

**CSS SPECIAL ISSUE EXPERT 40-PAGE GUIDE**

# designer™

Expert tutorials, techniques and inspiration



► **BUILD WITH GRID** • **ANIMATION TIPS** ▶ **OPTIMISE CODE**  
► **MASTER SASS** ▶ **HOT NEW PROPERTIES**

# 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

# MAKING YOUR AGENCY MORE EFFICIENT.

The STORM Hosting Platform will improve your page load times, streamline your workflow and allow you to spend more time on web development, and less time on server management.

Don't just take our word for it. Take it from our clients and try our **14-DAY FREE TRIAL** with full access to explore how our revolutionary Hosting Platform, STORM could be the solution to improving efficiency, and simplify how you manage your hosting. Visit [nim.host/STORMtrial](http://nim.host/STORMtrial).

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

Call, email or visit us at

**0203 126 6764**

[sales@nimbushosting.co.uk](mailto:sales@nimbushosting.co.uk)

[nim.host/clientstories](http://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

## CSS: style and substance

If you are reading this then there is a good chance you know what CSS is. We like to call it the king of styling, but Cascading Style Sheets in 2018 have so much more to offer than simple styling.

This issue we decided it was time to show off some of the immense talents that CSS has to offer and give an insight into some of the lesser-known but still essential elements that anyone needing to sharpen up their skills really needs to know.

So what is on offer? We kick off with eight pages all about CSS animation. This takes you from storyboarding through to implementing a simple project, while passing transitions,

keyframes, performance and SVG animation along the way. Want to know the ins and outs of CSS Grid? Then head to page 52 as we take you through the basics and beyond, and offer a few tips along the way.

Not as sexy as animation but just as important is optimisation. Head to page 58 to get quick and easy tips to lighten the code load and get faster-loading sites. CSS preprocessors let you extend CSS and make it even more powerful. We take a closer look at the very popular preprocessor Sass and show how to use it to style your sites. Everyone likes rules and best practices when it comes to code, so on page 72 Ahmed Abuelgasim delves into CSS methodologies and how to use them. Elsewhere, CodePen engineer Cassidy Williams shows off CSS art skills, and we look to the future with Houdini. Enjoy.

Using a CSS methodology, or a combination of many, can save you hours of development and debug time, leading to more organised code

## Highlight

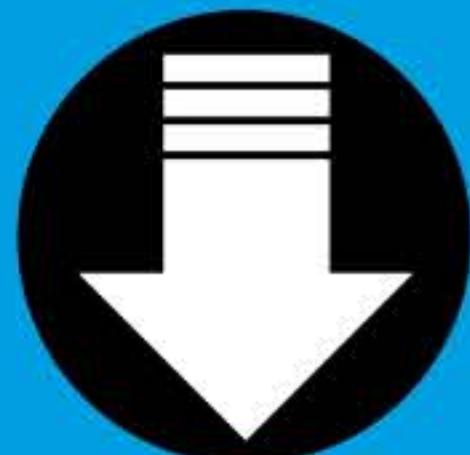


The types of work we have put out are intentionally very broad across verticals and expressions. We believe that broadness characterises us more than any singular piece of work

Web Designer finds out what's going on at Code and Theory. Page 32

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

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



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

**Video Tuition** - New chapters of expert PHP video guides from Killersites ([shop.killervideostore.com](http://shop.killervideostore.com))

**Assets** - 24 Luminous Presets and 10 iPad Pro (3G) mockups from Sparklestock ([sparklestock.com](http://sparklestock.com))  
- Tutorial files and assets



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

# This issue's panel of experts

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



**Steven Roberts**

Steven is a digital creative from Stockton-on-Tees. He is currently leading UX, Design and Frontend at [Asemblr.com](#). Steven has spoken at conferences about CSS animation this year, so for this issue we have gotten him to share some of his extensive knowledge with a host of tips and techniques.

**Page 42**

While it may seem a limited/lesser tool when it comes to animation, CSS is actually really powerful and is capable of producing beautifully smooth 60fps animations ■■■

## Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue he takes a look at CSS Grid and shows how to create grids, offering up some useful tips. **Page 52**

## Mark Shufflebottom



Mark is a professor of Interaction Design at Sheridan College near Toronto. In this issue he takes a peek at some of the hottest properties out there, which you should follow and great examples of cutting-edge CSS. **Page 76**

## Richard Mattka



Richard is an award-winning creative director, designer and developer, specialising in VFX and entertainment projects. In this issue he delves into the world of SVG and shows how to create and implement SVG for the web. **Page 80**

## Ahmed Abuelgasim



Ahmed is a self-taught front-end web developer currently working for Potato. In this issue he explains CSS methodologies, describes the popular ones and explains how to combine them to create a custom one. **Page 72**

## Simon Jones



Simon is a former corporate engineering director, now working at a tech startup. He's led full-stack teams working with many different technologies. This issue, he looks at how Sass can make it easier to manage stylesheets. **Page 64**

## Cassidy Williams



Cassidy is a senior software engineer at CodePen. She's active in the developer community and one of Glamour magazine's 35 Women Under 35 Changing the Tech Industry. This issue we get insight into her world and CSS art. **Page 68**

## Tam Hanna



Tam believes in taking a 360-degree view of things. He believes achieving high performance in websites is always a team effort. In this issue he demonstrates the techniques and tools needed to make your CSS blazingly fast. **Page 58**

## Matt Crouch



Matt is a software engineer at Vidsy, where they use Redux to power all of their React products. In this issue Matt demonstrates how to get up to speed with the concepts of Redux in the first of a five-part React series. **Page 86**

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

## Editorial

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

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

Contributors  
Ahmed Abuelgasim, Mark Billen, Leon Brown, Matt Crouch, Tam Hanna, Christian Heilmann, David Howell, Simon Jones, Richard Mattka, Steven Roberts, Andy Salter, Mark Shufflebottom, Cassidy Williams

## Photography

James Sheppard  
All copyrights and trademarks are recognised and respected

## Advertising

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

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

Subscriptions  
Email enquiries [contact@myfavouritemagazines.co.uk](mailto:contact@myfavouritemagazines.co.uk)  
UK orderline & enquiries 0344 848 2852  
Overseas order line and enquiries +44 (0) 344 848 2852  
Online orders & enquiries [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
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](http://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](http://www.futureplc.com)

Chief Executive **Zillah Byng-Thorne**  
Chairman **Richard Huntingford**  
Chief Financial Officer **Penny Ladkin-Brand**  
Tel +44 (0)1225 442 244

# PUT A PAUSE IN YOUR DAY

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



**PRESS PAUSE**  
ENJOY A MAGAZINE MOMENT

To find out more about Press Pause, visit:  
**pauseyourday.co.uk**

# **contents**

# Cutting-edge features, techniques and inspiration for web creatives

Chat with the team and other readers and discuss the latest tech, trends and techniques. Here's how to stay in touch...

webdesigner@futurenet.com • @WebDesignerMag • www.creativeblog.com

- 08 The state of JavaScript today**  
What's popular? What are the big earners?  
**Web Designer** looks at the latest SOJ report

**11 Keeping up with the web industry**  
**Web Designer** catches up with Christian Heilmann, who talks Twitter, Slack and more

**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

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

**26 Bon Voyage**  
Green Chameleon captain a site redesign for superyacht experience providers Asaro

**32 Decoded Design**  
**Web Designer** talks to Code and Theory about how they work and craft digital landscapes

**40 All you need to be a CSS Pro**  
Discover essential CSS animation tips, how to build with Grid, the best CSS methodologies, how to optimise your code and much more

**80 SVG on the web**  
SVG support is finally where it should be. Discover how to use in your projects

**92 Hosting listings**  
An extensive list of web hosting companies. Pick the perfect host for your needs

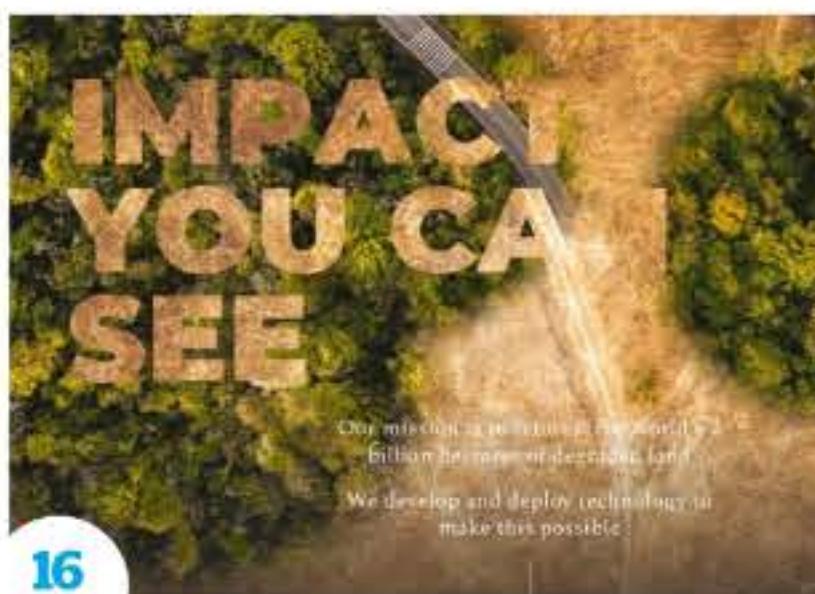
**94 Course listings**  
Want to start learning? Check out what courses are out there with this list

**98 Next month**  
What's in the next issue of **Web Designer**

**Cover focus**



40



# TUTORIALS

## 42 CSS animation tips & tricks

Learn the finer points of CSS animation and how to implement them into a project

## 52 Build with CSS Grid

Discover the basics and beyond. Get essential tips and techniques for better layouts

## 58 Get blazingly fast CSS

Cut out unnecessary code and slim down your stylesheets for faster-loading sites

## 64 Style sites with Sass

Extend your CSS and make styling easier with this popular preprocessor

## 68 CSS art with Cassidy Williams

Cassidy Williams, Senior Software Engineer at CodePen, shows how to illustrate with CSS

## 72 CSS methodologies

Discover what methodologies are, which ones are popular and create a custom one

## 76 Cutting-edge CSS

Stay on top of trends, find out what are the hot properties and what is Houdini

## 86 Manage React app data

Learn how to create an API-powered state application and discover the benefits of a centralised data store



# FILESILO

## 96 Get the latest must-have resources and videos

- New chapters of expert PHP video guides from Killersites
- 24 Luminous presets and 10 iPad Pro mockups from Sparklestock



## myfavouritemagazines

Visit the **Web Designer** online shop at [myfavouritemagazines.co.uk](http://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](https://bit.ly/2wetvGp)  
iTunes – [apple.co/2igtBYq](https://apple.co/2igtBYq)



**Subscribe today and save**

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

# Header

The tools, trends and news to inspire your web projects

## JavaScript: which are the popular choices?

The State of JavaScript 2018 survey gives a deep insight into who's using what in the world of JS



longside HTML and CSS, JavaScript is the key technology that is driving forward web development.

It is always moving forward, with new frameworks, tools and variations seemingly making an appearance almost weekly. While new additions add more power and panache to a designer's or developer's arsenal, the hard part is deciding which ones you need to be paying attention to. Who are the big players? Who are the newcomers to watch out for? Who's going to be the one that pays the big bucks? Choosing the right path is not an easy task, so any help newcomers and established developers can get is always welcome.

The State of JavaScript survey is an annual report that provides a wealth of information on what developers are using, what they're happy with and what they want to learn. The latest report surveyed over 20,000 developers. So where do all

these developers come from? The largest contributor is the US with nearly 25 per cent of respondents, with Australia and Germany offering over 5 per cent of respondents. The UK contributed 3.78 per cent. Africa had the fewest respondents, with Libya, Somalia, Angola and Mozambique only having one respondent each. As might be expected, the vast majority - over 18,000 - of those who participated in the survey were male. There were 931 female respondents. A representation of how the market stands?

JS developers rarely use a single library or framework. Even the popular choices have a companion. For example, React users also connect with Redux, Express, Mocha, Jasmine, Jest and of course ES6. So what are the popular front-end frameworks? As might be expected React tops the charts, with 65 per cent using and happy to use it again. It's not all roses though, as 6.7 per cent had used it and would not use it again. But 19 per cent

wanted to learn. Sitting in second place was Vue.js, closely followed by Angular. One interesting point was that over 46 per cent wanted to learn to use Vue.js while 33.8 per cent had used Angular but would not use it again. The remaining frameworks in the top six were Preact, Ember and Polymer, with around 25 per cent of respondents keen to learn Preact and Polymer. But what other libraries got a mention? The two standouts were Svelte and Aurelia, with jQuery still hanging on in there. Looking at back-end frameworks, Express is way out in front, with nearly 65 per cent saying they had used it and would again. Next was Next.js with nearly 9 per cent. Looking at mobile and desktop it was close between Electron and React Native, with over 50 per cent keen to learn both.

The State of JavaScript survey is a goldmine of JS-related information, a fascinating read and beautifully presented. We have only scratched the surface here so look at [2018.stateofjs.com](http://2018.stateofjs.com) to see in full.

## STAT ATTACK

### MOBILE VENDORS

Who are the big players in the European market?

#### Samsung

**32.18%**



Down 3% on the same time as last year

#### Apple

**28.81%**



Almost identical to the previous year

#### Huawei

**15.45%**



On the rise, 4% up on last year

#### Xiaomi

**3.78%**



A low rise, 2% up on last year

#### LG

**3.72%**



Slight downward spiral, by less than 1%



**Subscribe today and save**

**40%**

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

Source: <http://gs.statcounter.com>  
(Correct as of November 2018)

**HEAR FROM THE PROS**

**SIGN UP TO THE**

**web  
designer**<sup>TM</sup>

**NEWSLETTER TODAY!**



**Get weekly news, tips & inspiration**

**SIGN UP NOW!**

**[bit.ly/2K15b7Y](https://bit.ly/2K15b7Y)**

# Sites of the month



## 01. Trade Aid

[tradeaidchocolate.co.nz](http://tradeaidchocolate.co.nz)

A video background and overlaid graphics combine to reinforce the site's message

## 02. Old Spice - Big Lavender

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

A classic old-school shooter is a smart way to market Old Spice for a new generation

## 03. Human Forward

<https://bit.ly/2A2PqJg>

Big, bold and impressive animations make sure that the user is constantly engaged

## 04. MiM event

[gomim.com/en](http://gomim.com/en)

Subtle background animations combine with subtle movement as the users swipe or scroll

## Graphics Logotypes vol. 5

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

A collection of gorgeous hand-drawn logotypes from designer Martin Schmetzler



## Colour picker Sensei

<https://bit.ly/2OTKil5>



## Typesetter Fixture

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

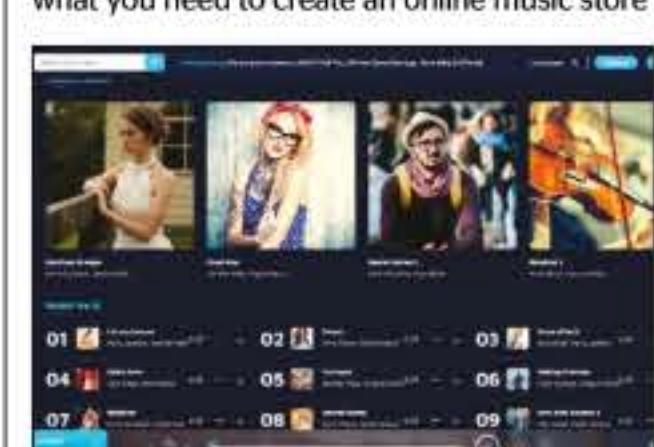
Based on 19th-century typefaces and posters, Fixture offers four widths ranging from Ultra Compressed to Expanded



## WordPress Miraculous

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

Starting a Spotify style site? This theme offers what you need to create an online music store



# Keeping up with the web industry

Web Designer recently caught up with Christian Heilmann at [Reasons.to](#)



**Christian Heilmann**

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

Twitter: [@codepo8](#)

“ I’m happy to pretend (or sometimes be honest) that I don’t know things, just to be an example for other people ”

**A**

lthough there is definitely value in social media, there is an issue with the amount of noise it generates. On Twitter, the problem is exacerbated by the fact that the timeline is not in sync any longer. It’s just a random presentation of what might be interesting for you, which means you miss a lot. I’ve got 64,000 followers on Twitter, and yet I get all my resources via RSS feeds. I can’t find much on Twitter because it gets lost in the noise.

The other problem that social media has caused is that instead of learning from each other, people try to create resources that get the most likes or retweets. We try to shortcut a lot. The priority is to make a cool headline that will get shared. I’ve blogged since 2005, and my blog posts are always long. This used to work, but now people just zone out. We’ve become less patient with learning. We’ve seen too many things come and go, and it feels like the investment in learning is not worthwhile given the speed of changes in the market.

It’s not all bad, though. There’s a lot of good stuff happening on Slack groups. And, although it’s not easy to use, I’d recommend trying Mastodon, which is like a more organised Twitter. There are also a lot of learning resources on GitHub. There are training courses you can take on other platforms, but GitHub is where people are channelling a lot of their efforts now. It also tends to get the most edits, so content there is generally up-to-date.

If you want to get started, taking part in open source projects is the best way. Helping with a typo or a bit of documentation, or making a demo for a product and asking the owner if they want to have it on the GitHub repo is a great way to get your name known. People publish great things as open source but often don’t have time to show the implementations or write documentation.

Although these platforms provide a good place for people to get involved in the industry, we need to work on being more welcoming to newcomers. The web industry is not accepting of people who don’t speak the right language. People and tutorials hide behind massive terms and clever-sounding messages. This could be a natural thing. When you’re an expert and you want to talk to other experts, sooner or later you develop a defined lingo, and that’s fine. But starting with this lingo and overwhelming newcomers that want to learn something is annoying. People are repeating the same sentences that they don’t even understand, just trying to fit in with the crowd.

Because of this peer pressure, newcomers often don’t dare to speak up when they don’t understand. I love that, as somebody who is in the know, I can do that for them. I interfere and say, “OK, but what does that mean?” I’m happy to pretend (or sometimes be honest) that I don’t know things, just to be an example for other people so they might find the courage to do the same.

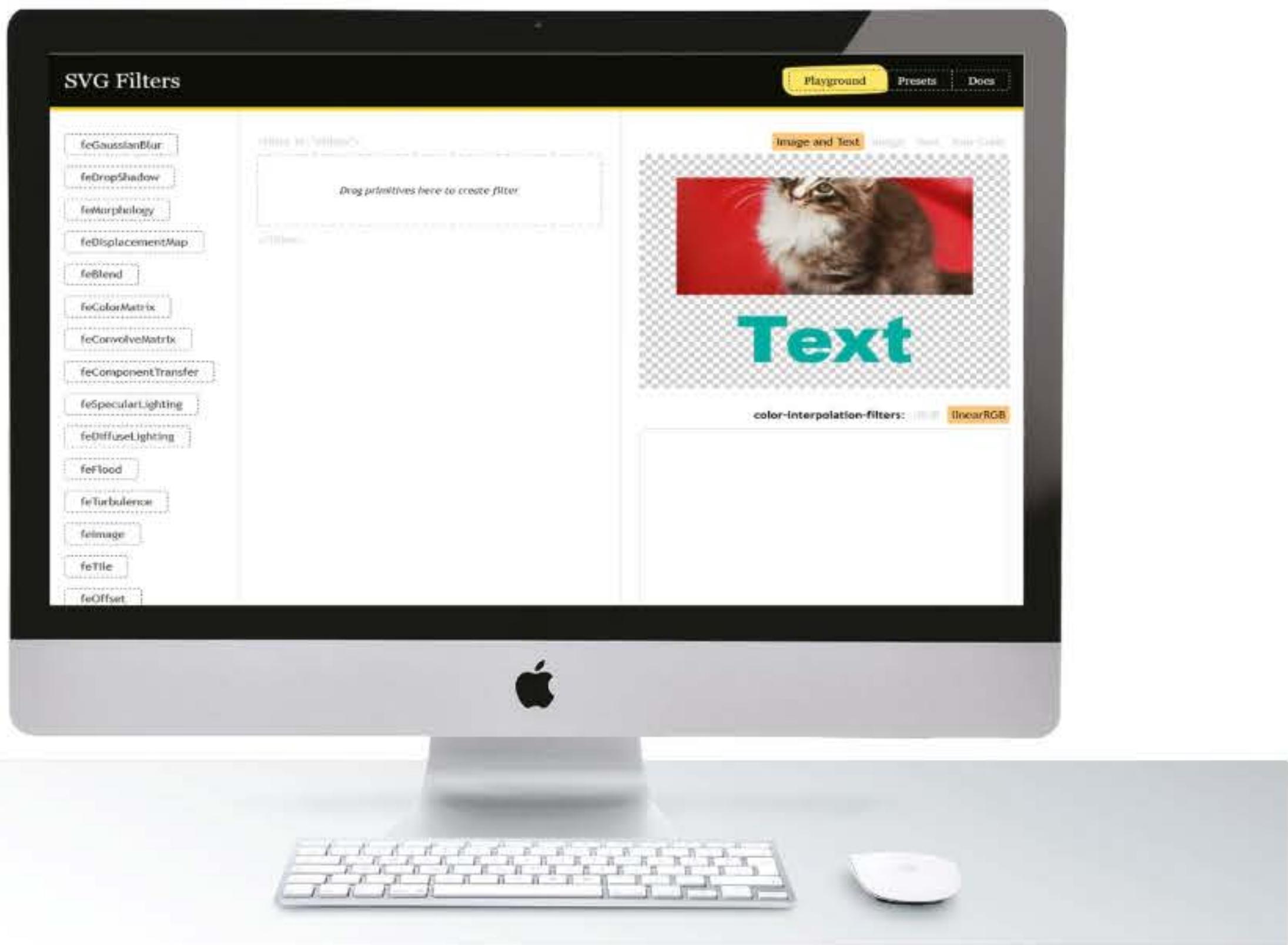
This is a particular problem with new technologies and buzz phrases. People make a lot of money by saying half-clever things about a technology rather than showing what it can be used for. Everything is Blockchain right now. 90 per cent of people on the planet can’t explain what Blockchain is. “We’re using Blockchain for that” – of course. The same way we used ‘the cloud’.

Being on the bleeding edge is interesting, but we’re chasing our own tails a bit and not giving new technologies enough time to mature and become useful. I want to see more products being created with these technologies. I’m currently working on a W3C proposal for Machine Learning in the browser. It would mean you could do facial recognition, audio recognition and similar ‘AI’ tasks on your own machine, rather than having to send it to Amazon, Google or Microsoft to get it analysed there. That way the privacy of your own data is much more given. There’s an open standard for the file format but not an open standard for the access to it. That’s something we’re proposing right now, and that’s what I want to see much more of.

Everybody sells their current solutions as magic rather than explaining them. I know graffiti painters who learned how to paint on canvas first because they need to know the craft before they can do the art. They are stunning at what they do as they chose to subvert the original art. We’re not patient enough right now to learn the ropes before calling ourselves experts.

# webkit

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



## SVG Filters

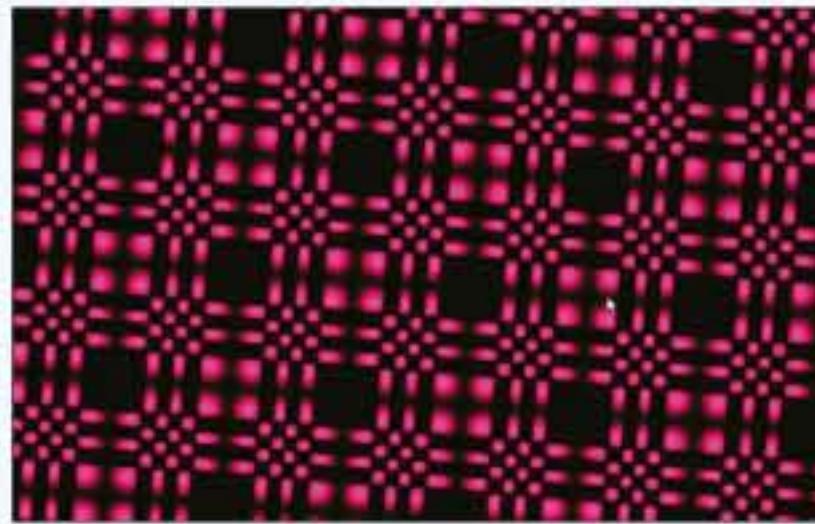
[yoksel.github.io/svg-filters/#](https://yoksel.github.io/svg-filters/#)

Want to see SVG filters in action without applying lines and lines of code? Then this rather neat online tool enables you to add different filters with a single click. Add

GaussianBlur to see the filter in action. Like it? Copy the code and use in your project. Experiment with the other filters to get the look you want.

## TOP 5 CODEPENS

Be inspired by this collection of smart and interesting codebases



### Relaxing Recursive React

[codepen.io/terabaud/pen/RqQmOX](https://codepen.io/terabaud/pen/RqQmOX)

An animated repeating pattern built with React that mesmerises and relaxes. Watch in a darkened room and feel the calm.



### Old-school DVD

[codepen.io/foleyatwork/pen/MzGKwq](https://codepen.io/foleyatwork/pen/MzGKwq)

If you have a DVD player, this a faithful recreation of the sleep screen. If not, watch and admire what you're missing.



### CSS file icons

[colorswall.github.io/CSS-file-icons](https://colorswall.github.io/CSS-file-icons)

Pure CSS file icons for popular extensions, including doc, docx, log, txt, wps, csv, dat, ppt, xml, mp3, wav, avi, mov, mp4. Available in different sizes.



### Day.js

[github.com/iamkun/dayjs](https://github.com/iamkun/dayjs)

This is a minimalist JavaScript library that parses, validates, manipulates and displays dates and times for today's browsers with a largely Moment.js-compatible API



### Video Crop

[video-crop.com](https://video-crop.com)

Want to crop your videos quickly and easily? This drag and drop tool is the answer and comes as part of a collection of video tools



### Smoothie Charts

[smoothiecharts.org](https://smoothiecharts.org)

Turn your data into live streaming graphs with this JavaScript Charting library. Read the Ten Minute Tutorial and create with the easy-to-use Chart Builder.



### Catch the Carrot

[codepen.io/MalaikaMI/pen/VVQpmW](https://codepen.io/MalaikaMI/pen/VVQpmW)

A simple game where you have to catch the carrot. Give it a try and see if you can beat our score of five.



### Periodic Table

[codepen.io/oliviale/pen/ZmvPPd](https://codepen.io/oliviale/pen/ZmvPPd)

A CSS based - slightly modified - periodic table that cleverly reflects how Olivia, the designer, wastes their time on the internet.

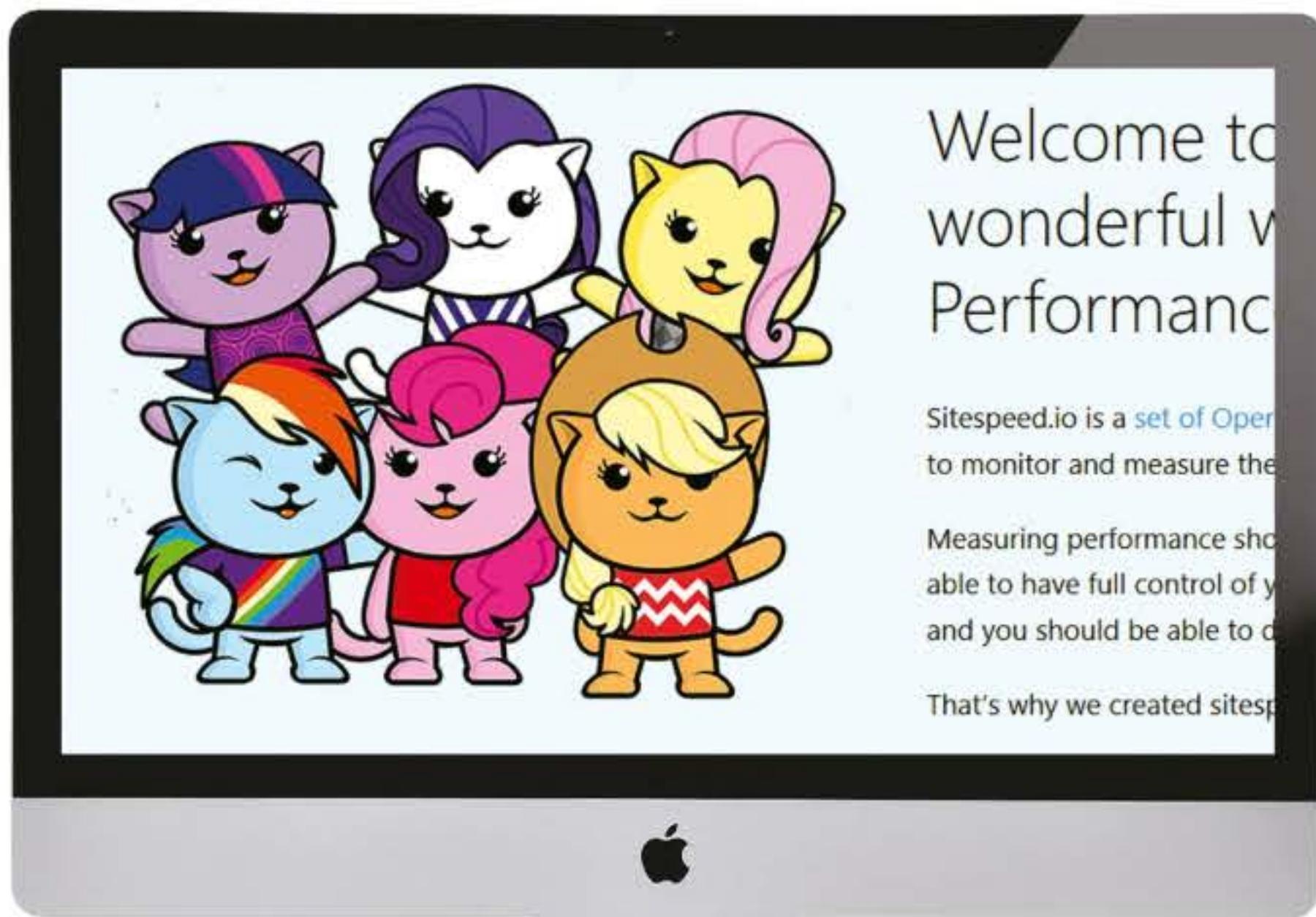


### Rounded Range Slider

[codepen.io/Artemis1/pen/GwywoI](https://codepen.io/Artemis1/pen/GwywoI)

A styled-up temperature gauge that shows Fahrenheit and centigrade. Use the slider to change or enter manually.

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



## sitespeed.io

[sitespeed.io](https://sitespeed.io)

Sitespeed.io is a set of open source tools you might or might not have heard of that make it easy to monitor and measure the performance of your website. Includes

helpful instructions on how to install and which tools to use for what. Only interested in timing metrics, then use Browsertime. Want to find issues, use The Coach.

## TOP 5 WORDPRESS THEMES

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



### Buzz Stone

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

A modern, contemporary image-led news/magazine style theme. Homepage options enable you to get a more custom option.



### Jawn

[mvpthemes.com/jawn](http://mvpthemes.com/jawn)

Big image, smart styles and modern fonts make this theme the ideal option for the modern-day blogger.



## SoundJS

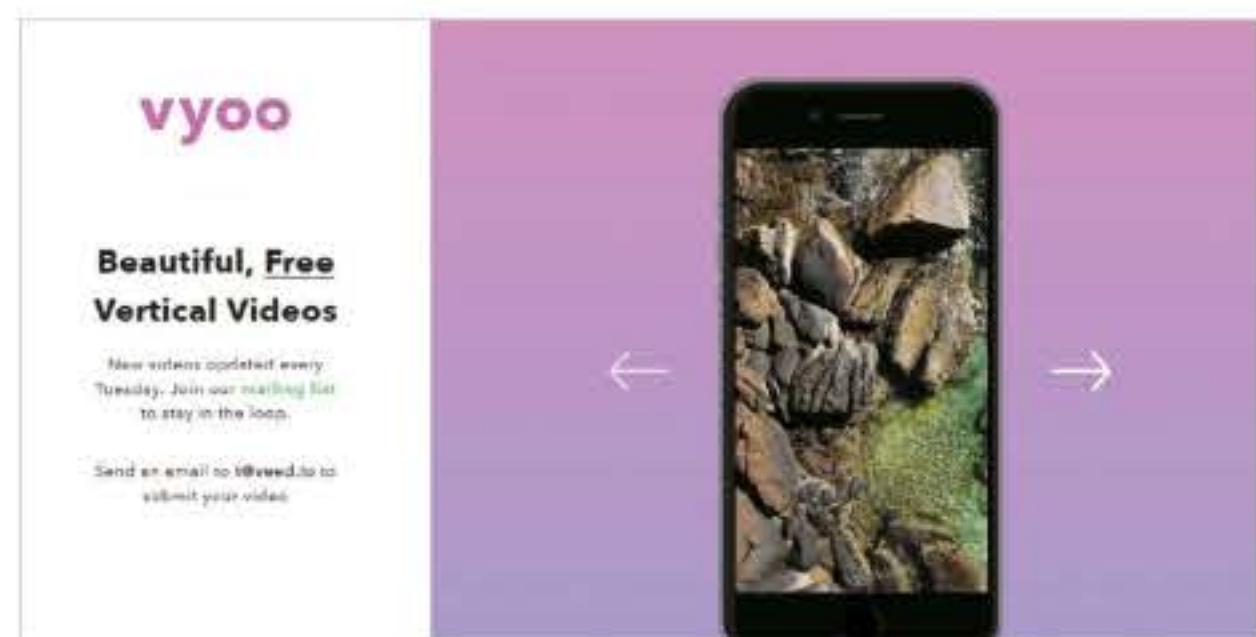
[createjs.com/soundjs](http://createjs.com/soundjs)

A JavaScript library that helps make working with audio "a breeze". It offers a simple API that abstracts HTML5 sound implementation.

## Colorblindly

<https://bit.ly/2DJKa8o>

Accessibility is important, and this Chrome extension simulates what users with colourblindness will see in the web browser. It offers eight different settings.



## Lookback

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

Want to know how people are using your site or app? Of course you do. Do real-time video testing, remote testing and in-person testing.

## Vyoo

[veed.io/vyoo](http://veed.io/vyoo)

This is simple. It offers beautiful, free vertical videos, with new ones uploaded every week. Download the video or edit with Veed.



## Insider

[darinka.design/insider](http://darinka.design/insider)

A classic, clean image-led layout with image animations to help engage users. A multipurpose theme with a touch of style.



## Flagrant

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

A modern image-led gallery theme. Ideal for photographers, artists, illustrators or any other image-led site.



## Hayden

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

An e-commerce theme aimed at children. As you might expect, it offers bright, bold colours and easy-to-navigate menus.

# Land Life Company

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

An aerial photograph of a winding road through a lush green forest. Overlaid on the image is the word "IMPACT" in large, bold, gold-colored letters. Below it, the words "YOU CAN SEE" are written in a smaller, gold-colored font. The text appears to be cut out of the forest, with the green foliage visible through the letters.

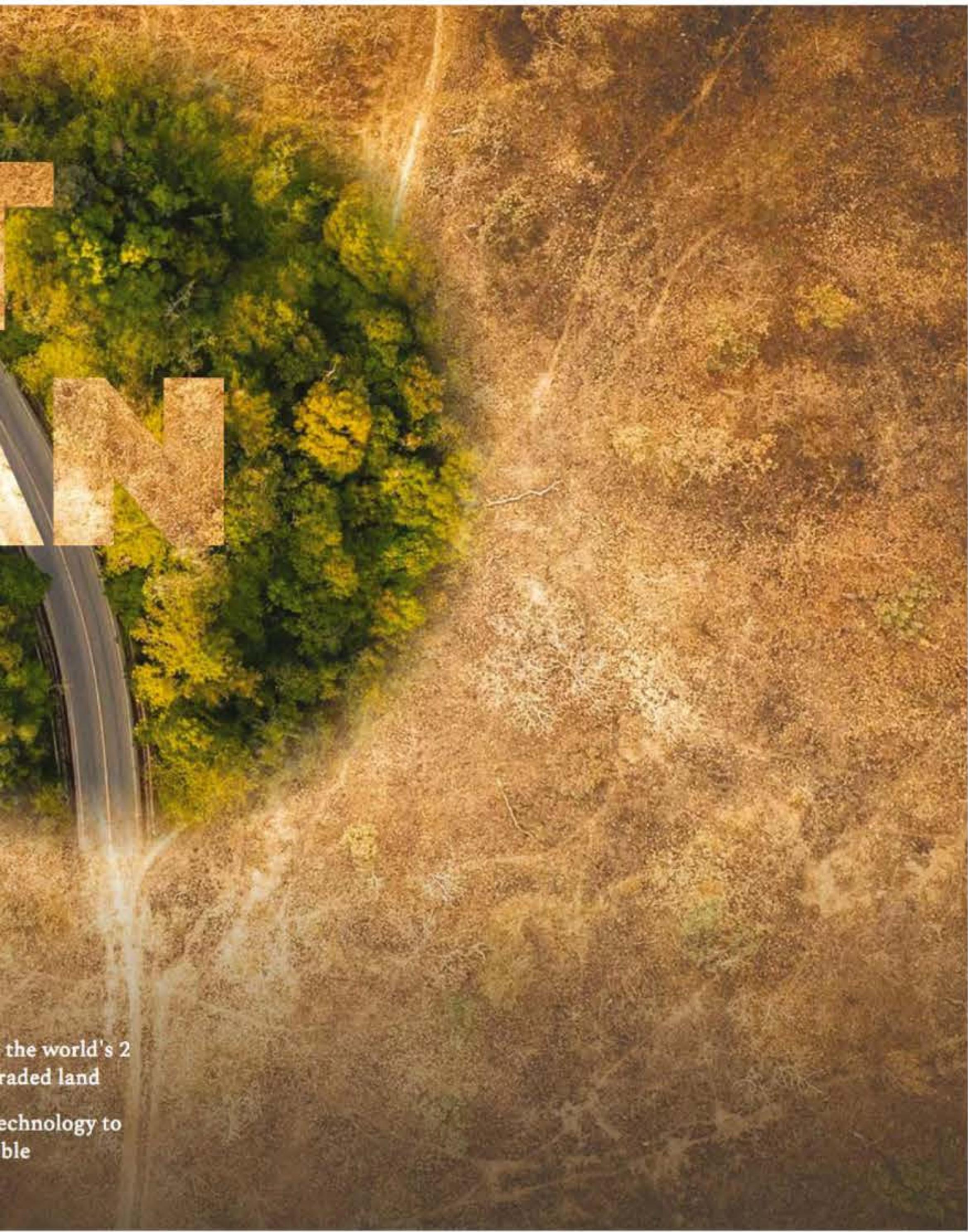
IMPACT  
YOU CAN SEE

Our mission is to reforest  
billion hectares of deg

We develop and deploy to  
make this possi

Designer:

Qikker Online [www.qikkeronline.nl](http://www.qikkeronline.nl)



the world's 2  
raded land

technology to  
ble

**“Based in Amsterdam and operating across 25 countries, Land Life Company’s mission is to help reforest 2 billion hectares of degraded land”**

#### Colours



#### Tools

WordPress, PHP, Yahoo UI (JavaScript), YouTube

#### Fonts

abcABC  
1234567890

*Avenir by Adrian Frutiger, plus Montserrat by Julieta Ulanovsky are both used to render page headings and paragraphs throughout.*

abcABC  
1234567890

*Spectral font by Production Type, also available via Google Fonts, provides the serif styled typeface found on links and quotes.*

Fix the planet



The desktop view shows a header with the logo and navigation links: Solutions, Case Studies, About. Below is a section titled 'Projects' with a grid of thumbnail images. One thumbnail for 'Grow Zambia!' is selected, showing a group of people in green shirts working in a field. A callout box for 'Saudi Arabia' discusses the 'Greener Desert' project.

The mobile phone view shows a 'Case Study' page for 'Building a Green Refugee Camp with UNHCR'. It features a large photo of a woman holding a plant, a 'Fast facts' section, and social media sharing icons.

**Above**  
Project locations are plotted on a rollover world map, with key area hotspots highlighted for viewing details on click

**Far left**  
Projects can also be viewed in a more linear layout, stacking a stream of photos and quotes under links for quick navigation

**Left**  
Beyond the opening header, a minimal colour palette is employed to allow beautiful location photography to be more impactful

# Create a cursor-controlled interactive spotlight effect

A spotlight effect allows users to search and reveal additional parts of background imagery

## 1. HTML document initiation

The first step is to define the webpage as a HTML document. This consists of the HTML document container used to contain the head and body sections. While the head section is used to incorporate the external CSS and JavaScript resources, the body section is used to store the content elements that are created in the next step.

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

## 2. Body content

The content placed inside the document body sets up a header element used to store content required for the effect. The div element is used as a container to control text and image content we want to place inside the header element. This HTML is defined to be SEO friendly by keeping elements and styling for the effect defined separately in the CSS and JavaScript files.

```
<header>
    <div>
        <h1>Hello</h1>
    </div>
</header>
```

## 3. JavaScript load event

Create a new file called 'code.js'. For the effect to work, JavaScript is used to modify the header element created in the previous step. This header element is only available after the page has loaded. An event listener is applied to the window to execute a function containing specific instructions after the page has loaded. This step finds the header section and inserts a span element as the first child.

```
window.addEventListener("load",function(){
    var header = document.
querySelector("header");
    var node = document.
createElement("span");
    var handle
    header.insertBefore(node, header.
```

```
firstChild);
```

```
    *** STEP 4 HERE
});
```

## 4. Movement listener

JavaScript is also used to attach an event listener to the header element found in step 3. This listener executes code to find the child element created in step 3, upon which new styles are calculated for its placement and background positions. Mouse co-ordinates relative to the header element are provided via the 'e' parameter.

```
header.addEventListener("mousemove",
    function(e){
        var shape = this.firstChild;
        var x = e.offsetX;
        var y = e.offsetY;
        shape.style.left = x + "px";
        shape.style.top = y + "px";
        shape.style.backgroundPosition = (0-x)+"px
        +(0-y)+"px";
    });
});
```

## 5. CSS: header styling

Create a new file called 'styles.css'. This step defines the styling for the header element, making it cover half of the web page. The background image for the element is also selected, with its size adapted to cover the full width of the header element. Overflow and cursor presentation are set to hidden to keep the effect tidy.

```
header{
    position: relative;
    display: block;
    height: 50vh;
    background: url(image.jpg);
    background-size: 100%;
    cursor: none;
    overflow: hidden;
    color: #fff;
}
```

## 6. Gradient overlay

A gradient is required to be placed over the background image. The 'before' virtual element is used to achieve this in a way that avoids additional HTML requirements. This element is sized and positioned to cover the full space of the header element with its own background image, created with a CSS-generated linear gradient.

