google Translate it will appear



reversed - ie the number at the start - but when you copy and paste it into the translation file it will return to the original order. This is happening because Arabic is an RTL language so the script is (almost) entirely mirrored. Google Translate does this by adding a 'dir="rtl"' attribute to the containing element. We'll learn how to do this in the next step. The rest of the translations are available in the demo repo, 'tutorial' branch.

19. Script direction

We need to manage the script direction in our app because Angular won't do this automatically for us. There also doesn't appear to be any way to detect if the current locale is an LTR or RTL language, so we'll need to hardcode this. It'd be great if Angular offered a built-in directive for this.

Open up 'app.component.ts'. Import 'Inject', 'LOCALE_ID' and 'HostBinding' from '@angular/core'. Then set up the 'HostBinding' as follows. This will add a 'dir' attribute to the AppComponent and set the default language direction to 'ltr':

```
@HostBinding('attr.dir') dir = 'ltr';
```

Next add a constructor and inject the 'LOCALE_ID'. Remember this is set by our configuration because we're using AOT.

```
constructor (@Inject(LOCALE_ID) private locale:
string) {}
```

And finally add the following snippet to the existing 'ngOnInit' method. Here we are checking if the 'LOCALE_ID', ie 'ar-IQ', starts with 'ar' and if it does change the direction to 'rtl' instead.

```
if (this.locale.startsWith('ar')) {
  this.dir = 'rtl';
}
```

If you plan to support more locales then you'll probably need to refactor this to make it more scalable, however, as there are only about ten RTL languages in use today this approach shouldn't be too unwieldy. Start the

Arabic app and you should now see that the UI is mirrored - the £ sign should be on the right.

20. Production

The final step is to generate and check our production builds. First, though, we need to make another quick modification to the 'angular.json' configuration.

In 'architect.build.configurations' duplicate the existing production object and rename it "production-ar-IQ". Then copy and paste the properties from the existing "ar-IQ" configuration into the object, so you have both the production options and the 'i18n' options.

You also need to update 'architect.serve.configurations' too. This time duplicate the existing "ar-IQ" object and rename it "production-ar-IQ" and change the 'browserTarget' value to point to your new 'production-ar-IQ' configuration.

Now you can build and serve your production Arabic locale with this command:

```
$ ng serve --configuration=production-ar-IQ
```

Okay, we're done! We've successfully internationalised our app, and localised it for 'en-GB' and 'ar-IQ' audiences. Angular makes the process remarkably straightforward for the developer, in fact, the hardest bit is figuring out what the translations should be - apologies to any Arabic speakers if anything is wrong!

Further reading

- There's a great guide on the Angular site about i18n. <https://angular.io/guide/i18n>
- Also take a look at the documentation for the extraction tool. <https://github.com/angular/angular-cli/wiki/xliff>
- If you want to find out how to serve your different locale bundles in Node.js take a look at this tutorial. <https://angular-templates.io/tutorials/about/angular-internationalization-i18n-multi-language-app>
- And finally there is a neat little tool for helping you to manage the merging of your translations files. <http://angular-translator.elol.fr/en>

OFFER ENDS
31 DECEMBER 2018



FROM £12.50 EVERY 3 MONTHS
(€73 /\$93 PER YEAR)

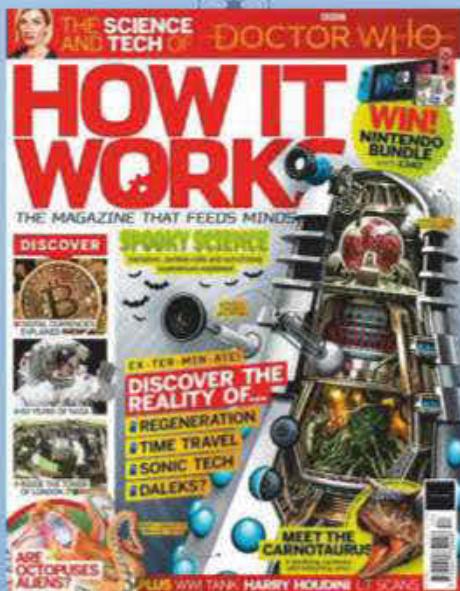


FROM £12.50 EVERY 3 MONTHS
(€104 /\$147 PER YEAR)

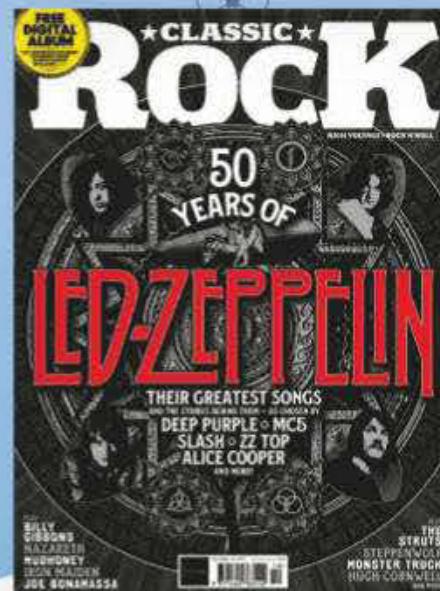


FROM £12.50 EVERY 3 MONTHS
(€104 /\$147 PER YEAR)

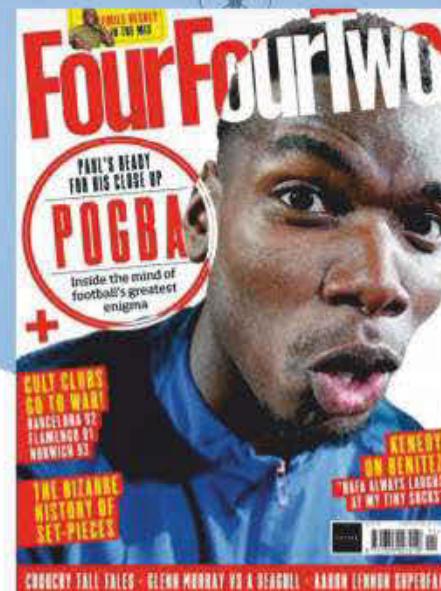
The perfect gift from just £7.50*



FROM £7.50 EVERY 3 MONTHS
(€70/\$87 PER YEAR)



FROM £8.75 EVERY 3 MONTHS
(€86/\$95 PER YEAR)



FROM £9.99 EVERY 3 MONTHS
(€60/\$62 PER YEAR)

Delivery included in the price

Free personalised e-card when buying for someone else

Buy as a gift or treat yourself!

Choose from a huge range of titles

SEE THE FULL RANGE AND ORDER ONLINE
www.myfavouritemagazines.co.uk/xmas187

ORDER HOTLINE: 0344 848 2852

PLEASE QUOTE XMAS187 WHEN ORDERING BY PHONE

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



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

Turn data into interactive diagrams

Discover the power of the Chart.js library to create dynamic charts and diagrams and make data more engaging



DOWNLOAD TUTORIAL FILES
www.filesilo.co.uk/webdesigner

In theory, creating diagrams is not difficult: handling the trigonometry required for a pie chart is among the most classic jobs used for training programmers.

Reinventing the wheel every time an aircraft is to be built gets tedious quickly – a lot of diagramming libraries vie for developer's attention.

This story is based on Chart.js: the open-source product tops usage statistics due to a unique trade-off between ease of use and advanced features. We will put the library through its paces, creating a set of printable diagrams from randomly generated data.

Thanks to the library, eight different chart types appear in your websites with minimal effort. The recently-added animations module enriches the diagrams with snazzy-looking visuals suited to amaze all but the most sophisticated users.

In addition to that, we will also risk a look across the pond by looking forward to learning more about visualisation theory and alternative products better suited to edge use cases.

