

FREE 100 CREATIVE
RESOURCES

CODE AN API-DRIVEN APP
WITH THE IONIC FRAMEWORK

LEARN HOW TO BUILD
WITH THE JAMSTACK

Expert tutorials, techniques and inspiration

designer

CSS

HOUDINI

MAKE DESIGN MAGIC WITH THE
APIs SHAPING CSS TODAY



CREATE AN
INTERACTIVE
WEBGL HEADER

SIMPLIFY ACCESS
TO LOCAL FILES
WITH THE FS API

INSIDE WEBXR:
AR, VR & MR
COME TOGETHER

MAKE A
DYNAMIC
CUSTOM CURSOR



Can you afford downtime, theft or fines?

65% of cyber attacks are targeted at smaller businesses.

Introducing Website Security from Heart Internet

Never worry about the safety of your site again



Action

Remove malware from your website as soon as it appears.



Alert

Get immediate updates about issues on your site.



Prevent

Stop future hacks with our Web Application Firewall.



Improve

Make your site even faster with our built-in CDN.

Introductory Offer Available Now

Call us on 0330 660 0255 or visit www.heartinternet.uk/website-security

Download our free ebook "Your Guide to Website Security" at www.heartinternet.uk/guide-to-website-security

HeartInternet

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

Arrivederci, ciao and goodbye

It's hard to believe, but after a blockbuster 140+ issues of **Web Designer**, I'm finally having to say goodbye to the magazine that I have loved and cherished, and hopefully anyone who is reading (and has read the mag) has enjoyed. I have travelled from the bottom to the top, starting out as a staff writer back on Issue 147, before finally taking over the reins on Issue 210. That's 83 covers I have worked on, with some amazing ones forever etched in my memory. I do have a sadly encyclopedic knowledge of **Web Designer** covers that I have worked on. Ask me what was on the cover of 213 and who designed it, and I'll give you an answer in less than 30 seconds.

While it's a day tinged with sadness, the latest issue still brings you a host of top quality articles, none more so than the cover feature. CSS offers more and more as every day goes by and Houdini has, and is going to have, a key role now and in the future. Matt Crouch takes a look at the key APIs shaping CSS: how they can be used, which ones can be used now and how to build with them. Keeping the future theme, we look at WebXR, what it is, great examples and the resources, tools and devices making it happen. Plus, we have an array of top tutorials, including cool text effects with Blotter.js and interactive WebGL headers. So, it's a final thanks to you all. And, you never know, **Web Designer** may well be back, but just in a more web-friendly form. Enjoy the last? issue. Sad face.

“The aim of Houdini is to open up CSS and allow developers to apply polyfills further along the pipeline and speed things up. They also open up new opportunities to create effects not previously possible.”

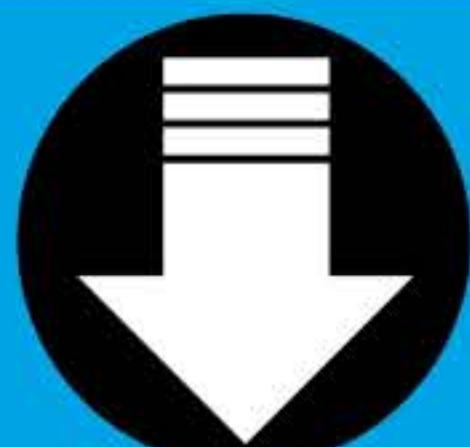
Highlight



“Communicating complexity is at the heart of our professional activity. We believe in making that information accessible.”

Italian talent The Visual Agency produce stunning, compelling visual feasts. **Web Designer** finds out more. [Page 38](#)

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



FREE - with your latest issue
100 Designer resources

Assets - 50 Retro Films Lightroom Presets (worth \$19) from Sparklestock
- 50 Editorial Lightroom Presets and LUTs (worth \$19)
- Tutorial files and assets



www.filesilo.co.uk/webdesigner

This issue's panel of experts

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



Matt Crouch

Matt is a software engineer at Vidsy in London. In this issue he takes a look at Houdini – the magical collection of APIs that are set to revolutionise how developers and designers approach CSS. See how they work, what they're capable of and how to start using them today.

Page 46

The Houdini task force consists of representatives from organisations such as Apple, Google and Microsoft, and other members of the CSS Working Group

Richard Mattka



Richard Mattka is an award-winning creative director specialising in innovative technologies, interactive experiences, web development and visual effects. This issue he delves into the world of WebXR to reveal more. [Page 84](#)

Mark Shufflebottom



Mark is a professor of Interaction Design at Sheridan College. This issue Mark is creating an interactive, colour changing image displacement filter in WebGL for the header section of a site. [Page 56](#)

Tam Hanna



Tam grew up using cross-platform development frameworks, JavaScript, being the one of most interest to him. This issue he looks at Google's newly introduced File System API and reveals what it has to offer developers. [Page 78](#)

Will Yoxall



Will is the MD of Untapped and specialises in hybrid app and web development projects. This issue he continues his guide to Ionic and shows how to connect an API and build an app for both iOS and Android devices. [Page 72](#)

Leon Brown



Leon is a freelance web developer and trainer who assists web developers in creating efficient code for projects. This issue, he recreates a host of techniques inspired by the top-class sites seen in Lightbox. [Page 19](#)

David Howell



David is a journalist with over 20 years' experience in publishing, and runs his own business, Nexus Publishing. In this issue, he talks to Italian Talent The Visual Agency, who turn information into beautiful, compelling visuals. [Page 38](#)

Joseph Ford



Joe has been building websites commercially since he was 16, and now works as a senior developer. In his tutorial this month he introduces the JAMstack and shows how to build an automatically deploying blog website. [Page 62](#)

Frank Kagumba



Frank is a tech enthusiast and front-end developer who's focused on making the web a better place for users every day. In this issue's tutorial, you will learn how to create cool text effects using the Blotter.js library. [Page 66](#)

Follow us!

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

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

Editorial

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

Art Editor Newton Ribeiro
Acting Editor in Chief **Claire Howlett**
Senior Art Editor **Will Shum**

Contributors

Richard Pilton, Mark Billen, Leon Brown, David Howell, Matt Crouch, Mark Shufflebottom, Joseph Ford, Frank Kagumba, Will Yoxall, Tam Hanna, Newton Ribeiro De Oliveira, Harriet Knight, Steve Mumby, Alexander Phoenix, Briony Duguid, Laurie Newman

Photography

All copyrights and trademarks are recognised and respected

Advertising

Media packs are available on request.
Commercial Director Clare Dove
clare.dove@futurenet.com
Senior Advertising Manager Mike Pyatt
michael.pyatt@futurenet.com
01225 687538
Account Director George Lucas
george.lucas@futurenet.com

International Licensing

Web Designer is available for licensing. Contact the Licensing team to discuss partnership opportunities.
Head of Print Licensing Rachel Shaw
licensing@futurenet.com

Subscriptions

Email enquiries contact@myfavouritemagazines.co.uk
UK Orderline & enquiries 0344 848 2852
Overseas order line and enquiries +44 (0) 344 848 2852
Online orders & enquiries www.myfavouritemagazines.co.uk
Head of subscriptions **Sharon Todd**

Circulation

Head of Newstrade **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

Managing Director - Prosumer **Keith Walker**
Chief Content Officer **Aaron Asadi**
Commercial Finance Director **Dan Jotcham**
Head of Art & Design **Greg Whittaker**

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

Distributed by Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU www.marketforce.co.uk Tel: 0203 787 9001

ISSN 1745-3534

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

All contents © 2018 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 20098865) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

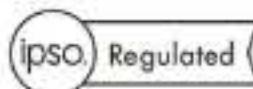
If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications in any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



Future plc is a public company quoted on the London Stock Exchange [Symbol: FUTR]
www.futureplc.com

Chief executive **Zillah Byng-Thorne**
Non-executive chairman **Richard Huntingford**
Chief financial officer **Penny Laddin-Brand**

Tel +44 (0)1225 642 244



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 and printer hold full FSC and PEFC certification and accreditation.