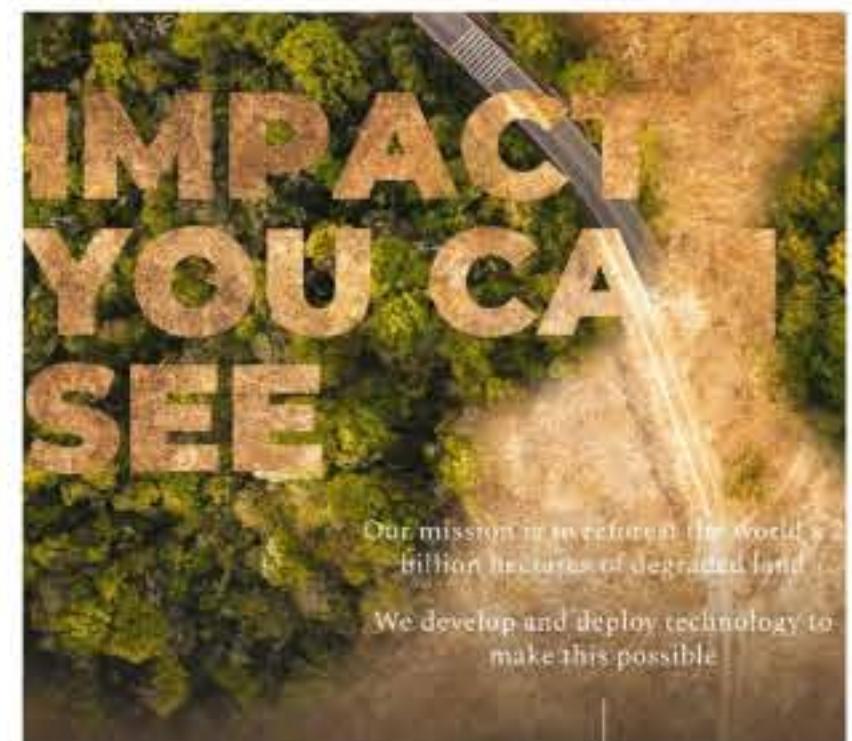
```
header::before{
    position: absolute;
```

```
content: "";
display: block;
width: 100%;
height: 100%;
background-image: linear-gradient(to
right,
    rgba(0,0,0,0) 0%,
    rgba(0,0,0,0) 30%,
    rgba(0,0,0,1) 50%,
    rgba(0,0,0,1)
);
top: 0;
left: 0;
}
```

## 7. Header children

All first-level children of the header element require absolute positioning in order to benefit from having a z-index. The first child is also styled with a width and height, along with a full border radius to appear as a circle. The same background used for the header is also applied - enabling this to be modified by the JavaScript in response to any mouse movement.

```
header > *{
    position: absolute;
}
header > *:first-child{
    display: block;
    width: 40vh;
    height: 40vh;
    background: red;
    border-radius: 100%;
    background: url(image.jpg);
    background-size: 100vw;
}
```



# The Adventure of Detective Moustachio

[zbeul.studio](http://zbeul.studio)



## Story Mode

**Designers:**

Renaud Rohlinger, Developer | Sofiane Hocine, Art | Lucas Fiorella, Music

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

**“More interactive experience than website, Detective Moustachio is a pixelated, side-scrolling platform game optimised for desktop and mobile”**



**Colours**



**Tools**

GSAP, core.js, Babel,  
Amazon AWS

**Fonts**

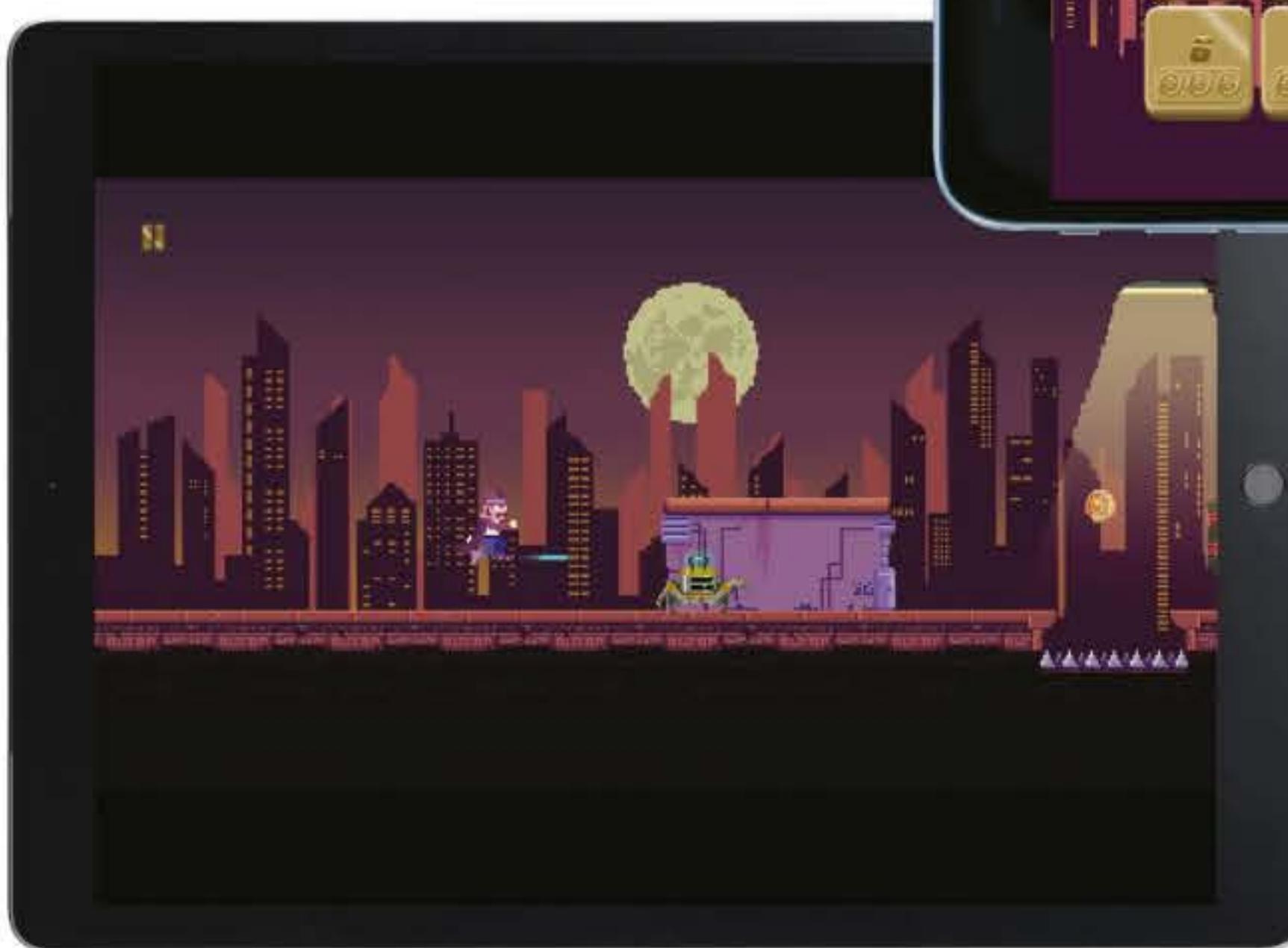
**abcABC  
1234567890**

*FFF Forward font by Fonts For Flash is the decidedly blocky typeface used for all the retro videogame text.*

**abcABC  
1234567890**

*Montserrat font by Julieta Ulanovsky is available via Google Fonts, and is also referenced within the CSS in 400 and 500 weights.*

**The cat of Marie-Pierre Tremblay' escaped again.  
The last time I found it I almost passed out.  
Another extreme investigation for the Detective Mustachio.**



**Top**

A skippable 'cut scene' intro presents what this HTML5 version of the game has in store for Detective Moustachio

**Left**

Essentially an auto-run platform game, player controls are kept simple on all devices, despite some clever upside-down physics at times

**Above**

The select level menu tracks progress, showing special coins collected and also enabling players to plot their own path

# Create a blip effect for hovered content

Use a visual blip effect to indicate interactive content elements

## 1. Initiate the document

The first step is to define the HTML document - i.e. the webpage. This consists of the HTML document container, which stores the head and body sections. The head section is used to load the external CSS file, and the body is used to store the webpage content created in step 2.

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

## 2. Page content

The page content consists of an article and two containers. The effect can be applied to any element using the 'blipbox' class. Images act in a way that stops the blip effect, and this example wraps each image inside a span element so the blip effect can work with images.

```
<article class="blipbox">
<h1>Hello</h1>
</article>
<span class="blipbox"></span>
<span class="blipbox"></span>
```

## 3. CSS: article

The example article is set to cover half of the vertical display space, with margin settings set to guarantee it to be centrally positioned. The styling for this element is separate and independent from the blip effect, although it is required to show how the effect would work with a content container element.

```
article{
    display: block !important;
    margin: 2em auto 2em auto;
    border: 1mm solid #000;
    padding: 1em;
    width: 50%;
    clear: both;
}
```

## 4. Blipbox definition

Elements using the 'blipbox' class are required to display as inline-block elements using relative positioning. This

enables any child elements to be placed in relation to their position, along with height and width properties becoming available. These are all required for the animation defined in step 7.

```
.blipbox{
    display: inline-block;
    position: relative;
}
```

## 5. Blip hover

A style rule is required to trigger an animation to elements using the 'blipbox' class whenever they are hovered over by the mouse cursor. This animation is applied to a virtual 'after' element, hence the use of the 'hover' and 'after' CSS selectors. The animation attribute specifies the name of the @keyframes animation and the duration to play it.

```
.blipbox:hover::after{
    animation: blipAnim .7s;
}
```

## 6. Overlay styling

The default settings of the 'blipbox' overlay are created using the CSS 'after' virtual element - as also referenced in step 5. This element is required to have default settings to define the starting state of the animation. It will cover the full area of the element, with semi-transparent black background. Opacity is set to zero in order to hide the element until the animation is triggered from step 5.

```
.blipbox::after{
    content: "";
}
```

```
display: block;
position: absolute;
top: 0;
left: 0;
background: rgba(0,0,0,.5);
width: 100%;
height: 100%;
opacity: 0;
}
```

## 7. Blip animation

The animation is required to move the 'after' element from the top left corner to slightly behind the top left corner while increasing its size. This provides the effect of the element increasing its size over the element. Another part of the blip effect is how the animation changes the opacity from fully visible to invisible.

```
@keyframes blipAnim{
    from{
        top: 0;
        left: 0;
        opacity: 1;
    }
    to{
        opacity: 0;
        top: -10%;
        left: -10%;
        width: 120%;
        height: 120%;
    }
}
```



# NEVER MISS AN ISSUE!



## Issue 281

Design, develop and create with top Google tools, add glitch effects with CSS, animate with GSAP and code UX patterns

SAVE UP  
TO  
WITH A DIGITAL  
SUBSCRIPTION  
SEE PAGE 50



## Issue 280

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



## Issue 275

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

## 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 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 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 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



## 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

## 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>



# BON VOYAGE

SUPERYACHT EXPERIENCE-PROVIDERS ASARO CHOSE GREEN CHAMELEON TO CAPTAIN A WEBSITE REDESIGN THAT WOULDN'T LEAVE GUESTS ALL AT SEA

ASARO  
[asaro.co.uk](http://asaro.co.uk)  
by  
Green Chameleon  
[craftedbygc.com](http://craftedbygc.com)

Project duration  
**4 months**

People involved

Nathan Riley  
**Design Director**

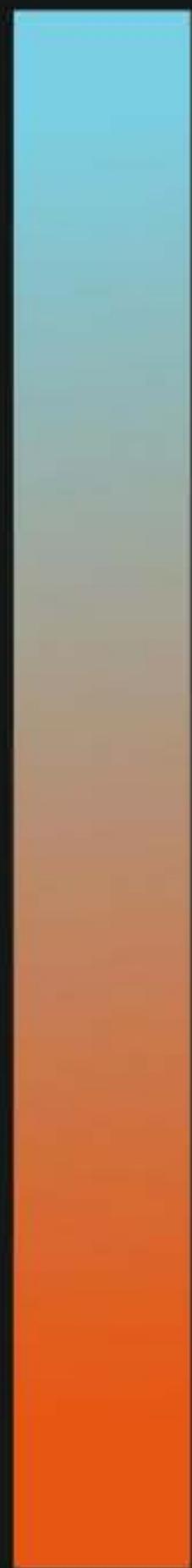
Tom Anderson  
**Creative Director**

Dan Linthwaite  
**Art Director**

Jack Harvatt  
**Designer**

Jake Whiteley  
**Senior Back-end Developer**

Jesper Landberg  
**Front-end Developer (Twotwentytwo)**



If you mused on the virtues of professional web design today, a big attraction must be variety. Because the internet and digital is now so vital and so ubiquitous in marketing, businesses and organisations of all conceivable kinds require quality websites. Take, for instance, a company that creates luxury immersive experiences on superyachts and megayachts - Asaro. This company designs and delivers bespoke, immersive theatrical experiences on an unimaginable scale all around the world. So when a client like that seeks out an online revamp, whoever answers the SOS call is in for a pretty diverse ride.

In this instance, the agency it reached out to is well-known within **Web Designer** pages for adapting to any project environment. Bristol-based Green Chameleon is a creative production studio that specialises in crafting extraordinary brand experiences that stand out, not blend in. Founded in 2012, this agency comprises a growing 'crew' of 13 talented designers, animators, developers, videographers and account managers. "We were approached by Asaro to redesign their website and help them better reflect their position within the seven-star luxury world," recalls GC's Design Director Nathan Riley. "They needed a creative way to visually represent their exclusive service offering, as photography was scarce due to the private nature of their events."

Asaro's audiences would be the captains, crew and owners of these immense vessels, as well as the guests who commission and enjoy the events they organise. So in order to speak their language while being discreet, the designers employed custom CGI renders, created to represent the client's event genres. "Inspired by water drop photography, we set out to create a striking and memorable aesthetic that embodied the spirit of each event and encapsulates the mystery and adventure weaved into all of Asaro's offerings." ▶



Nathan Riley  
Design Director



Tom Anderson  
Creative Director



Dan Linthwaite  
Art Director

## NAUTICALLY NICE

Green Chameleon knew that this kind of imagery would also tie the theatrical nature of the events to the nautical theme of yachting, which underpins everything Asaro does. As a perceptive agency, GC immediately recognised the potential for the project and the opportunity to work with a unique client in such a unique sector. "To say we were excited would be an understatement!" enthuses Creative Director Tom Anderson. "The brief and scope of work would draw on a huge breadth of our skills, allowing us to really utilise our team's creative and technical talents to deliver a stunning outcome, which is precisely the type of project we love to work on." They actually took the initial project call not in sunny Bristol but in Barcelona while attending the OFFF design conference. Already feeling inspired by the show, learning about the possibility of working on such an interesting and fun project got the gang particularly excited.

"After meeting with the Asaro founders and the wider team we felt the chemistry was there to make something extraordinary, and after submitting a proposal for the work we were thrilled to find out we'd been chosen to take the project on and realise our visions."

## PLOTTING COURSE

Making proper contact on their return home would be pivotal to touching base and talking. As an agency, Green Chameleon isn't exclusive in recognising that communication is key to any successful working relationship from the off. This catalyst for the delivery of great work across any given project has resulted in Green Chameleon structuring its design process around those principles.

"We make sure that from the beginning of every project, the client is appointed a member of our Client Services team to liaise with throughout, as well as being given exposure to all members of the creative team," says Nathan. "This ensures that creative requirements are understood and clearly interpreted our end and that any specific questions are answered directly and efficiently. We take an integrated approach to design and development, and the client is involved regularly to feedback over the course of the project."

With progress made on plotting the overall course, the focus could move onto discussing concepts. Presenting visual mood boards to convey initial ideas for the look and feel, the team explored colours and typography, along

with page layouts across print and digital. While offering the client insight into GC's vision and thought process, this phase also helped to hone creative direction in the early stages of the project and added context to the initial design presentation. "The biggest challenge we faced here was working out how to best visualise the Asaro offering," Anderson admits. "They operate in such a private and exclusive sector, and as a result photography and documentation of the actual experiences was very limited. It was this challenge that led us to create a series of computer-generated images to allude to the immensely detailed and immersive experiences that Asaro offers."

The art direction would draw on a blend of water drop photography and theatrical poster design, with the aesthetic of this photographic style seeming to fit nicely with the project's obvious nautical theme.

## EXTRA HANDS

Fresh from a conception phase that involved the

majority of the team, it was time to start facing the practicalities of building. Two members of the GC design team were assigned to handle the majority of the design work on the website moving forwards. The focus for creating the stunning 3D assets would fall on Art Director Dan Linthwaite, while Design Director Nathan handled the site and interaction design. In addition, they actually drafted in a helpful Swedish agency friend in the form of Twotwentytwo's Jesper Landberg. "We worked extremely closely with Front-end Developer Jesper to come up with the best interactions, keeping performance in mind throughout," confirms Nathan. "Sometimes this involved providing animated demos or specific micro-interactions but more often just chatting through ideas and seeing how best to achieve them."

This kind of extra technical assistance was appreciated, seeing as the WebGL underwater effect applied to the 3D renders would take considerable research and development to get right. "The code uses a displacement map to alter the intensity of the effect, so with this in mind we created several variations of the map to get the effect just right. In the CMS we built the ability to swap out the masks, meaning it was really efficient to try out a new version of the effect, cutting out a lot of needless back and forth during the testing phase."

Built upon a custom WordPress framework, the website would eventually be hosted on an Nginx/Apache hybrid DigitalOcean droplet running Ubuntu 18.04 LTS and PHP 7.2. As a rule ➤

**Adventure Reimagined**

Adventure without boundaries

Discover new worlds

Your trusted partners

What will you discover?

# Adventure Reimagined

WHAT WE DO

Incredible experiences

HOW WE WORK

Your trusted partners

*Asaro's website draws on the nautical theme while alluding to the mystery and adventure on offer*

## Discover new worlds

...sibilities, spectacular journeys, unimaginable adventures... the world at our fingertips, why settle for anything less?

## Adventure without boundaries

[Watch our video](#)


## Encounters



## AESTHETIC AMBITIONS

"We were driven by the ambition to create a striking visual that not only told a story but conveyed the idea of a mood without giving too much away," describes Creative Director Tom Anderson when quizzed on the initial design motivation. "It was here that we drew inspiration from theatrical poster design and chose CGI over photography, because we felt it gave us more freedom creatively, as well as ensuring consistency moving forwards." This inspiration would largely inform the project's art direction by mixing in a big dollop of water drop photography for good measure.

Underlining the nautical themes associated with a luxury superyacht experience, these metaphors applied to the 3D rendering, and WebGL work would also feed into other areas. "The website navigation and layout of the site was inspired by the journeys and maps that are at the heart of an Asaro experience. The site features an interactive dotted line that fills up as the user scrolls down the page, taking them on a journey of the Asaro offering. We also built an interactive slider to unlock the experiences section that evokes a sense of discovery and playfulness, a nod back to the Asaro experiences themselves."

## EARNING STRIPES

It stands to reason that all agencies like to be loved and place massive emphasis on satisfying their clients. The Green Chameleon crew are no exception of course, and a big part of the excitement of embarking on a project journey such as this is in seeing how the site fulfils the brief post-launch. "The website has generated positive feedback from Asaro's industry peers and their clientele," beams GC

Creative Director Tom Anderson. "The brochure was also well received at the Monaco Yacht Show, proving that the overall redesign worked to highlight Asaro's standing as a luxury brand with a modern approach among a field saturated in the usual hallmarks of opulence." This was hugely important to the client's overall campaign objective, with Asaro's rebranding drive being christened at such a Champagne-soaked industry event.

The site's production deadline was timed to have the website launched and live in order to premiere at the show, but perhaps the bigger reception has been how the online world has embraced the work. "We were extremely proud that the website won numerous notable design awards, including Site of the Day awards from Awwwards, FWA and CSS Design Awards and features on Communication Arts and Mindsparkle Mag, as well as a Mobile Excellence & Developer Award from Awwwards, and not to mention the very exciting FWAWWWARD."

**EXAMPLE EXPERIENCE**

### Enchanting on board entertainment



A troupe of mystical storytellers arrives on board to tell the story of Pandora's box. The children help Pandora calm the evil spirits using music and dance, and safely return them to the chest. Task completed, the storytellers leave the yacht with a promise to return the following day.

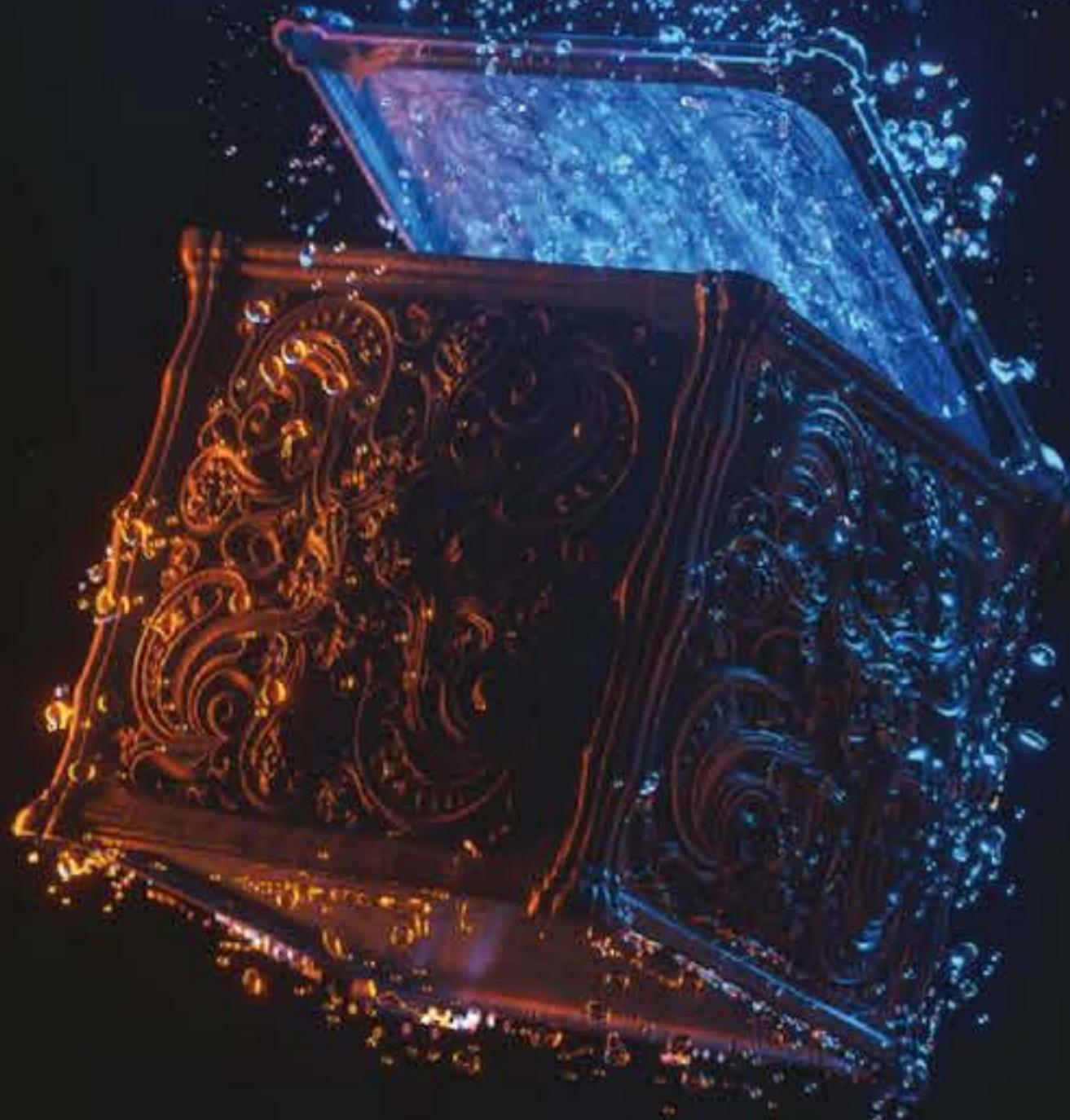
**Into the labyrinth**

A group of children explore the secret labyrinth on board the yacht. They follow the winding paths, and solve the puzzles to find the hidden treasure at the end of the tunnel.

**Start your journey →****Discover new worlds****Challenges**

Cloud-based games and challenges designed to stimulate, engage and entertain.

**Discover more**  
**ASARO****EMAIL**  
enquiries@asaro.com**PHONE**  
(+44) 0117 922 6892**SOCIAL MEDIA**  
Follow — @ in



► of thumb, the GC developers used plugins and third-party code as little as possible, preferring to build purpose-driven solutions in-house.

### JQUERY MAROONED

"With Asaro being so JavaScript heavy, this led us to completely drop jQuery for the project, which threw a lot of traditional WordPress form solutions out the window," begins Senior Back-end Developer Jake Whiteley. "We used the popular Axios library to handle the form submission, along with our custom framework to implement Laravel-like request validation on the backend." This decision to throw jQuery overboard would actually prove to be very popular with the developers, becoming a tactic they intend to actively pursue on future projects.

Keen to be as agile as possible, the GC coders also tried to keep back and front-end development happening in parallel. "Working in parallel enables us to respond to site structure and content changes as quickly as possible, and works well with our interaction design process. Plus the transference of ideas from our designers' heads into working animations can involve a lot of fine-tuning, and parallel development ensures that progress is being made, regardless of any blockers. With Asaro having such a tight turnaround time, this meant front-loading the back-end efforts to ensure that the site content structure was in place before the main bulk of work started."

### FIT FOR LAUNCH

Such a strategy was of course pivotal given the tight four-month deadline agreed between the

two parties. The finishing line would arrive in the form of the Monaco Yacht Show, an exclusive event for superyacht owners, charterers and decision-makers in the yachting and luxury industries. The client, Asaro, was naturally attending the show, with a view to launch their exciting new rebrand.

"The main challenge was ensuring that everything, from the business cards to the printed brochure, and most importantly the website, were all finalised and perfectly polished in time for the show," says Tom Anderson. "Plus, remember that the emphasis on the website's speed and performance was even greater in the instance of this project because we were constrained by the 3G or 4G connections at sea. With the majority of the target audience being captains of superyachts, we had to ensure the site's load speeds were fast and optimised for mobile and tablet viewing, as well as desktop."

Similarly, it was never going to be a case of just pushing the finished site out into the ocean of the world wide web. As part of such a shipshape relationship, there was a commitment to keep the thing afloat after setting sail. "Along with managing the hosting of the Asaro website, we're also on hand for any additions or maintenance that our designers or developers are required to provide support with," Nathan concludes. "The site was built on WordPress and so the majority of content is editable, which purposefully gives the client full control for future developments. However, we encourage an ongoing relationship with all our clients and have already begun discussing with the Asaro people about what's on the horizon next!"

## SITE HIGHLIGHT

We asked Green Chameleon's Design Director Nathan Riley to identify a favourite feature from the project

"The standout feature for us has to be the custom CGI renders that were created to represent Asaro's event genres. The striking visuals have proven to be a powerful visual mechanic that has created intrigue among Asaro's prospective clients. These assets have been a great success, but they were also a lot of fun to create and we were able to try out some new techniques in the process."



## WHO

Code and Theory

## WHAT

interactive products,  
campaigns, advertising,  
branded experiences,  
design, branding, user  
experience, interface  
design, graphic design,  
strategy, engineering,  
product UI, website  
design and development,  
motion graphics, rich  
media, and social media

## WHERE

1 World Trade Center,  
62nd Floor, New York,  
NY 10007

## WEB

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

## KEY CLIENTS

- [1] adidas
- [2] Comcast
- [3] NBC
- [4] Nasdaq
- [5] American Bar Association



Mike Treff (President)  
explaining his latest concept  
to an enthralled audience

# Decoded DESIGN

**With extensive experience in crafting digital landscapes, Code and Theory defines the relationship brands have with their customers across multiple channels. Connecting design and technology, Code and Theory creates user experiences that reinvent how companies can digitally transform**



The Code and Theory team admiring the inspirational view from their New York offices

**D**an Gardner, the CEO of Code and Theory, co-founded the agency in 2001 with his childhood friend Brandon Ralph. Dan and Brandon were working in the digital department of Draft, where they quickly realised that the full potential of digital did not exist in CD-ROMs, and so they left to form their own agency.

As a recent graduate of the University of Buffalo, Dan had received a BFA in Computer Art, while Brandon had dropped out of NYU to work full time. Realising that their ideas were not being put into practice, they decided to start their agency to create transformative digital experiences and products.

Initially working out of a one-bedroom apartment on the Lower East Side in New York City, they then moved into a Wi-Fi-less, abandoned floor of a downtown office building. Today, they are headquartered on the 62nd floor of the World Trade Center and have

additional offices in San Francisco, London and Manila.

Code and Theory craft new digital experiences. This level of understanding is reflected in the name Dan and Brandon chose for their agency, as Dan explains: "The name came first from a desire to play in the area where creativity and technology intersect. We wanted to imply our digital focus without being exceedingly overt and having a '.com' in the middle of our name. Code and Theory was a novel idea in 2001, so it wasn't hard to get the domain. Our name really speaks to why the agency was formed and our approach to creativity, through strategy and technology."

The website that Code and Theory currently maintains speaks volumes about not only the approach the agency takes to digital design, but also its skills with today's digital tools. Michael Treff (President) explains the importance of maintaining a powerful and



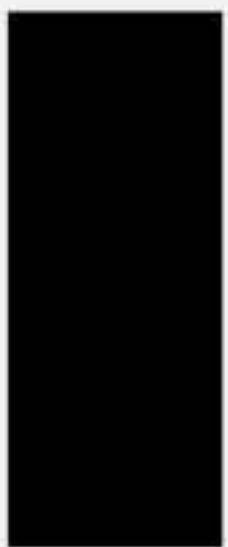
Keisha Brescia (COO, PMX Agency), Larry Muller (Chief Operating Officer), Naomi Soviecke (Financial Operations Manager), Naren Persaud (Financial Controller), Jonathan Chan (PMX) (Controller) and two other colleagues from PMX thrash out a new proposal



"Built upon a history of using digital as the catalyst for larger organisational change, the projects that enable us to take on thought-provoking questions that engage the development of our capabilities are the ones that define who we are. The types of work we have put out are intentionally very broad across verticals and expressions.

We believe that broadness characterises us more than any singular piece of work"

**Dan Gardner**  
CEO and co-founder



informative site: "It's what guides the first impression when someone is looking into us, so we think it's very important. Not only is the website a window into our agency, employees, culture, work and clients, but we also receive a lot of inquiries on our site from prospective clients, job candidates and press. Truthfully, we spend so much more time on our clients' work and their websites than we do our own, but I don't think that's unique to a company like ours, where we put our heart and soul into our clients' initiatives and strive for continuous improvement. That said, we have been working to overhaul our site and hope to launch a new one soon."

As an established agency, Code and Theory has been able to use its reputation across the marketplace to enjoy a steady stream of clients. "We're fortunate enough to have built a very good reputation in the industry over the years, so we receive a lot of referrals from current or former clients as well as inbound inquiries on our site," says Dan. "Additionally, we proactively identify verticals, clients, and most importantly new challenges that we want to work with, as well as participate in RFPs (requests for proposals) and pitches."

With a diverse portfolio of work and clients, Code and Theory has been able to choose which clients they work with. The selection process is detailed, involving a deep dive into the client's needs in order to understand how the project benefits both the potential client and Code and Theory.

"Typically, we're very selective to ensure that the relationship with the client is a true fit and that we genuinely connect with the company and the work," explains Michael. "We are also very realistic about our capabilities and resources, so we won't take on any piece of

work just for the sake of new business. That being said, we've been extremely fortunate to work with clients who trust us with work that has allowed us to challenge and expand our core capabilities, and we've been able to hire into those new projects.

"We like to think of ourselves as ambitious, since we do aim pretty high. Not to say we won't ever take on smaller projects, but they are usually a stepping stone into a larger engagement or, more often than not, a project or challenge we feel very passionate about."

Passion coupled with a deep understanding of today's digital communications channels means Code and Theory has been able to forge a reputation as not only technical experts but also a group of like-minded individuals that shape the digital landscapes we all traverse every day.

Dan explains their typical approach to each unique account they work on: "As a digital-first creative agency, we help our clients across the board to drive change and thrive in a time of digital disruption - so there really isn't one project that encompasses who we are as a company. Built upon a history of using digital as the catalyst for larger organisational change, the projects that enable us to take on thought-provoking questions that engage the development of our capabilities are the ones that define who we are. The types of work we have put out are intentionally very broad across verticals and expressions. We believe that broadness characterises us more than any singular piece of work."

As no two accounts are the same, Code and Theory is nothing if not flexible in its approach to each commission it takes on. David DiCamillo (Partner, Operations) explains how they manage a varied portfolio of clients: "Timelines

**NBC NEWS**

**NIGHTLY WITH LESTER**  
WEEKNIGHTS AT  
THE STORIES THAT MATTER

**TOP STORIES**

**HURRICANE IRMA**

**Irma churns toward Miami, could wash away buildings**

Hurricane Irma was on a track to slam Miami this weekend as it continued to cut a fatal and destructive swath through the Caribbean islands Thursday.

Hurricane Irma skirts Puerto Rico, leaves 1 million without power

Airlines scrap address price heavy demand Irma

**MACH**

**BETTER**

**THINK**

“As a digital-first creative agency, we help our clients across the board to drive change and thrive in a time of digital disruption”

**TODAY**

**TODAY**

**TODAY**

It can collect the cool air at night time and keep (the house) cool throughout the day.

Will Smith's evolution from 'Fresh Prince' to movie star

Many retailers offer student discounts where you can save up to 15 percent by showing a school ID. Additionally, Apple is making it easier for students to update their profile

## NBC NEWS

today.com

### THREE BRANDS, ONE SYSTEM: THE NBC NEWS DIGITAL REDESIGN

Code and Theory partnered with NBC News, one of the oldest and most established publishers with over 80 years experience, to revamp their digital ecosystem. As one combined team, they worked across a wide range of areas to rethink existing paradigms, including editorial workflows, design and development processes, design systems, advertising integration and technical approach.

The efforts started small and focused on the design and launch of four new flagship verticals: Mach, Better, Leftfield and Think. Work then expanded to encompass NBCNews.com, MSNBC.com and today.com. After the success of implementing a fresh design language and strategy, efforts were turned to NBC's Snapchat Show, Stay Tuned, the first daily news program of its kind.

As the ever-shifting media landscape continues to become more digitally focused, Code and Theory re-evaluated why, when and how people are consuming news, and sought to find a way to retain users' attention among all the bitesize content consumption options available. To do this, Code and Theory built a highly flexible set of storytelling tools that allowed NBC to focus on creating a strong editorial voice and bolstering confidence in their ability to deliver engaging and immersive original content to users, regardless of medium.

## Decoded Design



Defining and then creating each customer touchpoint delivered an integrated customer journey

# XFINITY MOBILE

xfinity.com/mobile

When Comcast wanted to enter the fiercely competitive mobile provider market, they asked Code and Theory to define, design and develop their mobile offering. They needed substantial support - from refining the vision and product concepts, establishing key value propositions and unique selling propositions to deal with the competition and overcome negative marketplace perceptions.

Xfinity mobile's e-commerce model necessitated the creation of an alternative product experience to contend with those employed by its retail competitors. A personalised solution was needed that

would address challenges related to in-home activation, custom order logistics, procedural implementation of technology and evolving mobile standards. Solving these issues required that essential information was supplied at efficient instances, metered through employing an elegantly simple, yet remarkably flexible unboxing experience.

In between online order placement and successful activation lie a series of logistical and user-dependent steps that must be correctly completed. Not only does this process require itself to be concrete, but the brand experience must also follow suit. As

reflected by the customer and independent feedback, the final design, experience and brand integrity scored high marks. The unboxing experience itself was met with the highest customer acceptance response across the entire brand journey.

This partnership with Comcast extended to nearly every touchpoint for Xfinity Mobile, defining an end-to-end consumer journey. Code and Theory designed and developed the complete mobile-first experience and created a packaging system based on the unique footprint of the phones themselves - including a special edition for the first 20,000 customers.

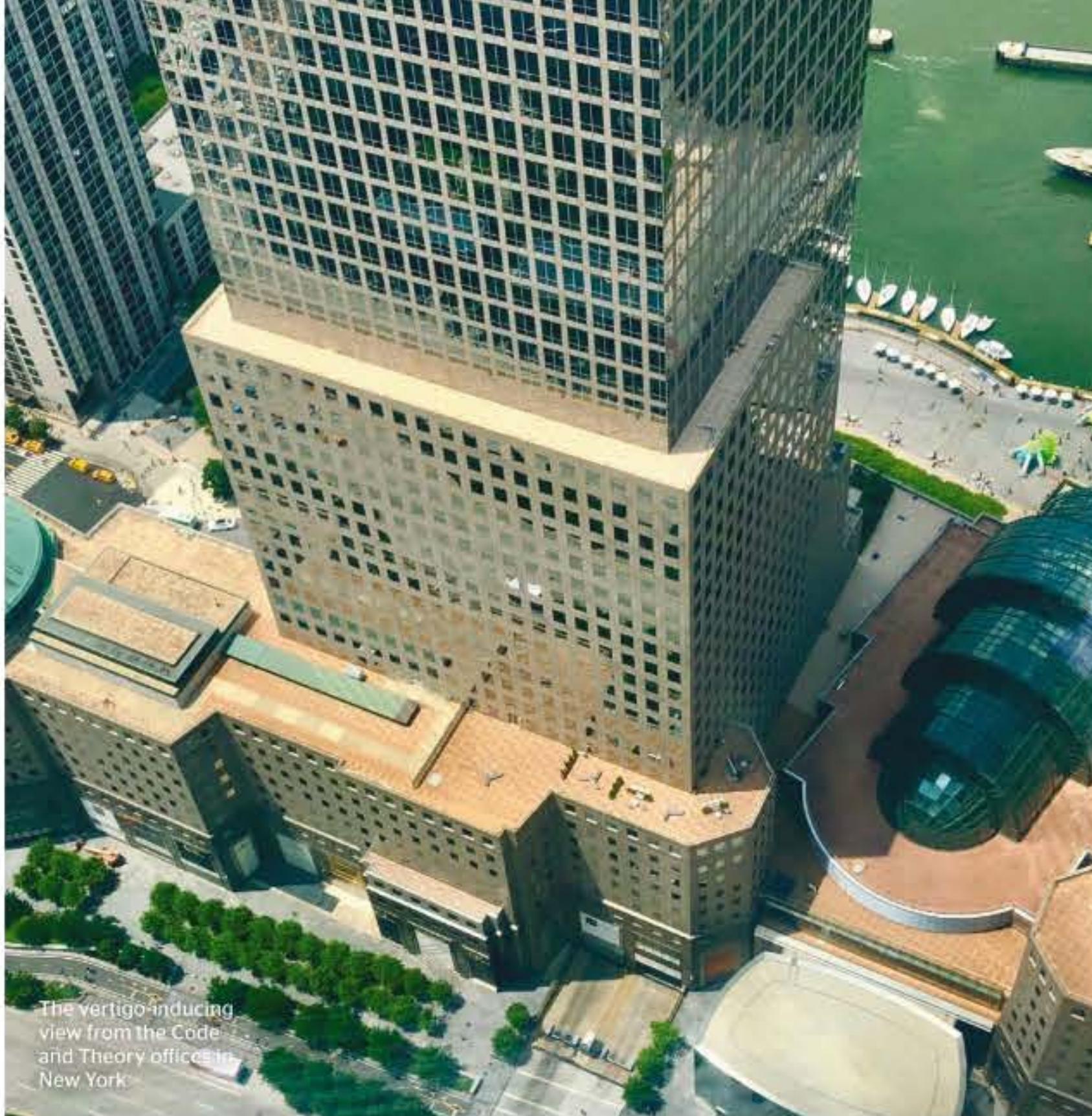
xfinitymobile Plan Shop Support My Account

Creating experiences around a brand is a core design skill that Code and Theory used on their work with Comcast

## XFINITY Mobile is here

Introducing a new kind of network designed to save you money.

Watch Video



and process vary greatly depending on the scope of a project. However, every single one of our projects is structured around a tight and dedicated interdisciplinary team we call 'Pods'.

"The Pods consist of, but are not limited to, skills such as Creative Strategy, Production, Content Strategy, Experience Strategy, Interaction Design, Visual Design, Data/Analytics and Technology. When the project requires additional scale or speciality outside the Pod, we add Flex Resources based on the requirements of the project phase. Within a Pod, we designate someone as the Directly Responsible Individual (DRI) to help drive leadership and vision across the team to prevent 'design by committee' decision-making, as well as ensure accountability for project success."

David concludes: "Depending on the scale and complexity of a project, the time-consuming part can vary. Sometimes that is more in the strategic upfront to identify or solve the problem, or sometimes it could be in the technology if there is some complexity or scale in the digital expression of the solution. Most importantly, we don't let process dictate an outcome to get to success - we use process as a facilitator to success."