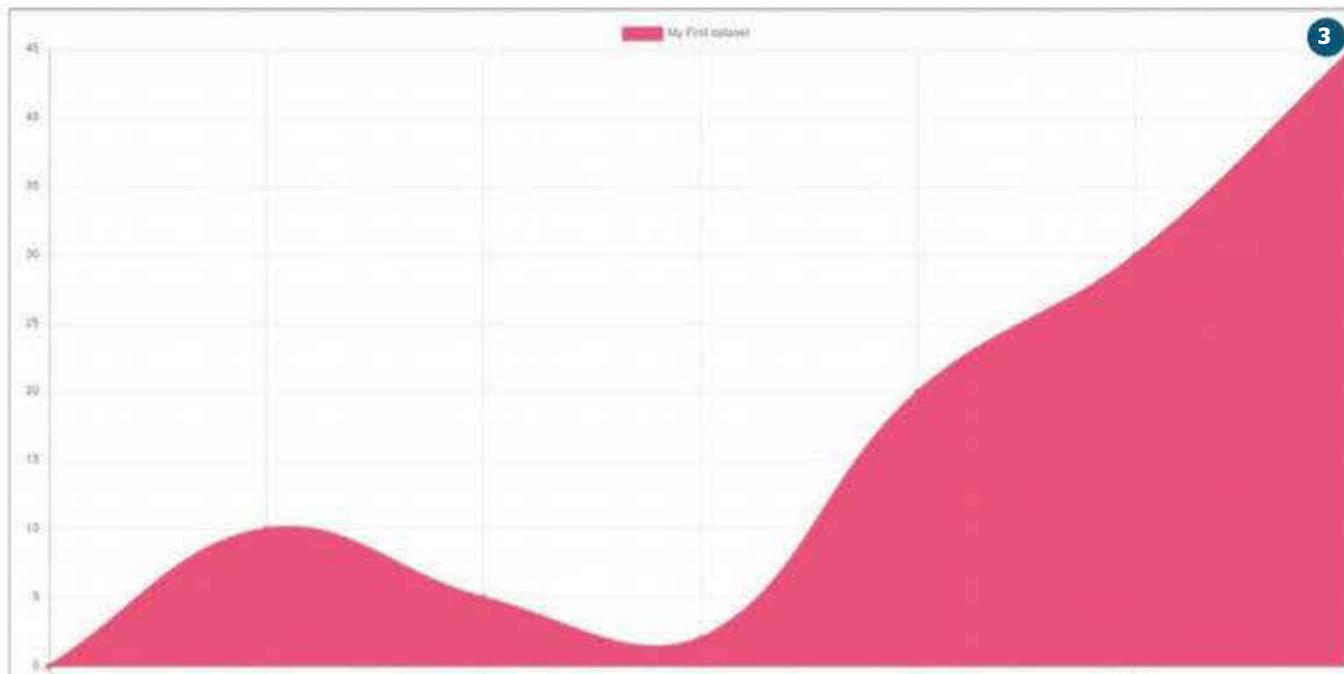
1. Deploy the library

Due to the library being widespread, you can find its minified version from various CDNs such as Cloudflare. Simply load it with a `<script>` tag, and ensure that a `<canvas>` object is nearby – Chart.js does its magic by using the rendering infrastructure provided in this widget.

```
<html>
  <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.4.0/Chart.min.js">
    </script>
  </head>
  <body>
    <canvas id="workArea">
    </canvas>
  </body>
</html>
```

2. Start initialisation

When the framework is loaded, the next step involves getting a Canvas2D handle pointing at the `<canvas>`



instance created in the preceding step. This can then be used to create a new instance of the 'Chart' class, which is responsible for data handling, rendering and interaction.

```
<script>
  (function() {
    var ctx = document.getElementById('workArea').getContext('2d');
    var chart = new Chart(ctx, {
      type: 'line',
      options: {},
      ...
    });
  })();
</script>
```

3. Add a data source

Chart classes take a data pointer which provides one or more 'DataSet' objects containing the actual measurement information. In the case of our line chart, only one DataSet is needed. It comes with the obligatory data element and a few optional parameters governing the actual display process.

```
var chart = new Chart(ctx, {
  type: 'line',
  options: {},
  data: {
    labels: ["January", "February", "March", "April", "May", "June", "July"],
    datasets: [{
      label: "My First dataset",
      backgroundColor: 'rgb(255, 99, 132)',
      ...
    }]
  }
});
```

```
borderColor: 'rgb(255, 99, 132)',  
data: [0, 10, 5,  
2, 20, 30, 45],  
}  
}  
));
```

4. Stop flickering

Chart.js comes with sophisticated resizing logic, which – unfortunately – tends to get screen sizes wrong from time to time. Limiting the size of the `<canvas>` element via CSS, sadly, does not do the trick. Instead, both aspect ratio maintenance and responsiveness must be disabled via the options field accompanying the Chart classes constructor.

```
<canvas id="workArea" style="width:800px;  
height:600px;"></canvas></canvas>  
<script>  
  (function() {  
    var ctx = document.getElementById('workArea').getContext('2d');  
    var chart = new Chart(ctx, {  
      type: 'line',  
      options: {  
        responsive: false, maintainAspectRatio: false},  
    });
  })();
</script>
```

5. Add layouts...

To paraphrase Andrey Tupolev: now that the small one is flying, let us bring up the bigger one. The scaffolding shown accompanying this step 'multiplies' our diagram – instead of working with one Canvas element, we now create a total of four of them and arrange them on the screen in a fashion similar to a LeCroy oscilloscope. Sadly, this does not quite work out as intended.

```
<canvas id="workArea"  
style="position:absolute; top:0%; left: 0%;  
width:49%; height:49%;"></canvas></canvas>  
<canvas id="workArea2"  
style="position:absolute; top:0%; left: 51%;  
width:49%; height:49%;"></canvas></canvas>  
<canvas id="workArea3"  
style="position:absolute; top:51%; left: 0%;  
width:49%; height:49%;"></canvas></canvas>  
<canvas id="workArea4"  
style="position:absolute; top:51%; left: 51%;  
width:49%; height:49%;"></canvas></canvas>
```

6... And tame rendering errors

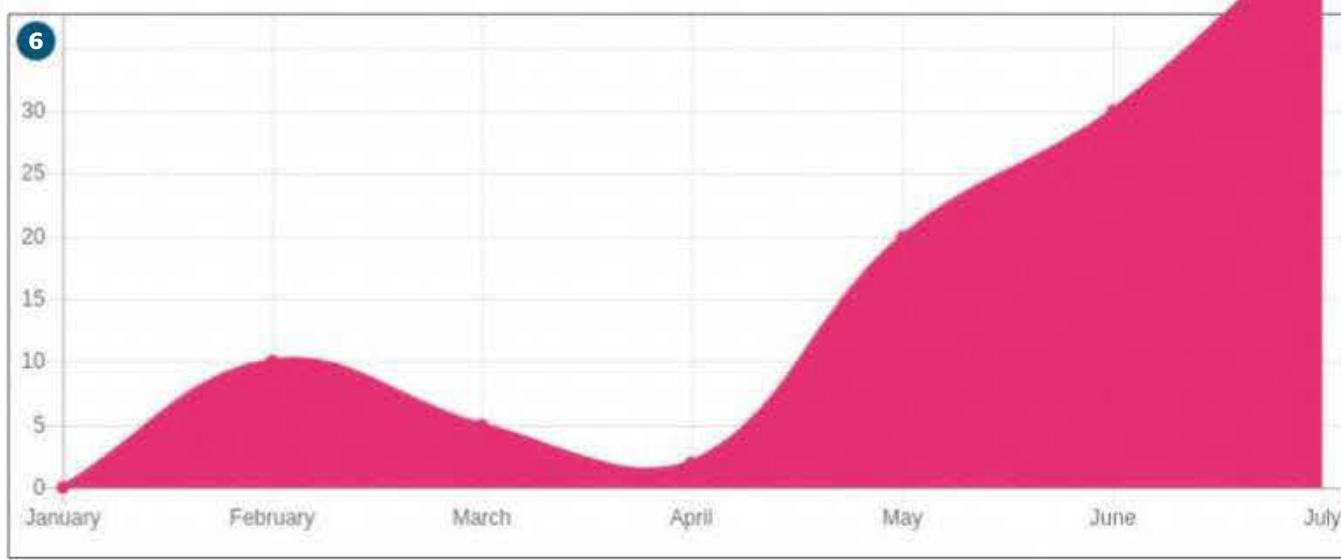
The safest way to handle Chart.js diagrams in complex layouts involves using wrapper `<div>` tags. They enforce a structure from the outside, thereby ensuring that the

Find more CDNs

Should you dislike Cloudflare for some reason, fret not: the product is also available from multiple other CDNs. Open <https://cdnjs.com/libraries/Chart.js> in a browser of choice.