Did you know that European forests, which provide wood for making paper and many other products, have grown by 44,000km² over the past 10 years? That's more than 1,500 football pitches 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



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 now and tomorrow for designers

Web Designer takes a look at the latest Design Census from AIGA and Google

11 Make apps accessible to all

Richard Pilton, MD at Kayo Digital tells us how apps can hit accessibility targets

12 Webkit: The best must-try resources out there

Discover the libraries and frameworks you need

16 Lightbox

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

30 Mystic making

Italian agency illo build a Tarot reader capable of predicting the future

36 Back issues

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

38 Information as Art

The Visual Agency turns information into beautiful, compelling visual experiences. **Web Designer** finds out more

46 CSS Houdini

Matt Crouch takes a closer look at the APIs that are shaping the future of CSS today

84 WebXR: unifying reality

Learn what WebXR is and valuable resources you will need to start using it

92 Hosting listings

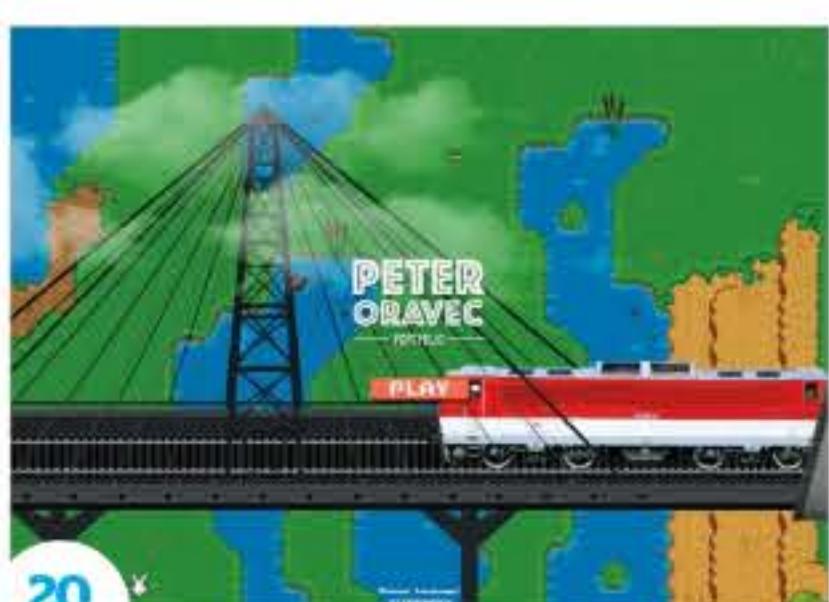
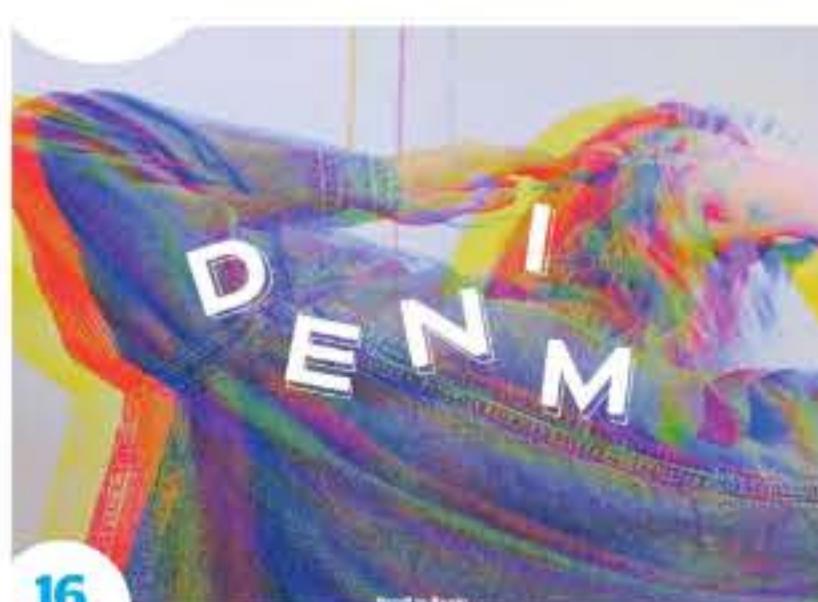
An extensive list of web hosting companies. Pick the perfect host for your needs

94 Course listings

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

98 Next month

Say hello to sister mag net. They're worth it



TUTORIALS

56 Create an interactive header

Build a dynamic image that transitions from black and white to colour on hover

62 Build with the JAMstack

Learn how to combine JavaScript and Markdown to build dynamic websites

66 Create cool text effects with JS

Learn how to use the Blotter.js library to create four interesting and engaging text effects

72 Build an API-driven app

In part two of this introduction to Ionic, learn how to connect an API and build and run on iOS and Android

78 Google's Native File System API

Take a look at a recently introduced API concept that lets sites access host file systems directly



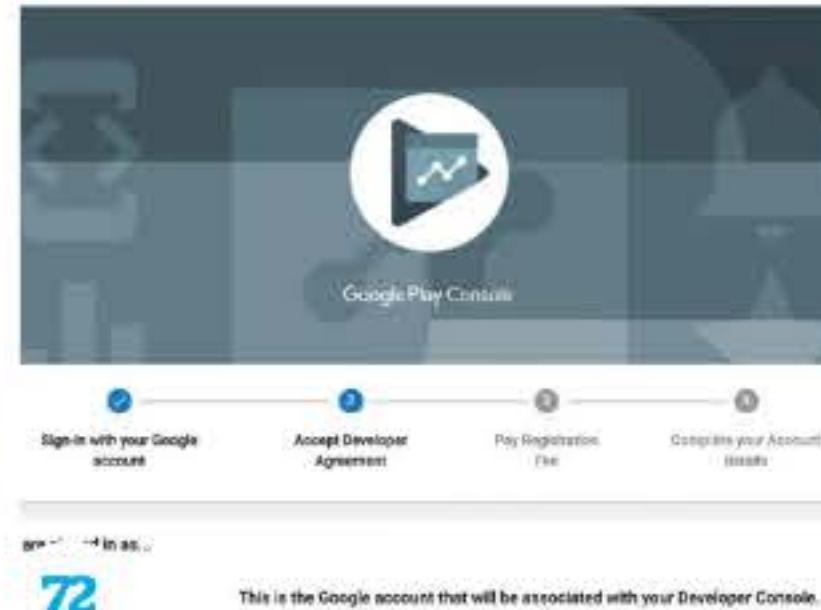
66



56



62



72



38

FILESILO

96 Get the latest must-have resources and videos

- 50 Editorial Lightroom Presets and LUTs (worth \$19) and 50 Retro Films Lightroom Presets (worth \$19) (sparklestock.com)



myfavouritemagazines

Visit the **Web Designer** online shop at
myfavouritemagazines.co.uk
for the latest issue, back issues and specials

DIGITAL EDITION

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

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



Subscribe today and save

<https://bit.ly/2AYtZ65>

Header

The tools, trends and news to inspire your web projects

What does the future hold for designers?

Web Designer takes a look at the recently released 2019 AIGA Design Census to see what it reveals

Designers, and we don't specifically mean web designers, are all part of the tapestry of the web.

Web designer has a broad remit, and can include UI/UX designers, graphic designers and 3D/animation artists. Why are we clarifying this? Because we are taking a look at 2019 AIGA Design Census, which includes a wide berth of design disciplines.

So what is the AIGA Design Census? It is an annual report put together by AIGA, the professional association for design, and Google. Here's a quick report summary where it reveals it will look 'deeply at who's designing today, the kind of work they're doing and how they're doing it, how much they're getting paid, how satisfied they are, what they anticipate for the future, and so much more.' The report goes on to say, 'We surveyed thousands of people who make up the design industry, which isn't only practicing designers; don't

forget all the students, educators, managers, business owners, and many other roles that make the design world spin. Still, for the sake of brevity, we refer to all respondents as "designers."

The majority of the Design Census looks at who's designing, how designers are working, what salaries designers are making and how satisfied designers are with their jobs today. Take a closer look at these stats at <https://designcensus.org/data/2019DesignCensus.pdf>.

The report is not exclusive to these elements – it does also include a section on what designers think of the industry, and what skills, technologies and issues are key to design now and in the future.

First up, we have taken a look at job satisfaction. This section sets out to 'explore what makes designers feel satisfied and valued (or anxious and discontented). A quick overview reveals that 'designers are a relatively happy bunch, with 65% saying they feel

satisfied in their current position'. But, to put this into context, in the 2017 report, 82% of designers reported feeling satisfied. The downturn is attributed to growing sense of instability in the industry. So how do designers get ready for the future of design? What skills can they acquire to make them more employable (and maybe more satisfied). The report reveals that artificial intelligence, augmented and virtual reality, and collaborative design software are poised to change the industry. So, what design skills will be needed? The most popular is simple – Adaptability, which is closely followed something more tangible, Multi-disciplinary skills. Further down the scale you will find Storytelling, Design research and Design frameworks. There is a lot of information in the 58-page report that will give you a great insight into the world of designers, so make sure you add it to your reading list today.