As an agency that creates digital communication assets that both fulfil their client briefs and also advance the digital environment, Code and Theory has a varied toolset it wields with absolute confidence, as Chris Delia (Group Creative Director) explains to **Web Designer**: "Our product design work is strictly done in the Sketch app, while we

prototype in InVision or Principle based on the fidelity of the prototype that we need to create. Sketch actually slowly worked its way into our design process a few years ago, after putting it through a lot of testing. There was a kind of breaking point with Sketch through one particular software update in which almost everything we were waiting for was implemented all at once, so we immediately switched over to Sketch.

"Prior to Sketch, we were using Photoshop or InDesign for our responsive design work, which sometimes caused friction between projects and designer taste. Both Photoshop and InDesign did the job just fine, but neither were truly built for modern digital design needs. There was a grey area missing in the tools we were in need of that Sketch filled for us. There are a number of reasons why we are sticking with Sketch, but what stands out the most is how much more productive we have become as a team since using it. It has greatly streamlined designing across platforms, to exporting assets, to working faster with developers. Unlike the Adobe suite of products, Sketch was built specifically for what we do in the product design space."

Chris concludes: "We have also done away with the time-consuming style guide process, which used to eat up a lot of time and money, by adopting Zeplin app into our process, which has cut the style guide production time down by 90 per cent. For content creation, we will use the Adobe suite. This is anything from photo composites, touch-ups, GIF creation, video effects, etc. For 3D we use Cinema 4D

## 2001

Dan Gardner and Brandon Ralph found Code and Theory.

**Employees: 2**

## 2002

Code and Theory develops the first-ever Flash Player website, for Sony Classical, and create the first Flash videos for Sony Music, setting off years of Flash video being the dominant way to consume online video.

**Employees: 3**

## 2006

Code and Theory opens its San Francisco office, brings one of the first large-scale content catalogues online for Charlie Rose and launches the largest online live streaming events in history at the time for Dr. Pepper.

**Employees: 15**

## 2007

Working beyond digital, Code and Theory launches an animation series in partnership with McCann Erickson for the Subway.

**Employees: 15**

## 2008

Launch of *The Daily Beast*, bringing together core strengths - advertising and content delivery - to execute the vision of Tina Brown and Barry Diller to reimagine online news and advertising.

**Employees: 15**

## 2012

Mashable: first-ever page-to-page continuous scroll on ('The Transporter module') launches, which is now standard behaviour on most publishing platforms (a world first).

**Employees: 15**

## 2013

Code and Theory establishes its Industrial Design practice, the first digital-first agency to move into physical hardware design.

**Employees: 15**

## 2014

Launches its first 23 magazine solution (Hearst: *Cosmo*, *Esquire*, *Elle*, *Good Housekeeping*, etc.). Launches the first Olympics site for the Sochi Winter Games (NBC). Launches *latimes.com*, at the time the largest fully responsive news site in the world.

**Employees: 15**

## 2015

Code and Theory is named one of Fast Company's 10 Most Innovative Companies in Design for work creating publishing platforms. Code and Theory acquires Mediurrent.

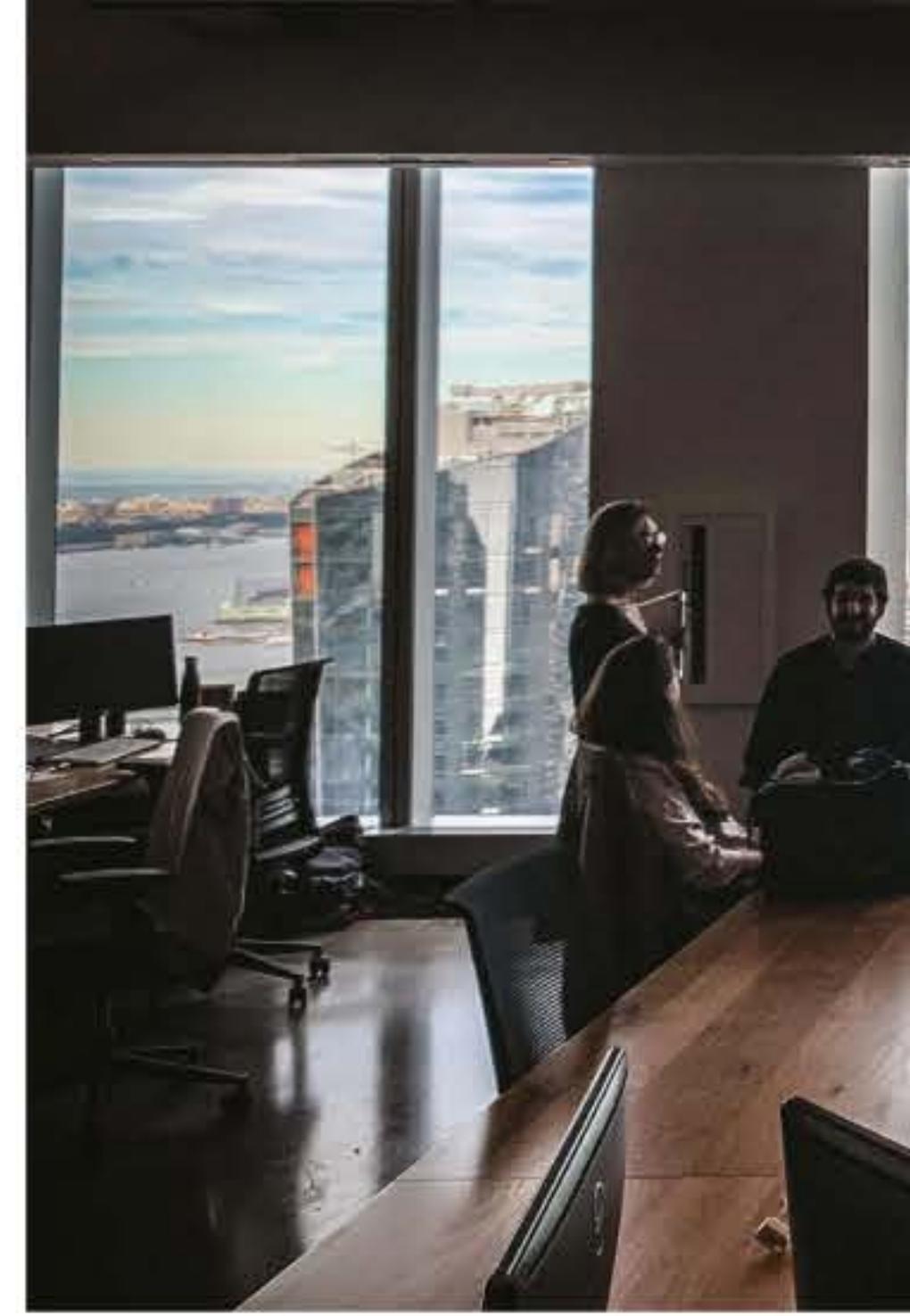
**Employees: 15**

## 2018

Code and Theory moves its NY HQ to 1 WTC

**Employees: 310**

Ren Vasey (Content Strategist) and Asli Aydin (Creative Product Strategist) discussing their latest client brief



and SketchUp. That said, we are a company that embraces change and improvement, so we are always open for an evolution in tools."

Using the latest tools to advance its vision for a client is commonplace across Code and Theory. Chris continues: "Whether we are considering tightly coupled or decoupled architectures, there is a growing tendency for websites to embed more of their logic on the client side. As such, the frontend technologies have to not only accommodate for advanced animations, interactions and responsive or fluid design in an increasingly complex device or browser environment, but they also have to account for more elaborate rules and flows.

"JavaScript is evolving quickly to be on par with these expectations: ES6 is introducing advanced notions, such as collections, promises, advanced inheritance, etc. that bring it closer to more traditional languages. This is all meant to allow for more complex experiences, which in turn means that these technologies will be more prevalent in the future. Frameworks such as AngularJS and React are already used by most major sites, in order to capture data from different sources and render the right experience for the user. Others, such as Three.js or D3.js, help developers create visually engaging experiences with advanced visual effects."

Mobile digital devices have clearly had a profound impact on digital communications. Laurent Pierre (CTO) explains how Code and Theory approaches these technologies: "As a pioneer in mobile and responsive design across industries, Code and Theory was among the first companies to design and deliver large-scale, responsive experiences in publishing (*Los Angeles Times*, the largest fully responsive newspaper in the United States), e-commerce (Moda Operandi, the first fully responsive



Tavit Geudelekian (Associate Director of Creative Strategy), Alexis Baran (Creative Product Strategist and Designer), Alexa Varsavsky (Associate Director of Creative Strategy), Erika Templeton (Associate Director of Content Strategy), Thomas Strickland (Associate Experience Strategy Director) and Nikhil Bhasin (Senior Engagement Manager) at work

luxury shopping experience targeted at high-net-worth clients), and mobile video consumption (MobiTV, at the time the largest white-label mobile video provider).

"To us, responsive design is about more than just responding to the user's device. Leveraging what we know about a user, we layer onto their experience mechanisms of adaptation that present the content, services, product offerings and resources most relevant to them."

As an agency that doesn't stand still, Code and Theory always has an eye on the future, as Dan outlines: "Powerful processing and accessible cloud computing have also allowed us to utilise artificial intelligence in various experience formats. Computer vision, machine learning, natural language processing, etc. are now all available for us to use at the mobile device level, and in turn we are able to create more tailored experiences for the user by bringing together a more contextually coherent experience. The access to readily available tools such as Google Cloud ML services and TensorFlow allow us to train our own models and are tools to help us to solve problems and create new ways for our applications to interact with users.

"And finally, there has been a great innovation in the creative development realm where tools such as Cinder, openFrameworks and TouchDesigner are paving the way to create all sorts of new physical and device-based experiences. These software programs, along with hardware micro-controllers and sensors, allow us to create things that connect the digital and physical world in ways we have not been able to before."

Code and Theory is a multifaceted agency that clearly understands what digital communications means today. Not content with following trends, this agency actively creates

agile spaces that consumers now move through. An appreciation of the link between great design, storytelling and a command of the latest tools ensures the brands it works with are able to innovate, astonish and understand their customers with well-crafted messages.

As Dan concludes, Code and Theory is a business that constantly pushes at the envelope of what digital design means today: "We continue to look at ways we can service our clients from the perspective of 'the user' and marrying that to business opportunities, so our agency's only limitation is our own creativity and not just tactical capabilities.

"Change is the constant, so we don't expect tomorrow's Code and Theory to be the same as today's, and that is what excites us. As for exciting upcoming projects, you'll have to wait and see."



"To us, responsive design is about more than just responding to the user's device. Leveraging what we know about a user, we layer onto their experience mechanisms of adaptation that present the content, services, product offerings and resources most relevant to them"

**Laurent Pierre**  
Chief Technology Officer

## CODE AND THEORY

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

### Founders

Dan Gardner  
Brandon Ralph

### Services

Data and Analytics

### Year Founded

2001

Platforms and Products

### Current Employees

310

Integrated Marketing

### Location

New York, USA | London, England | San Francisco, USA | Manila, Philippines

Connected Experiences

Business and Organisational Consulting





# What's inside?

Welcome to our CSS special issue. Take a look at what's on offer

## 42 CSS animation

Explore everything from reasoning and planning to implementation

## 52 Build with CSS Grid

Learn how to build basic grids before moving on to more advanced techniques

## 58 Optimise your CSS

Discover a host of tips and techniques to make your CSS blazingly fast

## 64 Style with Sass

Power up and extend your CSS with the popular preprocessor Sass

## 68 Create CSS art

CodePen engineer Cassidy Williams reveals how to create CSS art and offers some useful advice

## 72 Best CSS methodologies

Ahmed Abuelgasim explains what CSS methodologies are and how to use them

## 76 Cutting-edge CSS

The hot properties and APIs that you should be using today. Plus a peek at the future



---

# ANIMATION

---

ANIMATION IS AN IMPORTANT PART OF MODERN UX AND IS EASIER THAN EVER TO IMPLEMENT WITH CSS. IN THIS FEATURE WE'LL EXPLORE EVERYTHING FROM REASONING AND PLANNING TO IMPLEMENTATION



“ While it may seem limited or a lesser tool when it comes to animation, CSS is actually a really powerful tool and is capable of producing beautifully smooth 60fps animations ”

Steven Roberts, Head of UX, [Asemblr.com](http://Asemblr.com)  
@matchboxhero10 [codepen.io/matchboxhero](http://codepen.io/matchboxhero)

## BEFORE YOU BEGIN

Planning your animations and knowing when to use them can be just as important as the animation itself

### WHERE TO USE ANIMATIONS

Animations can become as important to your brand as your primary colour or logo.

#### ■ ANIMATE INTERACTIONS FOR BETTER UNDERSTANDING

Sudden changes to the state of your application or website can be jarring and can easily cause confusion. Animating interactions is a great way to help the user understand the change of state.

#### ■ DELIGHT YOUR USERS WITH MEMORABLE EXPERIENCES

Animation can be used to provide delightful experiences. You don't need to go overboard - something small and subtle can create a memorable experience, which can keep users coming back.

#### ■ INFORM USERS WITH PROGRESS INDICATORS

Progress indicators should be animated to inform the user that the system is still responding. Users have come to expect immediate feedback; when submitting data or loading content, a simple spinner can be enough to keep their attention.

#### ■ USE PERCENTAGES TO REPRESENT PROGRESS

Percentages when used to represent progress are a great place to use animation. Watching a bar or shape fill up is rewarding; visually representing a user's goal can subtly encourage them to complete it.

#### ■ DON'T ANIMATE EVERYTHING, IT'S NOT NEEDED

It's important to know how and when to use animation. You don't want to animate all the things and create the mess that GeoCities or MySpace became. Animation should be used sparingly.

## STORYBOARDING

### QUICKLY PLAN YOUR ANIMATION'S FLOW

Storyboarding is a technique used in many disciplines and industries. It is, for example, used for video games, TV shows, movies, music videos and any other discipline with a narrative and visual element to it.

### WHAT IS STORYBOARDING?

Storyboards are illustrations of the key frames or elements that will make up the final works. This allows for ideation of visual elements, which would take a lot of time to produce in full, to be quickly, easily and cheaply captured.

### WHAT ARE THE ELEMENTS OF A STORYBOARD?

**Panels:** Panels are the viewport for your animation. Your panels can be any size or shape you want, but need to match the container size and aspect ratio when you put them into the browser.

**Framing and composition:** This is choosing where in your panel to place the elements you're going to animate. This is

important for deciding if your panel edge will be visible or not. If you don't want it to be visible you need to keep everything in frame at all times.

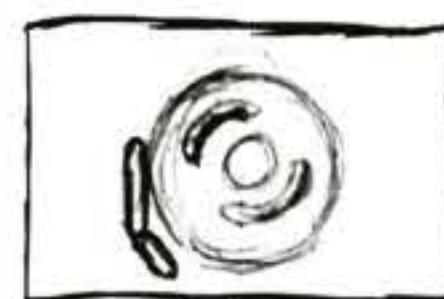
**Arrows:** Arrows in storyboarding are used to convey movement, showing the direction an element will be moving.

### HOW DO I CREATE STORYBOARDS?

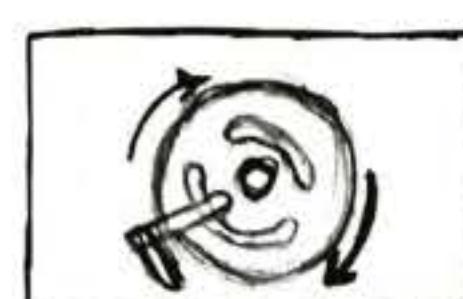
There are no solid rules when it comes to storyboarding, and you don't need to be able to draw either. The idea is to convey the timeline using as few sketches as you can. You can annotate your panels in order to explain certain features of your animation where needed.

### WHY STORYBOARD FOR WEB ANIMATIONS?

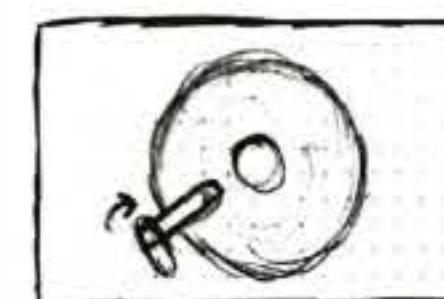
If you're creating anything more than a simple transition, storyboarding the elements can save you a lot of time. Before you even write any code, you understand how to break down the elements involved and where each of the key frames of your animation should lie on the timeline.



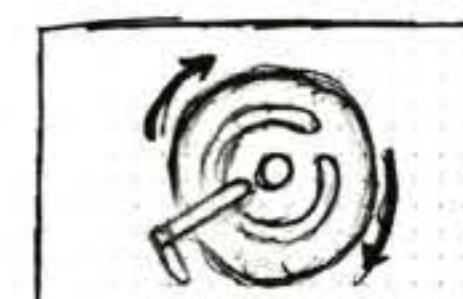
START W/ NO MOVEMENT  
ARM OFF TO THE SIDE



SPIN THE RECORD  
ARMED IN A  
CLOCKWISE  
DIRECTION



MOVE (FROM THE  
HINGE) THE ARM  
TO OVER THE  
RECORD



CONTINUE TO SPIN  
BUT AT A FASTER  
SPEED

Simple storyboard sketch showing how an interaction with a record player could animate

## CSS TRANSITIONS

CSS transitions can be used to achieve a number of animated effects, from a simple colour change to more complex transitions

### WHAT IS A TRANSITION?

Put simply, transitions are a way to provide animation between two property values. For the animation to trigger, something needs to change in the application or website.

### HOW DO TRANSITIONS WORK?

Transitions in CSS are simple, we just need to choose what elements to transition and when. For example, if we have a button and we want to change the background colour gradually instead of instantly when the user hovers over the button, we use a transition.

```
.button {background-color: Crimson; transition: background-color 500ms;} .button:hover {background-color: DarkRed;}
```

### TRANSITION SYNTAX

Transitions in CSS are made up of four properties, giving us control over how the transition will animate.

### TRANSITION-PROPERTY

The 'transition-property' property enables us to choose which properties we want to animate. We can transition a number of different properties. A full list can be found online at <https://mzl.la/2BpQJet>.

### TRANSITION-DURATION

The 'transition duration' property enables us to control how long the transition from one property value to another will take. This can be defined in either seconds (s) or milliseconds (ms).

### TRANSITION-TIMING-FUNCTION

Timing functions or easing enable us to adjust the rate of change over time. There are a number of keywords we can use. For example, the 'linear' keyword will transition from A to B at an equal temperament, whereas 'ease-in-out' will start slowly, speed up in the middle and slow down towards the end. Custom timing functions can also be defined using the 'cubic-bezier' property. A full list of timing keywords can be found at <https://mzl.la/2OXGB0e>.

### TRANSITION-DELAY

Transitions can be delayed using this property and is set using seconds or milliseconds.

### TRANSITION SHORTHAND

All of the transition properties can be combined into a shorthand statement using the 'transition' property;

```
transition: property || duration || timing-function || delay;  
transition: background-color 500ms linear 250ms;
```

We are free to omit the values we don't need in order to set the defaults.

### COMBINING TRANSITIONS

You can combine multiple transitions to create choreographed animations. Check this example:

```
transition:  
background-color 500ms linear,  
color 250ms ease-in 250ms;
```

## WHAT ABOUT BROWSER SUPPORT?

Support for transitions and animations in modern browsers is really good. Everything from Internet Explorer 11 or above is going to support the majority of the code needed for animation.

There are exceptions with some of the newer animation properties; CSS Motion Path, for example, or when using SVG or custom properties (CSS variables) as part of the animation.

## DO I NEED TO USE PREFIXES?

Prefixing for the most part is probably not needed unless we need to provide support for older browsers - Internet Explorer 10 and below.

There are several ways we can prefix code if needed. Auto-prefixer (<https://autoprefixer.github.io>) is a really useful tool that can be used as part of a build process or manually at the end of a project. It enables us to configure the browser support you need, then it will automatically prefix our code where needed.

We can also check the support for any property using the amazing tool 'Can I Use' (<https://caniuse.com>). Simply type in the property we want to use and see which browsers are supported.

### HAMBURGER ICON

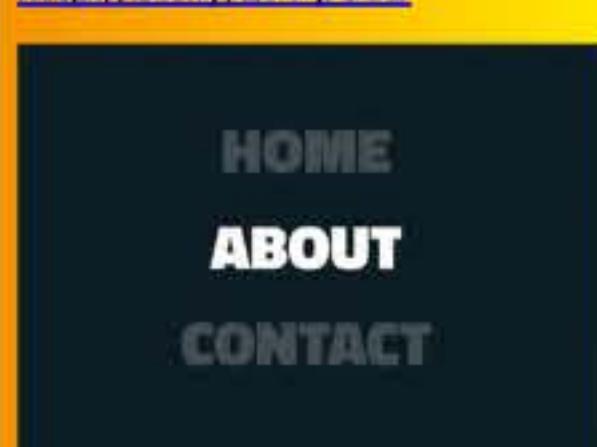
<http://bit.ly/2S8ptXI>



Upon hover the colour changes

### LARGE TEXT MENU

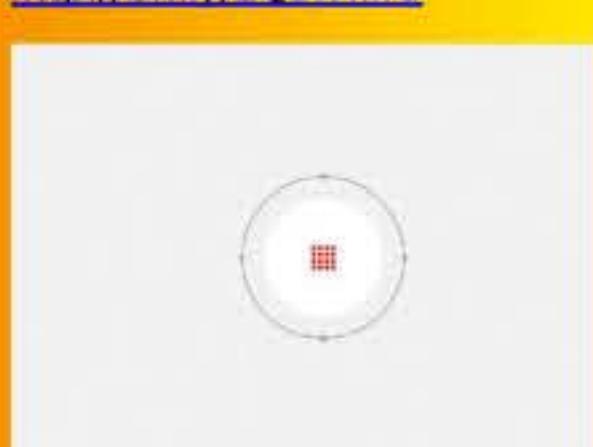
<http://bit.ly/2PPp3bZ>



The menu items light up

### CIRCULAR MENU

<http://bit.ly/2QaWIMG>



The middle icon changes colour

### ANIMATED CORNERS

<http://bit.ly/2FwujfI>



The triangular corner is animated

## CSS KEYFRAMES

CSS keyframes are used for complex or repeatable animations



### WHAT IS A KEYFRAME ANIMATION?

Keyframe animations in CSS enable us to define multiple property values to animate between. Keyframe animations can be reused and repeated, unlike CSS transitions.

### DEFINING KEYFRAME ANIMATIONS

CSS keyframe animations are defined using the '@keyframes' syntax. This works much like a media query where we nest elements inside of the '@' statement. Inside the keyframe declaration we have two options: we can use the keyword 'to' and 'from' or we can define our timeline using percentages.

### KEYWORD ANIMATIONS

When the animation we're creating only has two points to animate between, we can use the 'to' and 'from' syntax, in fact we can use just 'to', providing the original property value is set on the element we're going to be animating.

```
#record {
    transform: rotate(0deg);
    animation: rotate 1s;
}

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

### PERCENTAGE ANIMATIONS

When creating animations where we need to define more than one point to animate, we can use percentages. This enables us to have precise control over our animation.

```
@keyframes flash {
    25% { opacity: 1; }
    27% { opacity: 0.4; }
    29% { opacity: 1; }
    31% { opacity: 0.4; }
    75% { opacity: 1; }
}
```

### APPLYING OUR ANIMATION

Animation in CSS has a number of properties we can set in order to have precise control over the playback of our keyframe animations. Some, like 'animation-duration', 'animation-delay', 'animation-iteration-count', 'animation-play-state' and 'animation-name' are all fairly self-explanatory, while some of the other properties can be a little trickier to learn and utilise to their full potential.

### ANIMATION-TIMING-FUNCTION

Timing functions in animation are the same as transitions whereby we can use either keywords or set a custom timing function by using the 'cubic-bezier' value. A full list of timing keywords can be found at [developer.mozilla.org/en-US/docs/Web/CSS/animation-timing-function](https://developer.mozilla.org/en-US/docs/Web/CSS/animation-timing-function)

### ANIMATION-DIRECTION

When applying our animations, we have the ability to play them back in a number of ways. The default value is 'normal', which will play the animation forwards. We can also play the animation in reverse or alternate the animations playing forwards and backwards.

### ANIMATION-FILL-MODE

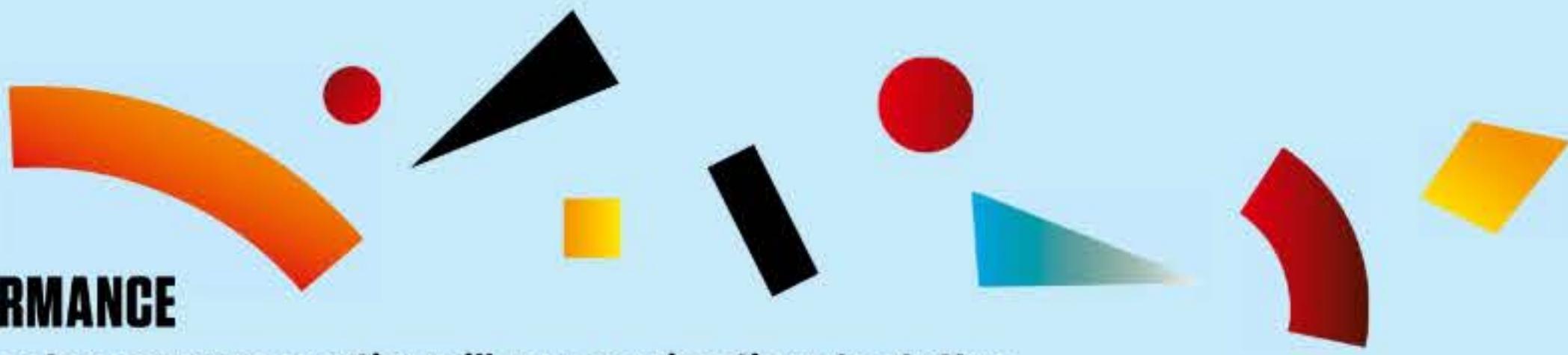
The fill mode value enables us to choose what should happen at the end of an animation to the value that we have changed. For example, setting the value to 'forwards' will keep the property values from the end of the animation, whereas the default value 'none' will return the elements to their original state after the animation has finished.

### ANIMATION SHORTHAND

All of the animation properties can be combined into a shorthand statement using the 'animation' property. We are free to omit the values we do not need and want to leave as the default values.

```
animation: duration || timing-function ||
           delay ||
           iteration-count || direction || fill-mode ||
           play-state || name;

animation: 500ms linear infinite flash;
```



## ANIMATION PERFORMANCE

Animating too much or too many properties will cause animations to stutter



## SVG Animated Birds

Page speed and performance is an important aspect of any application or website. If you are using animation as part of your project, it can be a good place to start when trying to optimise performance.

Firstly you can check that you're not using too many animations all at once on the page: as well as being bad for performance, it is also bad for your users' experience. Multiple animations on different parts of the page will be fighting for their attention as well as potentially causing performance issues. Being aware of the number of animations will help to address both of these potential issues.

### AVOIDING JANKY ANIMATION

In order to achieve a smooth 60fps animation, the browser has only 16.7ms (1000ms/60) to achieve all of the work that needs to be done per frame. If the browser can't complete all of the operations needed, it will stop and move on to the next frame, starting the calculation and redrawing process all over again. This is when we start to see dropped frames, causing janky or stuttering animations.

While the list of animatable properties is extensive, at the moment we can only safely animate a handful of these properties to keep within the necessary 16.7ms. These

properties are 'transform', 'opacity' and also 'filter'.

The reason for this is that animating any other property will cause the browser to have to repaint the page, and this is an expensive process in terms of performance and will likely take longer than 16.7ms to calculate and draw each change.

### GIVING THE BROWSER A HELPING HAND

We can also utilise the 'will-change' property in order to give the browser a heads up that this property is going to change. This enables the browser to perform some optimisations before your animation even starts.

### THE FUTURE OF WEB ANIMATION PERFORMANCE

The new Firefox Quantum project is taking amazing strides towards making every animatable property perform well, as well as many other improvements to rendering content on the web.

It's also worth noting that the newest versions of the iPad will play animations back at 120fps, which will reduce the time in which the browser has to calculate and paint each frame to 8.35ms.

## TIMING AND CHOREOGRAPHY

Utilising the correct timing and delays will produce much better-looking and easier-to-comprehend animations. When animating any elements, it's important to choose a duration that is appropriate to its context. For example, an animation that's applied to a button interaction should be short, usually less than a second.

Utilising long animations for common interactions is tedious and annoying to the user and can make your application or website feel sluggish and slow. However, providing an animation duration that is too short can cause confusion and provoke the wrong emotions and feelings for your audience. Once you have chosen a comfortable animation duration, you should use this in every aspect of your website or application.

When animating, multiple elements or property delays should be used to enhance comprehension. It's much easier to see what's happening in an animation when one thing happens at a time instead of all at once.

## ANIMATING SVG

SVG enables us to create intricate drawings and images, where all of the elements inside SVG are also animatable using CSS



SVG has many benefits – its vector nature is great, and we don't get any of the problems we get with bitmap images when displaying an image too big or too small or on devices with differing DPIs. SVG is also much smaller in size. SVGs are basically instructions for the browser to draw the image contained within.

### HOW DO I ANIMATE SVG?

Animation in SVG works exactly the same way as it does with any element on the page. We need some way to target the element we want to animate, and then apply the animation.

```
svg rect {  
    animation: 2s linear infinite rotate;  
}
```

### GOTCHAS

The main way in which animating SVG elements differs from regular elements is the 'transform-origin' property. Normally we would use percentages or keyword values

in order to set the point at which a transform operation takes place.

```
transform-origin: 50% 50%;  
transform-origin: center center;}
```

So if we can't use keywords and can't use percentages, how do we set our transform origin? The answer is to use pixel values. Further complications come into play because, unlike regular elements that would measure the pixels from the top left corner of itself, SVGs will measure from the top left corner of the parent SVG canvas. Jack Doyle, author of the GreenSock Animation Platform, has written a blog covering this topic in detail on the CSS Tricks blog: [css-tricks.com/svg-animation-on-css-transforms](http://css-tricks.com/svg-animation-on-css-transforms)

### BROWSER SUPPORT

CSS animation when used with SVG requires the SVG to be inline in the page for the majority of browsers. This means we can't use the image tag to include our SVG and perform animations; we need to have our SVG inside an SVG tag on the page.

## USING CUSTOM PROPERTIES

CUSTOM PROPERTIES, OR CSS VARIABLES, CAN BE USED TO CREATE CONFIGURABLE PARTS OF YOUR ANIMATION

### PREFERS REDUCED MOTION

Animations and movement can cause motion sickness in some users. We can use custom properties in order to effectively remove animations for users who have indicated preferring reduced motion.

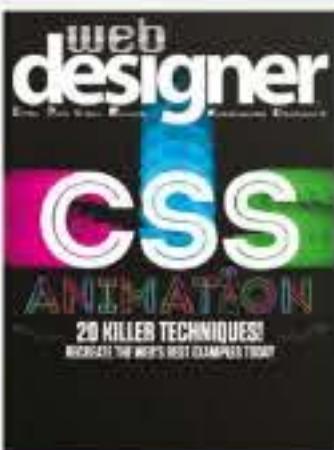
```
:root {  
    --timing: 500ms;  
}  
@media (prefers-reduced-motion: reduce) {  
    --timing: 0;  
}  
.animate {  
    animation: my-animation var(--  
timing) linear;  
}
```

By changing the timing to '0', we stop the animation from running when the user has requested it. While this media query isn't yet widely supported, it is by iOS.

### CHANGING OTHER PROPERTIES

We can use custom properties to define other parts of our animation, such as the colour or size and use. This is useful if we have a part that's configurable and we're using that property as part of an animation.

### WEB DESIGNER MAG 268 – CSS ANIMATED COVER



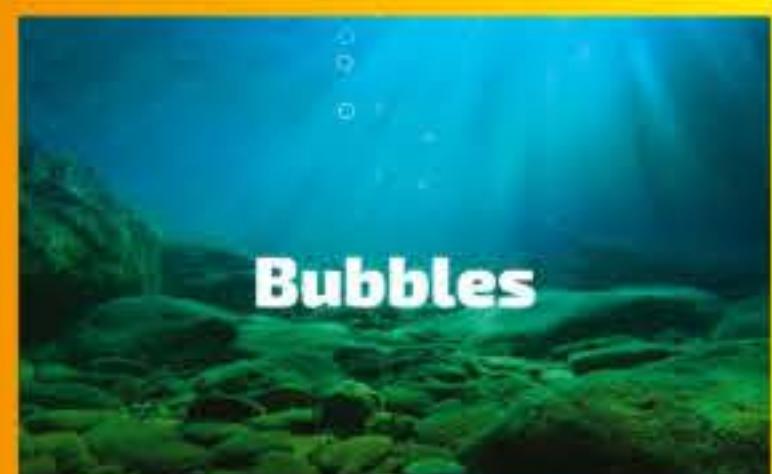
[codepen.io/matchboxhero/pen/qVWNZo](http://codepen.io/matchboxhero/pen/qVWNZo)

### SVG SNOW



[codepen.io/matchboxhero/pen/JrLJeb](http://codepen.io/matchboxhero/pen/JrLJeb)

### SVG BUBBLES



[codepen.io/matchboxhero/pen/LzdqOv](http://codepen.io/matchboxhero/pen/LzdqOv)

## BRING IT ALL TOGETHER

In this tutorial we'll look at how we can combine all of the elements we've covered in the article into one single animation



*The initial state of the animation, with the arm placed off the record and the record not moving*



*The first thing that happens is the arm moves into position over the record, and halfway through this animation the record begins to spin*



*Once our arm is in position and the record has rotated slowly once, we then spin the record faster, infinitely*



*When the user stops interacting the record spins backwards slowly and the arm moves back to its original position*

We'll be creating a record player animation where the arm of the record player moves into position over the record, as well as rotating the record itself at two different speeds. We'll create the animation using SVG transitions and keyframes, and we'll be using custom properties in order to make our animation configurable.

### CREATE AND EXPORT OUR SVG

The first step is to create and export our SVG code. You can do this using many different graphics editors (the example is Sketch for Mac). The shapes being used are simple - mostly straight and poly lines combined with circles. We want to create the SVG in the initial state of our animation.



### OPTIMISE OUR SVG

When we have the exported code from our editor we need to optimise that code to make it easier for us when creating our animation. Firstly we'll run it through [SVGOMG \(jakearchibald.github.io/svgomg\)](https://jakearchibald.github.io/svgomg) making sure to only optimise the parts we need. Once we've done this we'll paste our code into CodePen and create some basic page styles. We'll be using Sass to take advantage of the nesting capability.

<https://bit.ly/2PZVbd6>

### EDIT BY HAND

Next we will need to edit our SVG by hand. We are going to need to remove any inline transforms on elements that we are going to animate into our CSS. Doing this will make it easier to animate, because we'll be able to see all of our transform properties in one place.

```
#arm-top {
    transform: translate(9.5px, 3px)
    rotate(-90deg);
```

```
    transform-origin: -1px 11px;
}
#record-container {
    transform: translate(7px, 0)
    rotate(0deg);
    transform-origin: 24px 24px;
}
#record {
    transform: rotate(0deg);
    transform-origin: 24px 24px;
```

<https://bit.ly/2r7alya>

### ANIMATE THE ARM

We can achieve most of what we are trying to do using transitions. We will start with the record arm and animating the arm into position over the record. To do this we will need to rotate the arm of the record from '-90deg' to '0deg'. In order to transition rotation we need to use the transform property. This means we also need to keep any other transform properties the same when changing the rotation.

```
#record-svg:hover, #record-svg:active, #record-svg:focus {
    #arm-top {
        transform: translate(9.5px, 3px) rotate(0deg);
        transition: transform 1s;
    }
}
```

<https://bit.ly/2Qkew7Y>

## SLOWLY ROTATE THE RECORD

Next we can use another transition to complete the first, slower rotation of our record. To do this we will need to target the container and apply our transition code, much like we did in the previous step, except this time we will be adding a delay of a quarter of the time it takes for the arm to move into position.

```
#record-svg:hover, #record-svg:active, #record-svg:focus {
    #record-container {
        transform: translate(7px, 0) rotate(360deg);
        transition: transform 1s 500ms linear;
    }
}
```

<https://bit.ly/2Rcpw4r>

## SPEED UP THE ROTATION

In order to speed up the rotation, and for it to repeat infinitely, we'll need to use a keyframe animation. The animation we need to create is simple, we just need to rotate the record 360 degrees.

```
@keyframes rotate-record {
    to {
        transform: rotate(360deg);
    }
}
```

Then when we're applying our animation, we need to make sure to set the delay correctly so our animation starts at the end of the container stopping.

```
#record-svg:hover, #record-svg:active, #record-svg:focus {
    #record {
        animation: 400ms linear 1250ms infinite rotate-record;
    }
}
```

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

## IN REVERSE

Everything is working now and our animation is complete – until the interaction stops. At that point, both the arm and record stop animating and jump back to their original positions. This is where the setup of our SVG is important. The record element itself sits inside of a container. By animating the container using a transition we can also perform another transition in reverse with just a couple of lines of code.

```
#arm-top {
    ...
    transition: transform 500ms;
}
#record-container {
    ...
    transition: transform 500ms linear;
}
```

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

## CUSTOM PROPERTIES

Now we've got our complete animation, we'll make it configurable with custom properties. We can do this by setting our custom properties on the root element in our CSS.

```
:root {
    --background-color: #2A2A2A;
    --record-colour: white;
    --record-size: 12rem;
}
```

We can then apply them to the property values where needed, making sure to provide a fallback for each one.

```
.container {
    max-width: var(--record-size, 15rem);
    max-height: calc(var(--record-size, 15rem) * 1.166);
    ...
}
```

We can also use custom properties as part of 'calc()' functions, which is particularly useful for sizing and for creating durations and delays.



## CONFIGURABLE DURATION

We can utilise the 'calc()' function in order to make the animation duration configurable. Firstly we need to set a new custom property for our duration with a value, in seconds. We can then use this value in order to set all of the other animation time values.