Tutorials

Turn data into interactive diagrams



internal layout engine can not do more harm than necessary. In this case, however, ensure to re-enable the responsiveness feature.

```
<div style="position:absolute; top:0%; left:0%; width:49%; height:49%;">  
    <canvas id="workArea" ></canvas></div>  
  
<div style="position:absolute; top:0%; left:51%; width:49%; height:49%;">  
    <canvas id="workArea2"></canvas></div>  
  
<div style="position:absolute; top:51%; left:0%; width:49%; height:49%;">  
    <canvas id="workArea3" ></canvas></div>  
  
<div style="position:absolute; top:51%; left:51%; width:49%; height:49%;">  
    <canvas id="workArea4" ></canvas></div>  
  
<script>  
    document.addEventListener("DOMContentLoaded", function(){  
        var ctx = document.getElementById('workArea').getContext('2d');  
        var chart = new Chart(ctx, {  
            type: 'line',  
            options:  
            { }  
        })  
    })  
</script>
```

7. Bars are fun

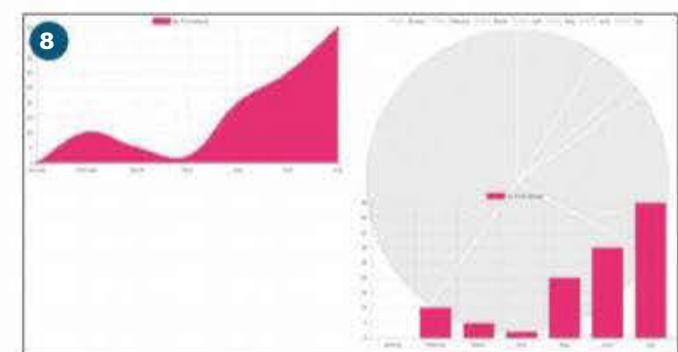
Always rendering line charts gets tedious quickly. Let's spruce things up by changing the type property to bar,

thereby yielding bar diagrams such as the one shown in the figure accompanying this step. We promote the data field to 'global' scope in order to eliminate reuse.

```
<script>  
    document.addEventListener("DOMContentLoaded", function(){  
        var myfield = {  
            labels: ["January", "February", "March", "April", "May", "June", "July"],  
            datasets: [{  
                label: "My First dataset",  
                backgroundColor: 'rgb(255, 99, 132)',  
                borderColor: 'rgb(255, 99, 132)',  
                data: [0, 10, 5, 2, 20, 30, 45],  
            }]  
        };  
  
        ctx = document.getElementById('workArea4').getContext('2d');  
        chart = new Chart(ctx, {  
            type: 'bar',  
            options: { },  
            data: myfield  
        });  
    })  
</script>
```

8. Pies cause trouble

In theory, a pie chart can be rendered along the same lines. Remove the colour properties to prevent uniform appearance, and set the type property to pie. Sadly, this



does not quite work out – when done, the pie chart will fill the entire screen. This is caused by a speciality of the pie renderer which uses the 'larger' of the two properties to determine pie radius.

```
var mypie = {  
    labels: ["January", "February", "March", "April", "May", "June", "July"],  
    datasets: [{  
        label: "My First dataset",  
        data: [0, 10, 5, 2, 20, 30, 45],  
    }];  
ctx = document.getElementById('workArea2').getContext('2d');  
chart = new Chart(ctx, {  
    type: 'pie',  
    options: { },  
    data: mypie  
});
```

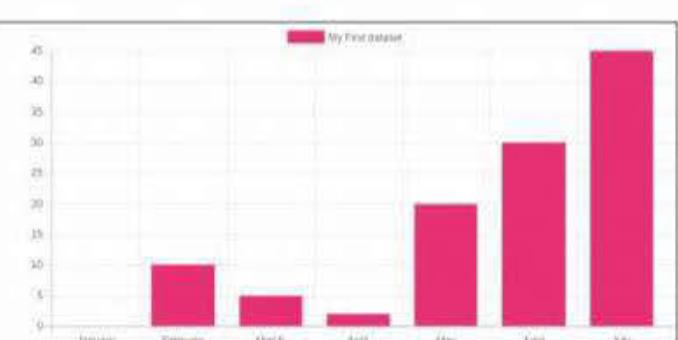
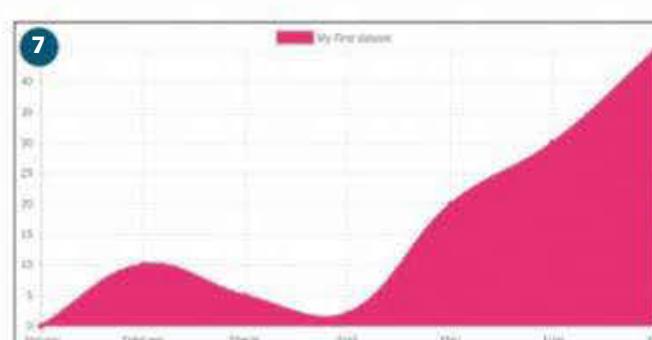
9. Solve the problem

Open the 'index.js' file created in step 1. Begin by specifying the graph type as line and adding the data to be visually represented as shown below.

```
<body>  
    <div style="position:absolute; top:0%; left: 0%; width:49%; height:49%;">  
        <canvas id="workArea" ></canvas>  
    </div>  
    <div style="position:absolute; top:0%; left: 51%; width:29%; height:49%;">  
        <canvas id="workArea2"></canvas>  
    </div>
```

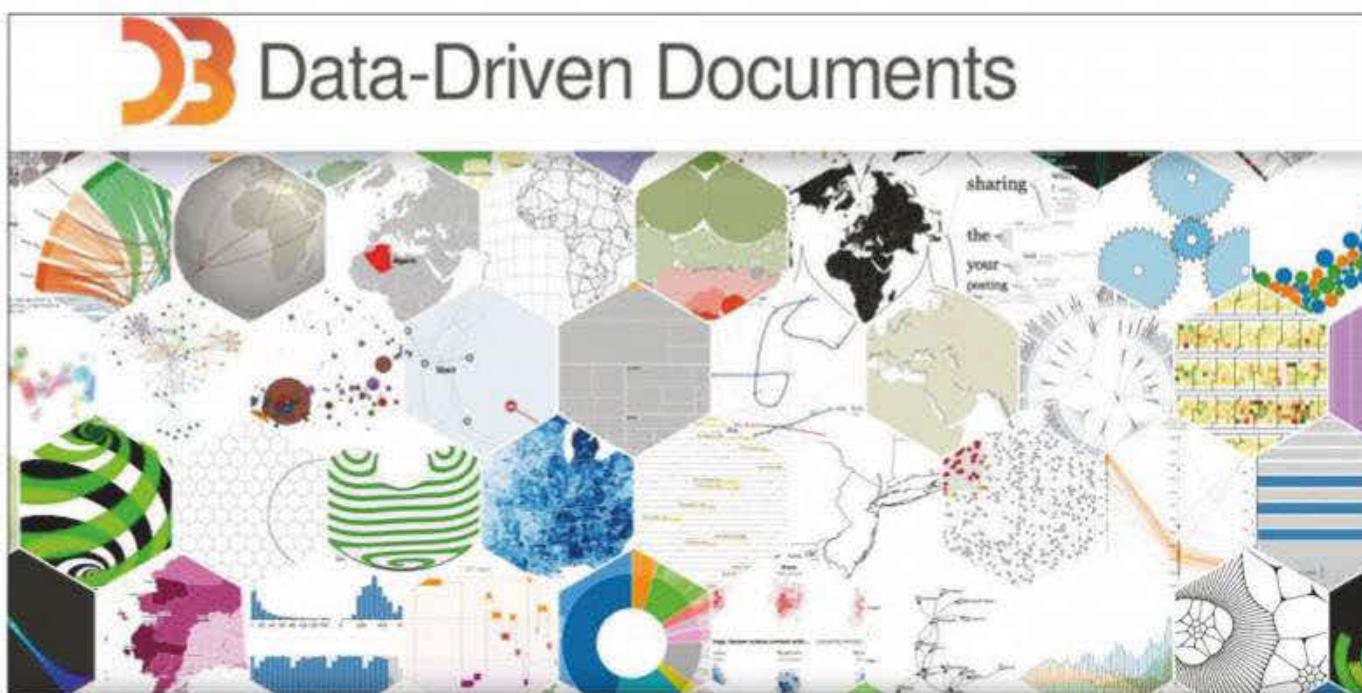
10. The smart approach...

Another approach to solve the problem involves re-disabling the 'maintainAspectRatio' feature. This way, the diagramming engine is allowed to rescale the



Don't go crazy

We can understand your frustration at the task ahead – after all, arranging elements 'by table' is not the best use of your time. Nevertheless, our target is the demonstration of possibilities for charting – ensuring perfect by-the-book CSS usage is a non-target.



Using D3 libraries

JavaScript tradeshows usually have at least one speaker who extols the values of the D3 library found at <https://d3js.org>. As the slides usually contain at least one insanely impressive slideshow, developers tend to 'take a stab'.

This usually ends in disaster for one simple reason: the product is immensely complex due to its strictly data-driven design. Think of a D3-based diagram like a 'shell' of `<div>` tags animated via data binding - an architecture as powerful as it is difficult to tame.