STAT ATTACK

MOBILE BROWSERS

Is the small screen ecosystem different to desktop?

Chrome

60.68%



No surprises that Google are number one

Safari

20.25%



Apple's browser shows the popularity of the iPhone

Samsung Internet

6.77%



Preinstalled on Samsung devices as standard. Android based

UC Browser

6.35%



More popular than Chrome in Asia's emerging markets

Opera

2.97%



Still hanging in there but popularity declining

CB CREATIVE BLOQ
creativebloq.com

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

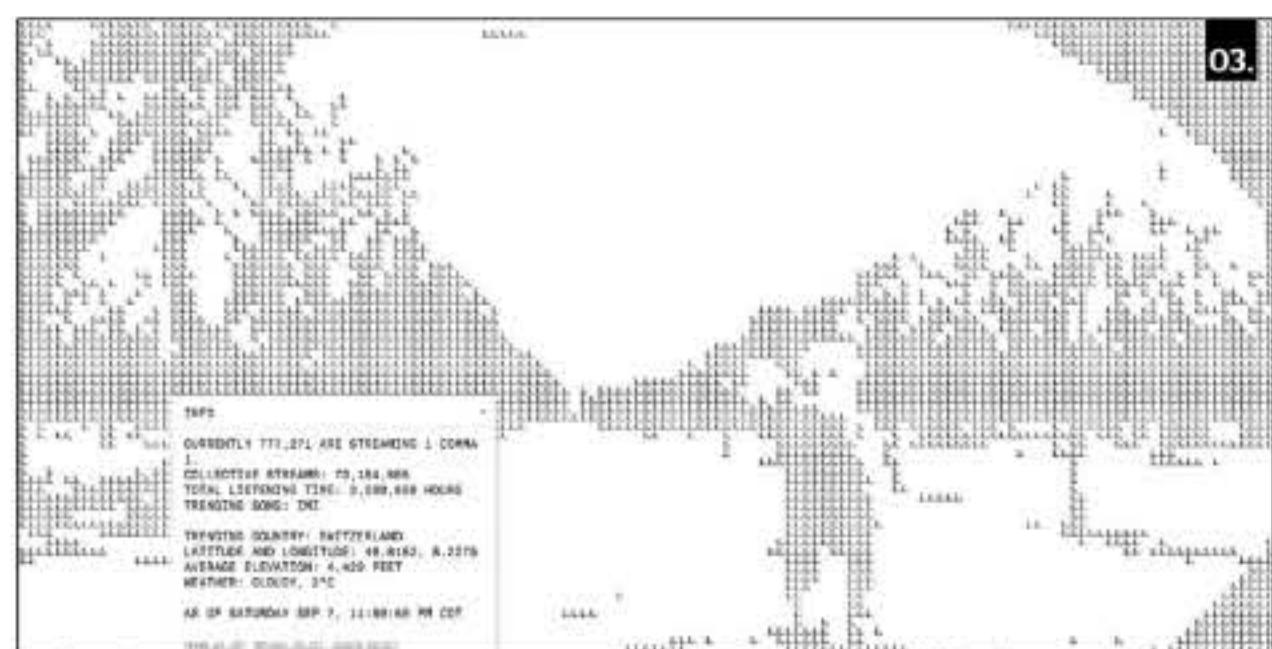


**Subscribe today and get
5 for £5**

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

Source: <http://gs.statcounter.com>
(correct as of Aug 2019)

Sites of the month



01. Solar Digital

<https://solar-digital.com>

A red overlay and sweet animated waves transition the user from one image to the next, and the next...

02. QI Catalog

<https://qodeinteractive.com/catalog>

A stylised bottom text navigation system reveals an image for each item as the user passes over the text.

03. ViiSUALiZER

<https://boniver.withspotify.com>

Bon Iver and Spotify turn data into an interactive dot pixel animation that mesmerises and engages in equal amounts. It's even better if you like them!

04. #24HourAce

<http://24hourace.gucci.com>

Gucci come up with another nod to web past with garish colours and a selection of excess design elements.

Graphics Logos and badges 2019

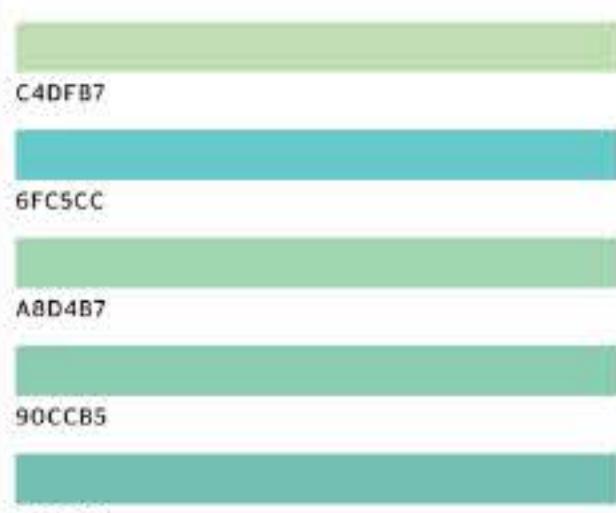
<https://bit.ly/2lyi6y9>

T-shirt illustrations, labels and patches from the talented hand of Alexander Shimanov.



Colour picker No Sensibility

<https://bit.ly/2mlUjf6>



Typesetter Campora

<https://bit.ly/2ndn0Rn>

Created by Chilean designer Salvador Rodriguez, Campora is an eclectic typeface with a retro flavour of the Eighties, featuring 19 styles.

ABCabc
0123456789

WordPress Gutentim

<https://bit.ly/2mDvt04>

A modern and clean Gutenberg WordPress blog theme featuring six smartly designed home page options.



HEAR FROM THE PROS

SIGN UP TO THE

**web
designer**TM

NEWSLETTER TODAY!



Get weekly news, tips & inspiration

SIGN UP NOW!

bit.ly/2KI5b7Y

Make your apps accessible to all

Accessibility is a key function that must be incorporated into any app



Richard Pilton

Managing Director
at Kayo Digital

<https://kayo.digital>

“Over formatted text can be difficult for dyslexic users to read, as those with impaired vision, so keep your formatting simple and consistent throughout your design.”

T

he ability to access, abbreviated as accessibility, refers to the design of products that can be used by people with disabilities. New government regulations mean websites and apps in the public sector must now be accessible to all.

Making your app accessible means you must consider a range of challenges a customer may face, ensuring that as many people as possible can use it.

For instance, those with impaired vision, motor difficulties, cognitive or learning difficulties as well as deafness or impaired hearing. Designing an accessible app means much more than simply making the design clearer. You must design an app that is adaptable to those throughout the spectrum and support those who need the apps extra, accessible functions.

So what needs to be considered when designing an accessible app? The initial step is to make sure you have a simple, clear layout. A cluttered design may be overwhelming, leading to a user who might give up on your app before they have even properly begun to explore it. Elements must be visible to those who may need to magnify the screen, so your design must be responsive to adapt to various screen sizes. Text and call-to-action buttons must also appear at an appropriate size, with the option to enlarge if required.

Following on with navigation, you must design an easy-to-follow app with clear indicators. Navigation should have short task flows and should be easy to find, as well as consistent throughout app stages. Again, this is especially important with your call-to-action buttons.

They must be clearly labeled, to smoothly guide users through your apps goal conversion. Navigation pointers should also be consistently positioned in a similar format throughout your app. For instance, menu bars and search boxes. This will allow users to quickly and easily explore your app in a logical order with no confusion.

Text is another element that must be carefully thought out. Dyslexia can affect a person's accuracy of reading, so you must really think about the structure of text used on your app. Uneven spacing, long sentences/paragraphs, italic fonts and the colour of text are just some features to consider. Over formatted text can be difficult for dyslexic users to read, as those with impaired vision, so keep your formatting simple and consistent throughout your design.