```
transition: transform 500ms;
```

Given the custom property '`—animation-duration`' being two seconds becomes:

```
transition: transform calc(var(--animation-
```

duration, 2s) / 4);

By doing this for every time value in our

animation, we can control the speed of the

entire animation by simply changing the

custom property at the top of our CSS.

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

## OPTIMISATIONS

Now we can add our accessibility options for '`prefers-reduced-motion`' and add the '`will-change`' property to all of the elements.

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

## TWITTER ACCOUNTS 5 TO FOLLOW

### SARAH DRASNER

[@sarah\\_edo](#)

Sarah is the author of the O'Reilly book *SVG Animations*.

### RACHEL NABORS

[@rachelnabors](#)

Interactive storyteller, cartoonist and author of *Animation at Work*.

### VAL HEAD

[@vlh](#)

Author of *Designing Interface Animation* and curator for 'UI Animation Newsletter'.

### LISI LINHAR

[@lisi\\_linhart](#)

She is passionate about web animation and talks about 'Practical Web Animation'.

### CODEPEN

[@CodePen](#)

Tweets great examples of animation that have the code for you to learn from.

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 to Friday 8am-7pm and Saturday 10am-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 to 63%

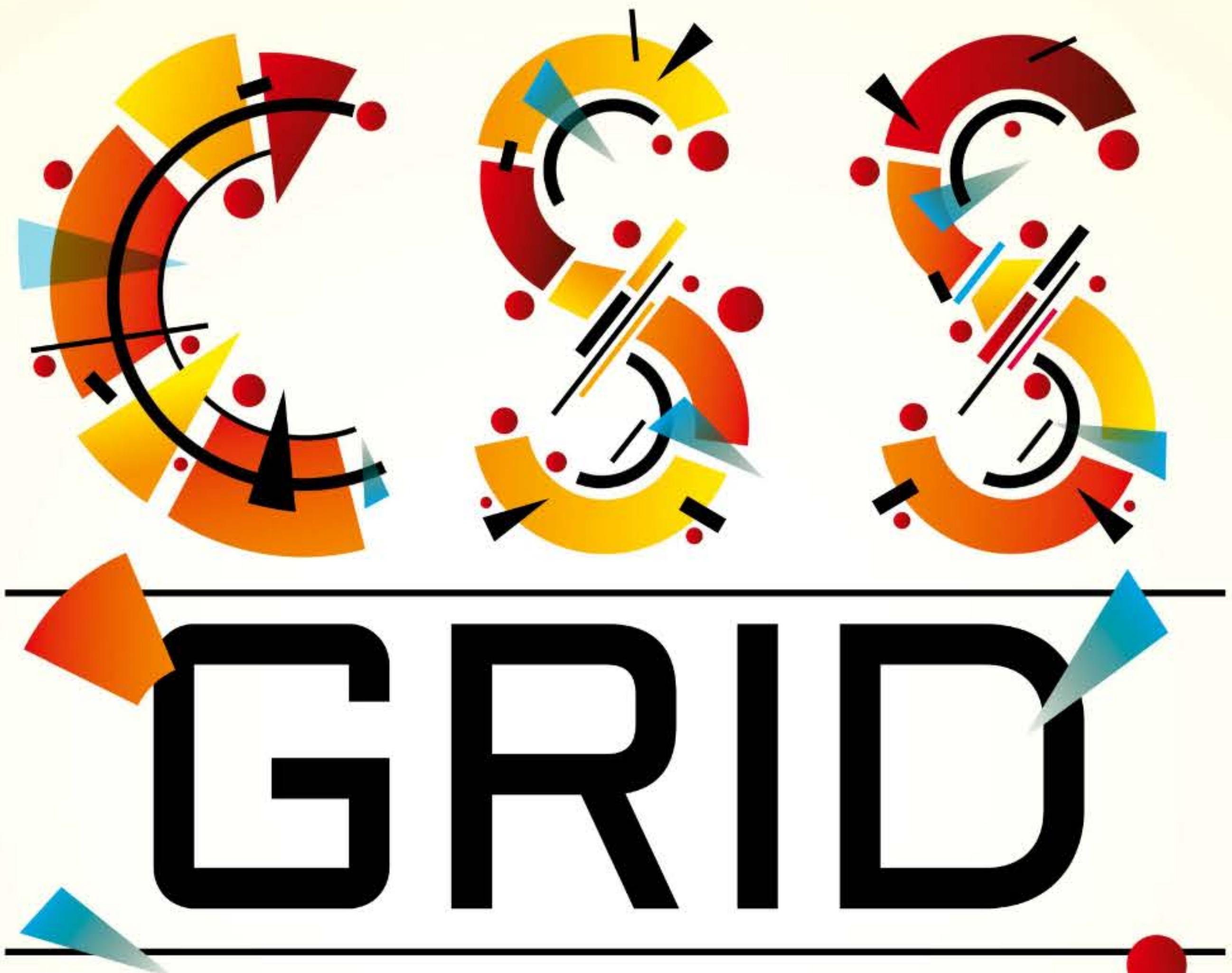


**SEE THE  
FULL RANGE AND  
ORDER ONLINE AT**

**[www.myfavouritemagazines.co.uk/pin/xmas182](http://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](http://bit.ly/magtandc). Offer ends 31st December 2018.



# GRID

CSS GRID ENABLES YOU TO DEFINE ADVANCED CONTENT LAYOUT FOR  
RESPONSIVE DESIGN WITHOUT THE COMPLICATED FORMATTING



“ CSS grid offers more flexible definitions for content layout. Unlike older alternatives, such as table layouts and floating elements, grid has been designed to benefit SEO, responsive design and accessibility ”

Leon Brown  
[twitter: @LeonBrown](#)

## WHY YOU NEED CSS GRID IN YOUR LIFE

CSS Grid lets you create advanced layouts while focusing on the user

To some, a wireframe may seem like a laborious and unnecessary task that can drag out a project with no real long-term benefit. Jumping straight into a design may seem like a compelling and rewarding alternative, but by doing so you could be forgetting the most important thing of all when creating a design: the user.

The ideation and wireframing stage of any design enables you to consider the layout and user experience from the very start of your project. By only using the core components for your web design – header, footer, navigation, buttons – to begin with, you can always ensure your focus is on the user, without getting distracted by which colour palette you had in mind.

There are plenty of tools out there to get started with wireframing, but today many web teams use Sketch. Its usability and unrivalled speed make it the obvious choice.

So a project has been set up and a font has been chosen for the

wireframe, but what should be added? This is where the core components are decided and added to the wireframe – the components that the design can't live without.

Starting with the header, what do we need if we are approaching this project mobile first? Is the logo the main focus? Should we consider using a 'burger' navigation to ensure best practice? These are the questions any designer should keep asking themselves throughout the process when building their wireframe components.

The first steps should be to create an artboard for mobile or desktop, and begin to build the components on the page with simple shapes in order to define the outline of the components like the header. To highlight areas of importance and interactivity in the wireframe, use varying shades of a monochrome colour palette while wireframing. For example, within the header, inserting a logo and navigation in a bold blue colour on

a pastel blue background will present them as the important elements in your component.

It's also important to consider a grid system, padding and margins as soon as possible when defining the wireframes. Showing rulers in a Sketch file and clicking on these rulers (down the left and above your artboard) will create guides, which in turn saves a lot of stress when sharing designs, as consistency is key when defining a design system.

When the essentials are in place for one component, move onto the next. Only include what you need. Keep in mind the bare minimum that is needed for the user to complete their task. Planning to use these components in more than one artboard? Why not create a symbol. Symbols act as super-components that update all the artboards they currently sit in. To create a component, right-click an element and choose 'Create symbol'. This will be saved into one packaged item manageable from the symbols page.

## WHAT ABOUT FLEXBOX?

Flexbox is another option that can be used to control the flow of content. The main difference is that it was created specifically for one-dimensional content flow, whereas grid is used for two-dimensional purposes.

While flexbox gives you the option of controlling the horizontal or vertical layout, CSS Grid controls both horizontal and vertical layout. In effect, you can consider CSS Grid to be the next step up from flexbox.

Although the grid feature can be used to define any type of layout, flexbox provides the advantage of keeping your CSS simpler for one-dimensional layout. It's worthwhile keeping in mind that requirements may change or expand to include a requirement for two-dimensional layout.

An example of this would be using a vertical one-dimensional layout for small-screen devices, and using a two-dimensional layout for view on big screens, such as a desktop. Flexbox does not offer the ability to adapt for this requirement, whereas grid does.

Both flexbox and grid are compatible with the content ordering feature, so changing from flexbox to grid should not be a significant problem if it's required.

Make sure that you keep styling in a stylesheet separate from the main HTML content if you want to maximise maintainability.

## BROWSER SUPPORT

MODERN FEATURES FOR MODERN BROWSERS

Grid and flexbox were not always part of the CSS specification, meaning that there is no support in older browsers. All of the major browsers have supported the full grid specification since 2017. The main concern for compatibility are Microsoft browsers (IE and Edge), which have high usage in businesses. They introduced limited support in Internet Explorer 10, but full support has only been available since Edge 16.

	1	2	3	4	5
1	Item				
2			Subitem		
3					

# ESSENTIAL PROPERTIES

THE PROPERTIES YOU NEED TO KNOW FOR CREATING GRID-BASED LAYOUTS

## GRID-AUTO-FLOW

Enables you to define the flow of grid cell layout using horizontal 'column' or vertical 'row', making the grid act similar to a table.

## GRID-GAP

Apply this to the grid container to set spacing between each grid cell. The default sets no spacing between cells.

## GRID-COLUMN

Sets the specific column to place the grid cell. Use this to target specific cells via class, ID or selector.

## GRID-ROW

Sets row location to place the grid cell. Use this to reference specific cells via class, ID or selector.

## GRID-AREA

Defines the grid area for a cell to cover using co-ordinates in the format startX/startY/endX/endY or a cell reference name.

## GRID-TEMPLATE-AREAS

A more advanced method for defining the grid layout using grid-area reference names given to the created content cells.

## GRID-TEMPLATE-ROWS

Used to specify measurements for individual rows within the design of the grid layout.

## GRID-TEMPLATE-COLUMNS

Similar to grid-template-rows, but used to specify measurements for individual columns within the design of the grid layout.

## CREATE A QUICK BASIC GRID

Examples of using CSS Grid to create a series of simple layouts

### 1. HTML CONTENT

To start, link the external CSS file, e.g. `<link rel="stylesheet" type="text/css" href="1.css" />`. Everything CSS Grid goes in the `<body>`. Three sections are defined - one for each example grid. The `data-grid` attribute is used to provide each section with a unique reference value that can be used by the CSS. Each child element of the grid will become a cell of the grid.

```
<section data-grid="1">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</section>

<section data-grid="2">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</section>

<section data-grid="3">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</section>
```

### 2. CSS GRID DEFINITION

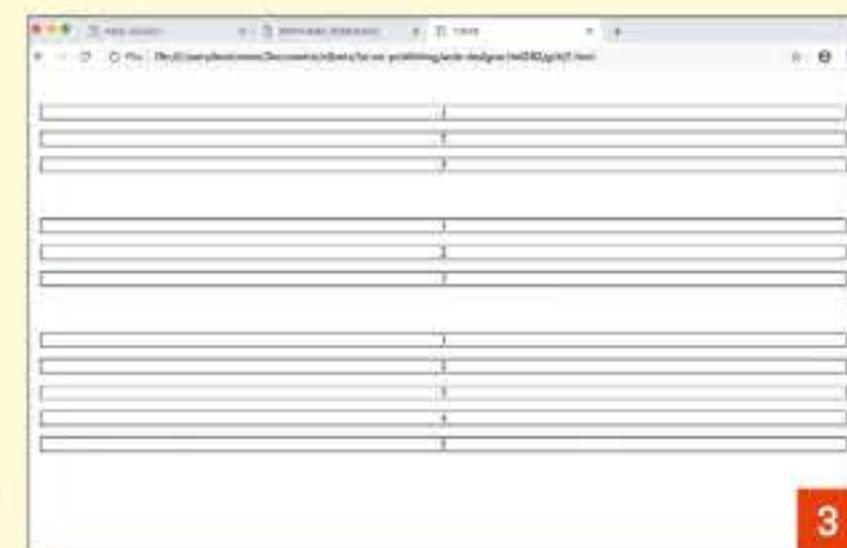
Create a file called '1.css'. This step sets formatting to be shared by all elements using the 'data-grid' attribute. Display is set as grid to make their first-level children act as the cells of the grid. Default background colour and text alignment for these grid cells are also set.

```
[data-grid]{
  display: grid;
  margin-top: 2em;
  padding: 1em;
  grid-gap: 1em;
}

[data-grid] > * {
  background-color: #fff;
  text-align: center;
  border: 1px solid #000;
}
```

### 3. HORIZONTAL COLUMNS

The columns template for grid 1 is set to contain three cells. Measurements for the cells use the `fr` unit to describe their width as a fraction of the



available space, so the middle column will be three times wider than the outer columns.

```
[data-grid="1"] {
  background: red;
  grid-template-columns: 1fr 3fr 1fr;
}
```

### 4. VERTICAL ROWS

The second grid is set to display with a vertical cell flow. As this is not the default flow for grid cells, the `grid-auto-flow` attribute is set to 'row'. The same approach is used to define the height of each row, using the `fr` unit measurement.

```
[data-grid="2"] {
  background: green;
  grid-auto-flow: row;
  grid-template-rows: 1fr 3fr 1fr;
}
```

### 5. MIXED GRID

The final grid is presented with a layout allowing for both vertical and horizontal content. Auto flow is set to 'column', setting the cells to flow horizontally. The first cell is set to span over two grid rows, guaranteeing the cells to flow horizontally over two rows.

```
[data-grid="3"] {
  background: blue;
  grid-auto-flow: column;
}

[data-grid="3"] > *:first-child{
  grid-row: span 2;
}
```





# GO TO THE NEXT LEVEL

An alternative approach to defining a grid layout by using reference names for content elements

## 1. DOCUMENT INITIATION

First define the webpage as a HTML document and link the external CSS stylesheet, e.g. `<link rel="stylesheet" type="text/css" href="2.css" />`. Everything CSS Grid goes in the `<body>`.

## 2. GRID HTML

The grid must be defined in a way that will allow its container and child cells to be referenced by CSS. This example uses semantic tags that have a unique name for each content area, although you could decide to use class names, data attributes or IDs as an alternative.

```
<article>
  <section>Content</section>
  <header>Header</header>
  <aside>Side</aside>
  <footer>Footer</footer>
  <nav>Menu</nav>
</article>
```

## 3. CSS: REFERENCE NAMES

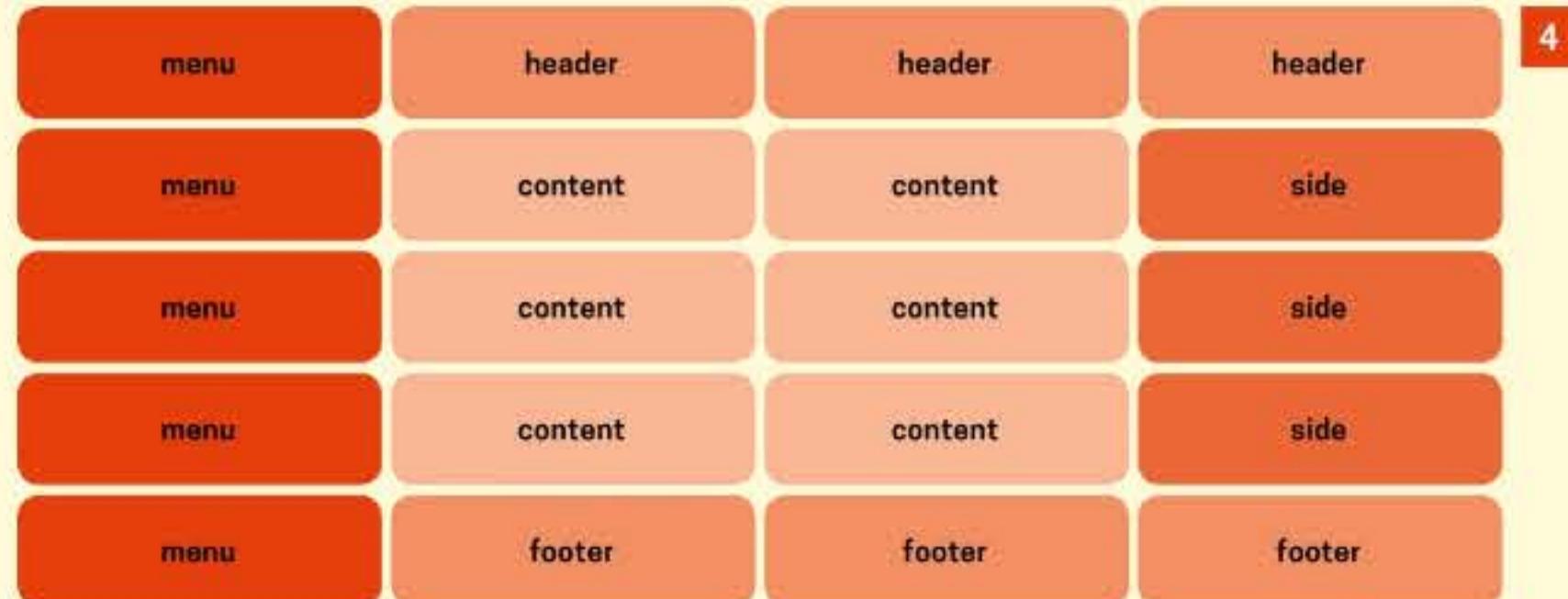
Create a new file called '2.css'. This step references each of the content elements inside the article container and uses the `grid-area` attribute to provide a unique reference name. This reference name will be used in the next step when defining the layout template for the grid.

```
article > aside { grid-area: side; }
article > nav { grid-area: menu; }
article > section { grid-area: content; }
article > header { grid-area: header; }
article > footer { grid-area: footer; }
```

## 4. GRID SETUP

The article in the HTML is used as the grid container. The `grid-template-areas` attribute defines the layout of the grid using the `grid-area` reference names identified in step 3. The setup works like cells within a spreadsheet, where names that are vertically or horizontally adjacent create one grid cell area.

```
article {
  display: grid;
  grid-gap: .5em;
  grid-template-areas:
    "menu header header header"
    "menu content content side"
    "menu content content side"
    "menu content content side"
```



```
"menu footer footer footer";
}
```

## 5. CONTAINER FORMATTING

With the grid layout now in place, the next step is to define additional presentation for the grid. This step sets the grid container's colours, along with the maximum width and minimum height. Some padding is also added to enable the grid to have a surrounding border space.

```
article {
  max-width: 25em;
  min-height: 20em;
  background-color: #333;
  color: #444;
  padding: 1em;
  font-size: 150%;
```

## 6. CELL FORMATTING

The final step is to set the formatting of the grid cells. This tutorial sets colours for the text and background, but you can add other CSS presentation attributes to meet your requirements. No new rules or limitations apply to grid cells, meaning you are still able to use your existing knowledge of CSS.

```
article > *{
  background: #fff;
  color: #000;
}
```



## MEASUREMENT PROPERTIES AND ATTRIBUTES



The grid feature offers many ways to define measurements for its cells, enabling you to define both fixed and responsive layouts

### FR

The `fr` measurement unit specifies the width or height of a grid cell as a fraction of the remaining space.

### SPAN

Sets the width or height of a cell to cover more than one cell space within the defined grid template.

### REPEAT

Used as shorthand to repeat the same width or height definition across multiple cells within the grid template.

### MINMAX

Allows you to define the minimum and maximum height or width of a grid cell; useful for controlling responsive layouts.

### PX / CM / MM / EM

Fixed unit measurements can be used in combination with `fr`, which results in fractional spaces being measured from the remaining space.

# WHAT ELSE DO I NEED TO KNOW?

Six tips to get more from CSS Grid

## 1. LAYERING

The z-index attribute can be applied to individual grid cells to allow them to be layered above or below other grid cells.

## 2. ALTERNATIVE LAYOUTS

Make use of CSS media queries to modify or redefine grid definitions for different screen sizes or media types.

## 3. TEMPLATE ATTRIBUTES

Properties whose name begins with 'grid-template' are used to define the design of the grid layout.

## 4. CELL ATTRIBUTES

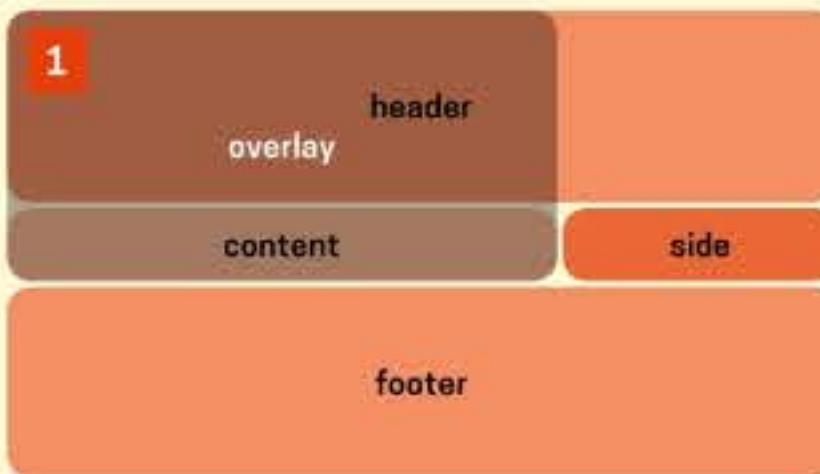
Properties that don't start with 'grid-template' are used to place cells within the grid.

## 5. CONTENT ORDERING

Grid allows you to order content independently from the HTML definition, enabling you to optimise HTML for SEO while also providing control for responsive design.

## 6. MAINTAINABILITY

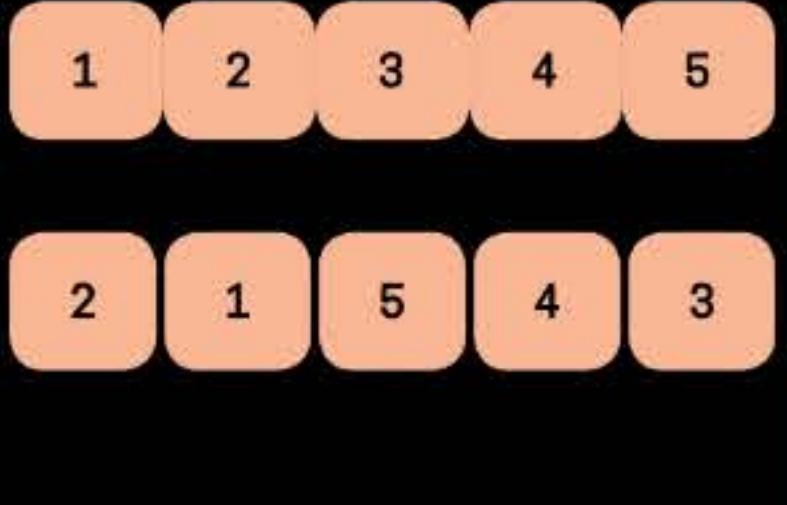
Use the grid flow attributes to allow future adaptions of the HTML to automatically adapt the grid without needing to update the CSS.



## WHAT ABOUT ACCESSIBILITY?

The same ability of grids to re-order content for responsive purposes can also be applied to increase accessibility. Whether it's changing or extending the stylesheet, or with the help of JavaScript, users can be given options to choose the order that best suits what they need.

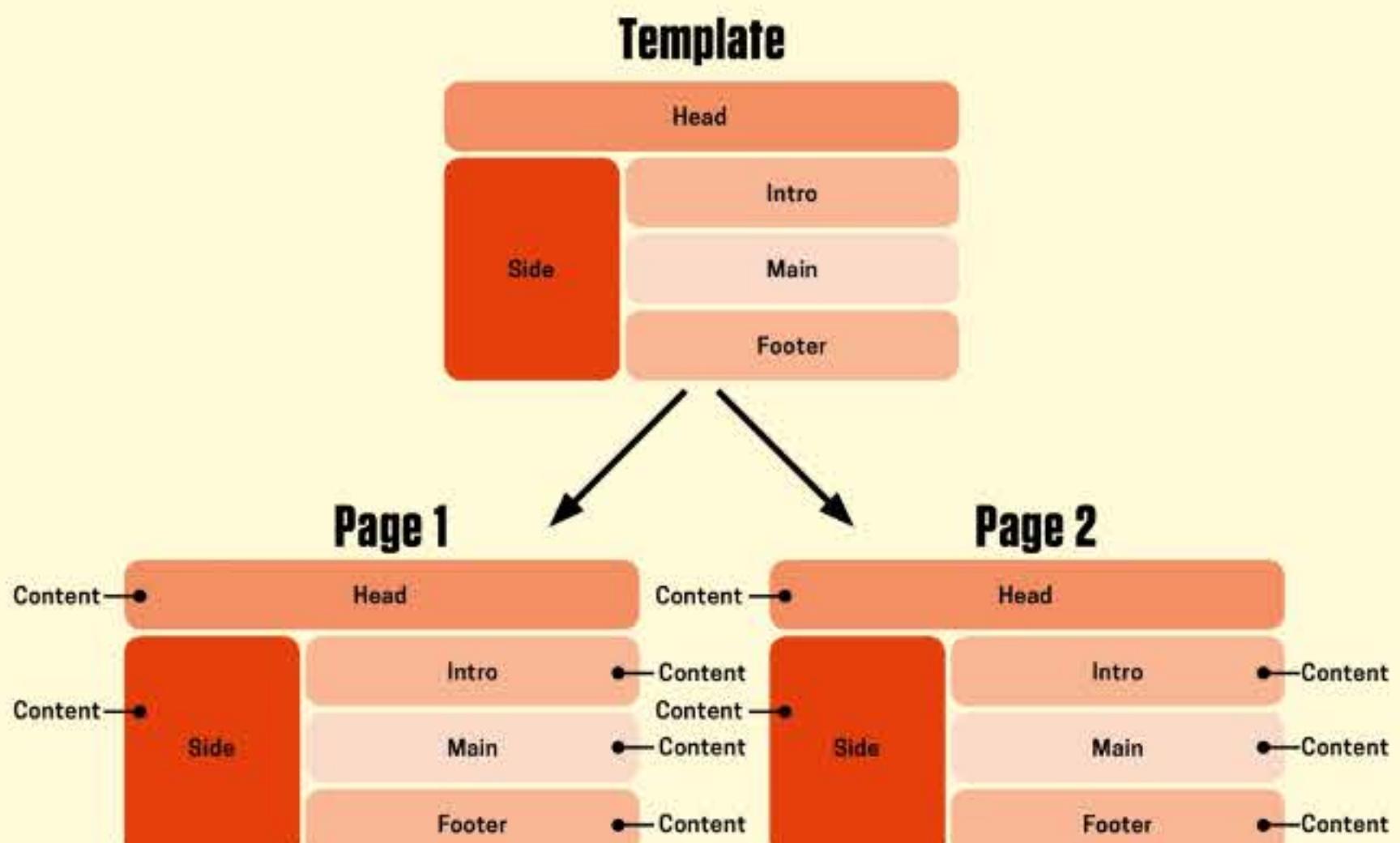
In its simplest form, this can be achieved by attaching additional stylesheets to load, possibly as a result of settings stored in a cookie, local storage or on the server. More advanced accessibility control can be executed as a result of interactions with a HTML form that triggers JavaScript to adapt the grid ordering.



## TEMPLATE VS CONTENT The two parts of the grid

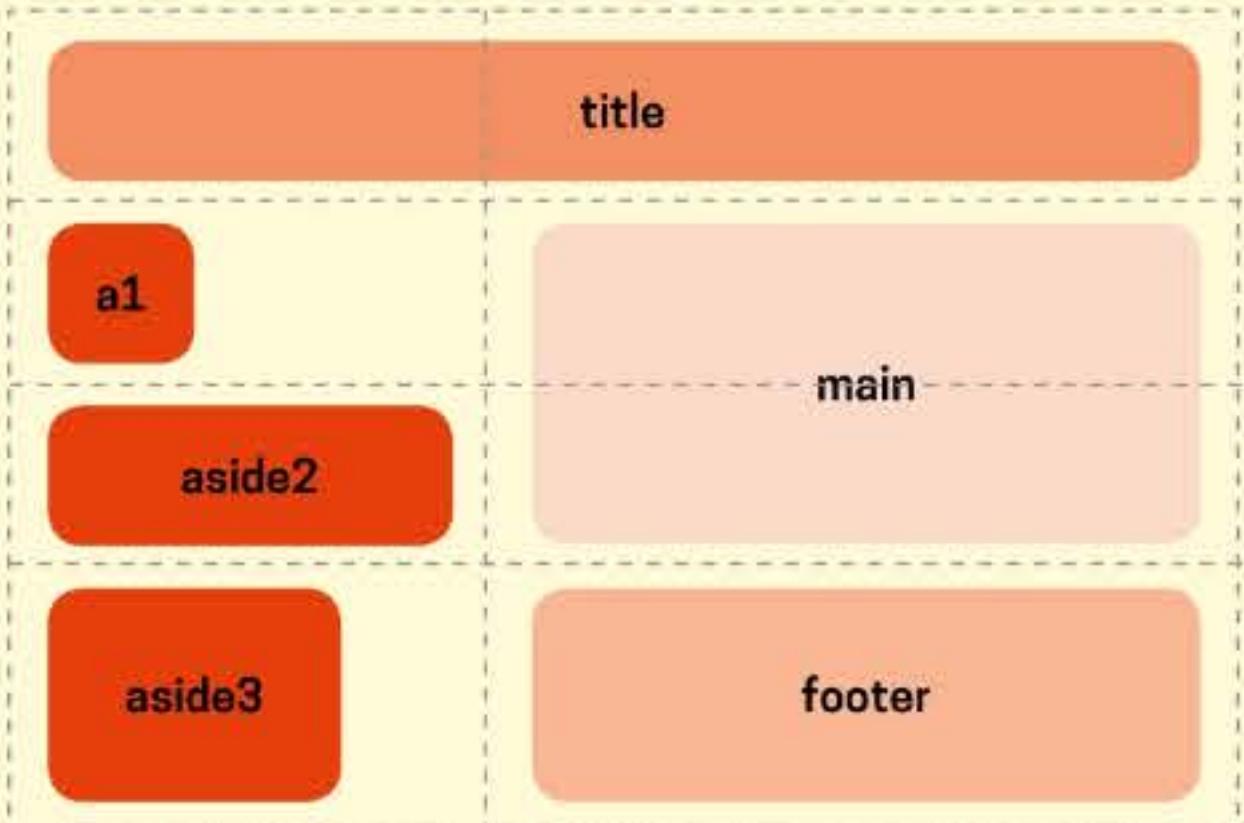
Grids consist of two parts – templates and content. This enables reusable grid layouts that can be called from HTML markup using tag names or attributes such as class or ID values. As well as being a benefit for responsive layouts, this reusability has many advantages for agile workflow strategies involving change

management. Once a template has been defined, you are able to apply it to a container element that stores content elements. Content flow within the grid appears in accordance to the settings of the grid template, which allows for custom rules to be applied directly to individual content elements.

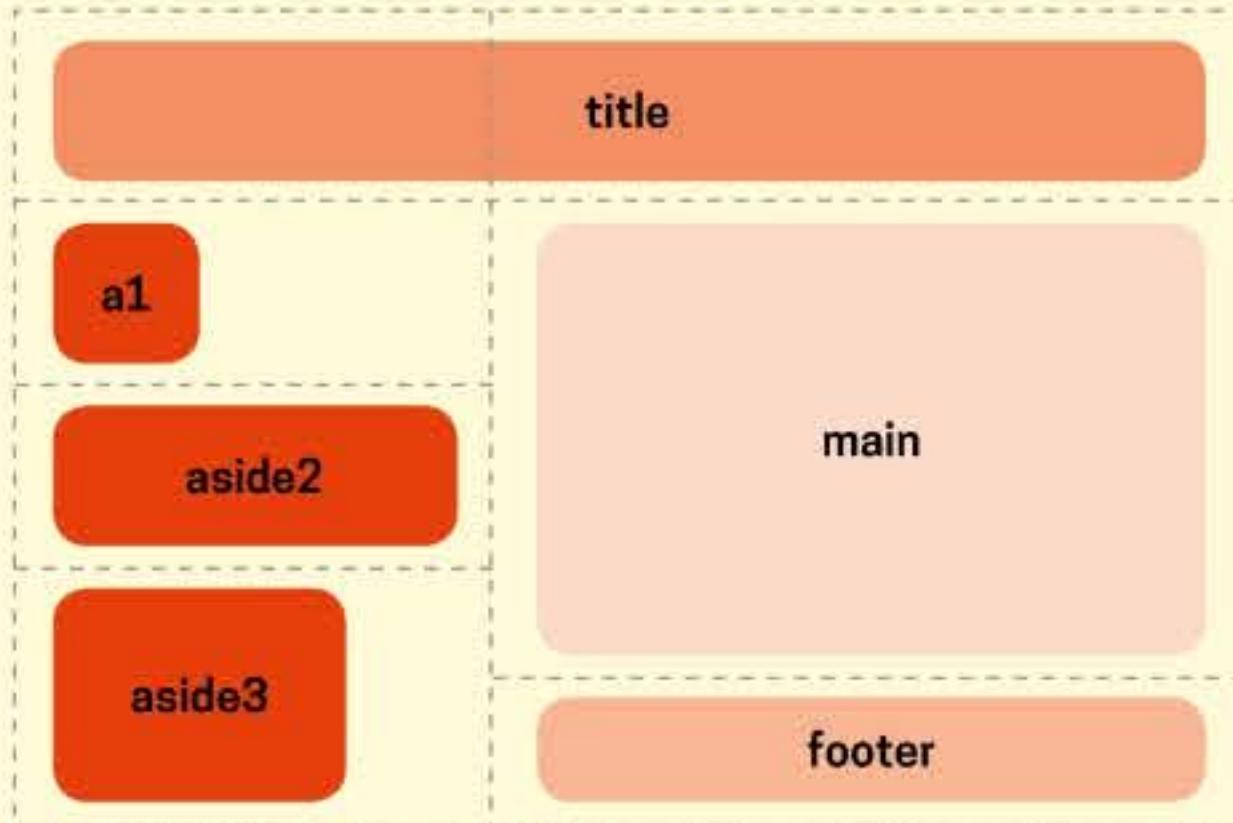




## subgrid



## nested grid



## GRID FOR AGILITY

Use CSS grid as a tool for agile project development

The ability of grid to keep layout definition entirely independent from the HTML content is not only good for responsive design and SEO but also for project management. A lot of talk exists about agile development being a project management methodology, but it's as much about how you implement your design and its code.

By using CSS grid effectively, you are able to protect your project

against unexpected changes that could otherwise escalate to derail your schedule and budget planning. An example of this would be a change request to alter the position of content, which would traditionally require you to modify both the CSS and HTML. This can be problematic for large projects where design and content are responsibilities of separate people or teams. Grid resolves this issue

by providing full control to the designers for defining the layout without concern for the HTML.

Future additions to the grid feature will allow advanced layouts to be defined with lower complexity. One example of this is the proposed subgrid feature, which will enable cells to act as a subgrid that automatically inherits the required cell layout from its parent. While subgrid layout is already possible

using nested grid, it is the feature's ability to adapt parent grids without concern for their content that will benefit your ability to be agile by eliminating risk and reducing testing requirements. In addition, the automatic management of subgrids by the browser will help to keep formatting simple and consistent across all developers, providing better ability to adapt your CSS whenever required.

## RESOURCES

### W3C SPECIFICATION

[w3.org/TR/css-grid-2/](https://www.w3.org/TR/css-grid-2/)

The latest details of the CSS grid specification. Includes details about proposed features, such as subgrid that will be implemented by future browser releases.

### GRID

[smashingmagazine.com/2018/07/css-grid-2/](https://smashingmagazine.com/2018/07/css-grid-2/)

Learn about the proposed subgrid feature in more detail, complete with full examples for using the proposed specification.

### POLYFILL

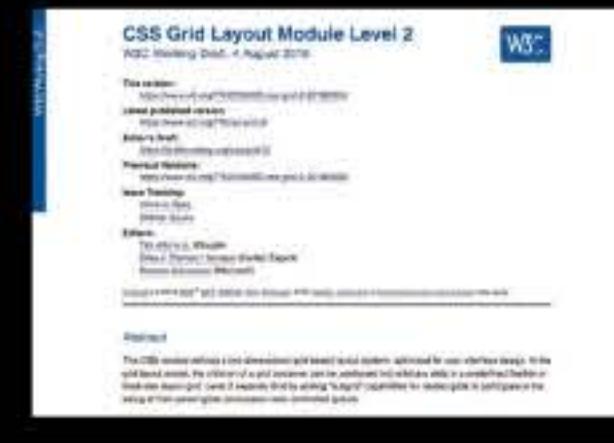
[github.com/FremyCompany/css-grid-polyfill](https://github.com/FremyCompany/css-grid-polyfill)

One of several polyfills available to bridge compatibility with older browsers. Make sure to take a look at the full description of its capabilities and limitations.

### GRID LAYOUT EXAMPLES

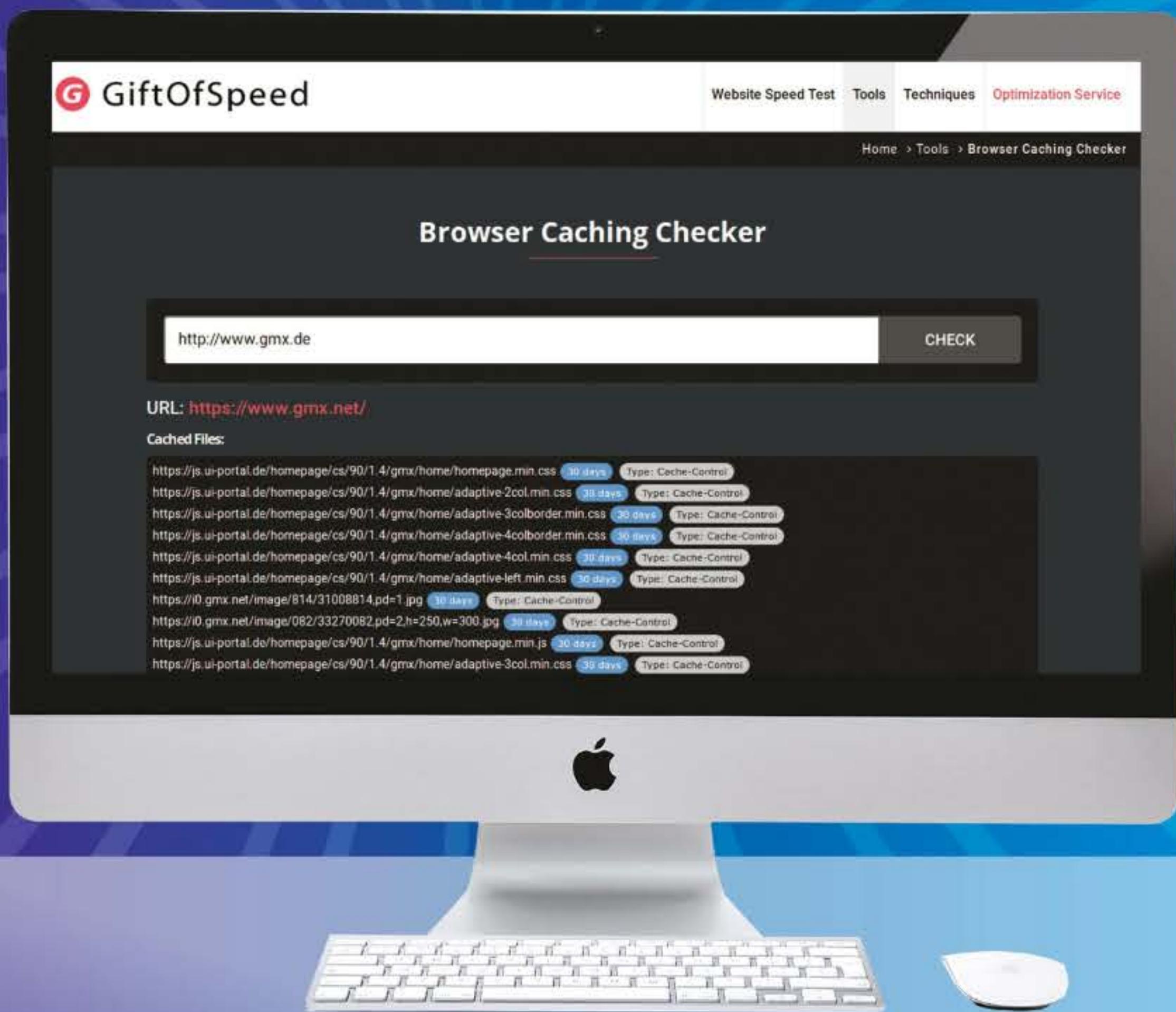
[gridbyexample.com/examples](https://gridbyexample.com/examples)

A selection of pre-made grid layouts that you can use to kickstart your web design or content section layout. Examples include both simple and advanced layout designs.



# Make your websites run faster with optimised CSS

While performance optimisations in JavaScript code are well-known, CSS is an often-overlooked part of the speed game





SS must pass through a relatively complex pipeline, just like HTML and JavaScript. The browser must download the files from the server and then proceed to parsing and applying them to the DOM. Due to extreme levels of optimisation, this process is usually pretty fast - for smaller web projects not based on frameworks, CSS usually makes up just a small part of total resource consumption.

Frameworks upset this equilibrium. Include a JavaScript GUI stack like jQuery UI, and watch CSS, JS and HTML sizes skyrocket. Often, developers usually feel the pinch last - when seated behind a powerful eight-core workstation with T3 internet, no one cares about speed. This changes as latencies or CPU-constrained devices come to play.

Optimising CSS requires a multi-dimensional approach. While hand-written code can be slimmed down using various techniques, going over framework code by hand is inefficient. In these cases, using an automated minimiser yields better results - developer time is not an unlimited resource.

The following steps will take us on a journey through the world of CSS optimisation. Not every single one might be directly applicable to your project, but do keep them in mind.

## 1. Use shorthand

The fastest CSS clause is one that never gets parsed. Using shorthand clauses, such as the margin declaration shown below, radically reduces the size of your CSS files. A lot of additional shorthand forms can be found by googling 'CSS Shorthand':

```
p { margin-top: 1px;
    margin-right: 2px;
    margin-bottom: 3px;
    margin-left: 4px; }

p { margin: 1px 2px 3px 4px; }
```

## 2. Find and remove unused CSS

Eliminating unneeded parts of your mark-up obviously causes immense speed gains. Google's Chrome browser has this functionality out of the box. Simply go to View>Developer>Developer Tools and open the Sources tab in a recent version, and open the Command Menu. After that, select Show Coverage and feast your eyes on the coverage analysis window highlighting unused code on the current webpage.

## 3. Do it in a more comfortable fashion

Navigating through the line-by-line analysis is not necessarily comfortable. Chrome's Web Performance Audit returns similar information - simply open it from the toolbar, View>Developer>Developer Tools>Audits and let it run. When done, a list of problematic elements will pop up.

## 4. Be aware of the problems

Keep in mind that an automated analysis of a CSS can always lead to errors. Perform a thorough test of the

entire website after replacing your CSS files with the minified ones - one never knows what mistakes the optimiser caused.

## 5. Inline critical CSS

Loading external stylesheets costs time due to latency - anyone remember the 'flash of unstyled content'? The most critical bits of code can therefore go into the header tag. Make sure not to overdo it, however. Keep in mind that the code must also be read by humans performing maintenance tasks.

```
<html>
  <head>
    <style>
      .blue{color:blue;}
```

```
</style>
</head>
<body>
  <div class="blue">
    Hello, world!
  </div>
```

## 6. Permit antiparallel parsing

@import adds structure to your CSS code. Sadly, the benefits are not for free: as imports can nest, parsing them in parallel is not possible. A more parallelisable way uses a series of <link> tags, which the browser can fetch at once.