Should you feel like using D3.js in anger, an interesting approach involves the use of pre-made D3 libraries such as C3 found at <https://c3js.org>. They act as a 'wrapper' around the underlying D3 library and provide an easy-to-use interface not dissimilar to the one in Chart.js.

diagram as it sees fit, ensuring that the entire circle shows up on the screen.

```
ctx = document.getElementById('workArea2').getContext('2d');
chart = new Chart(ctx, {
    type: 'pie',
    options: {maintainAspectRatio:false},
    data: mypie
});
```

11. Improve pie colouring

Chart.js does not contain a random colour generator - if you don't provide a colour array, colours won't change. As designing systems based purely on colours is inefficient - many (often high IQ and affluent) individuals suffer from colour blindness. A nice way around the problem is the 'patternomaly' library.

```
<script src="https://cdn.jsdelivr.net/npm/patternomaly@1.3.0/dist/patternomaly.js"></script>
```

```
var mypie = {
    labels: ["January", "February", "March", "April", "May", "June", "July"],
    datasets: [
        {
            backgroundColor: [
                pattern.
                draw('square', '#ff6384'),
                pattern.
                draw('circle', '#36a2eb'),
                pattern.
                draw('diamond', '#cc65fe'),
                pattern.
                draw('triangle', '#ffce56'),
                pattern.
                draw('square', '#1f77b4'),
                pattern.
                draw('circle', '#ff7f0e'),
                pattern.
                draw('diamond', '#2ca02c'),
                pattern.
                draw('zigzag-horizontal', '#17becf'),
            ]
        }
    ]
};
```

```
pattern.
draw('triangle', '#7f7f7f')
    ...
}
```

12. Using patterns

Actually applying the pattern is not difficult. As shown above, simply instantiate them using the name from the figure and a colour screen to be used as a background. Ensure that the array contains enough elements to cover each member of the data field.

13. Tooltip issues

Running the program with enabled developer tools finds an interesting problem. When passing the mouse cursor over the chart elements, errors pop up. This is caused by the tooltip window, which is not able to discern colour information from pie elements loaded with a pattern.

14. Overwrite information

The problem at hand can be remedied by overriding parts of the tooltip's element. Chart.js lets you submit event handlers that get called as a tooltip window pops up - overwriting 'labelColor' disables the snooping algorithm responsible for the emission of the warning seen before.

```
chart = new Chart(ctx, {
    type: 'pie',
    options: {
        maintainAspectRatio:false,
        tooltips: {
            callbacks: {
                labelColor: function(tooltipItem, chart) {
                    return {
                        borderColor: 'rgb(255, 0, 0)'
                    }
                }
            }
        }
    },
    data: myfield
});
```

```
backgroundColor: 'rgb(255, 0, 0)'
}
},
labelText
Color:function(tooltipItem, chart){
    return '#543453';
}
}
),
data: mypie
});
```

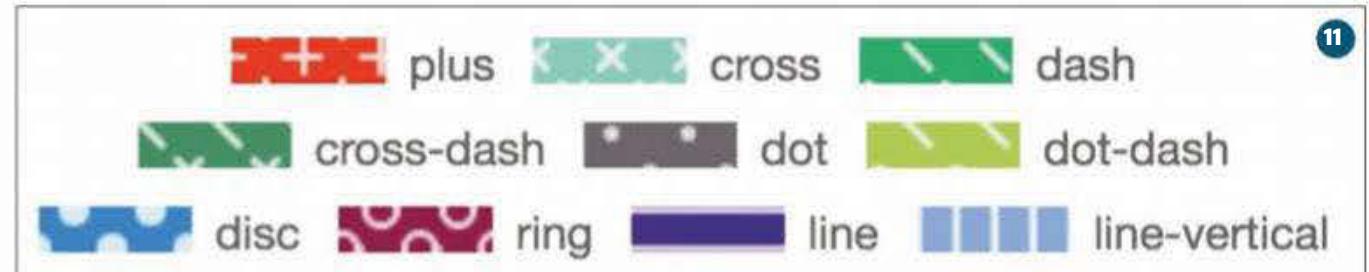
15. Add a title

Especially when diagrams are intended for export or saving, adding a title improves the meaningfulness of the information displayed. The code shown next to this step takes care of the problem effectively - additional customisation, such as the choice of fonts, can be accomplished with additional parameters.

```
var chart = new Chart(ctx, {
    type: 'line',
    options: {
        title: {
            display: true,
            text: 'Line Chart'
        }
    },
    data: myfield
});
```

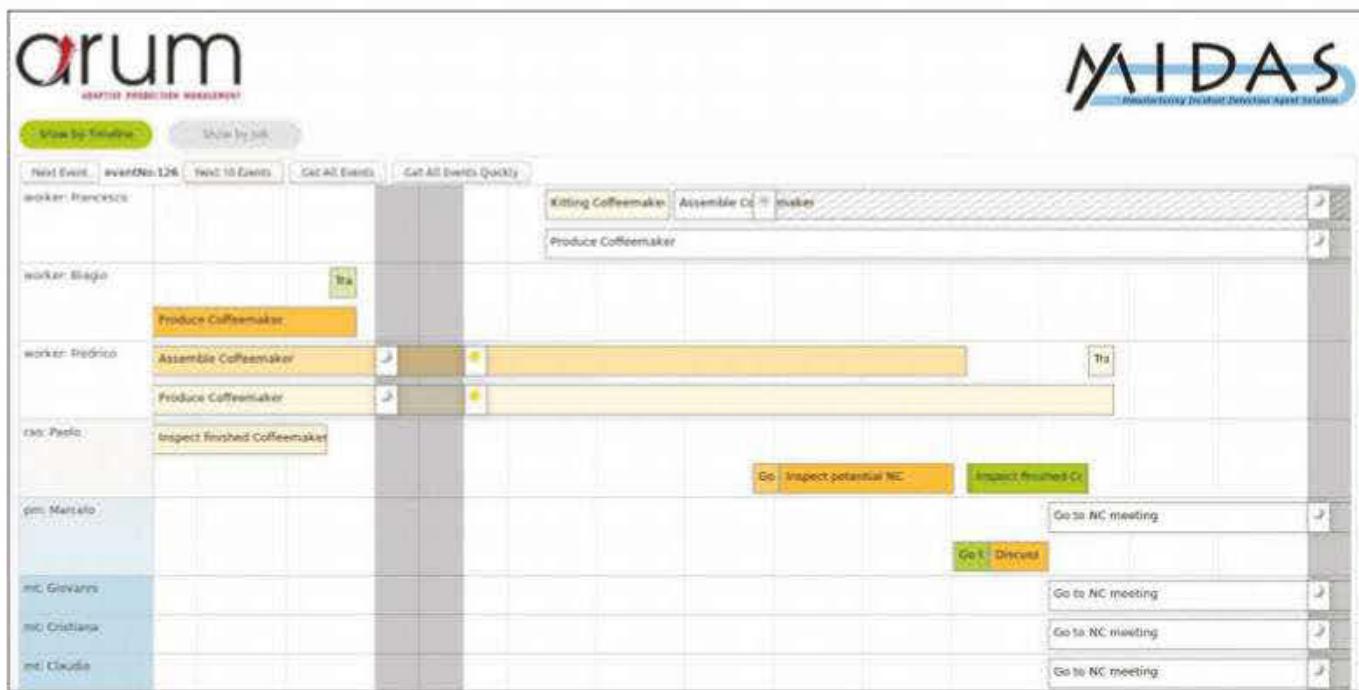
16. One more chart...

So far, our diagrams were limited to one bit of information at a time. Adding a second 'level' to a data field motivates Chart.js to create a chart made up of more than one data set. The Labels array is important, as its omission makes the program skip parts of the data.



Tutorials

Turn data into interactive diagrams



Business-focused diagrams

If your diagrams are dedicated to showing organisational or business process flow information, Chart.js is not the ideal candidate for the job.

<http://visjs.org> is an APACHE-licenced alternative dedicated to the creation of flow diagrams.

While the product is total overkill for 'simple' visualisation tasks, it excels at jobs such as the process overview diagram shown to the left.

The diagram updates itself automatically as the underlying dataset changes, thereby providing a real-time overview of the situation in the plant. This is accomplished via a custom dataset class dedicated to holding information. In addition to that, the developer team also provides a very powerful 3D diagramming engine.

```
var myfield2 = {
  labels: ["1", "2", "3", "4", "5", "6", "7"],
  datasets: [
    {
      label: "My First dataset",
      backgroundColor: 'rgb(255, 99, 132)',
      borderColor: 'rgb(255, 99, 132)',
      data: [0, 10, 5, 2, 20, 30, 45],
    },
    {
      label: "My second dataset",
      fillColor: "rgba(151,187,205,0.2)",
      strokeColor: "rgba(151,187,205,1)",
      data: [28, 33, 40, 19, 12, 27, 9]
    }
  ]
};
```

17. Scientific charting

Generating the label's array can get tedious. If your information is sourced from somewhere where ordinal information is readily available, the data array can also be populated with an array of 'Point[]' fields. In this case, use the syntax shown below.

```
data: [{}]
```

```
x: 10,
y: 20
}, {
x: 15,
y: 10
}]
```

18. Adjust placement of charts

The above-mentioned spacing problem makes positioning diagrams difficult. Chart.js addresses this problem via the padding attribute found in the options field - it allows you to declare a keep-out zone on each of the four margins of the container, thereby constraining rendering.