Creating an app which is interactive as well as accessible can be seen as a challenge, though creating a tick list of things you must consider will mean you can include media communications – for instance, video and audio, where appropriate. When incorporating these elements, users should be able to pause or stop, adjust volume and turn captions on and off. Abled users will also enjoy these options.

You may also need to consider adding subtitles, audio description and sign language for users who may be deaf.

Colour palettes are a top priority when designing any app as most businesses associate certain colours with their branding. To make your app accessible to colour-blind users, be sure to not solely rely on colour to communicate your message.

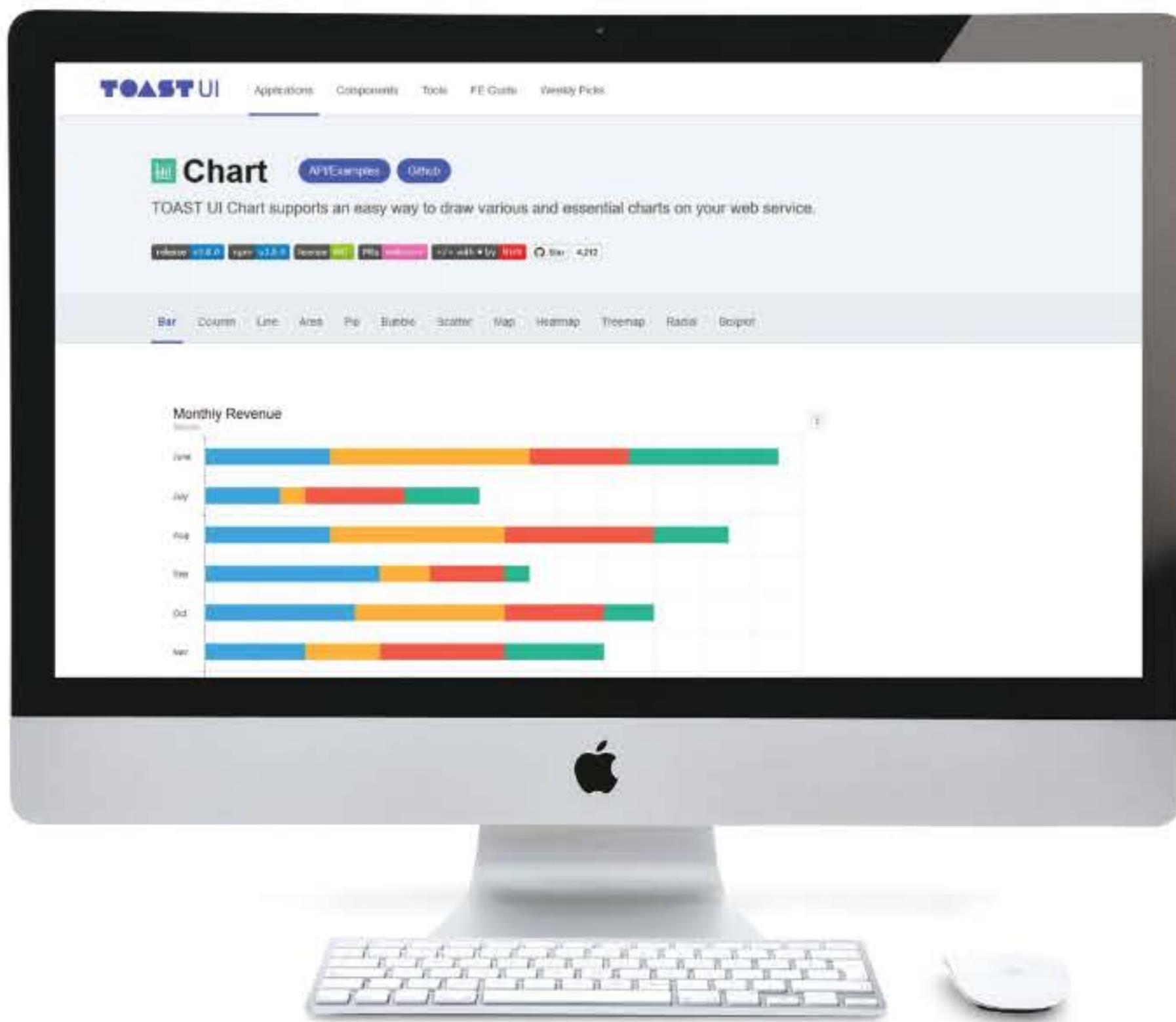
This doesn't mean that you should be avoiding using colour at all. As mentioned before, colour can reflect a brand, so just make sure that elements of your app are not purely identified based only on colour. Your palette should also be carefully selected so that users can fully interpret the information displayed.

Finally, remember to test! Following accessibility guidelines is a good start, but treating the development as a tick box challenge will not guarantee success. There are many methods to approach this, for instance engaging with real users at random and asking them to test elements of your app, or using digital tools that can offer detailed assistance on screen reading.

Designing an app is a great opportunity for creative minds to develop a strong platform for any business. Making it accessible should be not a constraint; it will allow you to create a positive brand image that can associate with everyone.

webkit

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



TOAST UI

<https://ui.toast.com/tui-chart>

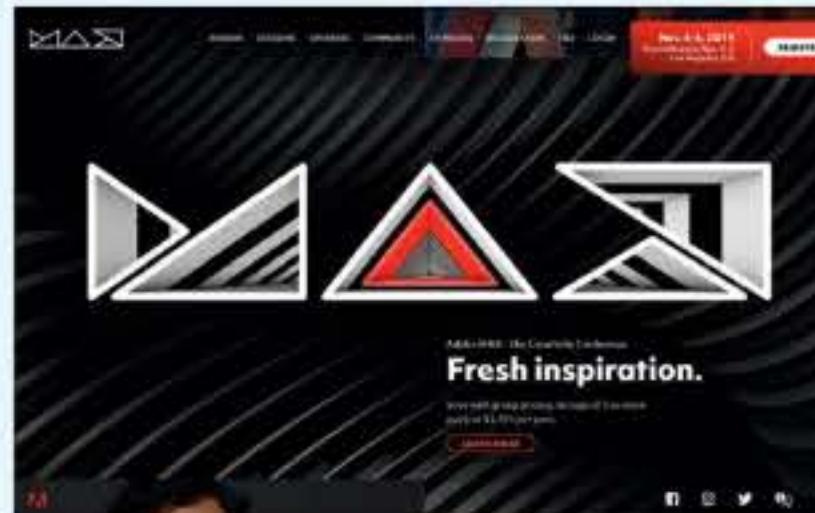
In a nutshell, 'TOAST UI Chart supports an easy way to draw various and essential charts on your web service'. It provides a host of chart types to visualise your data

including Bar, Column, Line, Pie, Area, Bubble, Doughnut and many more. It offers support for legacy browsers (yes, IE) and guarantees a uniform look in any browser.

TOP 5 Web conferences

Nov & Dec 2019

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



Adobe Max

<https://max.adobe.com>

A creative conference by one of the biggest in the business. There are over 300 sessions and a host of inspiring industry speakers.

AGILE in the CITY



Agile in the City

<https://agileinthecity.net>

This event focuses 'on delivering practical, useful, engaging content to build your skills and celebrate your passion'.

CONTACT US AT: webdesigner@futurenet.com | @WebDesignerMag



Glide.js

<https://glidejs.com>

This is a dependency-free JavaScript ES6 slider and carousel. It's lightweight, flexible and fast and designed to slide. Extend the power of the slider with your own custom plugins.



Loom

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

Loom is a Chrome extension that allows users to Capture a screen, record and narrate all at once. Once done, share with a simple link.



augmented-ui

<https://augmented-ui.com>

A collection of CSS that enables users to create cyberpunk inspired UI for any size element, on any web page or app. Little effort is needed to create, and it is free to use.



OGL

<https://oframe.github.io/ogl>

OGL is a small WebGL framework 'aimed at developers who like minimal layers of abstraction'. This makes it easier to use, plus it has zero dependencies and shares similarities with ThreeJS.



ffconf

<https://2019.ffconf.org>

A single day conference with eight sessions that see talks on community, performance, tools, debugging, IoT and Just JavaScript.



Meaning 2019

<https://meaningconference.co.uk>

Meaning is about connecting and inspiring people who believe in better business. See a host of industry experts deliver.



dotCSS

www.dotcss.io

Touted as the 'The world's largest CSS conference', expect top quality speakers presenting professional talks that will inspire.