```
@import url("a.css");
@import url("b.css");
```

**456 BERA ST**

Home About Archive Lab Reviews Contact

SUBSCRIBE / FOLLOW

RSS or @rogerjohansson

## Efficient CSS with shorthand properties

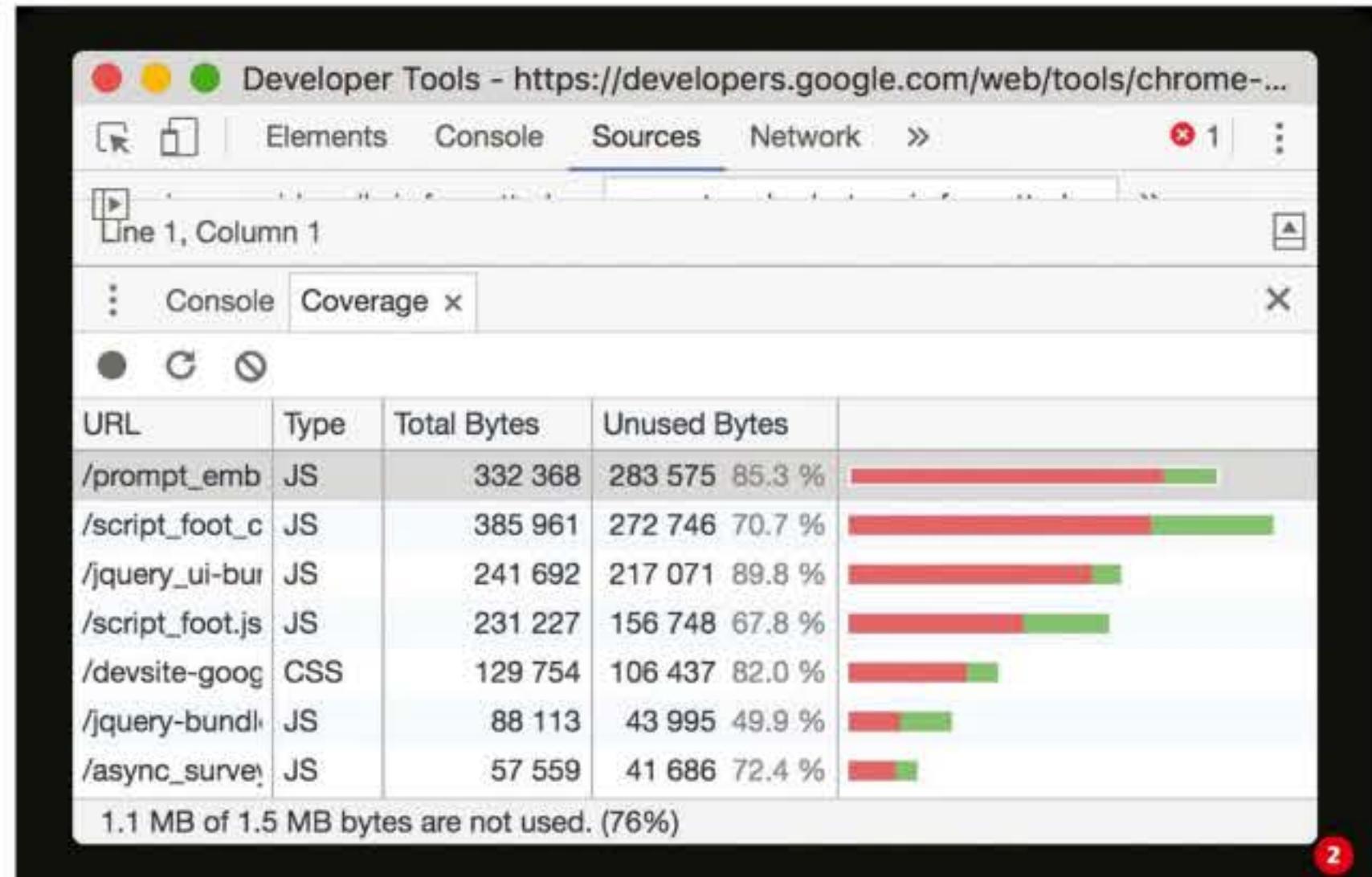
I get a lot of questions about CSS from people who aren't crazy enough to have spent the thousands of hours working with CSS that I have. Sometimes I'm asked to take a look at something they're working on to see if I can figure out why it doesn't work as expected. When I look at their CSS I often find that it's both bloated and unorganised.

One of the reasons for using CSS to layout websites is to reduce the amount of HTML sent to site visitors. To avoid just moving the bloat from HTML to CSS, you should try to keep the size of your CSS files down as well, and I thought I'd explain my favourite CSS efficiency trick: shorthand properties. Most people know about and use some shorthand, but many don't make full use of these space saving properties.

### Some background

Shorthand properties can be used to set several properties at once, in a single declaration, instead of using a separate declaration for each individual property. As you'll see, this can save a lot of space in your CSS file.

Quite a few shorthand properties are available - for details I suggest the W3C CSS specifications of the [background](#), [border](#), [border-color](#), [border-style](#), [border sides](#) (border-top, border-right, border-bottom, border-left), [border-width](#), [font](#), [list-style](#), [margin](#), [outline](#), and [padding](#) properties.



## Tutorials

Make your websites run faster with optimised CSS

```
@import url("c.css");
v<link rel="stylesheet" href="a.css">
<link rel="stylesheet" href="b.css">
<link rel="stylesheet" href="c.css">
```

### 7. Replace images with CSS

Years ago, a set of semi-transparent PNGs to create translucent effects on websites was commonplace. Nowadays, CSS filters provide a resource-saving alternative. As an example, the snippet accompanying this step ensures that the picture in question appears as a grayscale version of itself.

```
img {
  -webkit-filter: grayscale(100%); /* old
  safari */
  filter: grayscale(100%);
}
```

### 8. Use colour shortcuts

Common wisdom teaches that six-digit colour descriptors are the most efficient way to express colours. This is not the case – in some cases, shorthand descriptions or colour names can be shorter.

## Don't interfere with CDNs

While minifying stock CSS looks tempting, keep the costs in mind. A big file served from a fast CDN can be faster than a small file served from a slow server.

```
target { background-color: #ffffff; }
target { background: #fff; }
```

### 9. Eliminate unneeded zeros and units

CSS supports a wide variety of units and number formats. They are a thankful target for optimisation – both trailing and following zeros can be removed, as is demonstrated in the snippet below. Furthermore, keep in mind that a zero always is a zero, and that adding a dimension does not add value to the information contained.

```
padding: 0.2em;
margin: 20.0em;
avalue: 0px;
padding: .2em;
margin: 20em;
avalue: 0;
```

### 10. Eliminate excessive semicolons

This optimisation is somewhat critical, as it affects changes of code. CSS's specification lets you omit the last semicolon in a property group. As the savings achieved by this optimisation method are minimal, we mention this mainly for those who are working on an automated optimiser.

```
p {
  ...
  font-size: 1.33em
}
```

### 11. Use a texture atlas

Loading multiple small sprites is inefficient due to protocol overheads. CSS sprites combine a series of small pictures into one big PNG file, which is then broken apart via CSS rules. Programs such as TexturePacker simplify the creation process greatly.

```
.download {
  width:80px;
  height:31px;
  background-position: -160px -160px
}
.download:hover {
  width:80px;
  height:32px;
  background-position: -80px -160px
}
```

### 12. Chase down the px

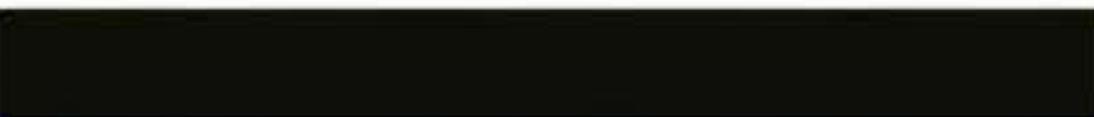
One neat way to increase performance uses a specialty of the CSS standard. Numeric values without a unit are assumed to be pixels – removing the px saves two bytes for each number.

```
h2 {padding:0px; margin:0px;}
h2 {padding:0; margin:0}
```

### 13. Avoid expensive properties

Analysis has shown some tags to be more expensive than others. The list accompanying this step is considered to be especially performance-hungry – avoid them when given the opportunity to do so.

```
border-radius
```

Color Name	HEX	Color	8
Black	#000000		
Navy	#000080		
DarkBlue	#00008B		
MediumBlue	#0000CD		
Blue	#0000FF		
DarkGreen	#006400		
Green	#008000		
Teal	#008080		
DarkCyan	#008B8B		
DeepSkyBlue	#00BFFF		
DarkTurquoise	#00CED1		
MediumSpringGreen	#00FA9A		

Not every hex code can be shortened. Typically code has to be in pairs to work

3 Defer offscreen images  
0.17 s

4 Defer unused CSS  
0.05 s

Remove unused rules from stylesheets to reduce unnecessary bytes consumed by network activity. [Learn more](#)

URL	Original	Potential Savings
...assets/styles.gmx-v2.min.css (s.maxcdn.com)	51 KB	47 KB
.teaser-stage{overflow:hidden;position:relative} ...	5 KB	3 KB

### Diagnostics

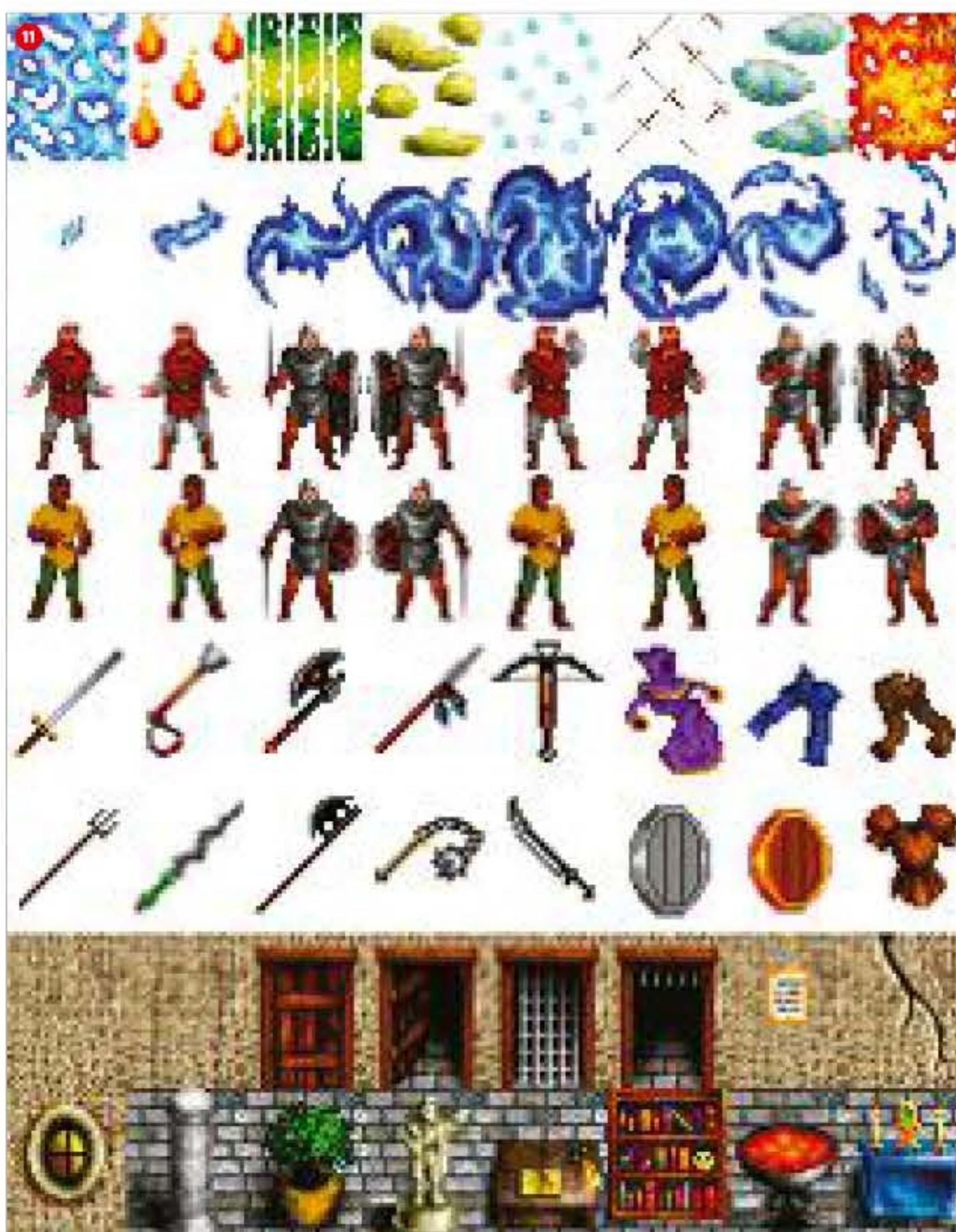
More information about the performance of your application.

- 1 Uses an excessive DOM size 1,853 nodes ▲
- 2 Has significant main thread work 4,920 ms ▲
- 3 ...

## Simplify debugging

Finding errors in minified or optimised CSS is tedious and should be avoided at all costs.

The safest workaround involves using the unoptimised CSS files during debugging, generating an optimised version only as the website is pushed live. Should this be unworkable for some reason, limit yourself to auto-generated optimisations. Many if not most optimisers produce so-called MAP files during execution. These can then be loaded into your debugger - when done, the debugger can correlate the optimised code to the unoptimised files found at another location in the file system.



box-shadow  
transform  
filter  
:nth-child  
position: fixed;  
etc.

## 14. Remove whitespace

Whitespace - think tabs, carriage returns and spaces - makes code easier to read but serves little purpose from a parser's point of view. Eliminate them before shipping. An even better way involves delegating this job to a shell script or similar appliance.

## 15. Eliminate comments

Comments also serve no purpose to the compiler. Create a custom parser to remove them before delivery. Not only does that save bandwidth but it also ensures that attackers and cloners have a harder time understanding the thinking behind the code at hand.

## 16. Use automatic compression

Yahoo's user experience team created an application that handles many compression tasks. It ships as a JAR file, available at [yui.github.io/yuicompressor](https://yui.github.io/yuicompressor) and can be run with a JVM of choice.

`java -jar yuicompressor-x.y.z.jar`

Usage: `java -jar yuicompressor-x.y.z.jar [options] [input file]`

Global Options

`-h, --help`

Displays this

information

`--type <js|css>`

Specifies the

type of the input file

## Don't go crazy

Premature optimisation is a dangerous game. While attempting to squeeze out the last bit of performance from your CSS is intellectually challenging, check if lower-hanging fruit are available before committing your time.

## Tutorials

Make your websites run faster with optimised CSS

**yuicompressor**  
2.4.8 • Public • Published 6 years ago

Readme 0 Dependencies 23 Dependents 5 Versions 16

### YUI Compressor - The Yahoo! JavaScript and CSS Compressor

The YUI Compressor is a JavaScript compressor which, in addition to removing comments and white-spaces, obfuscates local variables using the smallest possible variable name. This obfuscation is safe, even when using constructs such as 'eval' or 'with' (although the compression is not optimal in those cases) Compared to jsmin, the average savings is around 20%.

The YUI Compressor is also able to safely compress CSS files. The decision on which compressor is being used is made on the file extension (js or css).

#### Building

ant

Install  
npm i yuicompressor

weekly downloads  
2,606

version  
2.4.8

open issues  
122

homepage  
github.com

license  
none

pull requests  
19

repository  
github

last publish  
6 years ago

## 17. Run it from NPM

Should you prefer to integrate the product into Node.js, visit [npmjs.com/package/yuicompressor](https://www.npmjs.com/package/yuicompressor). The badly maintained repository contains a set of wrapper files and a JavaScript API.

```
var compressor = require('yuicompressor');
compressor.compress('/path/to/
file or String of JS', {
  //Compressor Options:
  charset: 'utf8',
  type: 'js',
```

## 18. Keep Sass et al in check

While CSS selector performance is not nearly as critical as it was some years ago (see resource), frameworks such as Sass sometimes emit extremely complex code. Take a look at the output files from time to time and think about ways to optimise the results.

**GiftOfSpeed**  
Website Speed Test Tools Technologies [RSS](#)

Browser Caching Checker

HTTP://www.giftospeed.de

HTTP://www.giftospeed.de

19

## 19. Set up caching

An old adage claims that the fastest file is one that never gets sent across the wires. Making the browser cache requests achieves this efficiently. Sadly, the setup of the caching headers must take place on the server. Make good use of the two tools shown in the screenshots - they provide a swift way to analyse the results of your changes.

## 20. Bust the cache

Designers often dislike caching due to fear of problems with upcoming changes. A neat way around the problem involves including tags with the file name. Sadly, the scheme outlined in the code accompanying this step does not work everywhere as some proxies refuse to cache files with 'dynamic' paths.

```
<link rel="stylesheet" href="style.
css?v=1.2.3">
```

## 21. Don't forget the basics

Optimising CSS is just part of the game. If your server does not use HTTP/2 and gzip compression, a lot of time is lost during data transmission. Fortunately, fixing these two problems is usually simple. Our example shows a few tweaks to the commonly used Apache server. Should you find yourself on a different system, simply consult the server documentation.

```
pico /etc/httpd/conf/httpd.conf
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/css
```

General

- The Keep-Alive header is deprecated.
- The server's clock is correct.
- The Content-Length header is correct.

Content Negotiation ([Content Negotiation responses](#))

- The compressed response is negotiated, but doesn't have an appropriate Vary header.
- The response body is different when content negotiation happens.
- Content negotiation for gzip compression is supported, saving 13%.

Caching

- The resource last changed 4 years 191 days ago.
- This response allows all caches to store it.
- This response allows a cache to assign its own freshness lifetime.

Validation: [ETag Validation response](#) | [Last Modified Validation response](#)

## Essential resources

### On the problems of automated CSS analysis

[css-tricks.com/heres-the-thin-about-unused-css-tools/](http://css-tricks.com/heres-the-thin-about-unused-css-tools/)

Automatically removing unused parts from stylesheets is not a computationally trivial task. Those interested in algorithmics will find this discussion highly interesting.

**CSS-TRICKS**

### Here's the thing about "unused CSS" tools

By CHRIS COYKIN | PUBLISHED 10 JULY 2011

There are a lot of tools that aim to help you remove "unused CSS" from your project. Never a week goes by that I don't see a tool for this being shared or promoted. It must strike some kind of perfect chord for some developers. I care about performance, and I know that reducing file size is good for performance. Indeed, it is. But we have CSS that is unused in our application. If we remove that, that's a performance win. It would be. We should acknowledge that. But while I'm not so sure.

There are other performance tooling players that play up this idea.

### CSS selector benchmark

[sitepoint.com/optimizing-css-id-selectors-and-other-myths/](http://sitepoint.com/optimizing-css-id-selectors-and-other-myths/)

Browsers keep getting faster and faster. This survey performed by SitePoint clears up a few old rumours about CSS selector length and the impact on performance.

### PageSpeed Insights

[developers.google.com/speed/pagespeed/insights](http://developers.google.com/speed/pagespeed/insights)

Google's PageSpeed Insights tool runs predefined checks on your webpage and tells you areas worthy of improvement.

https://www.bbc.co.uk/sport/football/scores/fixtures

72

Field data

File Download Time (HTTP/1.1)

File Download Time (HTTP/2)

File Download Size (HTTP/1.1)

File Download Size (HTTP/2)

### Cache-busting techniques

[css-tricks.com/strategies-for-cache-busting-css/](http://css-tricks.com/strategies-for-cache-busting-css/)

Our discussion of ways to force cached resources to update is limited. This tutorial provides additional hints.

### All about HTTP/2

[developers.google.com/web/fundamentals/performance/http2](http://developers.google.com/web/fundamentals/performance/http2)

Google's developer portal contains an excellent introduction into all things related to HTTP/2.

OFFER ENDS  
31 DECEMBER 2018

# The perfect gift from just \$23\*

CSS SPECIAL ISSUE EXPERT 40-PAGE GUIDE

web designer™  
Expert tutorials, techniques and inspiration



FROM \$23 EVERY 3 MONTHS (\$93 PER 13 ISSUES YEARLY)

Delivery included in the price

Buy as a gift or treat yourself!

Free personalised e-card when buying for someone else



[www.myfavouritemagazines.co.uk/pin/xmas187](http://www.myfavouritemagazines.co.uk/pin/xmas187)

ORDER HOTLINE: +44(0) 344 848 2852

PLEASE QUOTE XMAS187 WHEN ORDERING BY PHONE

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

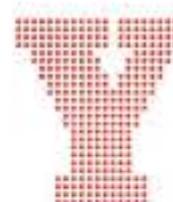


**Terms and conditions:** Savings calculated against the full RRP (single issue price x frequency). Dollar prices quoted are for the United States, other global territory dollar pricing may vary. This offer is for new subscribers only. You can write to us or call us to cancel your subscription within 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 terms and conditions please visit: [bit.ly/magterms](http://bit.ly/magterms). Offer ends 31st December 2018.

# Style a site using Sass

Sass is a style sheet language that extends CSS. It's common to take advantage of a CSS 'preprocessor' like Sass to make styling easier, and in this tutorial we'll show you how





You can do a lot with CSS – perhaps more than you might think – but the venerable style sheet language has its limitations.

In a modern web project, with npm modules, JavaScript frameworks, ES6 and more, it can feel somewhat anachronistic to fall back to writing vanilla CSS.

Fortunately, there are options out there that allow you to use other languages that compile to CSS. Referred to as ' preprocessors ', these tools integrate into your web build process and generate usable style sheets from whatever extension language you've chosen.

Sass (Syntactically Awesome Style Sheets) is one of the most popular of these options. Sass adds many valuable new language features that aren't (currently) available in CSS to help make your sites and apps more maintainable. These include things like 'mixins' and 'control directives', which sound daunting but are actually quite straightforward, and we'll look at these in this tutorial.

There are actually two different syntaxes for Sass, one that uses a '.scss' file extension, and one that uses '.sass'. The former looks more like CSS (in fact, all .css files are valid .scss files), while the latter eliminates CSS's brackets and semicolons in favour of indentation and newlines. We'll focus on .scss, but the choice is simply down to personal preference.

## 1. The compiler

Using Sass essentially requires a compiler. The simplest way to do this would be at the command line. You can do so using Homebrew. The Sass compiler is implemented in several different languages, and Homebrew will install the Dart version, which is fast.

```
brew install sass/sass/sass
```

## 2. Our first Sass file

Let's try creating a simple Sass file to see the compiler in action. One of the simplest concepts in Sass is variables, which can be specified once with a '\$' prefix and then used throughout your code. We'll create 'sass-input.scss'.

```
$text-color: #cccccc;
body {
  color: $text-color;
}
```

## 3. Command line compilation

Now we can run the Sass compiler at the command line to convert our .scss file into CSS output. You'll notice in the output file that the variables are gone and we're just left with standard CSS syntax that is usable by the browser.

```
sass sass-input.scss css-output.css
```

## 4. Automating build

This is great, but you don't want to run the Sass compiler manually every time you make changes. One option is to have it listen for changes to files in a directory and automatically recompile the output to a different directory (preserving filenames). This also effectively lets you segregate your source .scss file from your built CSS.

```
mkdir src
mkdir src/sass
mkdir public
mkdir public/css
sass --watch src/sass/:public/css/
```

## 5. A starter site

Now let's look at a simple starter site, which we can use to play around with Sass styling. We can get started by cloning an unstyled example site. The key thing to note is that our HTML page knows nothing about Sass. It has a single CSS file entry point, which we'll create next.

```
cd public
curl -o index.html https://raw.githubusercontent.com/simon-a-j/sass-tutorial/master/public/index.html
```

Here's the starter project we'll use to experiment with Sass styling

The official Sass website, which has many useful resources and comprehensive documentation

## 6. The main stylesheet

We're using styles.css as our main CSS entry point, which later on we'll use to import other modules. This means we need to tell Sass to generate this file, so we also need to create a styles.scss file in our src/sass folder. If you're running 'sass -watch' as before, this will automatically be compiled into CSS in the 'public/css' folder, and refreshing your site will show its changes. Try making some modifications and refreshing the HTML page in the browser as you go.

```
// styles.scss
body {
  font-family: sans-serif;
  text-align: center;
}
```

## 7. Colour scheme

Let's look at how Sass can help us manage a colour scheme for the site. It's common to have a palette of 5 or 6 colours for a webpage. We can externalise these in '\_colours.scss'. The underscore prefix tells Sass not to compile this into a new HTML file (a 'partial'). But we can use it in a slightly different way.

```
// _colours.scss
$colour-primary: #231651;
$colour-secondary: #2374AB;
$colour-light: #D6FFF6;
$colour-highlight1: #4DCCBD;
$colour-highlight2: #FF8484;
```

Our site now has a colour scheme thanks to a Sass partial

## 8. Using variables

To use these colour variables we've just set up, we can tell Sass to import the content of '\_colours.scss' into our main style sheet. We do this using an @import statement. Once you've done this, notice how the variables are resolved within the output CSS file.

```
// styles.scss
@import "_colours.scss";
body {
  font-family: sans-serif;
  text-align: center;
  background: linear-gradient(155deg,
```

## Build pipeline

We've looked at the simplest possible command-line usage of Sass, but in many real-world scenarios, you'll want it more neatly integrated into your build pipeline. A lot of popular framework starter kits do this already, but you can also set it up yourself with, say, Node – in which case you can use gulp or npm scripts and the node-sass module.

## Tutorials

### Style a site using Sass



We can introduce a responsive grid, which is easier to manage with Sass variables and nesting



A mixin helps us maintain backward compatibility with older browsers when we use CSS transforms like rotation

```
$colour-primary 70%, $colour-secondary  
70%);  
color: $colour-light;  
min-height: 100vh;  
}  
h1 {  
color: $colour-highlight1;  
}  
h2 {  
color: $colour-highlight2;  
}
```

### 9. Nesting styles

Another useful feature of Sass is the ability to nest styles. That is, you can specify a style for an element that is only applied if that element occurs within a parent element. Let's use this to differentiate our styling of links depending on whether they appear in the header or body.

```
a {  
color: $colour-secondary;  
}  
.profile-header {  
a {  
font-size: 16px;  
margin-left: 10px;  
margin-right: 10px;  
padding: 10px;  
border-radius: 5px;  
color: $colour-light;  
background-color: $colour-secondary;  
}  
}
```

### Mixins and placeholders

These might look quite similar, and there are many scenarios where you could use either. Generally mixins are most appropriate where you have parameters that will vary depending on where they are used. If the style snippet is always going to behave exactly the same way, with no variables passed to it to affect its behaviour, then a placeholder would make more sense.

### 10. Responsive grid

Now let's arrange our content into a responsive grid format. Sass has a couple of features to make this significantly easier to manage. As well as using variables to specify our breakpoints, we can nest @media queries within other styles, which makes behaviour specific to screen size much more readable.

```
$breakpoint: 800px;  
.profile-body {  
display: flex;  
align-items: stretch;  
justify-content: space-around;  
margin-top: 32px;  
margin-left: 10vw;  
margin-right: 10vw;  
@media screen and (max-width:  
$breakpoint) {  
flex-direction: column;  
}  
.profile-section {  
background-color: $colour-highlight1;  
color: $colour-primary;  
margin: 16px;  
border-radius: 10px;  
width: 340px;  
.profile-content {  
padding: 20px;  
}  
@media screen and (max-width:  
$breakpoint) {  
width: 100%;  
}
```

### 11. Introducing mixins

Mixins are another powerful Sass feature, which you can think of as a way of defining reusable stylesheet functions. A mixin is defined once, can take parameters, and can then be invoked anywhere in your Sass code. One use case for this is to handle vendor prefixing. If we want a CSS transform to work in older browsers, this might require a 'webkit' prefix for Chrome and Safari, for example. Let's define a mixin that takes care of this for us.

```
@mixin transform($value) {  
-webkit-transform: $value;
```

```
-moz-transform: $value;  
-ms-transform: $value;  
transform: $value;  
}  
.profile-logo {  
font-size: 60px;  
font-family: cursive;  
@include transform(rotate(25deg));  
}
```

### 12. Mixins and variables

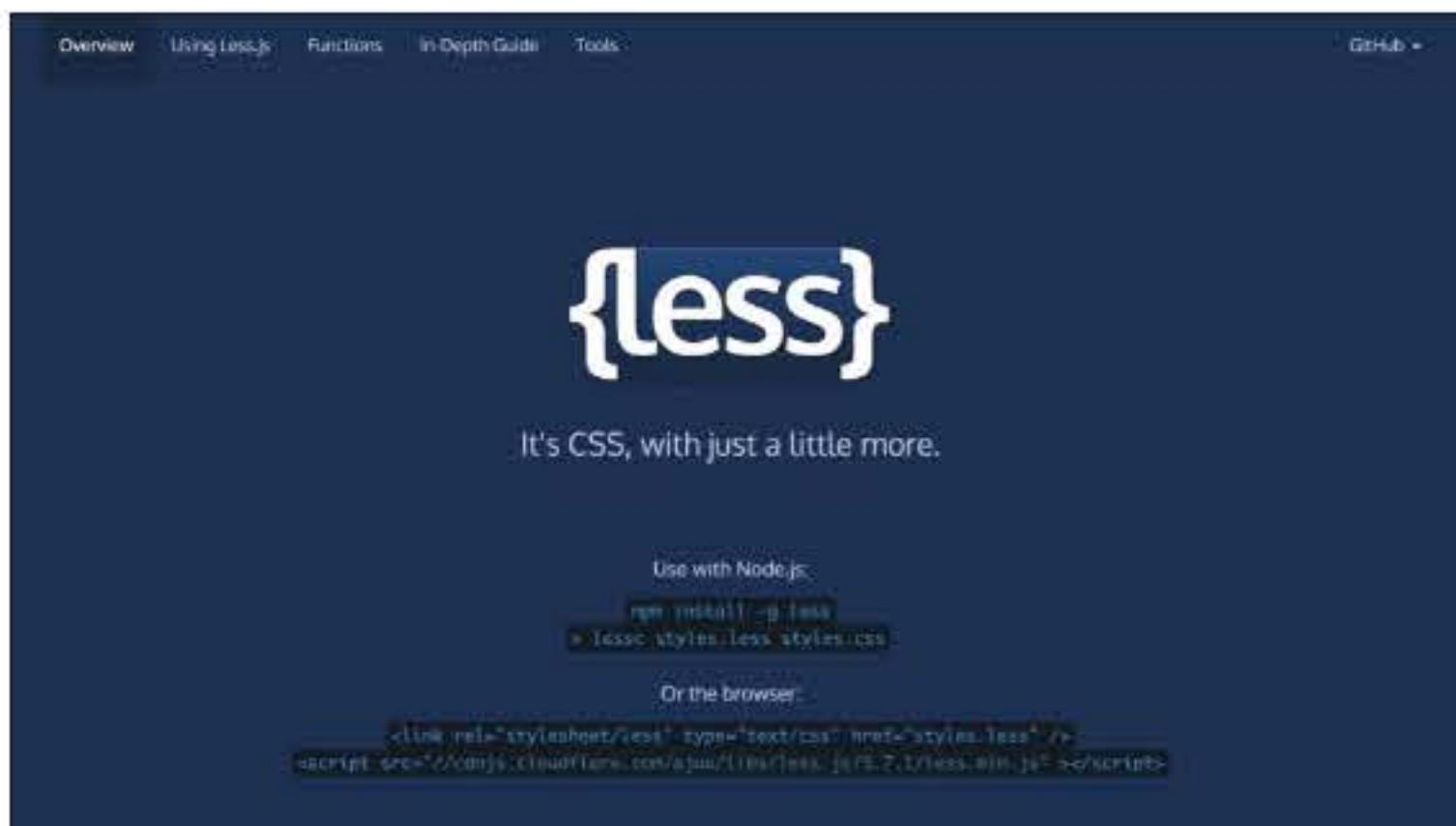
We can also use a mixin with multiple parameters, combined with some variables that we define, to more elegantly handle styling of various parts of the page. If we create a mixin that defines foreground and background colour, this will enable us to select appearances for different sections from a finite list of style variables.

```
$style1: (foreground: $colour-light,  
background: $colour-secondary);  
$style2: (foreground: $colour-primary,  
background: $colour-highlight1);  
$style3: (foreground: $colour-primary,  
background: $colour-highlight2);  
@mixin content-style($foreground,  
$background) {  
color: $foreground;  
background-color: $background;  
}
```

### 13. Using our new mixin

Now, we can use the content-style mixin to replace colour and background-colour statements elsewhere in our style sheet. When including it, we specify one of the two \$style variables as a parameter. Note the ... after the variable name, which tells Sass to pass the variable's content as a list, matching the number of parameters the mixin expects.

```
.profile-header {  
a {  
@include content-style($style1...);  
// ...  
}  
.profile-section {  
@include content-style($style2...);  
// ...  
}
```



## Other preprocessors

Sass is just one of the CSS preprocessors out there, and you might want to consider checking out the competition. Preprocessing is largely a stylistic preference, and the best choice is really the one that you find best suits your personal development style (or integrates most easily with other tooling you use).

Less (Leaner Style Sheets) is, like Sass, an extension of CSS that maintains backward compatibility. It's written in JavaScript so requires Node.

Stylus has flexible syntax support, letting you write either regular CSS, .sass style CSS, or omit braces, semicolons and colons completely and write very concise style sheets.

PostCSS takes a slightly different approach to the competition, allowing you to pick and choose which language extensions you need via a plugin system, rather than shipping everything at once.

The features supported by most of the preprocessing solutions are fairly similar - you'll generally get variables, nesting, mixins, functions and imports.

## 14. Inheritance

Another very powerful feature of Sass is inheritance. Right now we have two different styles for links in our page. If we want to use common styles across both, rather than copying and pasting CSS, why don't we use a placeholder class, denoted with '%', which can be extended by both, allowing them to inherit its styles?

```
.link-shared {
    font-size: 16px;
    margin-left: 10px;
    margin-right: 10px;
    padding: 10px;
    border-radius: 10px;
}
```

## 15. Extending classes

Now we can extend the link-shared class to define link styling throughout our site. This is starting to look quite elegant. We define what a link generally looks like just once, reuse it throughout, and specify colours from the palette for each link using a mixin.

```
.profile-header {
    a {
        @extend %link-shared;
        @include content-style($style1...);
    }
}
a {
    @extend %link-shared;
    @include content-style($style3...);
}
```

## 16. Modifying the theme

Now let's take a look at how easy Sass makes it to modify the theme of our site. The current colours might not be perfect. We cannot only modify the colour variables, but we can also use some Sass functions to procedurally generate colours that match a primary of our choosing.

```
// _colours.scss
$colour-primary: #2E1F27;
$colour-secondary: lighten($colour-primary,
25%);
$colour-light: lighten($colour-primary,
```

```
75%);

$colour-highlight1:
lighten(complement($colour-primary), 50%);

$colour-highlight2:
lighten(complement($colour-secondary), 50%);
```

## 17. Selecting a new set of colours

Now, we can modify the entire colour scheme for the site simply by specifying a new colour-primary value in the '\_colours.scss' file. Give it a try by experimenting with alternative colours. We could also have Sass randomise it (but remember this refers to the point at which your site is built, not runtime). You can also try adjusting the logic we're using to derive the other colours in the theme from the primary.

```
$red: random(255);
$green: random(255);
$blue: random(255);
$colour-primary: rgb($red, $green, $blue);
```

## 18. Using libraries

Sass's module system also makes it very straightforward to use 3rd-party libraries with minimal effort, and without shipping large runtime files to your end users. Let's try out the Angled Edges library, which we can use to create sloped edges for objects on our page.

```
git clone https://github.com/josephfusco/
angled-edges.git src/sass/angled-edges
```

## 19. The Angled Edges mixin

We can import Angled Edges the same way we did for our colour scheme partial. It's then usable via a mixin that ships with the library. Let's try it out in our profile-section class.

```
@import "angled-edges/_angled-edges.scss";
.profile-section {
    @include angled-edge("outside bottom",
"lower right", $colour-highlight1);
    @include angled-edge("outside top",
"upper right", $colour-highlight1);
    margin: 120px 16px 120px 16px;
    // ...
```

## 20. Output formatting

Let's finish up by looking at the output Sass generates. If you've been tracking your CSS files as we've made changes, you'll notice they remain quite readable. However, you can also have Sass build condensed CSS, which is less human-readable but still ready to ship. You can do this using the '--style' command-line flag.

```
sass src/sass/:public/css/ --style
compressed
```

## 21. More Sass

We've now explored quite a few features of Sass, and our site isn't looking too bad. Hopefully you're beginning to see how Sass helps us develop more maintainable style sheets. We haven't covered every feature of the language - there are many more useful functions shipped with it, and advanced features like control directives (such as @if, @for and @while) that are often used to create complex library functions. Overall, remember that Sass is entirely a stylistic preference. You can do everything we've seen with pure CSS if you like, but you should definitely think about preprocessing as your work becomes more complex.

[LinkedIn](#)
[Twitter](#)

13

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit



LET'S BREAK DOWN HOW TO MAKE ART WITH CSS BY MAKING A CUTE ICE CREAM BAR GRAPHIC, GOING THROUGH TECHNIQUES STEP BY STEP

## SAY HELLO TO CSS ART

Making CSS art can be intimidating, but it's easy once you break graphics down into simple parts

If you peruse websites like CodePen, DevArt and Dribbble, it's very easy to feel overwhelmed by the incredible things that developers and designers make, especially when you're just starting out. I made my first piece of CSS art after lurking on these sites for years and eventually mustered up the courage to try and do it. Now, a few years and over 100 experiments later, it's one of my favourite hobbies.

Making CSS art is interesting and fun. It teaches you nuances of CSS that you might not normally work with, helps you see graphics

(and problems in general) modularly and improves your speed when writing CSS. It's also a great way for designers and developers to get on the same page when working on a user interface together. As you practise and refine your skills making CSS art, you'll see an improvement in your day-to-day work (and you might even enjoy it more, too).

In this article, we're going to be making this cute ice cream bar graphic in CSS. I'll talk through and give you the code to follow, so that you can customise it and make it your own. Let's get started!

**As you practise and refine your skills making CSS art, you'll see an improvement in your day-to-day work (and you might even enjoy it more, too)**



### WHERE CAN I FIND...?

Want to make graphics after this article and not sure where to start? Check out Dribbble, Behance and CodePen Challenges for inspiration. If you end up using one of the graphics or ideas you find on those sites, don't forget to credit the work!

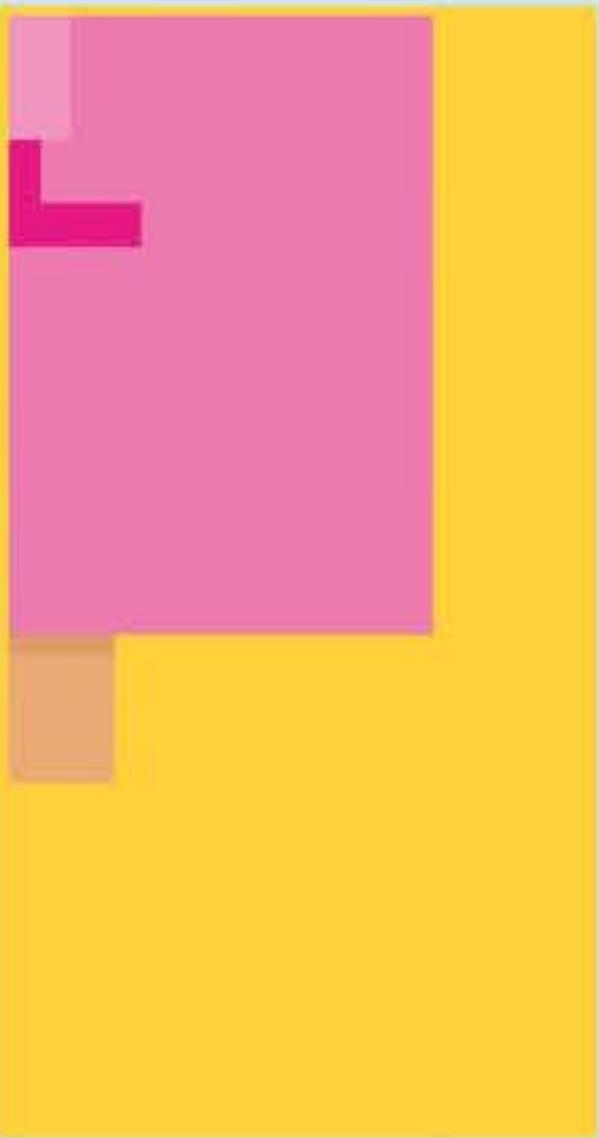
# LET'S MAKE ART WITH CSS

A step-by-step guide to creating an ice cream bar graphic

## 1. SETTING UP YOUR DOCUMENT

Before we write any CSS, we have to actually have something to style. You can use a preprocessor like HAML or Pug, or just straight HTML. I'll stick with pure HTML and CSS in our examples, but you're free to use what you're most comfortable with. Pull up a blank Pen on CodePen or an HTML document, and include the following for the ice cream bar in the <body>.

```
<div class='container'>
  <div class='ice-cream'>
    <div class='glare'></div>
    <div class='face'>
      <div class='eyes'>
        <div class='eye left'></div>
        <div class='eye right'></div>
      </div>
      <div class='mouth'></div>
    </div>
    <div class='stick'></div>
  </div>
```



## 2. COLOURS AND SIZES

Normally when you have a graphic in front of you to make, you will need the dimensions and colours of that graphic. In this particular case, I'll be providing you with those dimensions and colours. When you don't have me around though, I recommend getting yourself a colour picker tool (I personally use Digital Color Meter on Mac and ColorPic on Windows), and a screenshot tool to figure out your dimensions (the native ones for your operating systems work just fine). Let's add those dimensions and colours now to our document.

```
body {
  background: #FED550;
}

.ice-cream {
  width: 240px;
  height: 350px;
  background: #F982BF;
}

.glares {
  width: 35px;
  height: 70px;
  background: #FF98CC;
}

.eye {
  width: 18px;
  height: 18px;
  background: #FF2995;
}

.mouth {
  width: 75px;
  height: 25px;
  background: #FF2995;
}

.stick {
  width: 60px;
  height: 72px;
  background: #ECAA7D;
  border-top: 12px solid #DB9E74;
}
```

## 3. POSITIONING AND TRANSFORMING

Right now, our image is all blocky and in the corner. Before we shape everything better, let's add some positioning and transforms. Personally, my favourite positioning techniques (generally) are absolute positioning containers, and using flexbox for everything else. If you prefer using absolute and relative positioning for everything, or even CSS Grid, go for it! Now, we have to position the container in the middle of the page (I'll be using absolute positioning and a translate transform), and make sure the contents of the main container are centred and rotated as needed. You might notice that I also added a couple of dimensions for the '.face' and '.eyes' classes - that's for handling the spacing within that container. Go ahead and add these changes in yourself to each class:

```
.container {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  display: flex;
  flex-direction: column;
  align-items: center;
}

.ice-cream {
  display: flex;
  flex-direction: column;
  justify-content: space-around;
}

.glares {
  position: relative;
  left: 170px;
  bottom: 20px;
  transform: rotate(-38deg);
}

.face {
  display: flex;
  flex-direction: column;
```

```
align-items: center;
justify-content: space-between;
height: 58px;
}
```

```
.eyes {
  display: flex;
  justify-content: space-between;
  width: 145px;
}
```

## 4. PSEUDO-ELEMENTS

Our little ice cream bar is looking pretty good, if I do say so myself! One last concept that is incredibly helpful for making CSS art is pseudo-elements. Every single element on a page can actually be three elements: the main element, ::before, and ::after. You can style ::before and ::after as if they're separate <div>s within any element. For example, I could have changed the .eyes container to have no .eye <div>s at all, and just style a ::before and ::after for each eye. That would have been cool, but that would make styling the glare on each eye impossible. That being said, for the final missing glare on those eyes, let's add a ::before pseudo-element to make that possible. The key things to add to a pseudo-element are a display value and content: ". Check it out, and with that, we're done!