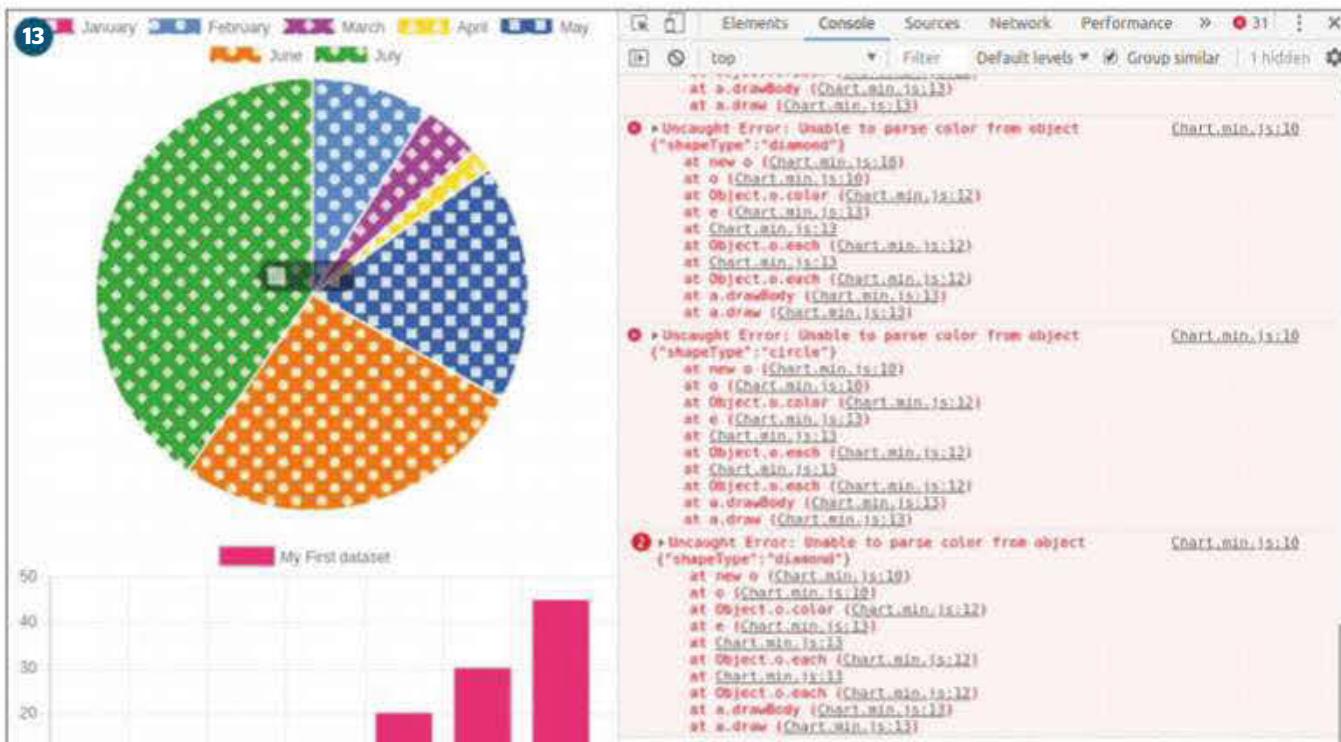
```
var chart = new Chart(ctx, {
  type: 'line',
  options: {
    layout: {
      padding: {
        left: 30,
        right: 30,
        top: 30,
        bottom: 30
      }
    }
  }
});
```



19. Set it globally...

Assigning layout settings to each diagram is tedious. A smarter approach involves the 'Chart.defaults.global' element. It exposes the default settings Chart.js will use for new diagrams, and can save a lot of code if multiple diagrams are to be hosted next to one another.

```
<script>
  document.addEventListener("DOMContentLoaded", function(){
    Chart.defaults.global.layout = {
      padding: {
        left: 30,
        right: 30,
        top: 30,
        bottom: 30
      }
    };
  });
}
```

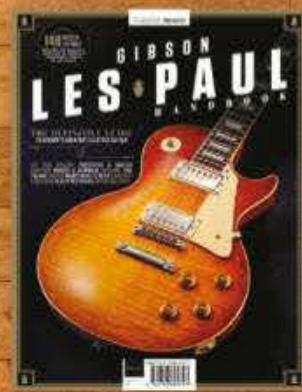
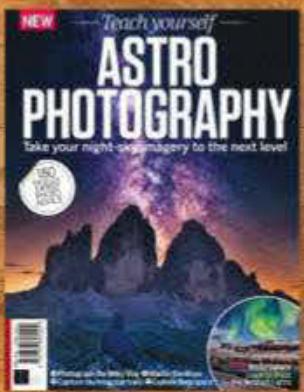
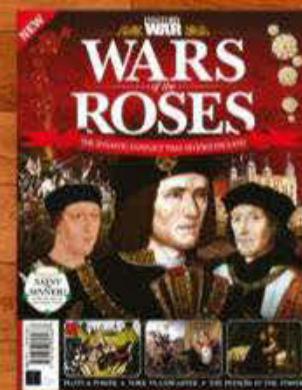


20. Look at examples

The Chart.JS developers provide a set of examples to show the framework in action. Simply open <http://www.chartjs.org/samples/latest> to take a look at the various options - the source code, usually, is commented well.

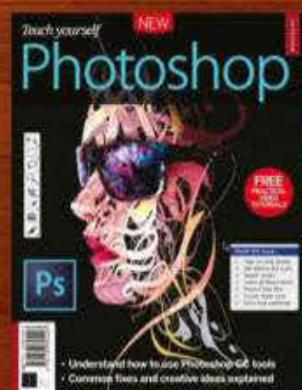
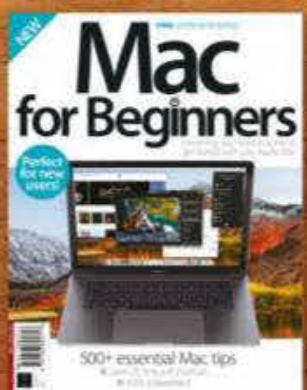
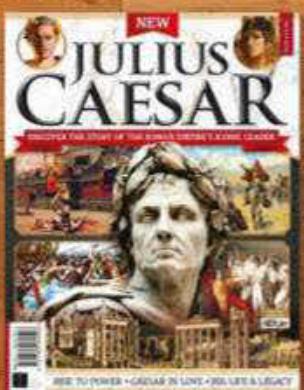
21. Learn even more

Chart.JS comes with extremely detailed documentation, which is hosted at <https://www.chartjs.org/docs/latest>. Simply open it in a browser of choice, and navigate to the sector which interests you the most.



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

F U T U R E

www.myfavouritemagazines.co.uk

Magazines, back issues & bookazines.



Get your listing in our directory

To advertise here contact *Chris*

chris.mitchell@futurenet.com

+44 (0)1225 687832

HOSTING LISTINGS



Featured host: Netcetera

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.

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.