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



Quasar Framework

<https://quasar.dev>

Cutting straight to the chase, what is Quasar? It's an "open-source Vue.js-based framework, which allows you as a web developer to quickly create responsive++ websites/apps in many flavours". These flavours include Single

Page Apps, Progressive Web Apps and Android and iOS apps. Quasar also has a host of features, including HTML/CSS/JS minification, ES6 transpiling and Code-splitting with lazy loading.

TOP 5 Codepens November 2019

Check out this collection of codebases to get some inspiration



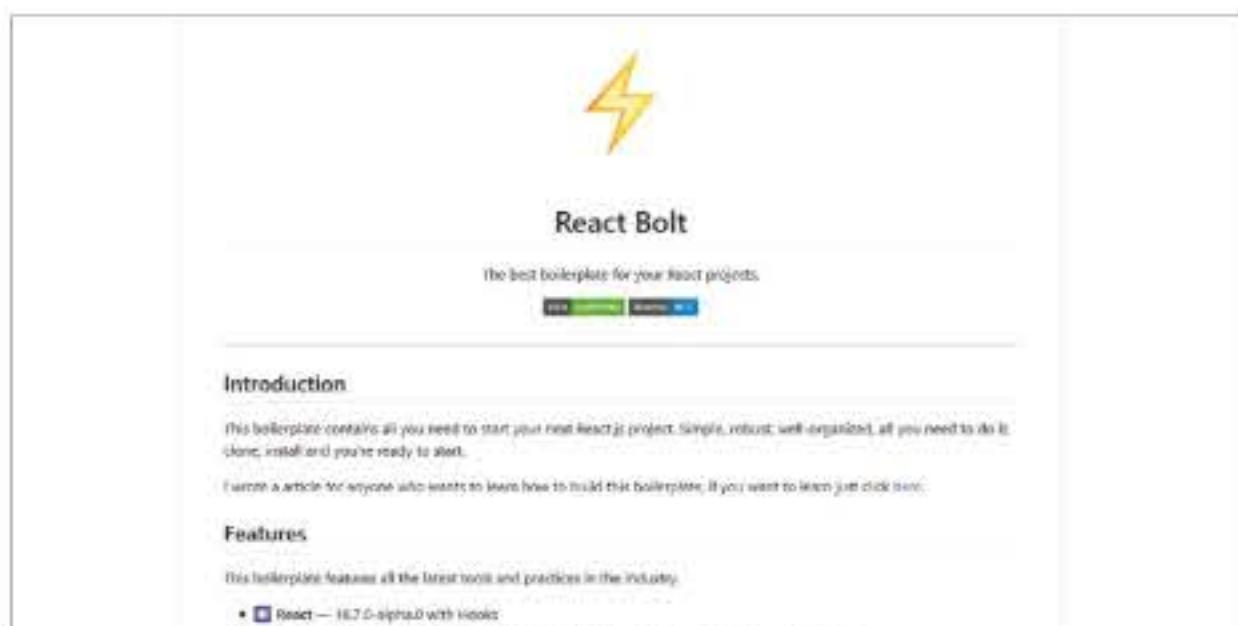
Fluid text hover

<https://codepen.io/robin-dela/pen/KKPYoBq>
Adds interest to headline typography with an interactive mask. On hover, watch as the background follows the cursor.



Test Shader #1

<https://codepen.io/AlainBarrios/pen/jJLVWv>
This effect is what a driver would see on a rainy night travelling along the motorway. Use the mouse to speed up/slow down.

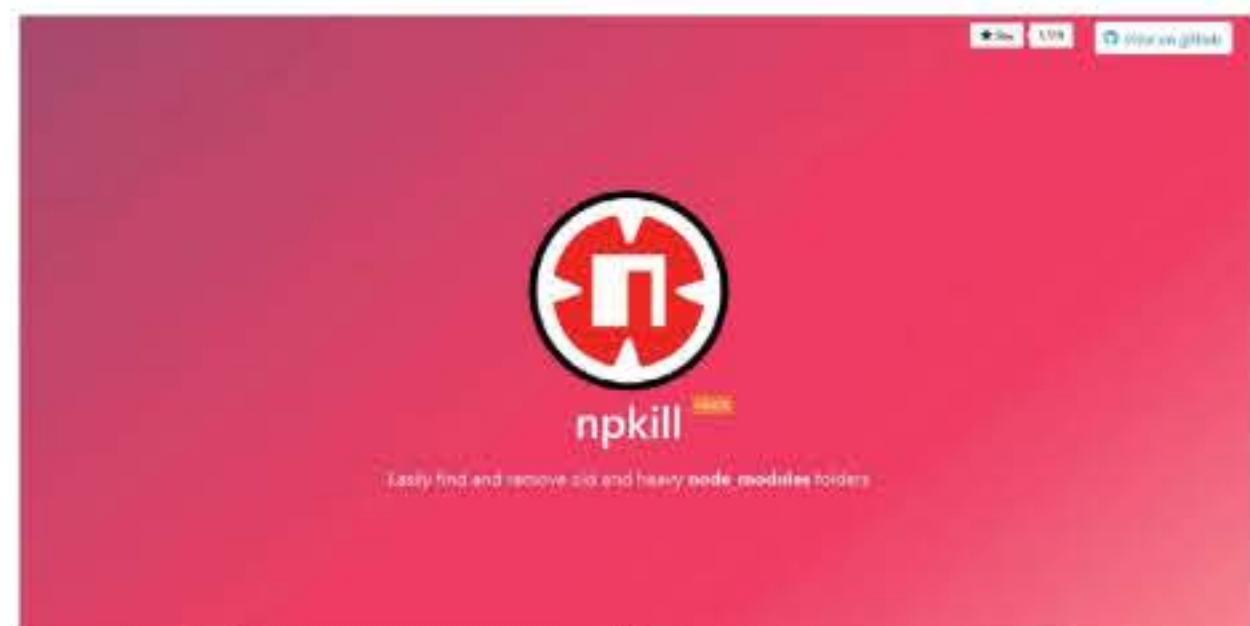


The screenshot shows the homepage of React Bolt. At the top is a yellow lightning bolt icon. Below it is the title "React Bolt" and a subtitle "the best boilerplate for your React projects." There are two small green buttons at the bottom.

Introduction
This boilerplate contains all you need to start your next React.js project. Simple, robust, well-organized, all you need to do is clone, install and you're ready to start.
I wrote a article for anyone who wants to learn how to build that boilerplate. If you want to learn just click here.

Features
This boilerplate features all the latest tools and practices in the industry.

- React — 16.7.0-alpha.0 with hooks
- Redux — 4.0.0-alpha.1 with Thunk, Babel and Webpack



The screenshot shows the npkill tool's interface. It has a red gradient background with a large white circular logo containing a red "n". Below the logo is the text "npkill" and "EASILY FIND AND REMOVE OLD AND HEAVY node_modules FOLDERS". At the bottom, there are two buttons: "Install" and "Find Folders".

React Bolt

<https://github.com/leonardomso/react-bolt>

A simple and robust boilerplate for your React projects. It features all the latest tools and practices in the industry including React, Redux with Redux Thunk, Babel and Webpack.

npkill

<https://npkill.js.org>

A quick summary of what npkill is, is perfectly demonstrated by its tagline, "Easily find and remove old and heavy node_modules folders." Type npkill in Terminal and the tool will do the rest.



The screenshot shows the Can I email website. At the top, there are tabs for "Home", "Features", and "News". Below that is a search bar with "Can I email" and a dropdown menu showing "HTML". A "Get Started" button is at the bottom right. The main content area shows a grid of email client logos: Apple Mail, Gmail, Outlook, and Yahoo Mail. Each logo has a green checkmark indicating support.

Can I email

www.caniemail.com

For those familiar with the Can I use site you will have a good idea what to expect. For those who don't, add in your tag or property and see which email clients support it.



The screenshot shows the Headroom.js website. At the top, there are links for "Headroom.js", "GitHub", and "@WickyNillia". The main heading is "Headroom.js" with the tagline "Give your pages some headroom. Hide your header until you need it.". Below that is an "Installation" section with download links for "Download (.zip)" and "Download (.tar.gz)". Further down is a "What's it all about?" section.