```
.eye {
  position: relative;
}

.eye::before {
  display: block;
  content: '';
  position: absolute;
  top: 20%;
  left: 50%;
  transform: translateX(-50%);
  width: 5px;
  height: 5px;
  background: #fff;
  border-radius: 100%;
```



## Q&A WITH CASSIDY

**Cassidy Williams tells us how she started making art with CSS and what inspired her**

### Hi Cassidy. Tell us a little about yourself and how CSS is part of your life.

Hi there! I'm Cassidy and I'm a senior software engineer at CodePen. CSS is one of the first things I taught myself how to code back when I was in middle school and first 'discovering' the ability to make things and put it on the internet (from my own personal sites to guild pages on Neopets). Obviously the internet is very different now, but CSS is still around and better than ever! I love experimenting with it and seeing new features come out.

### What inspired you to try out creating art with CSS?

I first got interested in making CSS art (with CodePen, actually) when I saw that Google launched their DevArt website ([devrt.withgoogle.com](http://devrt.withgoogle.com)) back in 2014. After experimenting with it a little on my own and following developers I admire, I decided to make it a goal to make CSS art regularly, around once a week, and eventually I was making it so often it led to me being introduced to the team at CodePen, and now I work there!

### What advice would you give to anyone wanting to get into the industry or those looking to progress?

It's cliché because it's true: Practise, practise, practise! Doing CSS art and experiments has made me a better developer. It forces you to think modularly and improves your ability to write UIs quickly. If you want to get better, keep building, even if it means trying to copy someone else's art (don't forget to give credit) or making something that might seem completely useless. Also, as someone new to the industry, it's a really interesting skill to have in your portfolio. Even though you might not use CSS art to actually get a job, it's a great way to get a foot in the door and show off your logical, creative code.



## YOU'RE AN ARTIST!

We made something really fun together today. If you'd like to check out my final result, you can find it on my CodePen profile ([codepen.io/cassidoo/pen/QJmxWX](https://codepen.io/cassidoo/pen/QJmxWX)). Making CSS art is really just a matter of applying the concepts we talked about in interesting and fun ways. You can take these techniques and run with them, and even add your own animations with keyframes and transitions. All this being said, I hope you learned something new, and this motivated you to try your hand at making something beautiful.

The screenshot shows a CodePen interface with three tabs: HTML, CSS, and JS. The HTML tab contains a single line of code: <div>. The CSS tab contains the following:

```
body { background: #F1D5E8; }

.container {
  position: absolute;
  top: 50px;
  left: 50px;
  transform: translate(-50%, -50%);
  display: flex;
  flex-direction: column;
  align-items: center;
}

.ice-cream {
```

The JS tab is empty. Below the interface, there is a small, cute, pink, teardrop-shaped CSS art character with a smiling face.

# CSS METHODOLOGIES

Ahmed explains what CSS methodologies are, the most popular ones and how they may be combined together to create a custom methodology



Using a CSS methodology, or a combination of many, can save you hours of development and debug time, leading to more organised code that is easier to write, scale and maintain

**AHMED ABUEL GASIM**

Front-end web developer at Potato  
[hungryPhilomath.com](http://hungryPhilomath.com)

# WHAT ARE CSS METHODOLOGIES?

**C**SS methodologies refer to sets of guidelines for writing modular, reusable and scalable code. Although CSS is an easy language to write, without an agreed-upon convention the code gets messy almost as fast as it is written. Since each CSS declaration is defined on its own line, files get huge quickly, making them a nightmare to maintain. Preprocessors, such as Sass, Less and Compass, have done wonders to mitigate this problem by allowing selectors to be nested and blocks of code to be replaced with single-line 'mixin' declarations. While this helps, large projects can still require thousands of lines of code. Fortunately, preprocessors also allow CSS to be split across smaller files, or 'partials'. But what to include in each partial and how they are named must be agreed upon by a team, otherwise their use can do more harm than good.

## MANAGE SPECIFICITY

Another potential problem experienced with complex projects is managing specificity. CSS assigns a weight to each style rule, so when multiple rules are used on the same element, the highest weighted rule is considered more specific and is therefore applied. When multiple, equal-weight rules are used the lowest one wins. Specificity is calculated using four number groups represented as 0-0-0-0, where numbers do not overflow from one group to another, so 0-1-2-1-5 is valid. Each element or pseudo-element in a selector increments the right-most group, e.g. 'h1' is 0-0-0-1 and 'div::before' is 0-0-0-2. Each class, attribute or pseudo-class increments the next group, e.g. '.some-class.another-class' is 0-0-2-0 and 'section.some-class.another-class:hover' is 0-0-3-1. IDs increment the next group, e.g. '#some-id#another-id' is 0-2-0-0 and 'ul#some-id img.some-class:active' is 0-1-2-2. Inline styles applied using HTML style attributes increment the leftmost group and are therefore the most specific selectors. The higher the overall number, the more specific the selector. So if one developer uses 'div.some-class' to apply styles to an element, it is not possible to override them lower down the code using '.some-class' on its own. It is therefore common practice to use only single-class selectors when possible.

To solve these and other CSS implementation issues, groups of coders around the world have developed different CSS methodologies, or sets of standard practices, each with their own focuses, advantages and disadvantages. They are not frameworks or libraries, rather rules for writing CSS code that encourage developers to stick to conventions that make code easier to write and maintain, saving hours of development time. These methodologies are not mutually exclusive and can be used together in a way that best suits developers.

In this article we will take a look at a few of the most popular CSS methodologies, highlighting their pros and cons, and how they can be combined together in a custom methodology.

```
# styles.css x
113
114 /* Media block */
115 .media-block {
116   background: #ccc;
117   color: #414141;
118   padding: 24px;
119   width: 400px;
120 }
121
122 .media-block:hover {
123   background: #eee;
124 }
125
126 .media-block--small {
127   width: 200px;
128 }
129
130 .media-block__media {
131   margin-right: 24px;
132   border: 1px solid black;
133 }
134
135 .media-block__content {
136   font-size: 14px;
137 }
138
139 .media-block__content--featured {
140   font-size: 16px;
141 }
142
143 .media-block__title {
144   font-weight: bold;
145 }
146
147 .media-block__details {
148   margin-top: 24px;
149 }
```

Methodologies provide rules for writing CSS code

# Popular CSS methodologies

Of the vast number of CSS methodologies, we will explore the most popular, discussing their advantages and disadvantages

## 1. OBJECT-ORIENTED CSS (OOCSS)

Divide layout into objects, then abstract their CSS into modules

OOCSS involves identifying objects on a page and separating their structural and visual CSS styles into two declaration blocks. These blocks can then be reused by different elements, and changes need only be made in one place, leading to better consistency.

Declaration blocks are applied to elements using single-class selectors to avoid specificity issues. This technique also separates content from container, so objects look the same wherever they appear. Classes also decouple mark-up from CSS. Using '`.title`' instead of '`h2`' for heading '`<h2 class="title">`' allows it to be changed to '`<h3 class="title">`' without changing the CSS. To further separate HTML and CSS, class names should not include property values. A class '`blue`' would require renaming in HTML and CSS if the colour changed.

Using OOCSS a button's CSS and mark-up can be defined as:

```
.button { padding: 10px 16px; }
.primary-skin { color: blue; }
.secondary-skin { color: green; }

<button class="button primary-skin">primary skin button</button>
<button class="button secondary-skin">secondary skin button</button>
<div class="primary-skin">primary skin div</div>
```

OOCSS introduces many useful concepts, but its lack of rules leads to variations in interpretation that can result in inconsistencies. It has, however, been used as inspiration for stricter methodologies.

## 2. ATOMIC CSS (ACSS)

Create a class selector for every repeating CSS declaration

ACSS encourages developers to define single-purpose class

selectors for every reusable declaration. Unlike OOCSS, which discourages CSS property values in class names, ACSS welcomes it. Using ACSS styles can be defined and applied to elements as:

```
.mb-sm { margin-bottom: 16px; }
.mb-lg { margin-bottom: 32px; }
.color-blue { color: #1e90ff; }

<div class="mb-lg">
  <p class="mb-lg color-blue">Blue text</p>
  <img class="mb-sm" />
</div>
```

There are programmatic approaches to ACSS that automatically generate CSS based on classes or attributes that users add to the HTML. Atomizer ([acss.io](https://acss.io)) is one such tool, allowing the previous HTML to be redefined as:

```
<div class="Mb(32px)">
  <p class="Mb(32px) C(#1e90ff)">Blue text</p>
  <img class="Mb(16px)" />
</div>
```

automatically generating the following CSS upon build:

```
.Mb\16px\ {
  margin-bottom: 16px; }
.Mb\32px\ {
  margin-bottom: 32px; }
.C\#1e90ff\ { color: #1e90ff; }
```

The main benefit of ACSS is the ease of maintaining consistent code and not having to invent classes for components requiring a single CSS rule.

However, ACSS used on its own can lead to an unmanageable number of classes and bloated HTML files. It is therefore common to only use ACSS principles to create helper classes that define consistent, reusable declaration blocks.

## 3. BLOCK ELEMENT MODIFIER (BEM)

Use a standard naming convention for classes

BEM encourages developers to divide layouts into blocks and

nested elements. Variations from the average appearance of a block or element should also be identified and applied using modifiers.

CSS declarations are applied using a single class name of format '`block-name`' for blocks and '`block-name_element-name`' for elements, with two underscores in between. Modifier names are appended to classes, prefixed with an underscore or two hyphens for better clarity, for example '`block-name_element-name_modifer-name`' or '`block-name_element-name--modifer-name`'. An object is a block if it can exist without ancestors, otherwise it's an element.

Blocks can have nested blocks and elements, but elements cannot. Modifiers must be used alongside block and element classes, not instead of them.

BEM can be applied to a list, where '`list-block--inline`' and '`list-block__item--active`' display lists horizontally and highlight items respectively:

```
<ul class="list-block list-block--inline">

  <li class="list-block__item">Item 1</li>
  <li class="list-block__item">Item 2</li>
</ul>

<ul class="list-block">
  <li class="list-block__item list-block__item--active">Item 1</li>
  <li class="list-block__item">Item 2</li>
</ul>
```

BEM is a highly effective naming convention that creates predictably behaving CSS that is easy to manage, maintain and scale. BEM does have downsides, however, including the difficulty in inventing class names for deeply nested objects, the ridiculously long class names and bloated HTML that may sometimes result,

and also the lack of consistency that is caused by the inability to share CSS between objects.

## 4. SCALABLE AND MODULAR ARCHITECTURE FOR CSS (SMACSS)

Split CSS code across multiple files for better performance and organisation

SMACSS works by dividing CSS into five categories - base, layout, module, state and theme - commonly split into separate files.

Base styles override the default styles and are mainly applied using element selectors:

```
h1 { font-size: 20px; }
a { text-decoration: none; }
```

Layout styles are for major objects like headers and sidebars. They are applied using IDs or classes with generic helper declarations optionally prefixed with '`l-`':

```
#header { height: 50px; }
.l-right { float: right; }
```

Module styles are for smaller, reusable objects like buttons and lists, each commonly with its own file. They are applied using classes, with nested items classes commonly prefixed with the ancestor class:

```
.list { ... }
.list--icon { ... }
.list--text { ... }
```

State styles are for changeable states, like hidden or disabled. They are commonly applied with class names prefixed with '`is-`' or '`has-`' and chained to other selectors:

```
.button { ... }
.button.is-disabled { ... }
```

Theme styles are optionally used for changing the visual scheme.

SMACSS provides well-organised CSS code split logically across multiple files.

Using SMACSS does, however, introduce specificity traps by allowing IDs and relying on selector chaining for state and some layout declarations.

# Combining CSS methodologies

CSS methodologies can be used together to create a custom methodology that works best for you

**A**s you have probably gathered from this article, each CSS methodology comes with its own benefits and drawbacks. It is, however, possible to combine aspects of multiple methodologies together to create your own custom one that works best for you.

Let's look at one way of combining the four methodologies discussed so far, for a site with a homepage and a button component, using Sass as a preprocessor. Applying SMACSS principles, we can divide our code across multiple Sass partials as shown opposite.

Then import them into 'styles.scss' that will be converted to 'styles.css' by Sass, as follows:

```
@import 'base';
@import 'helpers';
@import 'components/button';
@import 'pages/home';
```

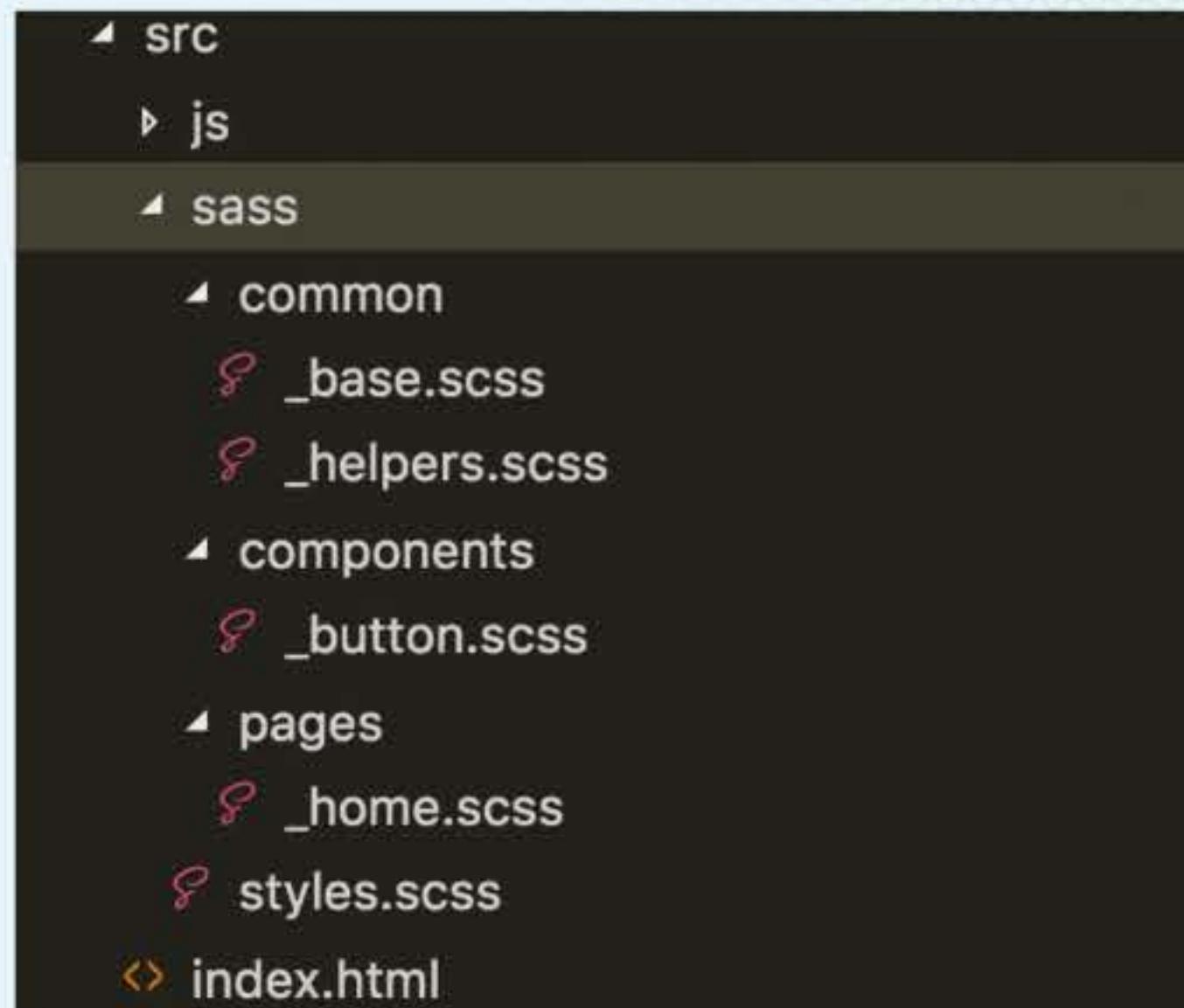
Next we can add any styles that override the browser defaults to '\_base.scss', allowing mainly element selectors and their pseudo-classes:

```
* { box-sizing: border-box; }

html,
body { height: 100%; }

a {
  &, &:active, &:focus, &:hover,
  &:visited {
    text-decoration: none;
  }
}
```

Selector chains are sometimes required to override unwanted styles applied by external frameworks. For example, the 'Materialize.css' framework applies padding to grid columns using a two-class selector chain '.row .col', making it impossible to override with a single-class BEM selector. Such overrides should also be added to '\_base.scss', for example



'.row .col {padding: 16px}'. For this to work make sure external libraries are sourced in the HTML before 'styles.css'.

Using ACSS ideas we can create helper classes that apply consistent styles to any element, eliminating the need to create a new class name and component file for elements requiring a single CSS declaration. Instead we can apply the helper class directly to the HTML. For example, we can create a responsive, top margin helper class in '\_helpers.scss':

```
.h-margin-top {
  margin-top: 16px;

  @media (min-width: 720px) {
    margin-top: 24px;
  }
}
```

For each component we will have a separate file in the 'components' directory and use a BEM methodology. We will allow BEM formatted single-class selectors, their pseudo-elements and an infinite number of chained pseudo-classes. For example, the CSS of buttons can be defined in '\_button.scss' as following, with modifier 'button--is-disabled' greying out the button and showing a tooltip with the message 'disabled' on hover:

'button.scss' as following, with modifier 'button--is-disabled' greying out the button and showing a tooltip with the message 'disabled' on hover:

```
.button {
  position: relative;

  &__icon { width: 12px; }

  &__text { text-transform: uppercase; }

  &--is-disabled {
    background: grey;

    &::after {
      content: 'disabled';
      position: absolute;
      top: -150%;
      left: 16px;
    }

    &:not(:hover)::after {
      display: none;
    }
  }
}
```

Finally, we can add page-specific overrides to a corresponding file in the 'pages' directory. To ensure that these overrides are always applied

**As you can see, combining methodologies is easy and can lead to a personalised, consistent approach to CSS development that scales effortlessly and is easily maintained**

to our elements and those from external libraries with potentially multi-class selectors, we will give each page a unique ID. For example, we can hide disabled buttons on the homepage by adding the following to '\_home.scss':

```
#home {
  .button--is-disabled {
    display: none;
  }
}
```

As you can see, combining methodologies is easy and can lead to a personalised, consistent approach to CSS development that scales effortlessly and is easily maintained. The custom methodology detailed above is just a suggestion, and my advice is for you to develop your own. Combine aspects you like from as many methodologies as you can find, adapting them to your liking, and stick to them. There is little point in creating a methodology if you constantly deviate from it. If you find yourself doing so, then incorporate these deviations into your methodology in a way that is consistent and easy to understand.

# THE CUTTING EDGE OF

# CSS

Stay on top of CSS trends, new features and new properties to develop your design skills and make award-winning websites

**C**SS is one of the most important tools of any web designer, given that it adds the design styling to pages, but it has become far more important than that. Having the ability to put animation on pages and react to interactions is the central part of creating a compelling user experience. CSS now performs many tasks that

previously needed JavaScript, such as off-screen menus, drop downs, lightboxes, accordians and parallax scrolling. This makes your site more performant. Staying on top of trends and new features is essential to expanding your design skills towards making award-winning websites. Check out the examples opposite.

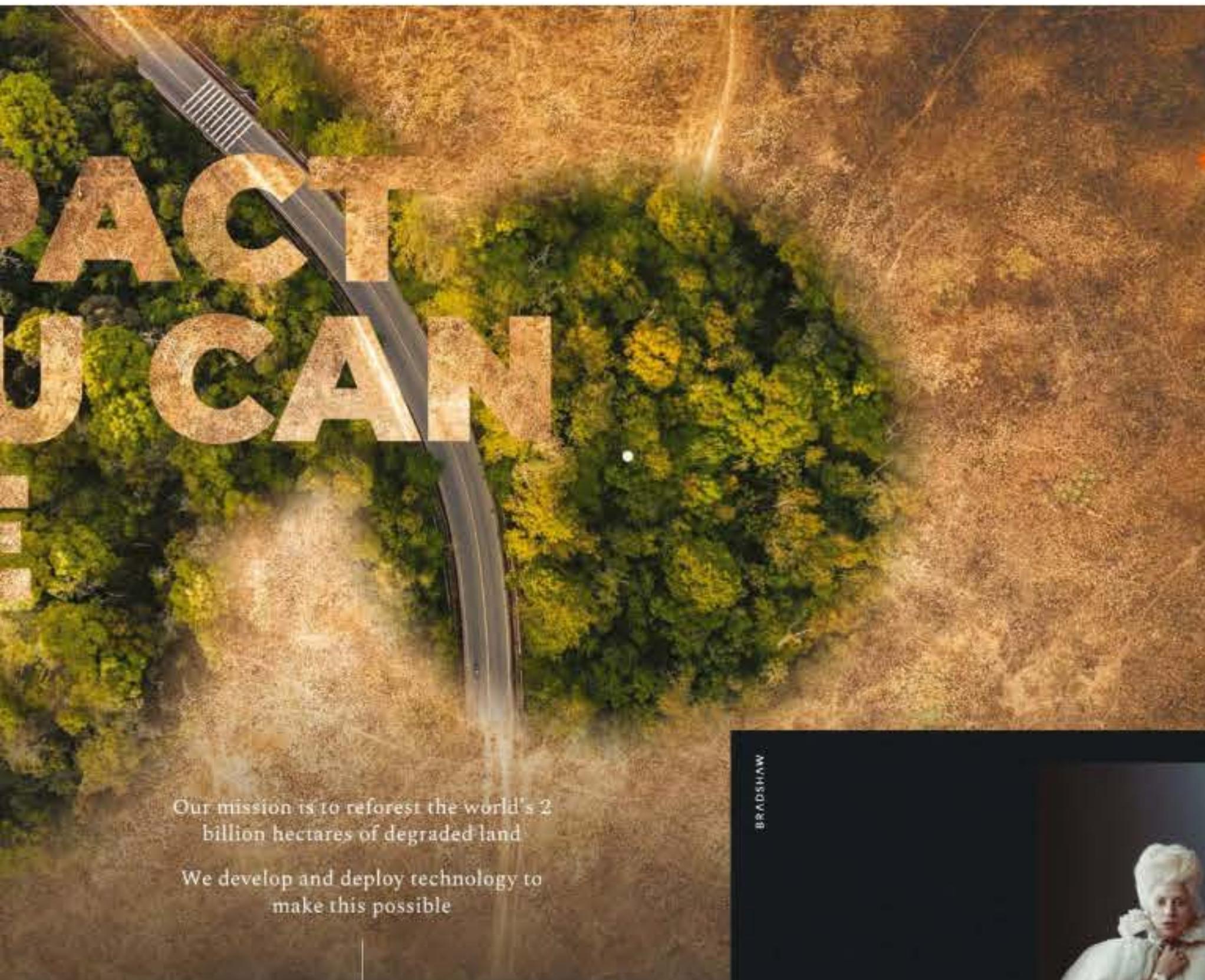


**CSS is the designer's swiss army knife, specifically created to add the design to your pages, it's the magic that elevates your design above your competitors**

**MARK SHUFFLEBOTTOM**

Interaction Design Professor  
[twitter.com/webspaceinvader](http://twitter.com/webspaceinvader)





### CSS AND SVG CLIPPING MASKS

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

Land Life Company makes extensive use of SVG clipping masks with CSS to create this fully interactive image mask with text masking for this interactive discovery site.



### HEAT DISTORTION EFFECT

[tympanus.net/Tutorials/HeatDistortionEffect](http://tympanus.net/Tutorials/HeatDistortionEffect)

This effect isn't that visible in the image, but SVG filters can be used by CSS and animated so that heat 'wobble' displacement effects can be added to images.



### BI-DIRECTIONAL SCROLLING

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

Unexpected scrolling always grabs the user's attention. Photographer Dean Bradshaw's website moves subtle background text horizontally across the screen as the user scrolls downwards.

**RACHEL ANDREW**

@rachelandrew

Rachel is a CSS working group expert and has been a strong advocate for web standards.

She is currently vocal in promoting the use of CSS Grid to design pages.

**ANA TUDOR**

@anatudor

Ana is an international speaker, covering workshops on cutting-edge CSS features.

You'll also find her demonstrating features of the CSS conical gradient.

**MARY LOU**

@crnacura

Mary is constantly working with CSS to develop unique user experiences, and these are often pushing the boundaries of what is currently possible. Many of her examples often show up on award-winning sites.

**JAMIE COULTER**

@jamiecoulter89

Jamie pushes CSS to its absolute limits by creating adventure games solely in CSS, without any JavaScript - which sounds like an impossibility - and these games are insanely cute!

**JEN SIMMONS**

@jensimmons

Jen is Designer Advocate at Mozilla, responsible for creating the Grid Inspector. She teaches all over the web and at conferences on the power of CSS for design.

# WHAT'S HOT RIGHT NOW

Here's three CSS properties for you to try today

## DESIGNING FOR DISPLAY CUTOUTS

[developers.google.com/web/updates/2018/09/nic69](https://developers.google.com/web/updates/2018/09/nic69)

As the advent of phones with 'cutout' sections or 'notches' on the display rises, we as web designers have to think about what that means for our designs on those pages. Currently web browsers on these devices add an extra margin to accommodate these cutouts. Web designers often like to use the full screen, and now there is a way to do that, with Chrome leading the way. Add this meta tag to the document:

```
<meta name='viewport' content='initial-scale=1,
viewport-fit=cover'>
```

Then use the CSS safe environment variables to layout your content:

```
.content {
  padding: 16px;
  padding-left: env(safe-area-inset-left);
  padding-right: env(safe-area-inset-right);
}
```

Now you just have to test your design in landscape and portrait on the device.

The screenshot shows a blog post titled "Display cutouts (aka notches)". It includes a sidebar with a list of posts and a main content area with code snippets and explanatory text. A diagram of a smartphone with a notch is shown.

## CSS CLIP PATH

[codepen.io/bbx/pen/GYLOZd](https://codepen.io/bbx/pen/GYLOZd)

The CSS clip path is one of those properties that has really helped push visual effects. Originally this was CSS clip, and is used to clip out part of the element. There are a number of ways to use this, including inset, which would inset the element by however many pixels on each side:

```
clip-path: inset(10px 20px 30px 40px);
```

There is also polygon, which would enable multiple points to be added so some more obscure shapes can be made. Circle and ellipse are fairly straightforward, but it's also possible to reference a path from an SVG image as well:

```
clip-path: url(path.svg#c1);
clip-path: polygon(5% 5%, 100% 0%, 100% 75%, 75% 75%, 75% 100%, 50% 75%, 0% 75%);
clip-path: circle(30px at 35px 35px);
clip-path: ellipse(65px 30px at 125px 40px);
```

The screenshot shows a login interface with a purple hexagonal input field for the password. The input field has a white border and rounded corners. The background of the page is a gradient from teal to blue.

## CSS CONIC GRADIENTS

[leaverou.github.io/conic-gradient](https://leaverou.github.io/conic-gradient)

The conic gradient is very similar to a radial gradient. However, you can't create a colour wheel in radial gradient, but you can do that with a conical gradient. Where radial gradients radiate between points, the conical moves around a circle and blends between colours. At first you may think this isn't really anything special, but with a little creativity to add stops in the right places there are some great results that can be achieved. At present this can only be achieved using a polyfill. Check out Ana Tudor's work with conical gradients.

The screenshot shows a 3D hexagonal prism with a conical gradient applied to its faces. The gradient transitions through various shades of grey and black. The background is a light grey.

**“**

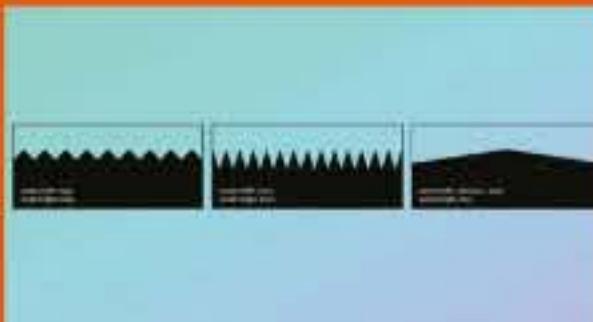
Its plan is to introduce a new series of APIs that will open up the power of CSS to the browser in JavaScript for the first time. This means that CSS features can be coded by the designer or developer and have all the native speed of regular CSS

## WHAT IS HOUDINI?

Every web designer has probably tried to implement a new feature of CSS but has been unable to because browser support isn't there. Houdini is here to end that

Houdini is a new W3C task force, and its goal is to take away the problem of poor support of new features on CSS. Its plan is to introduce a new series of APIs that will open up the power of CSS to the browser in JavaScript for the first time. This means that CSS features can be coded by the designer or developer and have all the native speed of regular CSS. All of this might sound a little like overkill, to suddenly enable designers to extend CSS, but it will allow for consistent coverage of a new CSS feature and reduce the need for complicated polyfills that aim to backfill on older browsers where there is spotty coverage.

You may be wondering if this is all CSS, and the answer to that would be no. The way to think of this is that Houdini opens low-level JavaScript APIs for the browser's render engine. It's not ready for use yet, and in order to get access you will need to go to 'chrome://flags' in Chrome and enable 'Experimental Web Platform features' to use it.



The hello Houdini mini site has some good visual examples. Here you can see a sawtooth edge being used



Ana Tudor shows how Houdini can be used to animate, transform and create some sophisticated 3D animation



Google shows some of its use cases for Houdini. This example demonstrates parallax scrolling running as a native CSS command

# CSS APIs

## LAYOUT API

### MASONRY LAYOUT

[googlechromelabs.github.io/houdini-samples/layout-worklet/masonry](https://googlechromelabs.github.io/houdini-samples/layout-worklet/masonry)

This is one of the Google Chrome Lab samples. In this example a native layout is created in CSS for a masonry-style layout. The Chrome Labs site is a good resource, as all of the currently implemented APIs have source code and examples available at this site.

## ANIMATION API

### CONTROLLING ANIMATION WITH SCROLLING

[houdini.glitch.me/animation](https://houdini.glitch.me/animation)

This not only demonstrates the scrolling effect on the page but also shows exactly how it was created, with a tutorial explaining how the Animation API works. The page also contains customisable code to experiment with. Changing values will cause elements to scroll at different speeds on the page. The rest of the site has great resources for explaining Houdini APIs.

There are six APIs associated with what Houdini offers to the browser. Here three of those are explored

## PAINT API

### RIPPLE EFFECT

[youtube.com/watch?v=BX\\_qv2yKSUk](https://youtube.com/watch?v=BX_qv2yKSUk)

The ripple effect shown in this example is the kind of effect that is found natively on Android devices, where one part of something is tapped and the animation radiates from that point. The example is useful because it's on YouTube so is easy to see, and it demonstrates custom 'painting' to the browser renderer.



## TO FOLLOW

### CSS-TRICKS

@css

CSS-Tricks has long established itself as a website that showcases CSS features in great detail. Its Twitter account gives the latest, up-to-date news on CSS.

### RESPONSIVE DESIGN

@rwd

Ethan Marcotte wrote the first book on responsive design, and this Twitter account gives you all the CSS responsive information in one easy-to-find place.

### CSS AWARDS

@cssawards

Selecting the best websites that are developed with CSS from around the world, this twitter account gives a great information feed on what can be designed with CSS.

### THE CSS WORKING GROUP

@csswg

The CSS Working Group is the group that writes the CSS specifications for the World Wide Web Consortium (W3C) so that all internet browsers adhere to the same standards. It is a great resource on the latest CSS.

### CODROPS

@codrops

Codrops is a useful resource for all things web design-related, but more specifically codrops has a strong slant towards the design styling of CSS.



# SVG ON THE WEB

SVG support is finally where it should be. So if you're still using PNG for logos, icons or interact elements, it's high time you learn about the power of Scalable Vector Graphics



SVG is now widely supported on all major browsers and devices. They are super small, searchable, modifiable via code and scalable. They look great at all sizes and can be used just like images or inline right in your HTML

**RICHARD MATTKA**

Creative Director, Developer, Designer  
[richardmattka.com](http://richardmattka.com)

# WHY USE SVG?

THERE ARE A HOST OF REASONS TO IMPLEMENT SVG. FIND OUT WHAT THEY ARE

When it comes to logos, icons, interface elements and vector-based artwork that you'd like to look crisp, animate or make responsive, there are plenty of reasons to use SVG.

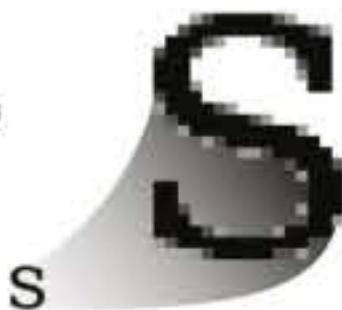
## 1 PIXEL-PERFECT SCALING

Using PNGs or rasterised images means you are restricted to pixels. It also means you are likely to have multiple images for various retina devices or resolutions. SVG is not affected by resolution or whether or not you have a retina display. SVG images scale and always look perfect because they are drawn with code, not pixels.

## 2 MODIFIABLE

SVG can be animated and styled with CSS. You can change colours, sizes, fonts and so much more. Elements within the SVG can be responsive as well to user interaction. Animations that you use on HTML elements can also be used on SVG elements.

Raster  
.jpeg .gif .png



Vector  
.svg



## 3 SMALL FILE SIZE

PNGs can get large very quickly, especially when you need high-definition. The larger the file size, the longer it takes to load and to render. JPGs are better but also large. That means slower sites. SVG is just code, which means very small file sizes. Use it for icons, logos and anything that can be created as vector art.

## 4 ACCESSIBILITY

SVG files are text-based and can be searched and indexed. This makes them readable by screen readers, search engines and other devices. Each element within the SVG is also accessible for search but also manipulation.

With a desire for fast, responsive experiences, SVG can make dramatic improvements to your site. There are several great times to use SVG

# WHAT IS SVG USED FOR?

LOGOS, UI ELEMENTS, ICONS AND ARTWORK ALL FIT THE PROFILE

SVG is heavily underused, partly because it took a long time for widespread support across browsers. With a desire for fast, responsive experiences, SVG can make dramatic improvements to your site. There are several great times to use SVG.

## LOGOS AND ICONS

Logos and icons often share a need to be clear and sharp at any size. Whether it's a simple button or a large, screen-filling billboard, you want the quality consistent and crisp. SVG enables precise control of every shape, line and element. SVG icons and logos are more accessible, which means they're easier to position and manipulate in real-time.

## CHARTS, GRAPHS, INFOGRAPHICS

SVG can be updated dynamically to plot data based on user actions or other data-driven events. This makes SVG perfect for things like interactive maps or graphics that illustrate data.

## VISUAL EFFECTS

Many real-time visual effects can be created using SVG, including shape morphing and transitioning from one shape to another. You can transform letters into shapes, logos into other logos and much more.

## ANIMATIONS

SVG can work with CSS animations, so you don't have to learn new animation skills. SVG can also use its own built-in SMIL animation ability, which is even more robust if you want to make really creative results.

## TRADITIONAL DRAWINGS & ILLUSTRATIONS

Many traditional illustrations translate really well to SVG, provided they are not overly complex in details. Sketches and diagrams to explain features of a product or illustrate a concept can work very well.

## INTERFACES AND APPLICATIONS

SVG is ideal for complex interfaces that can be integrated with web-based applications and rich internet applications (RIAs). They are lightweight, easy to animate, and each element can capture user interaction events.

# BROWSER SUPPORT

MOST MODERN WEB BROWSERS SUPPORT SVG. GREEN IS GOOD

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android
6-8		2		3.1				2.1-2.3			
9-10	12-16	3-62	4-69	3.2-11.1	10-55	3.2-11.4		3-4.3			
11	17	63	70	12	56	12	ALL	4.4-4.4.4	7	12-12.1	69
18	64-65	71-73	TP						10	46	

# ADD SVG TO THE PAGE

THERE ARE SEVERAL WAYS YOU CAN USE SVG ON YOUR WEBPAGES. LET'S HAVE A LOOK AT HOW A FEW OF THESE WORK

## USE THE <IMG> TAG

To embed an SVG via an `<img>` element, you just need to reference it in the `src` attribute as you would with any image. You should define a height and width as well, the same as when you add images. Here is an example of what the HTML code looks like:

```

```

The advantages to this method are that it is very quick and easy to add. You can also wrap it in an `<a>` tag to make it a hyperlink like a regular image. You can use CSS to affect the SVG image the same as any other, but you cannot use JavaScript or external CSS to manipulate the content of this image. You can still use inline CSS in the SVG code itself if you edit the SVG code.

## USE SVG IN CSS

SVG images can be used as `background-image` in CSS as well, just like PNG, JPG or GIF images. Here is an example of including an SVG image in CSS:

```
.element { background-image: url(/images/
image.svg); }
```

The advantages of SVG, such as crispness at any size, are retained with this method. You can also do anything a raster graphic can do, like repeat, scale, position and more.

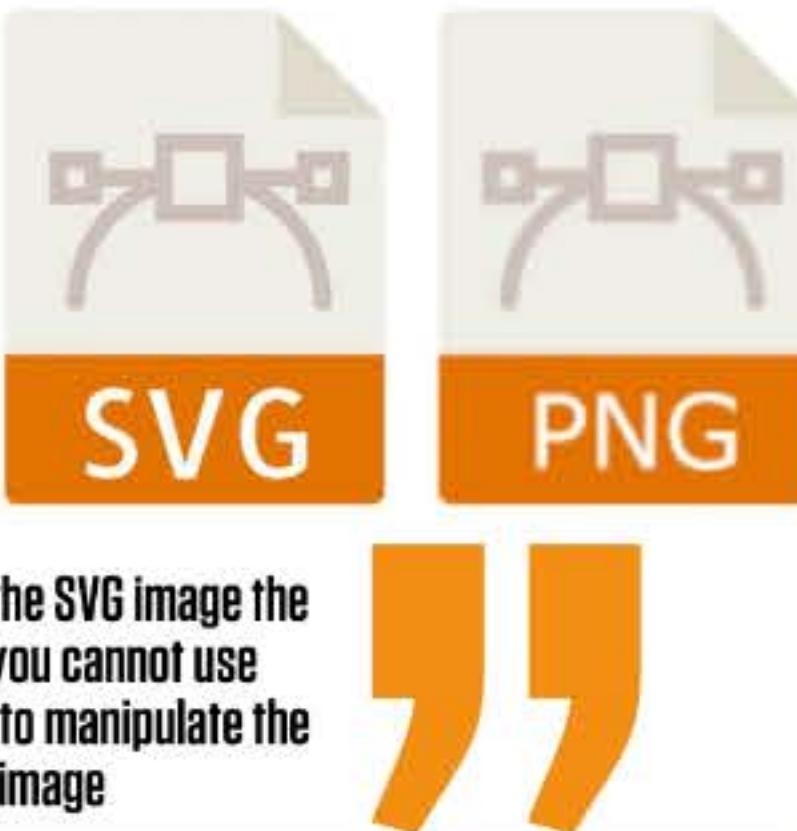
**You can use CSS to affect the SVG image the same as any other, but you cannot use JavaScript or external CSS to manipulate the content of this image**

## INLINE SVG IN HTML

You can also open up the SVG file in any text editor, copy the SVG code and paste it into your HTML document. We call this **inlining SVG** or using it inline. SVG code elements begin and end with the `<svg></svg>` tags. Here is a simple example of what you could paste into your document:

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="black"
  stroke-width="4" fill="red" />
</svg>
```

Putting your SVG inline saves HTTP requests and therefore can reduce your loading time. You can assign classes and ids to SVG elements and style them with CSS, either within the SVG or wherever you put the CSS style rules for your HTML document. In fact, you can use any SVG presentation attribute as a CSS property.



## CODING SVG

YOU CAN CODE SVG IN ANY TEXT EDITOR

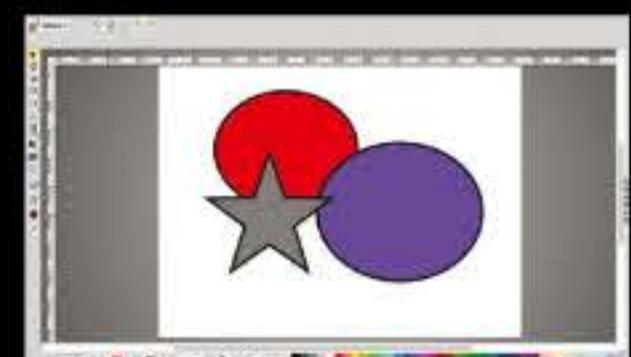
SVG uses an XML-based language for describing vector images. It's a markup language, like HTML, except that you have a wide variety of elements for defining the shapes to create these images, along with effects to apply to these shapes as well.

You can code SVG right along with your HTML in your favourite code editor. You can code them as separate files and include them, or you can code them inline in your HTML. As a simple example, the following code creates a circle and a rectangle:

```
<svg version="1.1"
baseProfile="full"
width="300" height="200"
xmlns="http://www.w3.org/2000/svg">
  <rect width="100%" height="100%" fill="black" />
  <circle cx="150" cy="100" r="90" fill="blue" />
</svg>
```

## APPS FOR SVG

Since SVG is a text file, you can use any text editor you like to edit SVG manually. There are also a number of great apps for quickly creating and working with SVG



### ADobe ILLUSTRATOR

[adobe.com/products/illustrator](http://adobe.com/products/illustrator)  
Illustrator is easy to use and creates beautiful results. However, the resulting SVG can have some quirks that make it necessary to post-process to be perfect.