3. 24/7 technical support

Netcetera has a committed team of

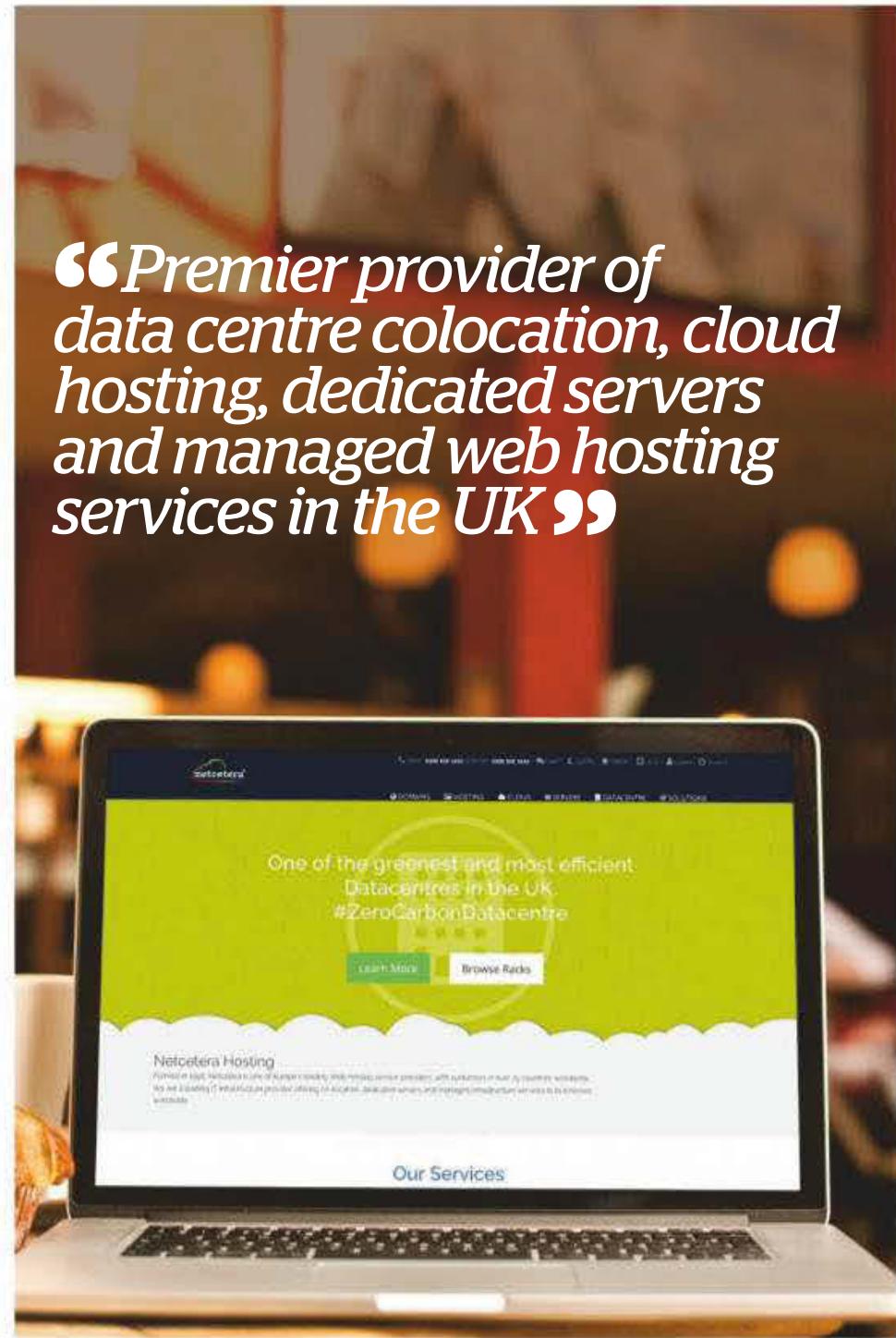
knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

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
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
08455279345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow, then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

Cluster web hosting



fasthosts.co.uk
08081686777

UK-based and operating 24/7 from dedicated UK data centres. Fasthosts keep over one million domains running smoothly and safely each day. Services can be self-managed through the Fasthosts Control Panel.

- Dedicated servers
- Cloud servers
- Hosted email



Budget hosting



hetzner.com
+49 (0)9831505-0

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997, the company has provided private and business clients

with high-performance hosting products as well as the infrastructure for the efficient operation of sites. A combination of stable technology, attractive pricing, flexible support and services has enabled Hetzner Online to strengthen its market position nationally and internationally.

- Dedicated/shared hosting
- Colocation racks
- SSL certificates



All-inclusive hosting



1and1.co.uk
03333365509

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
08432892681

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

Agency hosting specialist



nimbushosting.co.uk
02031266781

Nimbus Hosting have partnered with agencies to develop our revolutionary platform STORM. With a team dedicated to outstanding support our 5 star Google reviews truly speak for themselves. Join the thousands of agencies and freelancers who are benefitting from a control panel that speeds up your website development as well as your client's websites. Super charge your digital projects today with STORM.

- 30 second WordPress install
- Deploy directly from GitHub
- Easy team management