Headroom.js

<http://wicky.nillia.ms/headroom.js>

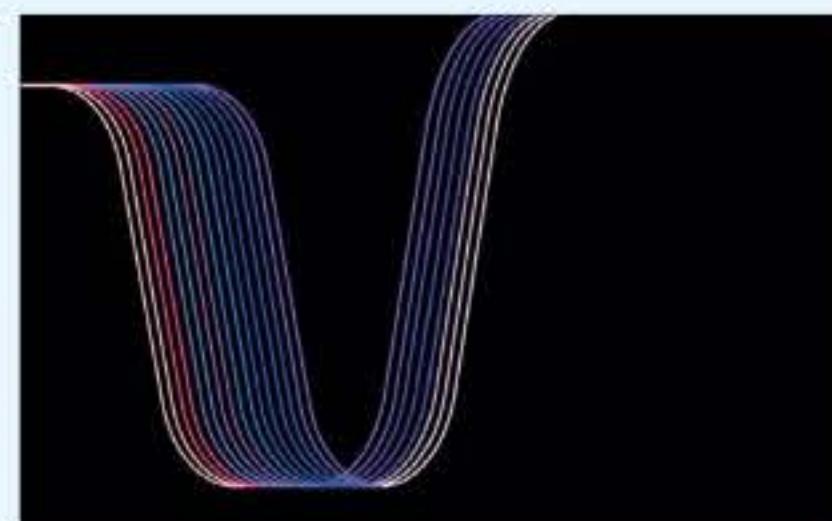
Headroom.js is a lightweight, high-performance JS widget with no dependencies. Scroll down to see your header appear, scroll back up and see it reappear.



Arrowtrix

<https://codepen.io/faisal-jawed/pen/ZEzZjKJ>

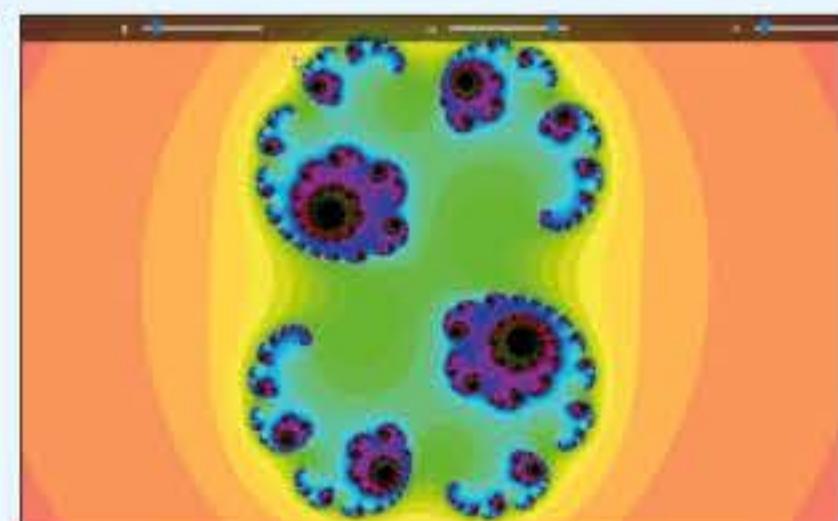
Subtitled Matrix Arrows, this is the effect it's recreating. Various-sized green arrows falling down the screen and fading out create depth.



letter-v

<https://codepen.io/ruidovisual/pen/ZEzPPXr>

HTML and CSS create an undulating letter V with the lines rippling across the letter and changing colour to create realistic movement.



State of my brain

<https://codepen.io/terabaud/pen/VwZRrRL>

Psychedelic shaders are the order of the day here. The colours change and patterns expand and grow. Play with the sliders.

Denim - A Data Deep Dive



[Scroll to Begin](#)

www.lyst.com/denim-report

Designer:

Drench www.drench-design.com



“Fashion hub Lyst presents a rich digital report detailing the data captured around denim buying habits and worldwide search trends.**”**

Colours



Tools

jQuery, CSS3, Soundcloud

Fonts

abcABC
1234567890
abcABC
1234567890

Brown font by Aurèle Sack appears in its Regular and Bold forms for supplementary text paragraphs.

abcABC
1234567890

Adieu font from Good Type Foundry is used in the Black style for lead paragraphs and titles.

Share

A Decade of Denim

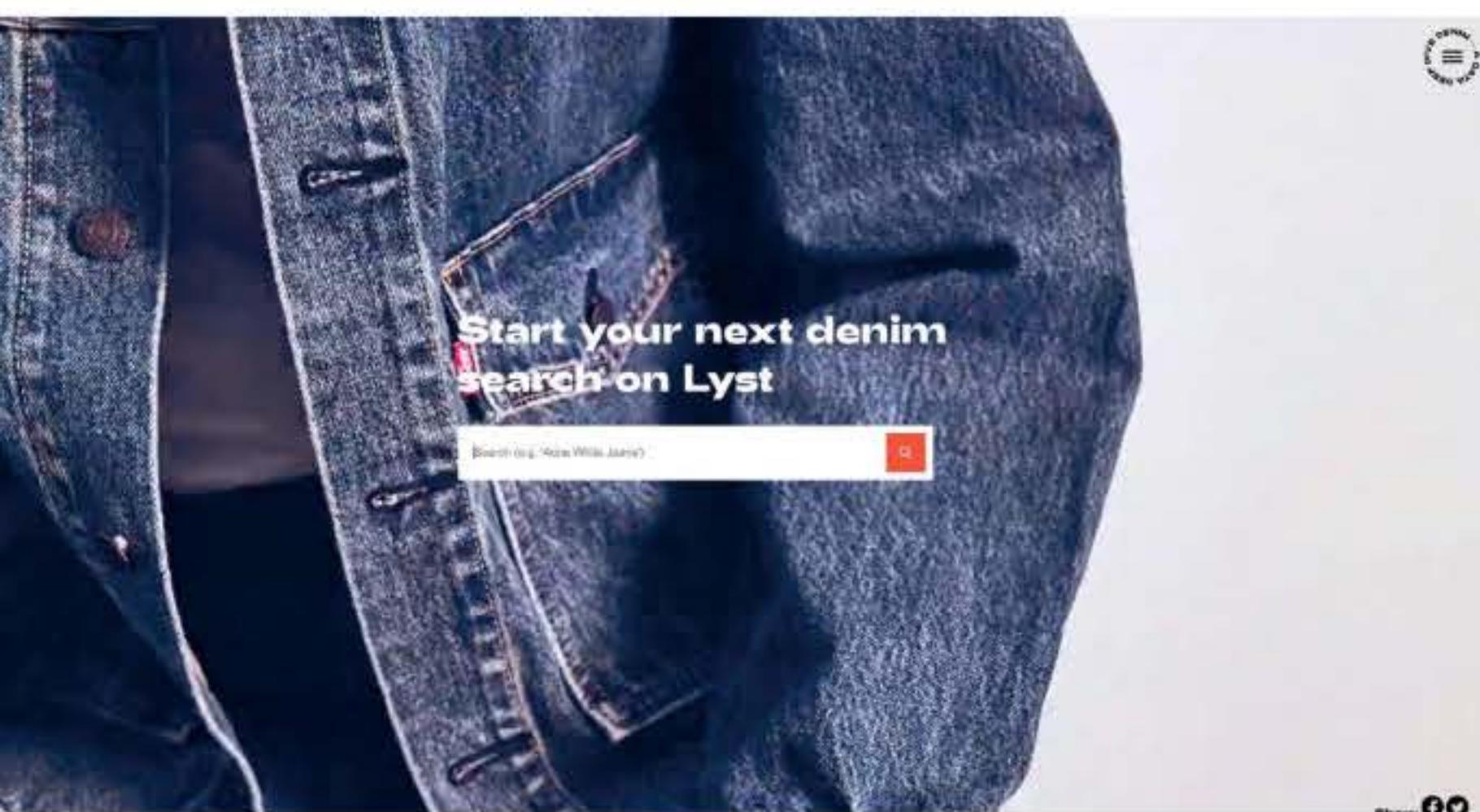
Share:

01 02 03 04 05 06

ASTEL



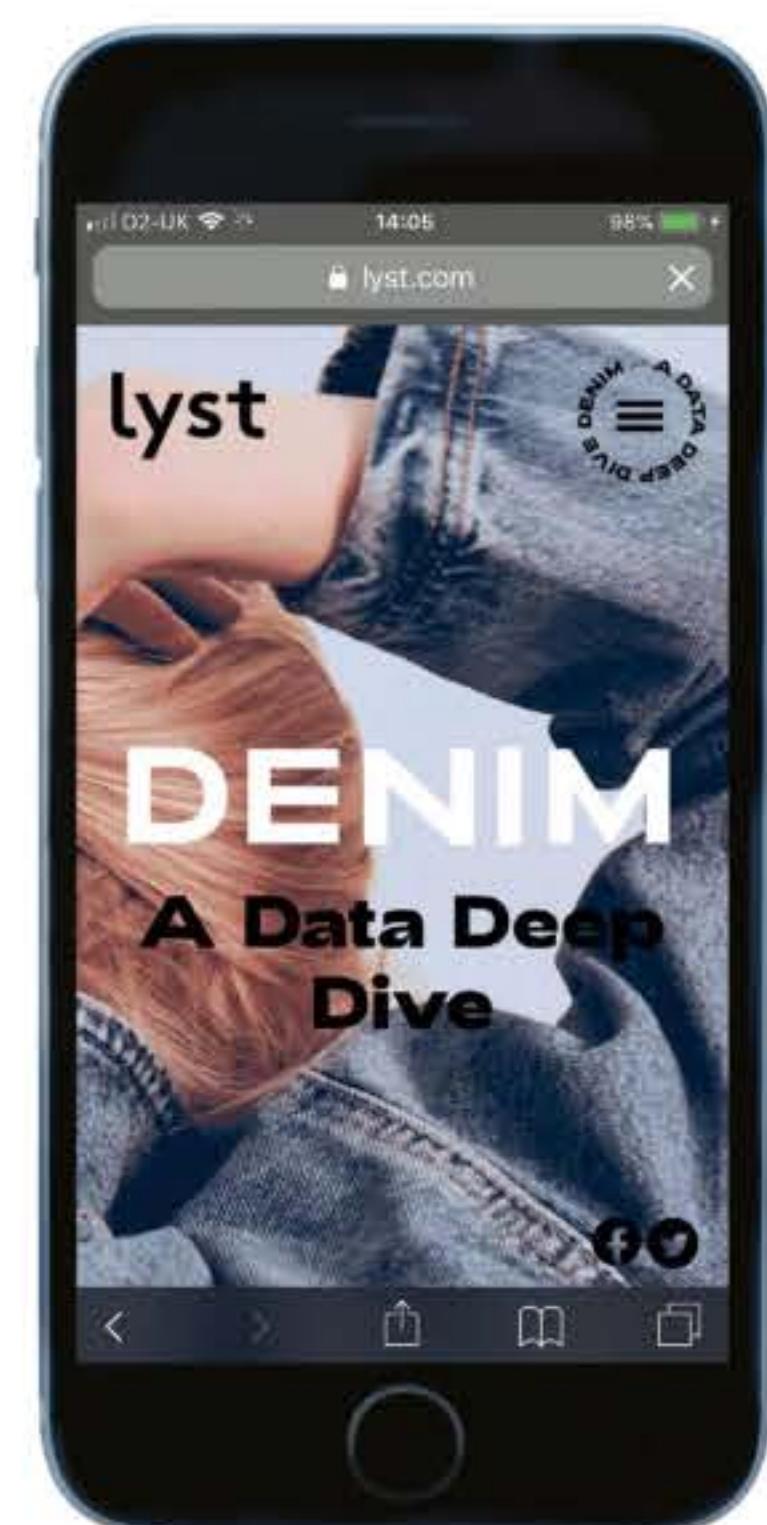
Searches: Trending:

**Left**

The site inventively illustrates how denim styles are being searched and where they are trending the most.

Above

Jerkily animated, staccato video clips provide jaunty points of interest throughout and remind one of denim's enduring durability.

**Left**

Within the footer of the single page is a search bar for placing subsequent denim searches straight from Lyst.

Build a text animation that reacts to page scrolling

An effect that animates the individual letters of a page title in response to user page scrolling

1. HTML content

The first step is to define the web page's HTML. This step's HTML defines the page document with its head and body sections. While the head section is used to load the external CSS and JavaScript resources, the body section stores the title element that will be used for the animation.

2. CSS: container elements

Create a new file called "styles.css". This step defines the visual presentation settings for each container used in the example. The HTML document and its body are set to use a minimum height that's 8 times the height of the screen to guarantee scrolling ability. Meanwhile, the h1 element used for the title is set to display its text using fixed positioning to guarantee visibility independently of scroll position.

```
html, body{
    min-height: 800vh;
}

h1{
    position: fixed;
    text-align: center;
    width: 100%;
    font-size: 4em;
}
```

3. CSS: title children

Each of the title letters will be represented as an individual element inside the h1 element. Relative positioning is used so that later steps can modify the position of these elements in relation to their text-flow position. Setting the display mode to inline-block causes each child element to be placed with standard text flow positioning.

```
h1 > *{
    position: relative;
    display: inline-block;
    top: 0;
}
```

4. JavaScript: initiation

Create a new file called "code.js". The JavaScript code is dependent on the HTML body content being available, hence the Javascript code being placed inside a "load" event listener on the browser window. This step finds the h1 content element as the parent and stores its text as a variable string. The h1 parent is then set to empty - ready for step 5.

```
window.addEventListener("load", function(){
    window.lastScrollY = 0;
    var parent = document.querySelector("h1");
    var txt = parent.innerText;
    parent.innerText = "";
}

*** STEP 5 HERE
});
```

5. JavaScript: create letter elements

Each of the letters stored under the txt string from step 4 are created as individual child elements inside the h1 parent. Each character element is assigned properties for describing a random speed and starting position, current position and opacity. The generated DOM node has the letter set as inner text and the required top style for positioning applied before being appended to the parent h1.

```
for(var i=0; i<txt.length; i++){
    var letter = document.createElement("span");
    letter.speed = Math.floor(Math.random()*10)+2;
    letter.y = Math.floor(Math.random()*400)+2;
    letter.originY = letter.y;
    letter.opacity = 1;
    letter.innerText = txt[i];
    letter.style.top = letter.y+"px";
    parent.appendChild(letter);
}

*** STEP 6 HERE
```

6. JavaScript: scroll event

An event listener to detect page scrolling is added to the browser window. When triggered, a check is performed to identify if the page was scrolled up, along with a search for every child node within the h1 container. Step 7 is executed if the scroll position is less than double the screen height - otherwise it is step 8 that's executed. The lastScrollY is also updated for use with step 7.

```
window.addEventListener("scroll", function(){
    let scrollUp = (window.lastScrollY < window.scrollY ? true : false);
    var nodes = document.querySelectorAll("h1 > *");
    if(window.scrollY < window.screen.height*2)
    {
```

```
    *** STEP 7 HERE
} else{
    STEP 8 HERE
}
window.lastScrollY = window.scrollY;
});
```

7. JavaScript: reposition letters

A for loop is used to reference each of the nodes found in step 6. If the scroll direction was upwards, the element's vertical position will be reduced by the assigned speed, with a check guaranteeing it doesn't move under the target position. Where scrolling isn't upwards, the node's vertical position is increased by its speed, along with a check to make sure it hasn't moved beyond its starting position.

```
for(var i=0; i<nodes.length; i++){
    var speed = nodes[i].speed;
    if(scrollUp == true){
        nodes[i].y -= speed;
        if(nodes[i].y <= 0)
            nodes[i].y = 0;
    } else{
        nodes[i].y += speed;
        if(nodes[i].y > nodes[i].originY)
            nodes[i].y = nodes[i].originY;
    }
    nodes[i].style.top = nodes[i].y+"px";
}
```

8. JavaScript: fade letters

Again, a for loop is used to reference each of the nodes found in step 6. Each item has its opacity recalculated based on the direction of the scrolling and its index position within the h1 container. Opacity is decreased when scrolling is up, and increased when scrolling down. The calculation is then applied as the node's opacity style.

```
for(var i=0; i<nodes.length; i++){
    if(scrollUp == true && nodes[i].opacity > 0)
        nodes[i].opacity -= 1/(1+i*10);
    else if(scrollUp == false && nodes[i].opacity < 1)
        nodes[i].opacity += 1/(1+i*10);
    nodes[i].style.opacity = nodes[i].opacity;
}
```

Peter Oravec

<https://peteroravec.com>



Designer:

Peter Oravec (in-house) <https://peteroravec.com>



Inspired by retro RPG videogames, Front-end Developer Peter Oravec transforms his portfolio into an interactive fun quest