### SVG-EDIT

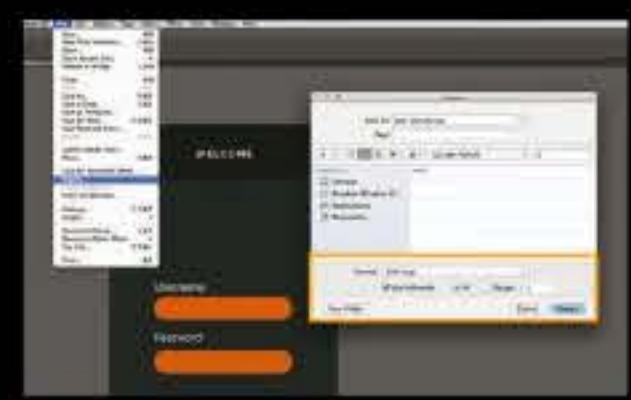
[svg-edit.github.io/svgedit/releases/latest/editor/svg-editor.html](http://svg-edit.github.io/svgedit/releases/latest/editor/svg-editor.html)  
SVG-edit is a fast, web-based, JavaScript-driven SVG drawing editor that works in any modern browser.

### INKSPACE

[wacom.com/en/products/apps-services/inkspace](http://wacom.com/en/products/apps-services/inkspace)  
Ink to vector - keep your creative ideas alive with this editable digital ink. Export your sketches and drawings to a scalable SVG format for further editing and enhancing for web and more.

### APACHE BATIK

[xmlgraphics.apache.org/batik](http://xmlgraphics.apache.org/batik)  
Batik toolkit is written in JavaScript and offers almost complete SVG support. Batik includes a viewer (Squiggle), a rasteriser for PNG output, an SVG pretty printer to format SVG files, and a TrueType-to-SVG-Font converter.



# MAKING SHAPES

SVG MAKES IT EASY TO CREATE A NUMBER OF BASIC SHAPES

## CIRCLES

There are three attributes to use to create a circle. They include:

- r - The radius of the circle.
- cx - The x position of the centre of the circle.
- cy - The y position of the centre of the circle.

The code to draw a circle is very easy to use. You can set the stroke colour and fill colour inline or separately as well. Here's an example:

```
<svg width="200" height="250">
  <circle cx="400" cy="300"
    r="200" stroke="red" fill=
    "transparent" stroke-
    width="5"/>
</svg>
```

## RECTANGLES

In a similar way to making circles, you can create rectangles, using x and y for position, then height and width attributes to define the size. The code looks like this:

```
<rect x="100" y="100" width=
  "400" height="300" stroke=
  "black" fill="transparent"
  stroke-width="5"/>
```

## LINES

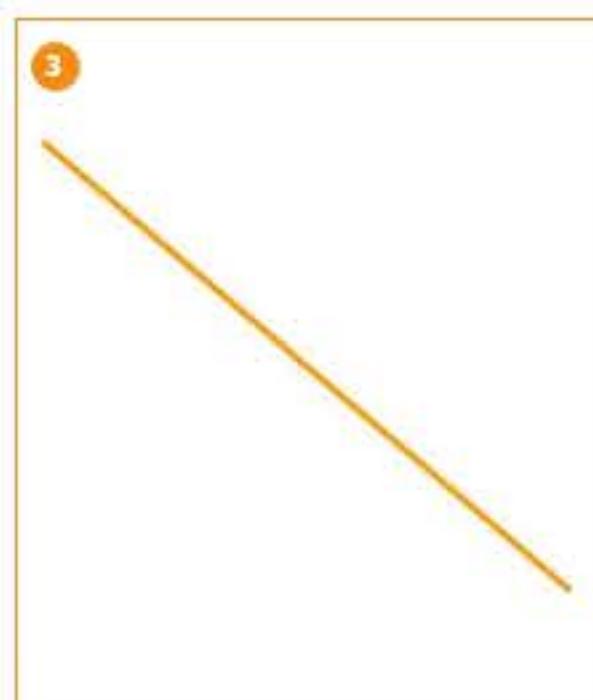
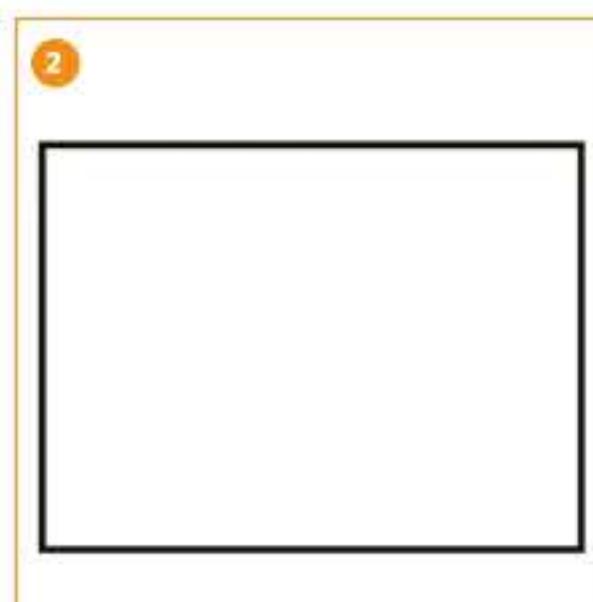
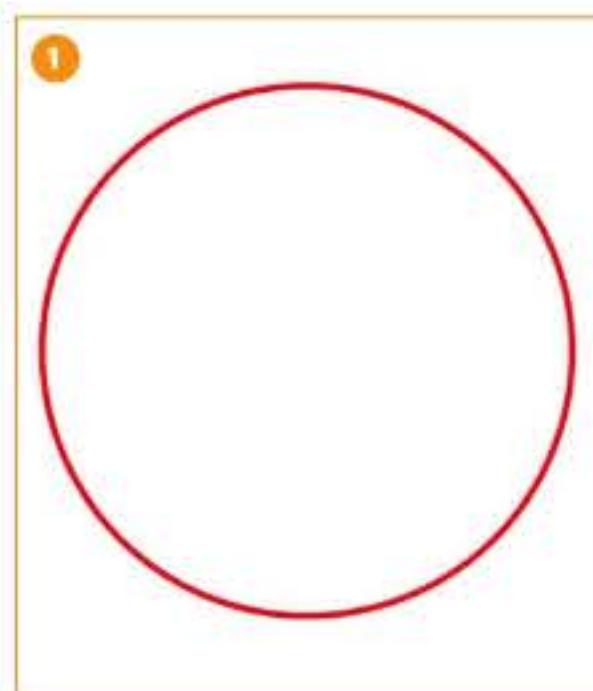
Lines in SVG are for straight lines. They take as attributes two points that specify the start and end point of the line. Here's an example:

```
<line x1="100" x2="500" y1=
  "110" y2="450" stroke="orange"
  stroke-width="5" />
```

## POLYGONS

You can also create any kind of polygon shape using the polygon element, which will conveniently return to the first point for you after the end of your list of points. Here's a quick example of drawing a polygon:

```
<polygon points="50 160, 55
  180, 70 180, 60 190, 65 205,
  50 195, 35 205, 40 190, 30
  180, 45 180"/>
```



# MAKING PATH & CURVES

The `<path>` element is extremely powerful for making shapes. You can use it to create lines, curves, arcs and more. You can even use it with text to create text that flows along paths.

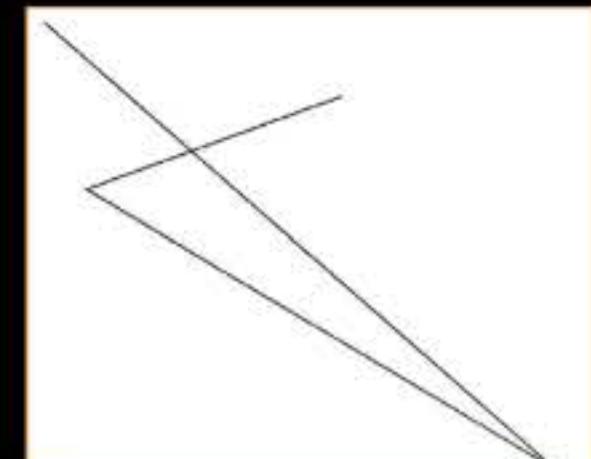
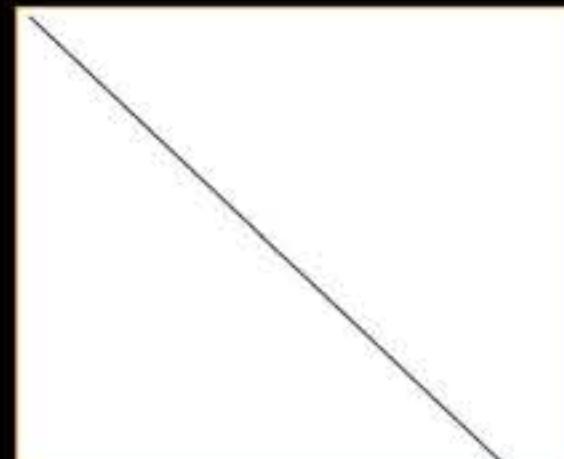
## DRAWING LINES

To draw a line or curve, you use the `path` element and 'define' the path using the 'd' attribute. You can 'move' the start of a line using the 'M' command for absolute positioning and the lower case 'm' for relative positioning. You use the 'L' command to draw a line to a new point. Here is a quick example of a line:

```
<path d="M 10,10 L 250,250" fill="transparent"
  stroke="black"/>
```

Here is an example of multiple lines, to show you how easy it is:

```
<path d="M 10,10 L 250,250 30,100 150, 50"
  fill="transparent" stroke="black"/>
```



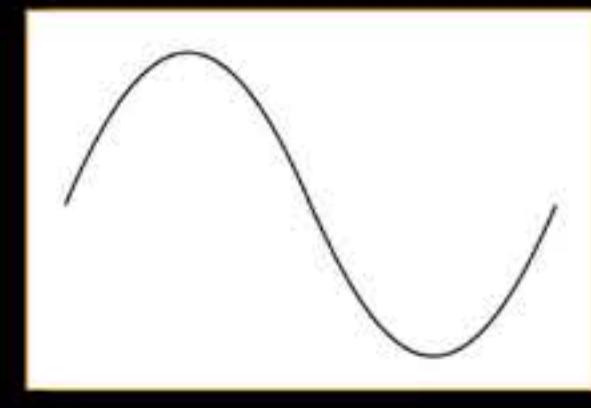
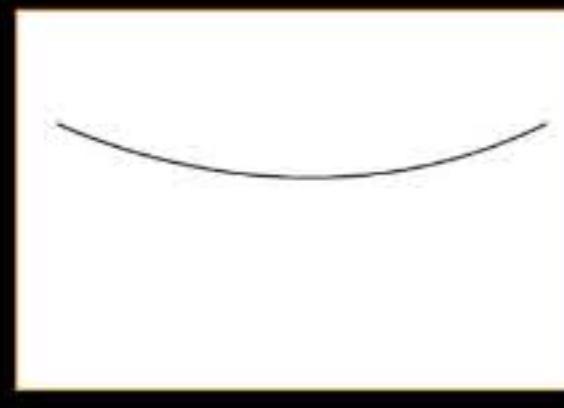
## DRAWING CURVES

You can also use the 'C' command to draw curves. Use the 'M' command to set the start position, and then list the start, the anchor point and the end point like this:

```
<path d="M50 50 C 50 50, 150 100, 250 50" stroke="black"
  fill="transparent"/>
```

You can also string curves together using the 'S' command to make even more complex curves like this:

```
<path d="M10 80 C 40 10, 65 10, 95 80 S 150 150, 180 80"
  stroke="black" fill="transparent"/>
```



There are plenty of curves and paths you can create. There is a great resource for learning even more about paths here: [developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths](https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths)

# MAKING TEXT & PATH

CREATE TEXT QUICKLY AND EASILY AND CUSTOMISE WITH A FEW ATTRIBUTES

Making text is easy in SVG. You use the `<text>` tag to define it. Here is an example:

```
<text x="10" y="10">
This is text.</text>
```

The `x` and `y` determine the position of the text. Like with the shape elements, text can be coloured with the `fill` attribute and given a stroke with the `stroke` attribute. You can even use gradients and patterns as strokes and fills.

## SETTING FONT PROPERTIES

Each of the following properties can be set as an attribute or via a CSS declaration: `font-family`, `font-style`, `font-weight`, `font-variant`, `font-stretch`, `font-size`, `font-size-adjust`, `kerning`, `letter-spacing`, `word-spacing` and `text-decoration`. Like the `<span>` elements `<tspan>` can also be used to select web sections of your text. A typical use-case might be to bold some text like this:

```
<text>
This is <tspan font-weight="bold" fill="red">bold and red</tspan>
</text>
```

This is **bold and red**

You can also set strokes, fills, rotations, direction and much more. Here is a quick example of setting the text to draw outlines only, and to switch the direction of the text:

```
<text x="350" y="50"
style="fill: none; stroke:
#000000; font-size: 48px;
direction: rtl; unicode-bidi:
bidi-override;">
This is text
</text>
```

This is text

txet si sihT

We use the `direction` style and set it to `'rtl'`, which means right to left. We also have to set the `unicode-bidi` style to `'bidi-override'`.

## USING TEXTPATH

This element uses the `xlink:href` attribute to connect to a path and aligns the characters along this path. This enables you to draw virtually any path you wish, and then attach your text to it, so it follows it. Here's an example:

```
<path id="my_path" d="M 40,40
C 200,100 10,100 520,200"
fill="transparent" />
<text>
<textPath xmlns:xlink=
http://www.w3.org/1999/xlink"
xlink:href="#my_path">
This Text is Curved
</textPath>
</text>
```

## RESOURCES

Learn more with these great resources

### MOZILLA DEVELOPER NETWORK (MDN)

[developer.mozilla.org/kab/docs/Web/SVG](https://developer.mozilla.org/kab/docs/Web/SVG)

A great resource for learning the ins and outs of SVG, with many examples to help get you up and running.

### CSS-TRICKS - EVERYTHING YOU NEED TO KNOW ABOUT SVG

[css-tricks.com/lodge/svg/](https://css-tricks.com/lodge/svg/)

CSS-Tricks hosts a complete SVG video tutorial series here. It covers what SVG is, why use it, how to use it and much more.

### ENVATOTUTS - HOW TO HAND CODE SVG

[webdesign.tutsplus.com/tutorials/how-to-hand-code-svg-cms-30368](https://webdesign.tutsplus.com/tutorials/how-to-hand-code-svg-cms-30368)

envatotuts has a nice in-depth tutorial that dives into many great examples of hand-coding SVG, including reusing SVG in your page.

## WHAT ABOUT SVG 2?

SVG 2.0 - a brand new SVG

SVG 2.0 is currently in candidate recommendation stage, which means it is not supported by browsers yet, and it may still be some time before it is. It boasts numerous improvements and features. It also removes or deprecates some features of SVG 1.1 and incorporates new features from HTML5 and Web Open Font Format.

SVG 2.0 removes several font elements, such as `glyphs`, and they are replaced by the WOFF font format. Also the `xml:space` attribute is deprecated in favour of CSS.

It reached candidate recommendation stage on 15 September 2016. The latest draft was released on 18 October 2018. You can read more about it and follow its progress here: [w3.org/TR/SVG2](https://w3.org/TR/SVG2)

“

SVG 2.0 is currently in candidate recommendation stage, which means it is not supported by browsers yet, and it may still be some time before it is. It boasts numerous improvements and features

## CSS & SVG

You can use CSS styles inline, inside SVG elements. Here is an example:

```
<svg width="350" height="60">
<text>
This is <tspan font-weight="bold" fill="red">bold and red</tspan>
</text>
<style><![CDATA[
text{
dominant-baseline: hanging;
font: 28px Verdana, Helvetica, Arial, sans-serif;
}
]]></style>
</svg>
```

You can also use CSS separately, just like with any other elements

Use inline or external CSS

on your page. For example, if you set a class attribute on your SVG shape, you can set its colour through regular CSS like this:

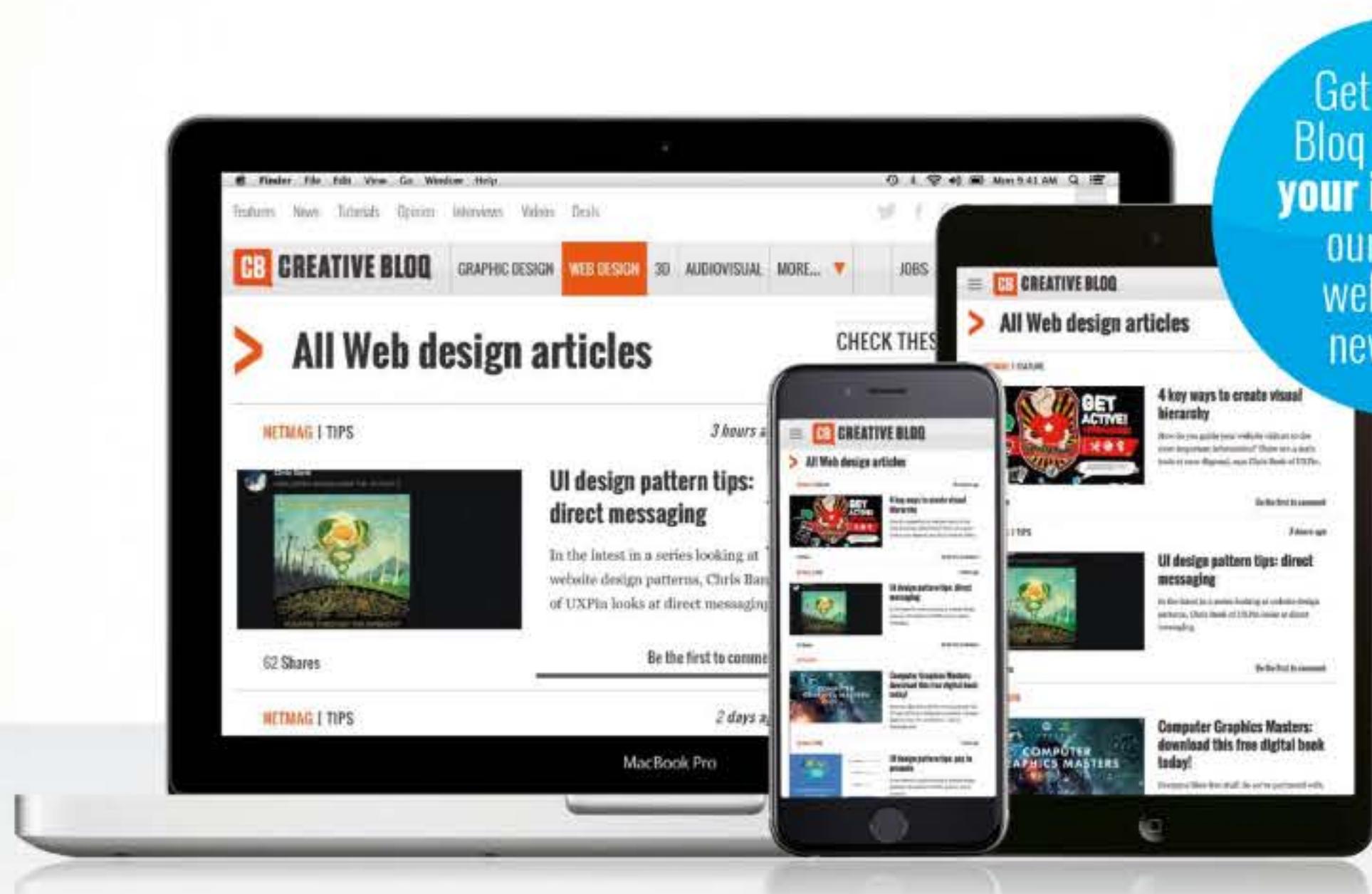
```
<svg class="logo" width="400" height="400" viewBox="0 0 400 400">
<rect class="box" x="0" y="0" width="400" height="400" fill="#56A0D3" />
</svg>
```

And the CSS would be:

```
.box { fill: red; }
```

Notice we had inline colouring that we then overrode with our CSS. Pretty cool!

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



Get Creative  
Blog **direct to your inbox** with our weekly web design newsletter

**CB** CREATIVE BLOQ

Graphic design

Web design

3D

Digital art

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

# Manage React application data with Redux

Create an API-powered state application and discover the benefits of a centralised data store



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



The appetite for single-page applications is greater than ever. Instead of a complete page reload each time the user clicks a link or fills a form in, we can provide real-time feedback using JavaScript and do the heavy lifting in the background. As developers, we have greater control over the data that gets transferred, and the user enjoys a faster, seamless experience - it's a win-win.

But without that constant connection back to the server, the state of the application has to live on the client-side. Frameworks like React are great at storing state at a component level but start to struggle when dealing with large sets of application-wide data. Explicitly passing that data through as props can clutter up the code and increases the surface area for bugs to appear.

Redux takes the state of an application and places it in one location. By combining with React, we can have the whole interface powered by a centralised data store. This makes an application more reliable, more testable, and when issues do arise they become easier to replicate.

This tutorial is the first in a five-part series where we will be building PhotoShare - an application that allows visitors to comment directly on a set of photos. While it seems simple on the outside, there is lots to keep track of, from photos to the users and their comments.

In this part of the tutorial we will cover the basic concepts around Redux, such as actions, reducers and the data store. Later in the series we will use these concepts to power interactions such as form filling or server communication.

## 1. Install dependencies

Before we get started, we need to download all the files needed to run this application. Inside 'package.json' we can see the dependencies, including React, Redux and the 'react-redux' package that deals with the connection between the two. Download the project files, then head

to the directory in the command line and install the dependencies. Yarn is required to manage those dependencies, if this hasn't been downloaded already. Then start the development server.

```
> yarn  
> yarn start
```

## 2. Initialise the store

The store is the heart of Redux. It holds the state of the application and orchestrates all communication to and from it. There should only ever be one store per application. Open up 'index.js' and create a store. This is done using the 'createStore' function from Redux. Pass in the root reducer, which we will come to later.

```
import { createStore } from "redux";  
import rootReducer from  
  "./reducers/root";  
const store = createStore(  
  rootReducer);
```

## 3. Pass in the store

Before React can use any data from the store, it needs to be given access to it. The 'react-redux' package has a <Provider> component that can provide access to any component that requests it. Import the <Provider> component and wrap it around the rendered content. Now we can start pulling in the state data wherever the application needs it.

```
import { Provider } from "react-redux";  
[...]  
ReactDOM.render(  
  <Provider store={store}>  
    <BrowserRouter>  
      <App />  
    </BrowserRouter>  
  </Provider>,  
  document.getElementById("root")  
)
```

## 4. Set up root reducer

A reducer is a function that knows how to update the store. The store only deals with one reducer. We can use Redux's 'combineReducers' method to split that into separate, smaller reducers that each have a specific role, such as storing information about the photos or the state of the interface. Our first reducer will hold the photos within the gallery. Open up 'reducers/root.js' and import that reducer in.

```
import { combineReducers }  
from "redux";  
import photos from "./photos";  
const rootReducer = combineReducers({  
  photos  
});  
export default rootReducer;
```

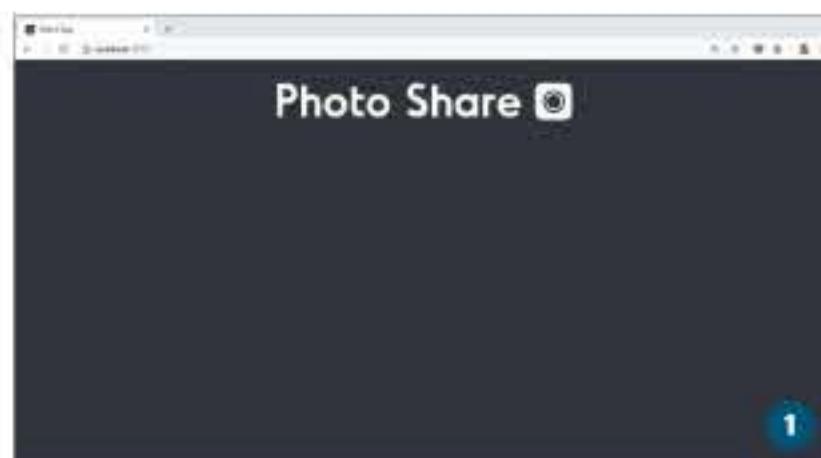
## 5. Create photos reducer

When a reducer is called, it gets passed the current state from the store and an object describing a change to the state known as an 'action'. The return value of the reducer is what then becomes the store's new state. Open up 'reducers/photos/photos.js' and create our first reducer. For now, every action will return the state straight back without any updates.

```
const initialState = {};  
export const reducer =  
(state = initialState, action) => {  
  return state;
```

## Use more than switch

Redux does not specify the shape of a reducer. While a switch statement is most common, use what makes sense for the application. The structure does not matter so long as the state object is returned.

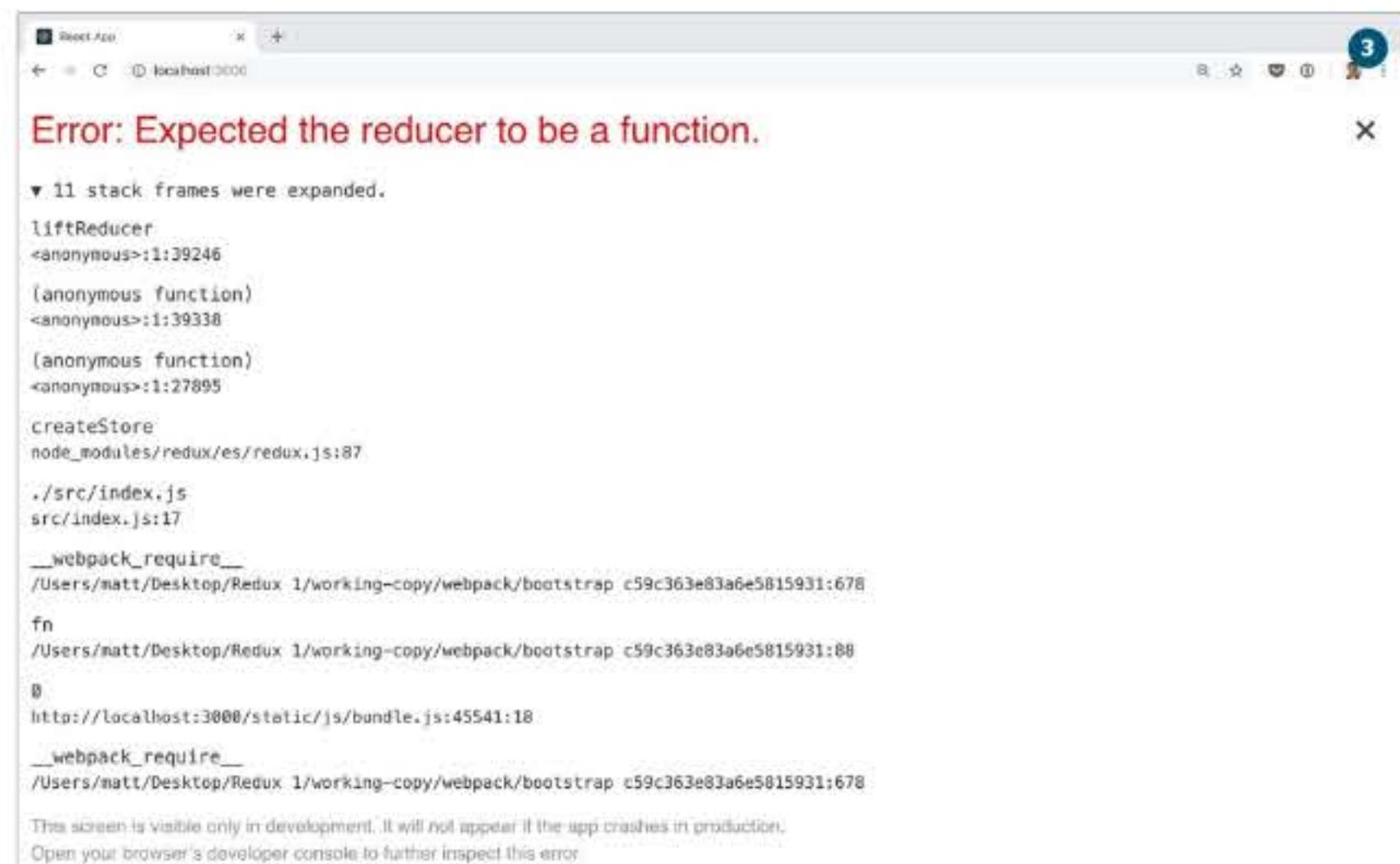


### Above

Starting the development server should also open up the application in the default browser. If it does not, head to <http://localhost:3000>

### Right

While developing the application it will display full-screen errors as well as the usual console ones. When built for production, these do not appear

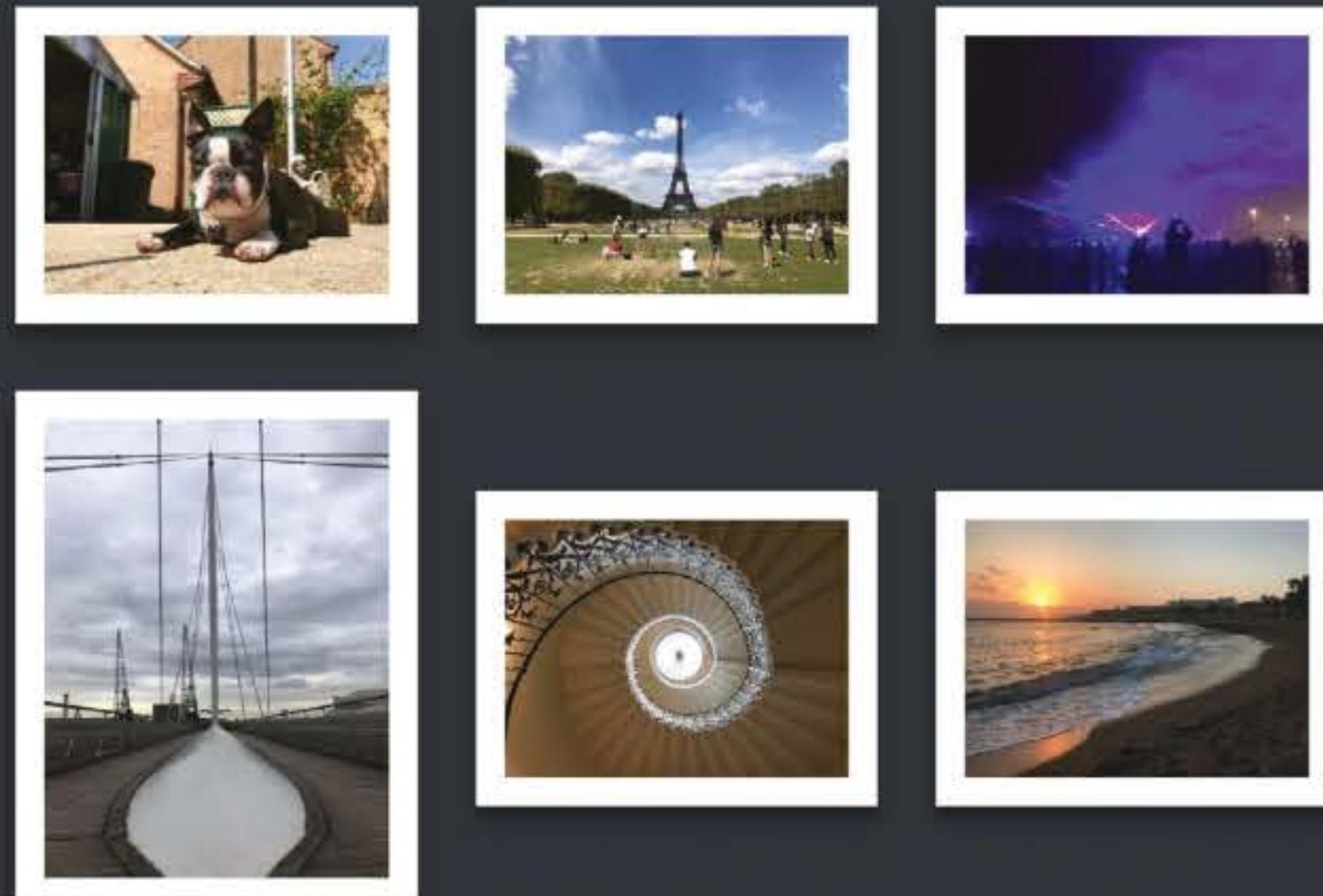


## Tutorials

Manage React application data with Redux

9

# Photo Share



```
};  
export default reducer;
```

## 6. Apply initial state

Each reducer is likely to have data stored in an expected shape, such as an array or an object to hold specific information about that part of the state. We can initialise it with a set of values from the outset. For this application we will pre-fill the photo data locally. Usually this would come from a remote server and would initially be empty. Import the example response and provide it as initial state to the reducer.

```
import exampleResponse from  
"../../example-response";  
const initialState = exampleResponse;
```

## 7. Select the photos

While not strictly required by Redux, selector functions are a useful way to get exactly what you need out of the store. They filter out the rest of the data to leave only what is needed for each component. Open up 'selectors/

```
photos/photos.js' and create a selector that gets all the photos for the gallery.
```

```
export const getGalleryPhotos =  
state => Object.values(state.photos);
```

## 8. Connect with Redux

For React components to access the store we created in step 3, we need to connect them to Redux. To do this we use 'connect' from the 'react-redux' package. 'connect' is an example of a higher-order component - an established React pattern that provides functionality to a passed component. Head to 'components/container/Gallery/Gallery.js', import 'connect' and use it to replace the exported component.

```
import { connect } from "react-redux";  
[...]  
export default  
connect()(GalleryContainer);
```

## 9. Map state to props

There are two parameters we can pass to 'connect'. The first is a special function that allows us to pass parts of the state from Redux into components. The 'connect' function gets the state from the store and passes it to selectors. Selectors then get what they need and return it to the component as a prop. Create this 'mapStateToProps' function and use it to pass the 'photos' prop into <GalleryContainer>. We can now see the images start to appear in the gallery.

```
import { getGalleryPhotos }  
from "../../selectors";
```

```
export const mapStateToProps =  
state => ({  
photos: getGalleryPhotos(state)  
});  
export default connect(mapStateToProps)  
(GalleryContainer);
```

## 10. Select a photo

With the gallery complete, we still need a way to view specific photos in more detail. Clicking a photo in the gallery shows us a <Photo> component on its own page, but it's currently blank. Create another selector in 'selectors/photos/photos.js' to select a specific photo. We can re-use the logic of 'getGalleryPhotos' like we would any other function.

```
export const getPhoto = (state, id) =>  
getGalleryPhotos(state)  
.find(photo => photo.id === id);
```

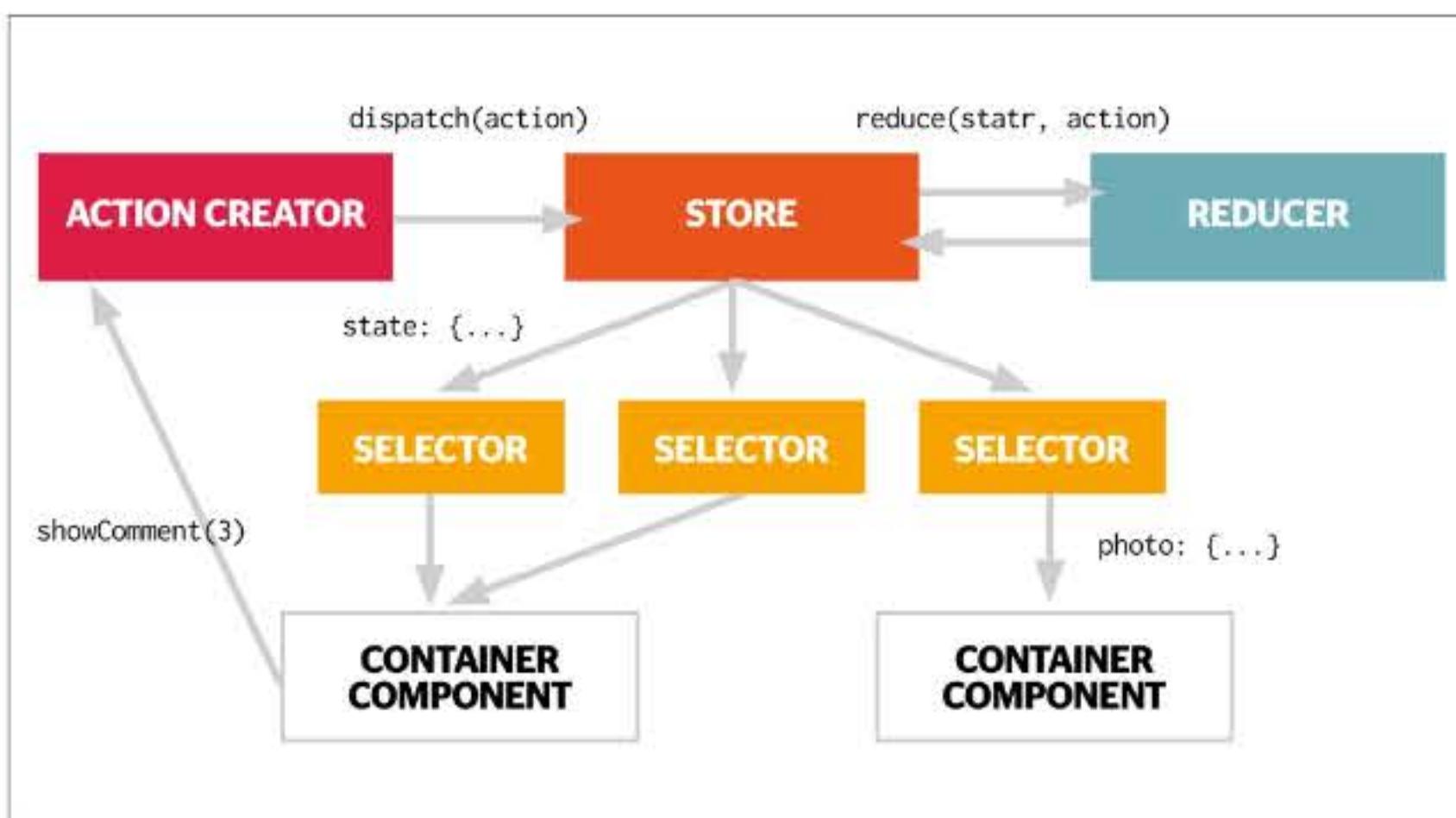
## 11. Link single photo

As this component is not linked to <Gallery>, we need to connect it to Redux separately. We can use the same approach to pull information out using 'mapStateToProps' and the new selector. Open 'components/container/Photo/Photo.js' and connect this component to the store.

```
import { connect } from "react-redux";  
import { getPhoto } from  
"../../selectors";  
export const mapStateToProps =  
(state, props) => ({  
photo: getPhoto(state)
```

## How many connections?

Connecting too many components couples an application closer to Redux, while too few re-renders parent components unnecessarily. It's important to find a balance that works best in each application.

**The Redux data lifecycle**

At its core, Redux is a set of functions and objects with specific jobs to do. By ensuring each part stays separate and communicates in a defined way, Redux can help keep the state of an application up to date all from one place.

The application's state lives within the store. We can use selectors to pick out the relevant parts of that state and pass it on to the components that need it.

The only way to update the data in the store is through an action, which is typically triggered by a user. The action creators make objects based on parameters, which are then passed to each reducer connected to the store. They check the action, perform some modification to the state and return a new copy for the selectors to use.