Flexible cloud servers



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

+44 (0)1225 687832

COURSE LISTINGS



Featured: **Northcoders**

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

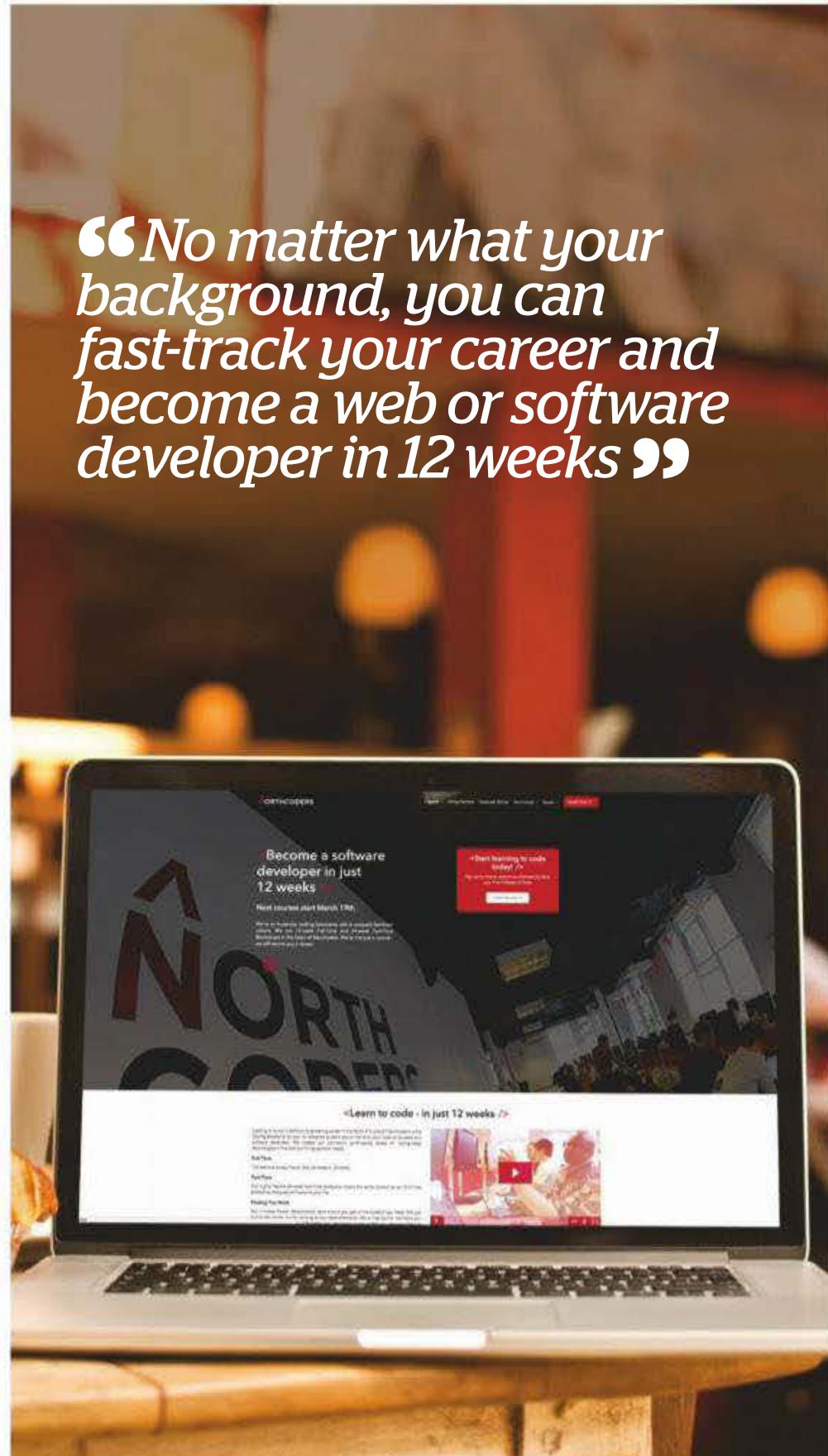
Once you've decided it's right



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



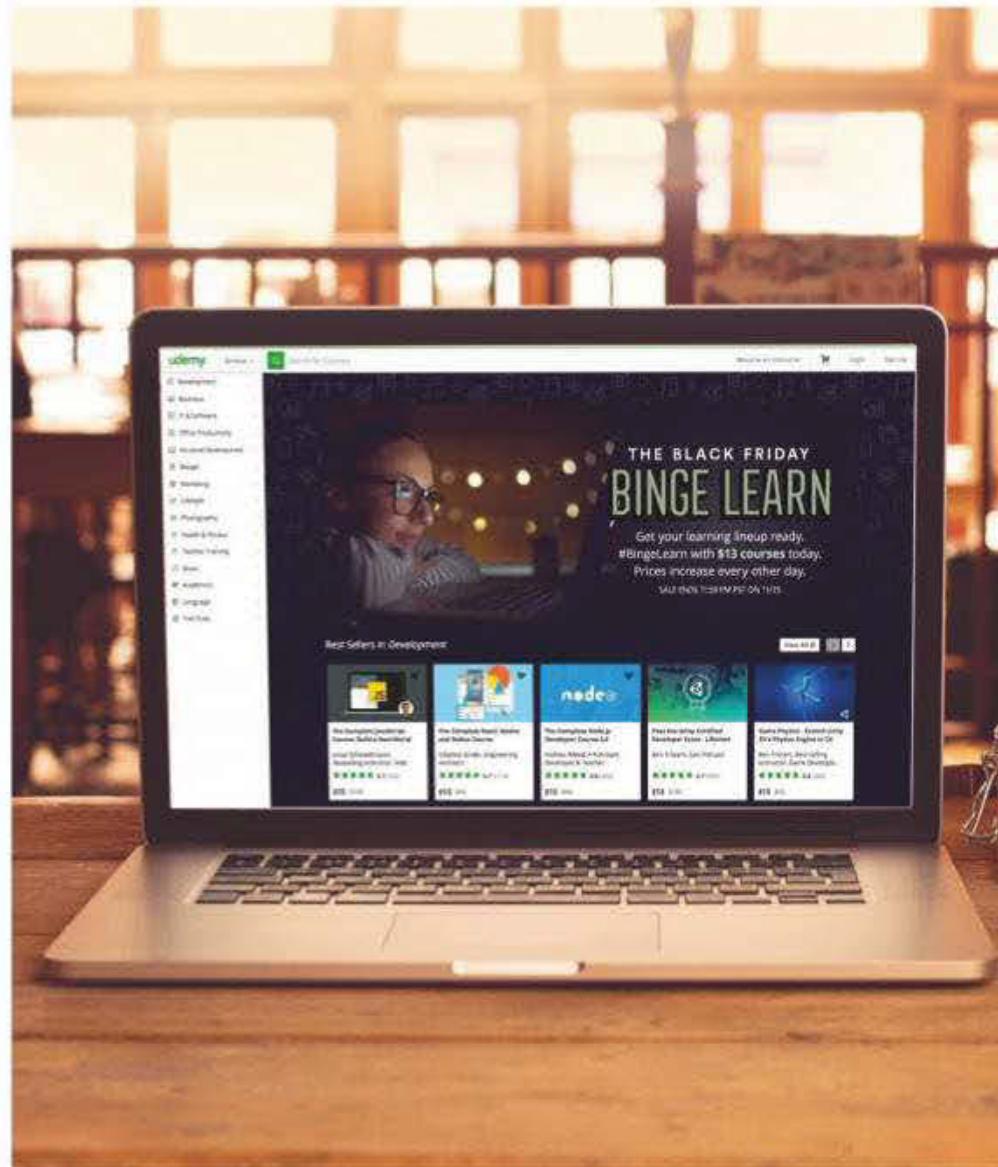
UDEMY

udemy.com

Twitter: [@udemy](https://twitter.com/udemy)
Facebook: [udemy](https://facebook.com/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.



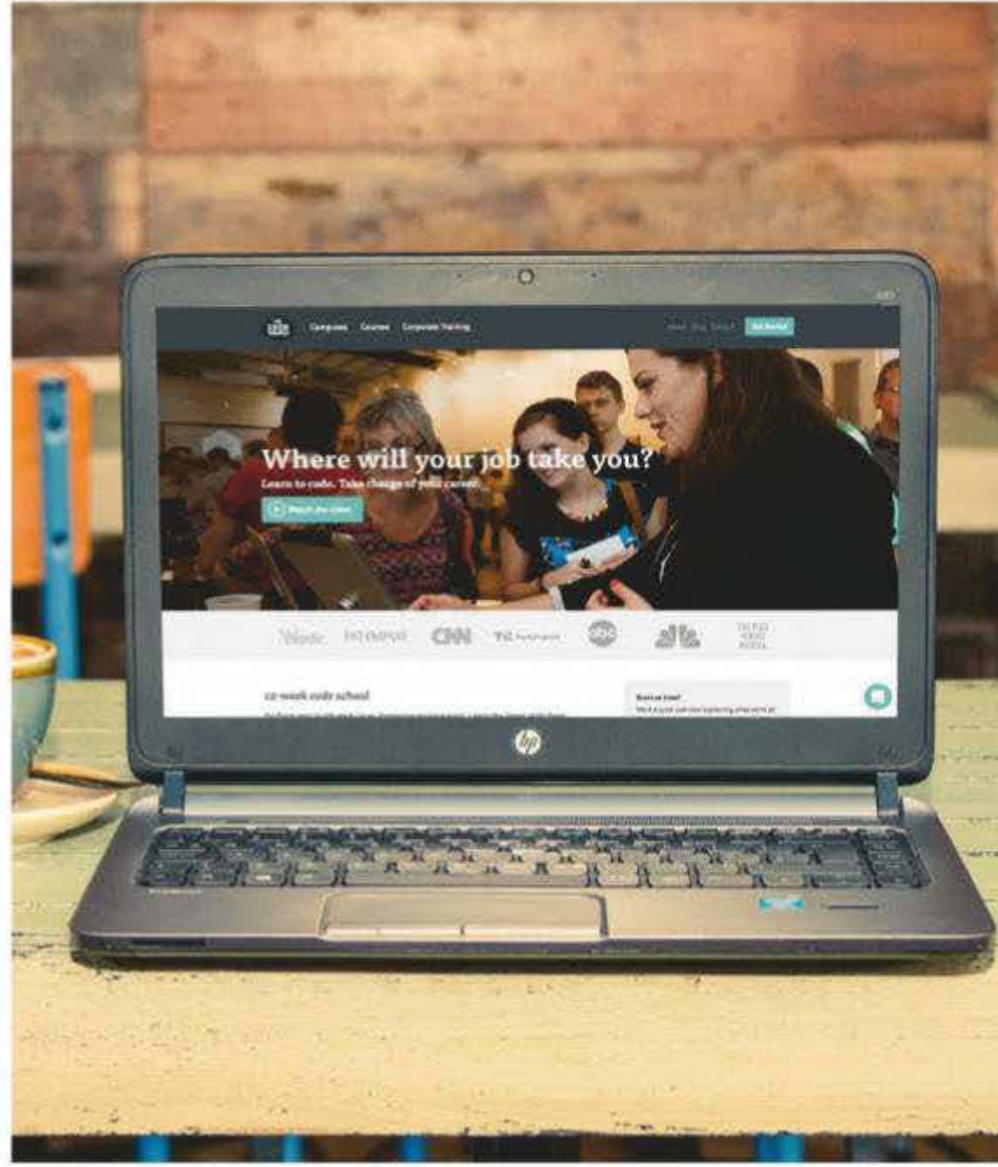
THE IRON YARD

theironyard.com

Twitter: [@TheIronYard](https://twitter.com/TheIronYard)
Facebook: [TheIronYard](https://facebook.com/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](http://wegotcoders.com)

wegotcoders.com
hello@wegotcoders.com

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training 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



[Future Learn](http://futurelearn.com)

futurelearn.com
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



[GYMNASIUM](http://thegymnasium.com)

thegymnasium.com
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.

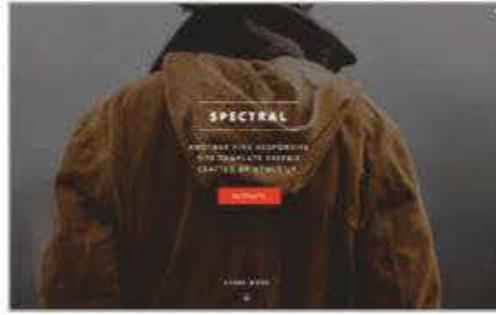
- Gain real-world skills
- Get expert instruction
- Career opportunities

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



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
- 81 minutes of expert PHP video courses from Killersites (shop.killervideostore.com)
- 40 Instant Instagram filters and Sky Replacer Photoshop actions from Sparklestock worth \$53 (www.sparklestock.com)

→ Log in to 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

An incredible gift for subscribers



Subscribe today & unlock the free gifts from more than 60 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

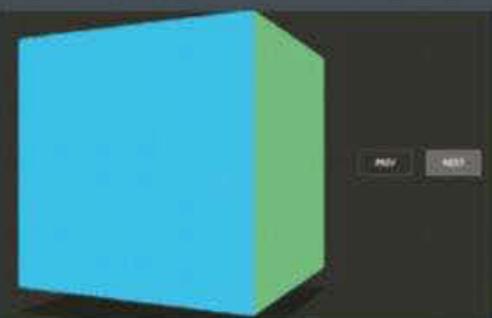
Over 60 hours of video guides

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 28 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

ALL YOU NEED TO KNOW TO BECOME A **CSS PRO**

MASSIVE GUIDE TO ALL THE ESSENTIALS FOR BUDDING DESIGNERS & DEVELOPERS

MUST-KNOW CSS ANIMATION TIPS

A collection of the essential and smart techniques to add style and dynamism to designs