Each part of this process can be built upon and expanded without affecting the rest. This means that as the application grows the data store can scale with it.

```
});  
export default connect(  
  mapStateToProps)(PhotoContainer);
```

**12. Pass through ID**

The selector still requires an ID to get the correct photo from the store. The 'mapStateToProps' function has a second argument that passes through the component's current prop values. The routing within the application provides the ID of the photo, which is passed as an 'id' prop. Update 'mapStateToProps' to use this new ID. Clicking on the first photo in the gallery now shows a larger version.

```
export const mapStateToProps =  
(state, props) => ({  
  photo: getPhoto(state, props.id)  
});
```

**13. Show comment**

In this application, users can leave comments on a specific part of the photo. These will be displayed as an overlay on the image itself. Open up 'components/presentational/Photo/Photo.js' and start adding comments after the <img> component. Pass all the data it needs as separate props.

```
import Comment from  
"../../container/Comment";  
[...]  
{comments.map(({comment, id, left,  
top, user}) => (  
  <Comment comment={comment}  
    id={id} key={id} left={left}  
    top={top} user={user}  
/>>))}
```

**14. Hide other comments**

Inside the <Comment> component, the 'comment\_box-open' class is causing all of the comments to display at once. We can use Redux to only show the active comment. Head to the component at 'components/presentational/Comment/Comment.js'. Update the

className to only show up when the 'isCommentOpen' prop is true.

```
<div className={`comment_box  
comment_box--${direction} ${  
  isCommentOpen ? "comment_box--open"  
    : ""  
}>  
>
```

**15. Create UI reducer**

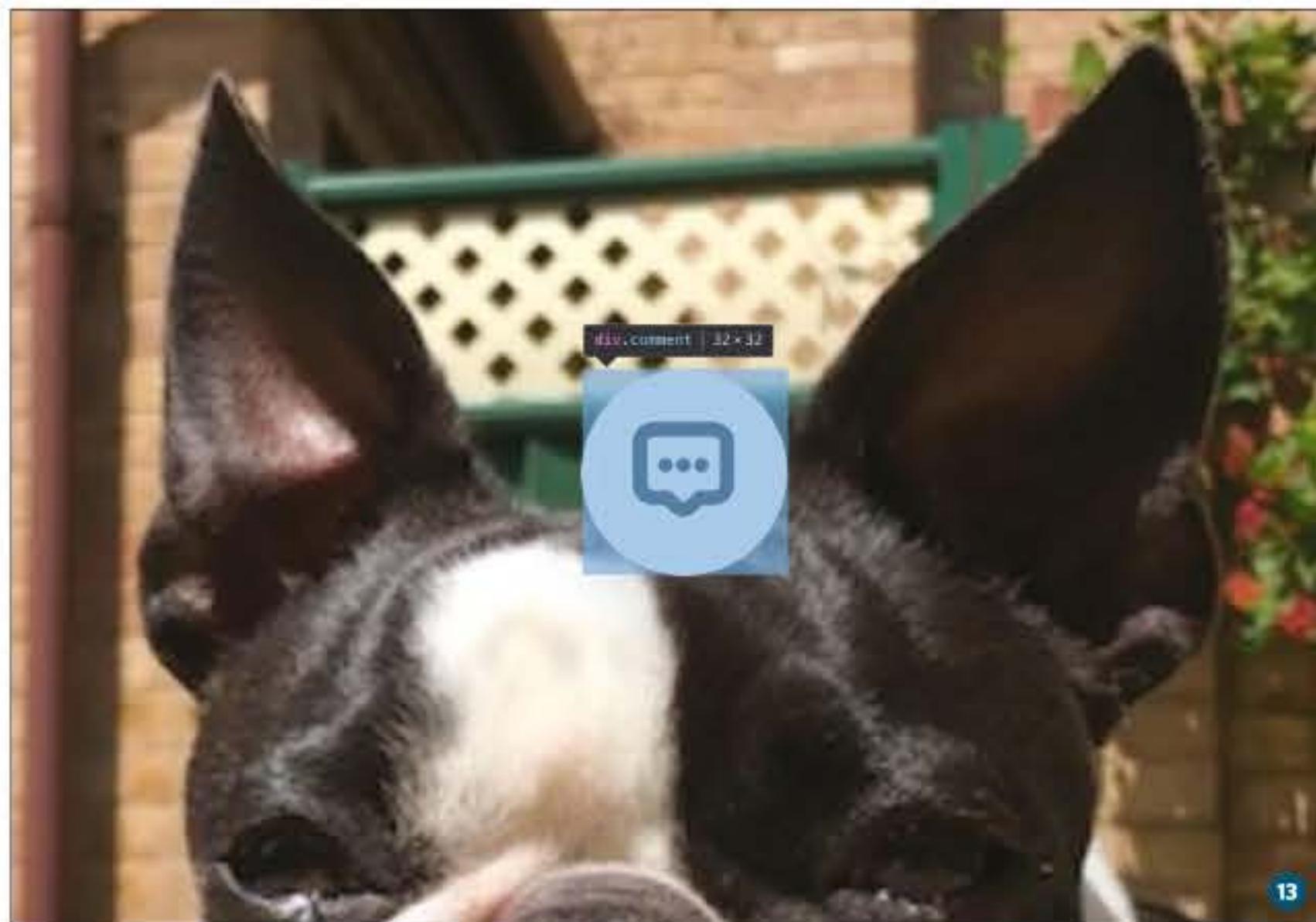
We need to record which comment is currently open in the store. As it's not directly related to a photo, we should create a new reducer to handle this part of the state. Open 'reducers/ui/ui.js' and create a reducer similar to the one for photos.

```
const initialState = {
```

```
  commentOpen: undefined  
};  
export const reducer = (  
  state = initialState, action) => {  
  switch (action.type) {  
    default: return state;  
  }  
};  
export default reducer;
```

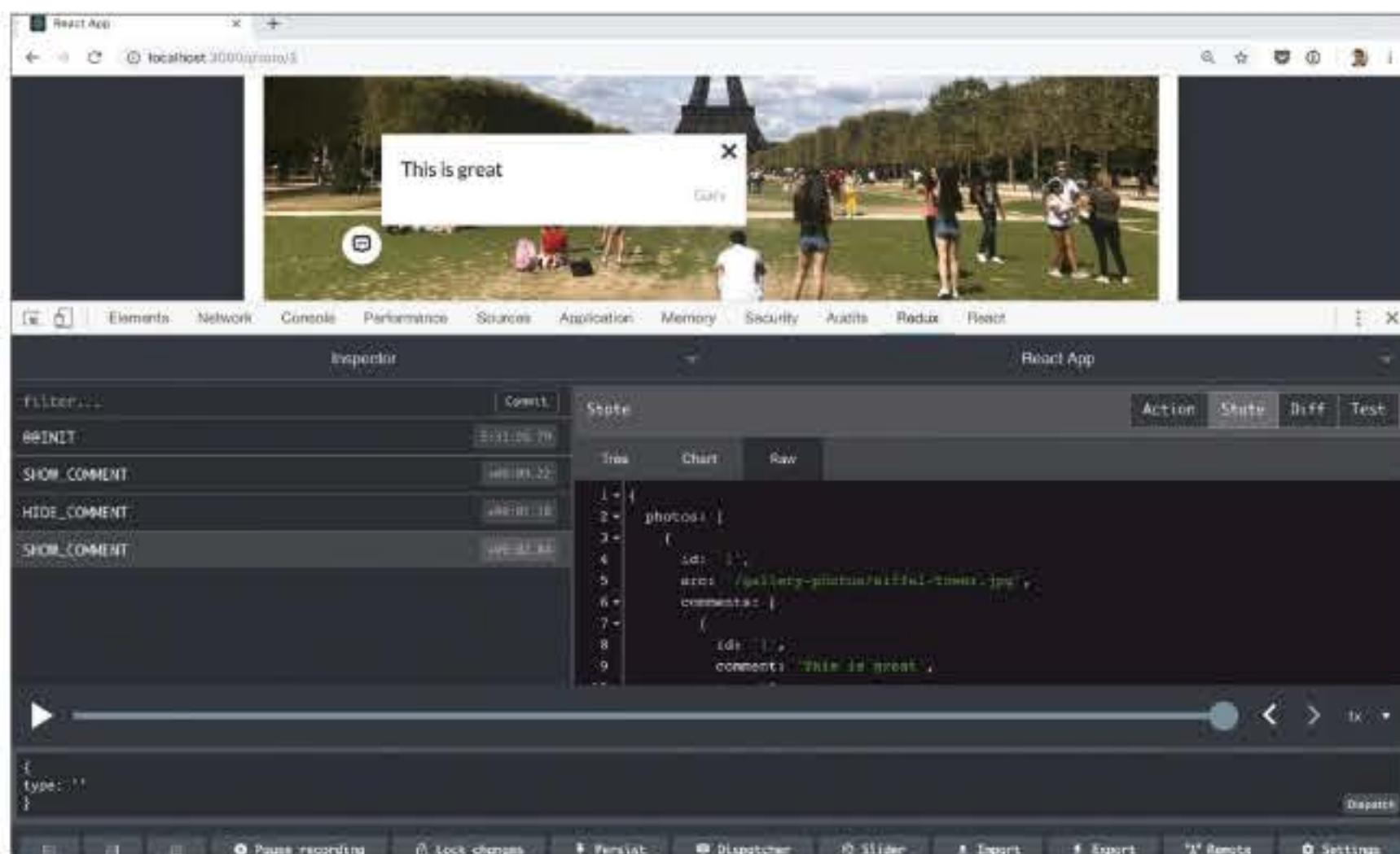
**16. Link the reducer**

Before Redux can access our new reducer, we need to pass it through to the root reducer. Go back to 'reducers/root.js' and add the new UI reducer into the call to 'combineReducers'. Any UI data inside the store can now be accessed under the 'ui' property.



## Tutorials

### Manage React application data with Redux



### Debugging with Redux DevTools

With Redux quietly running in the background, it can be difficult to know exactly what's going on. We could log out each new state as an action comes in, but this seems unsuited and noisy.

Thankfully there is a browser extension available that can tap into a Redux store and even manipulate it in real time. Not only can it track actions and show the current store state, but also undo those actions to help see how the application changes.

It taps into the store through an 'enhancer' - a module with access to the API behind Redux to extend its behaviour. It is recommended that this enhancer is removed when building the application for production.

The extension is available for Chrome and Firefox. For more information head to [github.com/zalmoxisus/redux-devtools-extension](https://github.com/zalmoxisus/redux-devtools-extension)

```
import ui from './ui';
const rootReducer = combineReducers({
  photos,
  ui
});
```

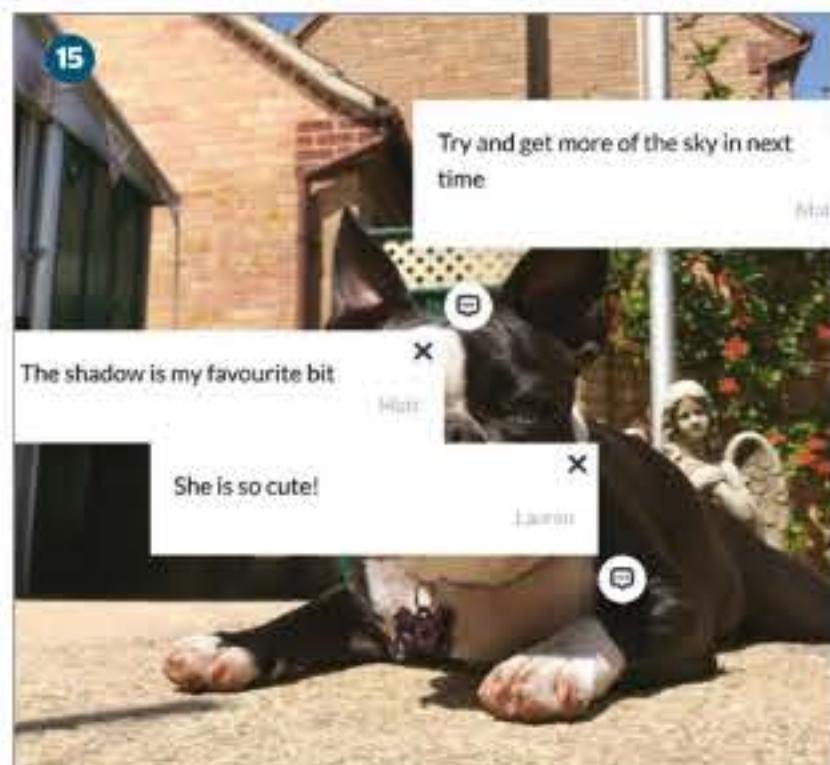
### 17. Create an open selector

With the UI state now in place, we can check to see which comment is open. Go to 'selectors/ui/ui.js' and add an 'isCommentOpen' selector. By passing the ID of the current comment, we can check it against the state. If the comment is open, then the selector returns true and the comment appears.

```
export const isCommentOpen =
  (state, id) =>
    state.ui.commentOpen === id;
```

### 18. Connect the comments

Now everything is set up, we can connect each <Comment> component to the store, like with <Gallery>. Head over to 'components/container/Comment/Comment.js' and connect everything up like before. Pass the new selector inside 'mapStateToProps'.



```
import { connect } from "react-redux";
import { isCommentOpen } from
  "../../../../selectors";

export const mapStateToProps =
  (state, props) => ({
    isCommentOpen: isCommentOpen(
      state, props.id)
  });

export default connect(mapStateToProps)
(CommentContainer);
```

### 19. Toggle comment actions

To tell Redux which comment should be visible, we need to dispatch an action to the reducers. By using action creator functions, we can be sure that the action object looks the same each time. Add a couple of action creators in 'actions/ui/ui.js' to show and hide a comment by ID. We use constants for the type so they can match up with the reducer later on.

```
import { HIDE_COMMENT, SHOW_COMMENT } from
  "../../../../constants/actions";
export const hideComment = () => ({
  type: HIDE_COMMENT
});

export const showComment = id => ({
  type: SHOW_COMMENT,
  payload: id
});
```

### 20. Update the UI reducer

When either action is dispatched it ends up passing through the UI reducer we created earlier. By using the 'type' property in the action, we can match an action to a change in the store. Add cases for the actions in 'reducers/ui/ui.js'. These will either record or clear the open comment ID.

```
import { HIDE_COMMENT, SHOW_COMMENT } from
  "../../../../constants/actions";
...
```

```
case HIDE_COMMENT:
  return {
    ...state,
    commentOpen: undefined
 };
```

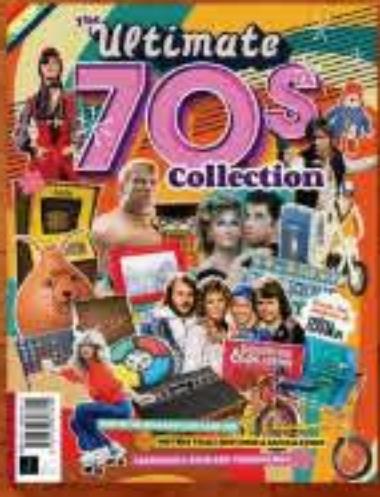
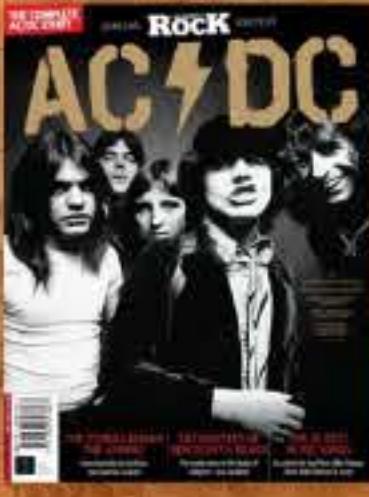
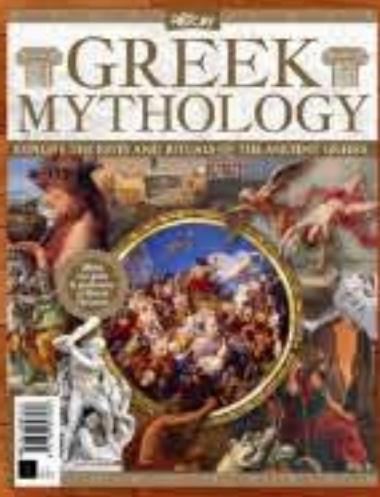
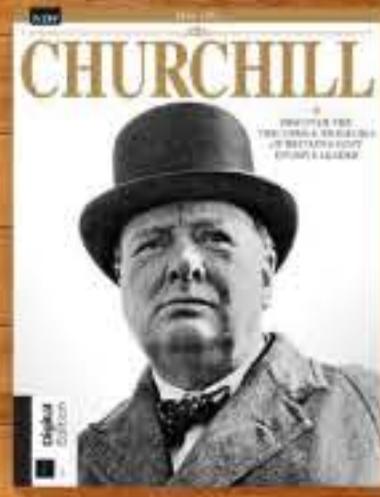
```
case SHOW_COMMENT:
  return {
    ...state,
    commentOpen: action.payload
 };
```

### 21. Dispatch new actions

The 'connect' higher-order component takes a second function, which gives access to the 'dispatch' method from Redux. In short, this 'mapDispatchToProps' function works similar to 'mapStateToProps' but allows us to dispatch actions. Back in 'components/containers/Comment/Comment.js', hook up 'mapDispatchToProps' to pass through functions that can be called when a comment is clicked.

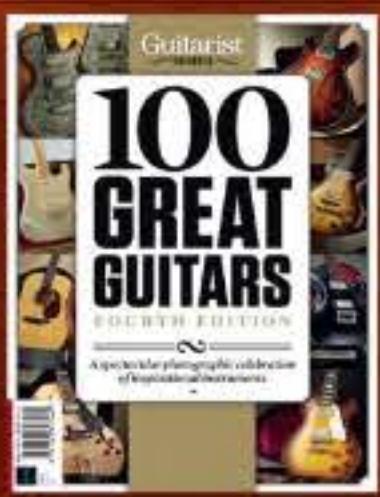
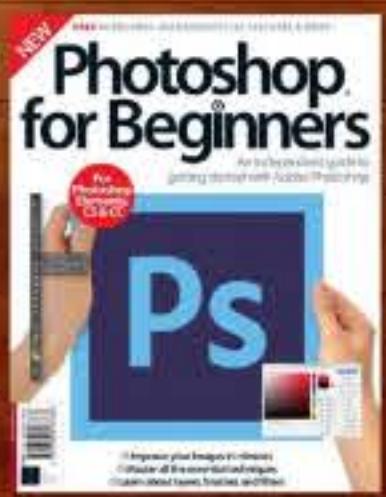
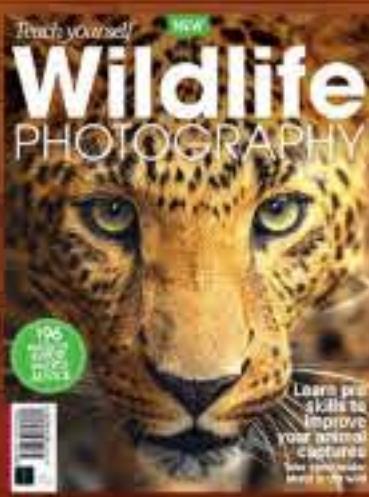
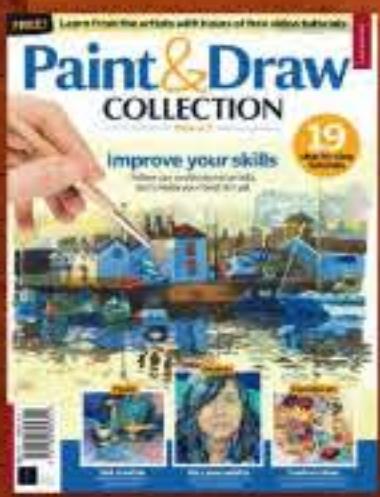
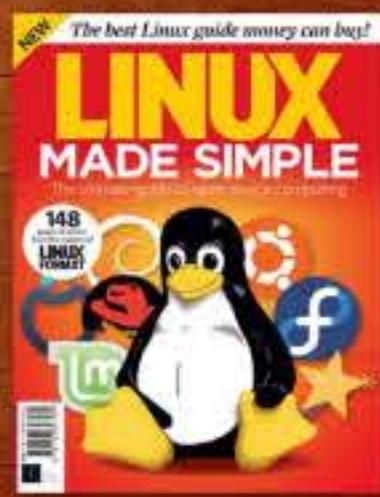
```
import { hideComment, showComment } from
  "../../../../actions";
export const mapDispatchToProps =
  (dispatch, props) => ({
    hideComment: () =>
      dispatch(hideComment()),
    showComment: () =>
      dispatch(showComment(props.id))
  });

export default connect(
  mapStateToProps,
  mapDispatchToProps
)(CommentContainer);
```



## 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](http://www.myfavouritemagazines.co.uk)  
Magazines, back issues & bookazines.

**Get your listing in our directory**  
To advertise here call the ads team on 01225 442244

# HOSTING LISTINGS



## Featured host: Netcetera

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

03330 439780

### About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide.

As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services designed to more effectively manage IT

infrastructures. A state-of-the-art data centre environment enables Netcetera to offer your business enterprise-level colocation and hosted solutions.

Providing an unmatched value for your budget is the driving force behind our customer and managed infrastructure services. From single server to fully customised data centre suites, we focus on the IT solutions you need.

### What we offer

- **Managed hosting** - A full range of solutions for a cost-effective, reliable, secure host.
- **Cloud hosting** - Linux, Windows, Hybrid and Private Cloud Solutions with support and scalability features.

- **Data centre colocation** - Single server through to full racks with FREE setup and a generous bandwidth.

- **Dedicated servers** - From QuadCore up to Smart Servers with quick setup and fully customisable.

### 5 tips from the pros

#### 1. Reliability, trust & support

Reliability is a major factor when it comes to choosing a hosting partner. Netcetera guarantees 100 per cent uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

#### 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](http://cwcs.co.uk)  
08001777000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and are passionate about what they do.

- Colocation hosting
- VPS
- 100 per cent network uptime

## UK-based hosting



[cyberhostpro.com](http://cyberhostpro.com)  
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](http://fasthosts.co.uk)  
08081686777

UK-based and operating 24/7 from dedicated UK data centres. Fasthosts keep over 1 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 ONLINE

[hetzner.com](http://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



[land1.co.uk](http://land1.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](http://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](http://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. Supercharge your digital projects today with STORM.

- 30-second WordPress install
- Deploy directly from GitHub
- Easy team management

## Flexible cloud servers

# elastichosts

[elastichosts.co.uk](http://elastichosts.co.uk)  
02071838250

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 call the ads team on 01225 442244*

# COURSE LISTINGS



## 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

## Featured: **Northcoders**

[northcoders.com](http://northcoders.com)  
Twitter: @northcoders  
Facebook: Northcoders

full-time bootcamp, or fit their course around your life with their 24-week part-time bootcamp. Their internal career support team will help find you work as a developer, setting up interviews with your choices of Northcoders Hiring Partners across the north of England.

## What we offer

- **Full-time:**  
Fast-track your career in just 12 weeks

- **Part-time:**  
Fit our curriculum around your life in 24 weeks

## 5 tips from the pros

### 1. Get started with coding

The best way to know if coding is for you is to just try it! We recommend the free, online JavaScript track of Codecademy to get you started with the basics.

for you, set aside a few evenings each week to really start making progress! If coding is for you, this should be fun.

### 4. Be prepared

We'll be with you every step of the way when you apply. Make sure you go through all the materials we recommend and ask for help if you're stuck.

### 5. Get social

With Northcoders, you're not just on a course, you're part of a community that will stay with you long after you graduate. Make the most of it!

### 2. Do your research

Make sure you read plenty of student reviews to make sure you're applying somewhere reputable. Read their blog and have a look at their social channels.

### 3. Throw yourself in

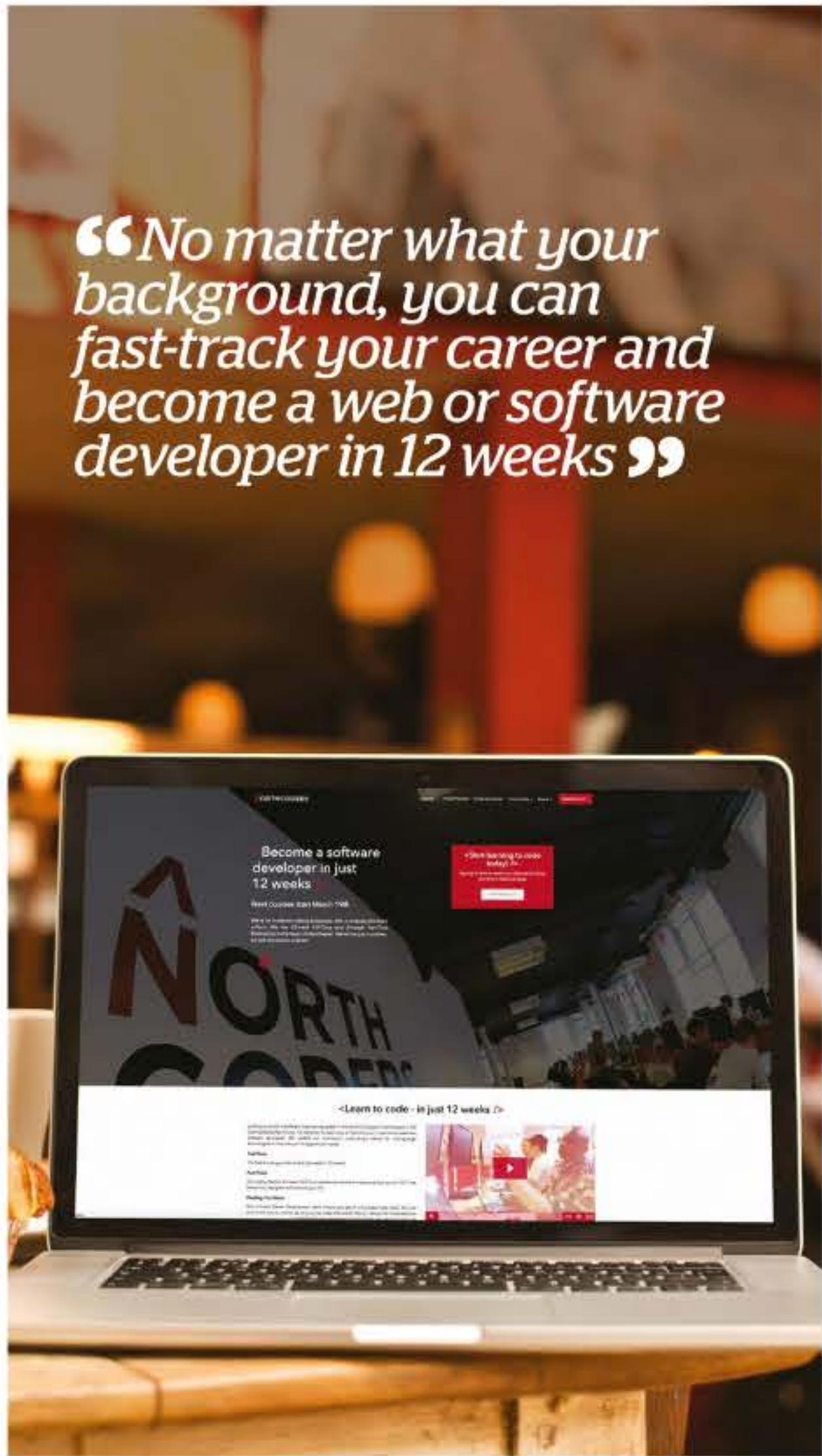
Once you've decided it's right



Becoming part of this vibrant, caring community was something I hadn't expected before the course, but now I couldn't be without it. To be a Northcoder is to be enlightened, inspired and supported

**Joanne Imlay**

*Primary school teacher to software developer at CareIcon*



Northcoders delivered their part of the bargain in spades. They provided tremendous assistance in turning me into the full product - a well-rounded, capable future tech employee - and they have the contacts to deliver the opportunities for such people.

**Joe Mulvey**

*Maths teacher to software developer at Auto Trader*

# udemy

## UDEMY

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

Twitter: [@udemy](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.



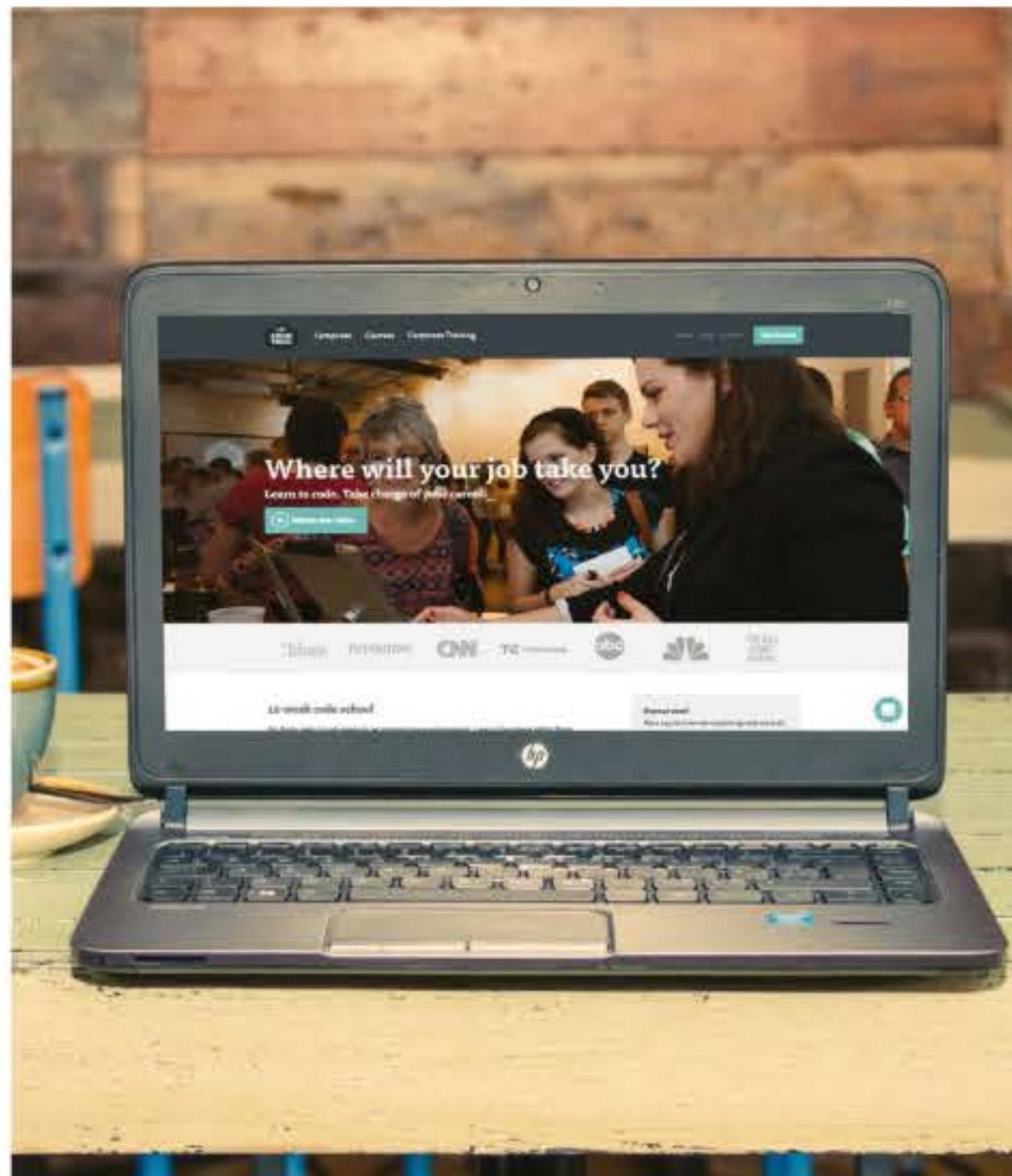
# THE IRON YARD

[theironyard.com](http://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 back-end engineering, front-end engineering, mobile engineering and design. The Iron Yard exists to create real, lasting change for people, their companies and communities through technology education. The in-person, immersive format of The Iron Yard's 12-week courses helps people learn to code and be prepared with the skills needed to start a career as junior-level software developers.

**12-week code school:** Learn the latest skills from industry pros.  
**Free crash courses:** One-night courses, the perfect way to learn.



# WE GOT CODERS

## WE GOT CODERS

[wegotcoders.com](http://wegotcoders.com)  
[hello@wegotcoders.com](mailto:hello@wegotcoders.com)

We Got Coders is a consultancy that provides experts in agile web development, working with startups, agencies and government. Take one of their 12-week training 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

[futurelearn.com](http://futurelearn.com)  
[feedback@futurelearn.com](mailto:feedback@futurelearn.com)

Choose from hundreds of free online courses, from Language & Culture to Business & Management, Science & Technology to Health & Psychology. Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access

# GYMNASIUM

## GYMNASIUM

[thegymnasium.com](http://thegymnasium.com)  
[help@thegymnasium.com](mailto:help@thegymnasium.com)

Gymnasium offers free online courses, designed to teach creative professionals in-demand skills. Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

- Gain real-world skills
- Get expert instruction
- Career opportunities

# Free with your magazine

Instant access to these creative resources...

## Essential assets and resources

Get textures, fonts,  
backgrounds and more



## Exclusive video tutorials

Learn to code/create  
with HTML, CSS & JS



## 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
- New chapters from Killersites expert PHP video courses ([shop.killervideostore.com](http://shop.killervideostore.com))
- 24 Luminous presets and 10 iPad Pro hi-res mockups from Sparklestock worth \$36 ([www.sparklestock.com](http://www.sparklestock.com))

→ Log in to [www.filesilo.co.uk/webdesigner](http://www.filesilo.co.uk/webdesigner)

Register to get **instant access**  
to this pack of must-have  
creative resources, how-to  
videos and tutorial assets

**Free  
for digital  
readers, too!**  
Read on your tablet,  
download on your  
computer





The home of great downloads – exclusive to your favourite magazines from Future!

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, in one place

## Get started

Everything you need to know about accessing your FileSilo account



- 01** Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.



- 02** You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.



- 03** If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at the address below:  
[filesilohelp@futurenet.com](mailto:filesilohelp@futurenet.com)

# An incredible gift for subscribers



## Subscribe today & unlock the free gifts from more than 70 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

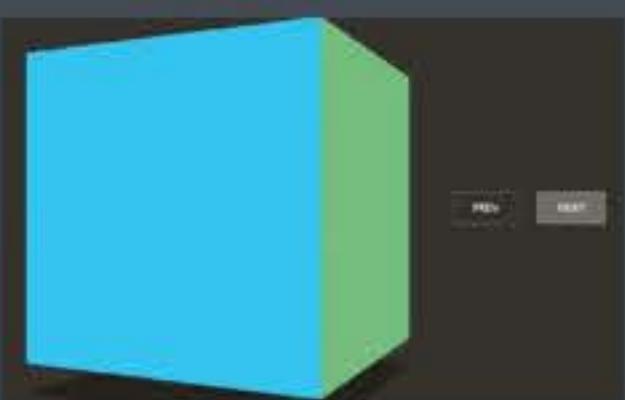
### Over 70 hours of video guides

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 50 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

---

# JAVASCRIPT 2019

---

**25 APIs, frameworks and tools that designers and developers will need in their toolbox**

## WHAT'S NEW IN WORDPRESS 5?

What does the Gutenberg editor have to offer, and what are the pros over previous versions?

## CSS CUSTOM PROPERTIES

Learn how to use new media query '@media (prefers-color-scheme: dark)'

## REACT AND REDUX THUNK

Learn how to deal with fetching/receiving data from a server, handling latency and errors

Visit the **WEB DESIGNER** online shop at



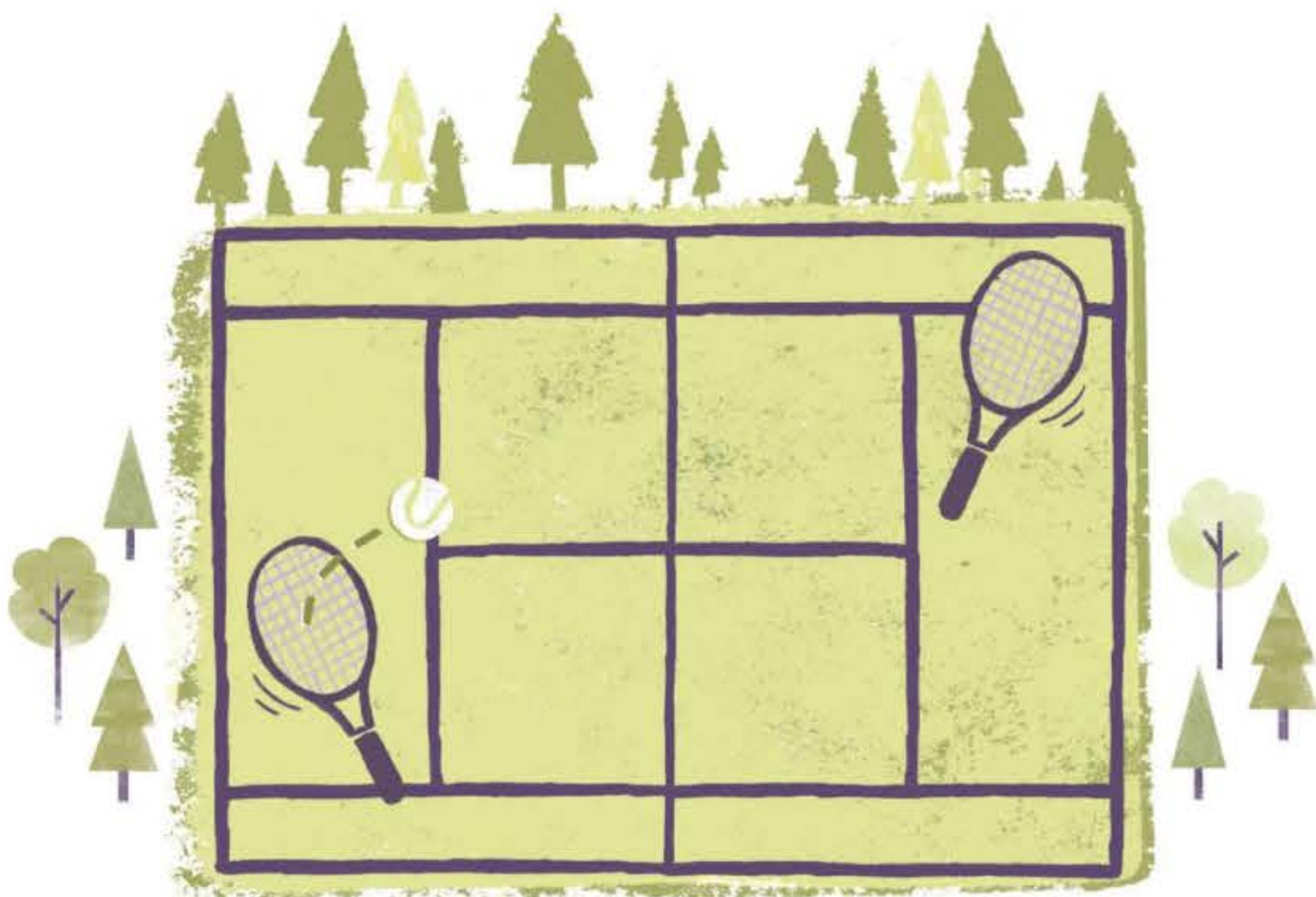
**myfavouritemagazines**  
[myfavouritemagazines.co.uk](http://myfavouritemagazines.co.uk)

for the latest issue, back issues and specials

**ALL IN YOUR NEXT  
WEB DESIGNER**  
Issue 283 on sale  
**Tuesday 8 January 2019**

**SUBSCRIBE TODAY** Go to page 50 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!

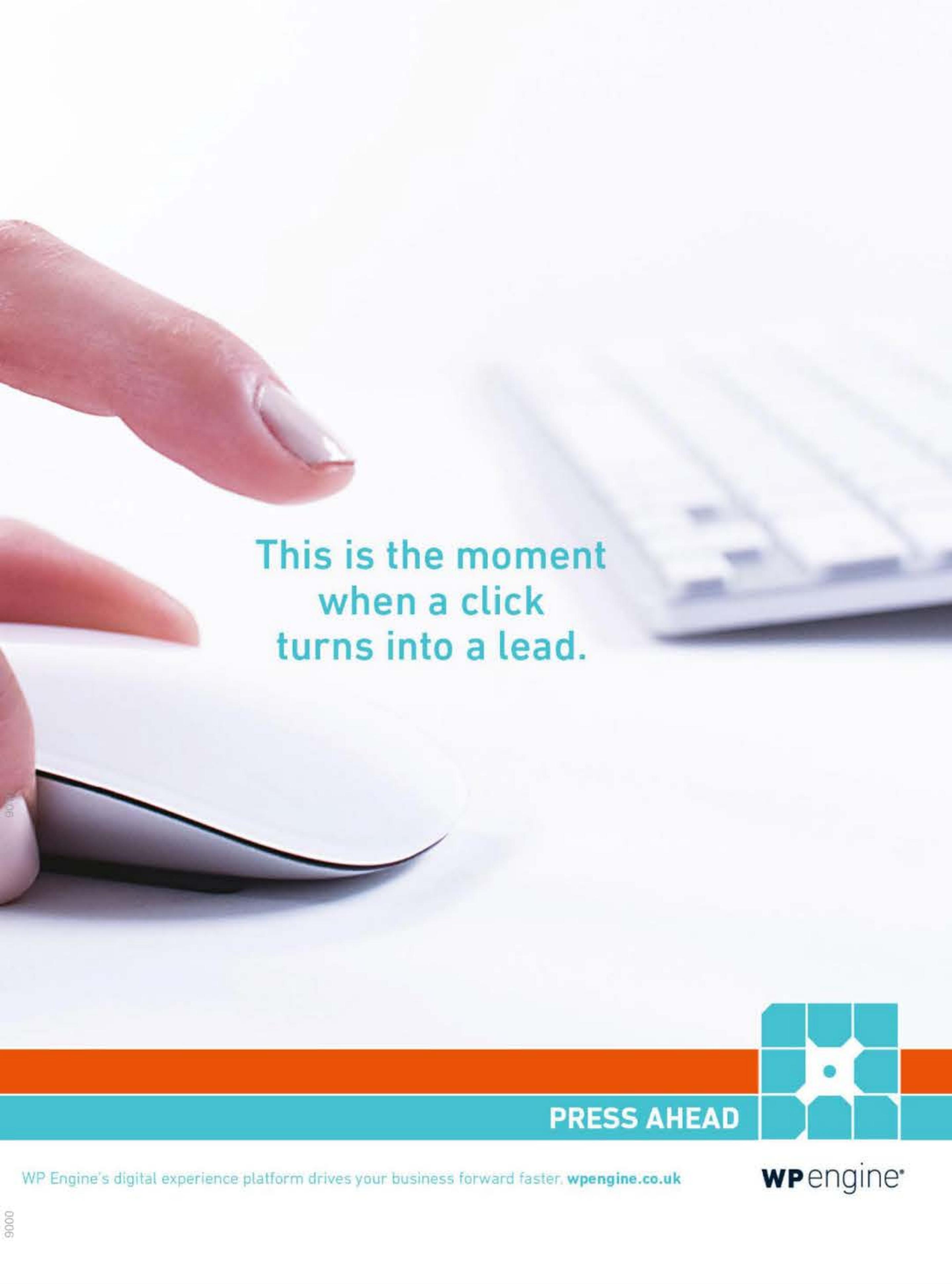
Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood.

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.

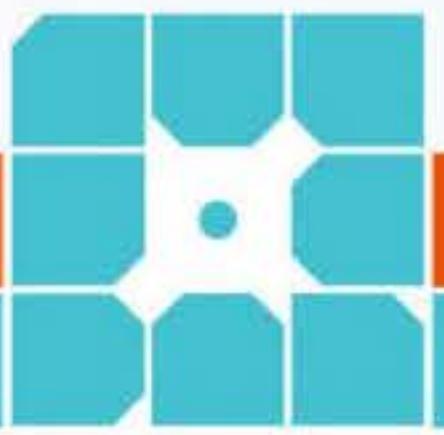
There are some great reasons to **#LovePaper**  
Discover them now,  
[twosides.info](http://twosides.info)





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](http://wpengine.co.uk)

**WP** engine®