GET TO GRIPS WITH GRID & SUBGRID

Learn how to create contemporary web layouts and see what's next for CSS Grid

NEW REACT TUTORIAL SERIES

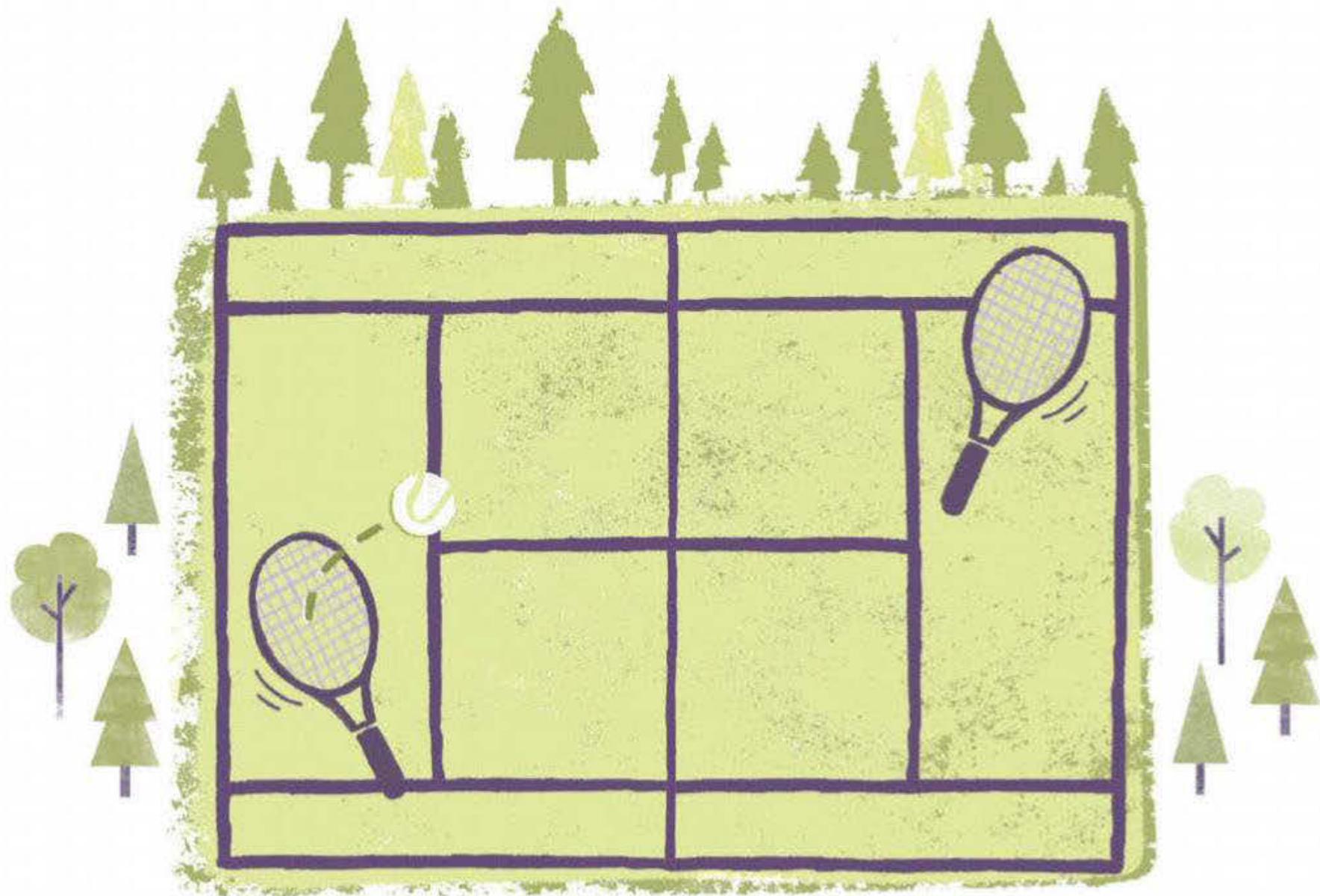
Make sure that you don't miss the start of our new five-part series on React & Redux

Visit the **WEB DESIGNER** online shop at
 **myfavouritemagazines**
myfavouritemagazines.co.uk
for the latest issue, back issues and specials

ALL IN YOUR NEXT
WEB DESIGNER
Issue 282 on sale
Tuesday 11th December 2018

SUBSCRIBE TODAY Go to page 28 to learn more

45 THOUSAND TENNIS COURTS EVERY DAY!



Did you know that European forests, which provide wood for making paper and many other products, have been growing by over 45 thousand tennis courts every day!

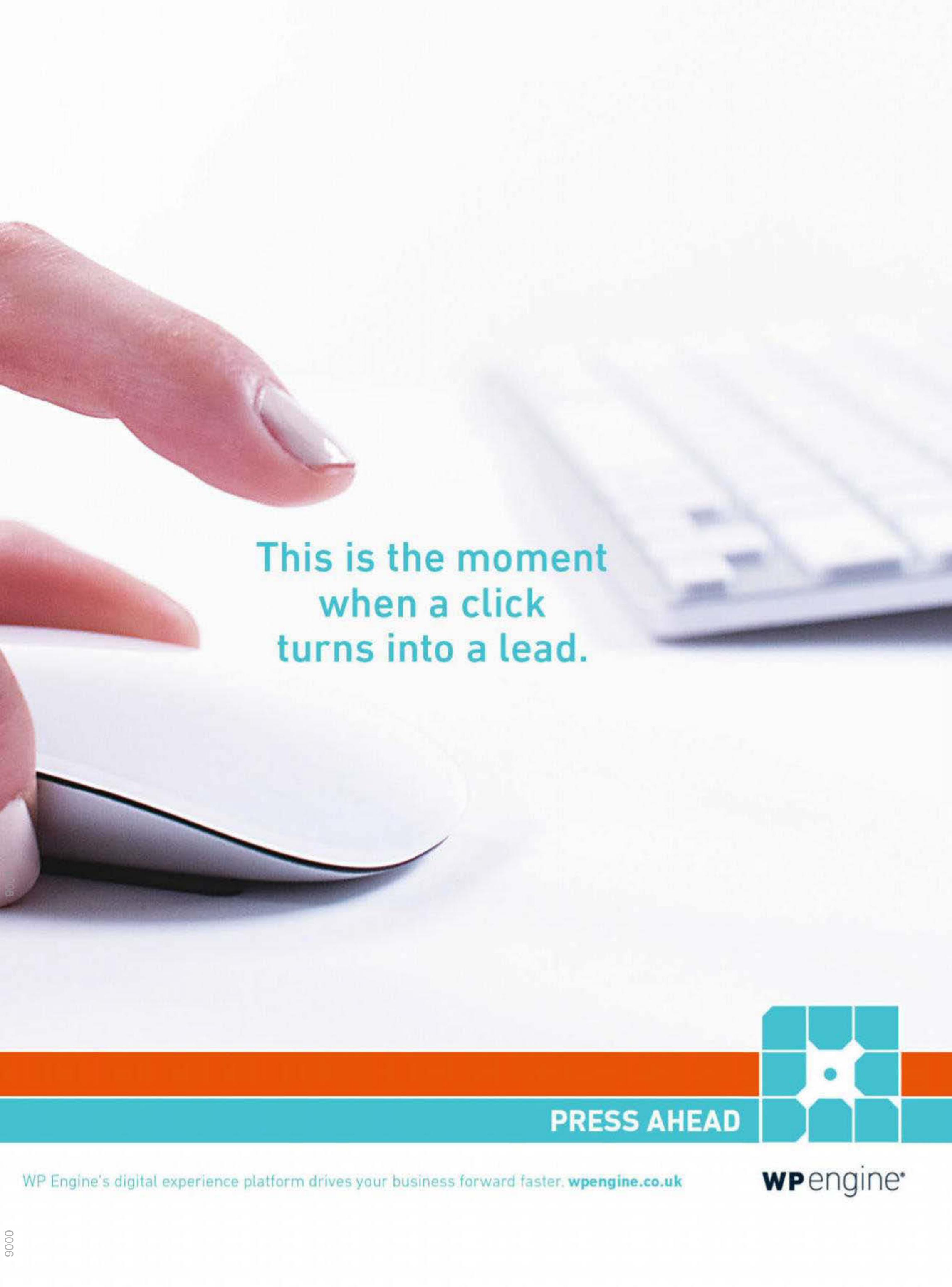
Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood.

UNFAO, Global Forest Resources Assessment 2005-2015

Two Sides is a global initiative promoting the responsible use of print and paper which, when sourced from certified or sustainably managed forests, is a uniquely powerful and natural communications medium.

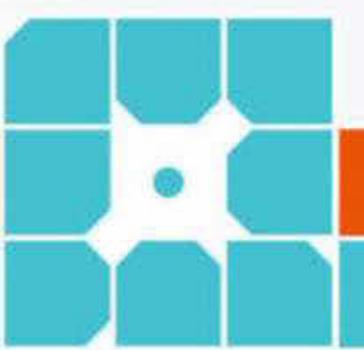
There are some great reasons to **#LovePaper**
Discover them now,
twosides.info





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

WP engine*