Colours

#2E8135	#146C99
#C49854	#C81503

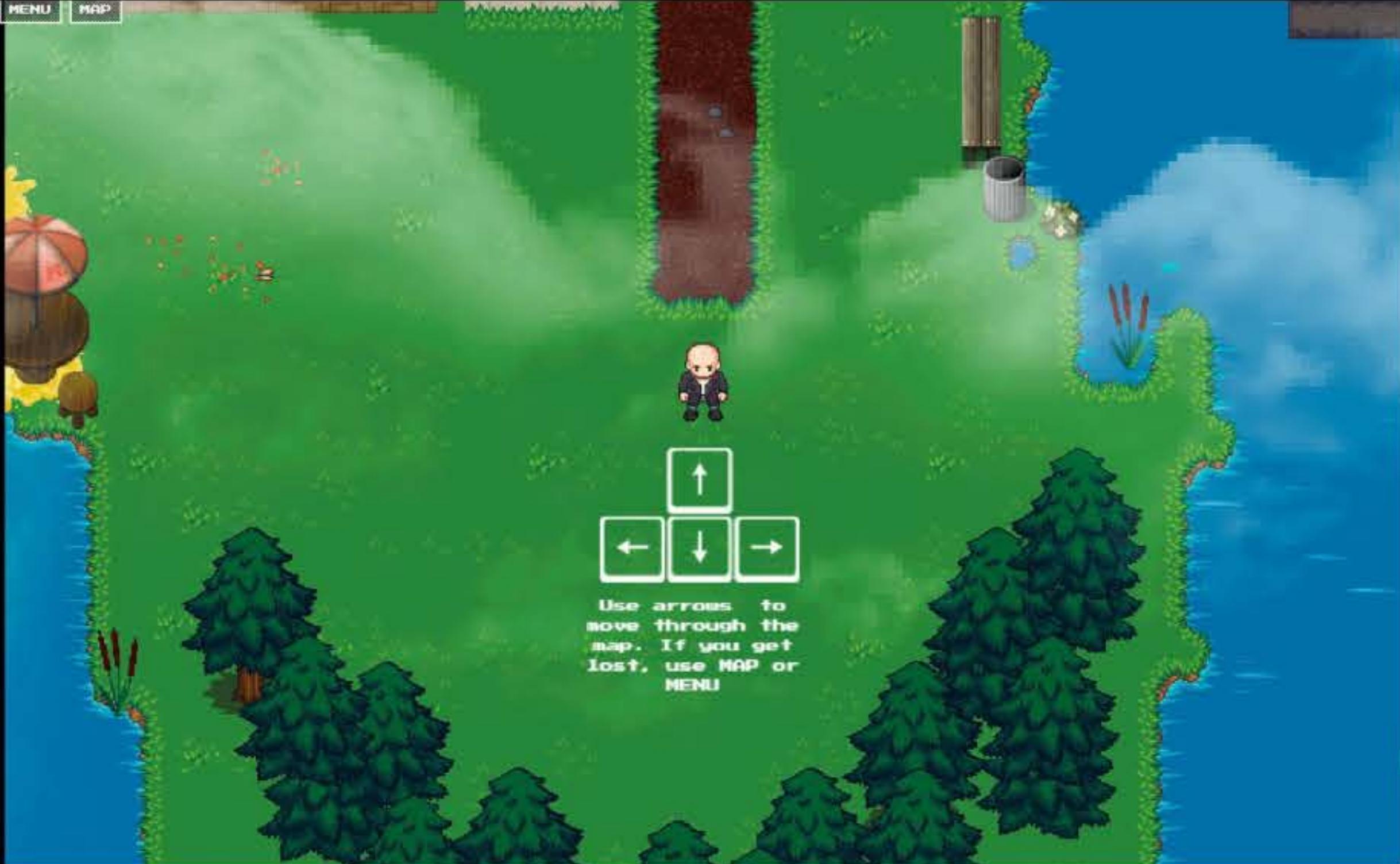
Tools

HTML5, jQuery, Phaser

Fonts

**abcABC
1234567890**

Press Start 2P by CodeMan38 is a bitmap font based on retro Namco arcade games and available via Google Fonts.

**Above**

Visitors navigate the site by walking the Peter Oravec sprite around a map via the arrow keys or an onscreen joystick.

Left

Numerous hotspots scattered throughout the world reveal key details about Peter and also hints on the site's build.

About me

From Lotus Notes to Javascript

My Cup of Coffee

**Left**

The Map overview speeds navigation with a top-down display, showing key areas that can be clicked to fast travel.



Code a content popup lightbox UI component

A user interface control that will automatically generate a popup lightbox to display targeted content

1. Document initiation

The first step is to initiate the document structure, which consists of the HTML document container storing sections for head and body. Firstly, the head section is used to load the external CSS and JavaScript resource created in steps 3 – 7. The body section is used to store the content elements created in step 2.

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

2. Body content

The page body content is made from a list of anchor links and a collection of hidden sections. Each anchor link uses the data-popup attribute to reference the ID of the hidden section to display when clicked. This feature will be activated through CSS and JavaScript in the following steps.

```
<ul>
<li><a data-popup="A">Page 1</a></li>
<li><a data-popup="B">Page 2</a></li>
</ul>
<div style="display:none">
<section id="A">
<h2>One</h2>
<button data-popup-close>Close</button>
</section>
<section id="B">
<h2>Two</h2>
<button data-popup-close>Close</button>
</section>
</div>
```

3. Links and lightbox

Create a new file called "styles.css". This step sets the anchor links to appear as links when they don't have a href attribute. The lightbox element that is to be created by JavaScript will have a lightbox class. This will be used to set the element to display with fixed

positioning that covers the full screen with a semi-transparent background colour.

```
.wa { color: blue; }
a:hover{ cursor: pointer; }
.popup{
  position: fixed;
  background: rgba(0,0,0,.75);
  width: 100vw;
  height: 100vh;
  top: 0;
  left: 0;
}
```

4. Lightbox window

The JavaScript will place a single element inside the popup container to be used as the content window. This element will also use absolute positioning, but will be positioned and sized to display at half width height in the middle of the screen. A white background is applied to make the window stand out from the semi-transparent background of the parent popup overlay.

```
.popup > *{
  position: fixed;
  top: 25vh;
  left: 25vw;
  width: 50%;
  height: 50%;
  background: #fff;
```

5. JavaScript: Link Search

Create a new file called "code.js". The JavaScript requires the body content to be available, hence all code is placed into a load event on the browser window. A querySelector search is used to find elements containing the data-popup attribute. A forEach loop is used to reference each of the resulting nodes to add a click event listener to execute the code in step 6 when clicked.

```
window.addEventListener("load", function(){

  let nodes = document.querySelectorAll('[data-popup]');
  nodes.forEach(function(item){
    item.addEventListener("click",function(){
      *** STEP 6 HERE
    });
  });
});
```

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

6. Popup initiation

A search for the target content is performed based on the ID referenced by the clicked element's data-popup attribute. A new container element is created, with the popup class applied – as defined in step3. A box element is also created to store a duplicate of the previously identified target element.

```
let target = document.getElementById( this.getAttribute("data-popup") );
let container = document.createElement("div");
container.classList.add("popup");
let box = document.createElement("div");
box.appendChild( target.cloneNode(true) );
*** STEP 7 HERE
```

7. Popup element

A search for a child element inside the box that uses the data-popup-close attribute is performed. This element is assigned a reference to the container under its popup field. A click event listener is applied to this button, referencing the popup field to remove the container when clicked. Finally, the box is attached as a child of the container – and the container is attached to the page's document body.

```
let closeBtn = box.querySelector("[data-popup-close]");
closeBtn.popup = container;
closeBtn.addEventListener("click",function(){
  this.popup.remove();
});
container.appendChild(box);
document.body.appendChild(container);
```



Title Loans - Autonomy

<https://en.autonomy.finance>

= AUTONOMY



= auto

TITLE LOANS AT A MONTHLY INTEREST RATE FROM 2.5%

Push and drag to the right

Designer:

Red Collar <https://redcollar.digital>



“Autonomy demonstrates the ease of its microloan products with a minimal yet bold site that defies gravity”

Colours

#AFC9CE

#FBE880

#273437

#F8FBFB

Tools

PHP, GSAP, Bitrix

Fonts

abcABC
1234567890

abcABC
1234567890

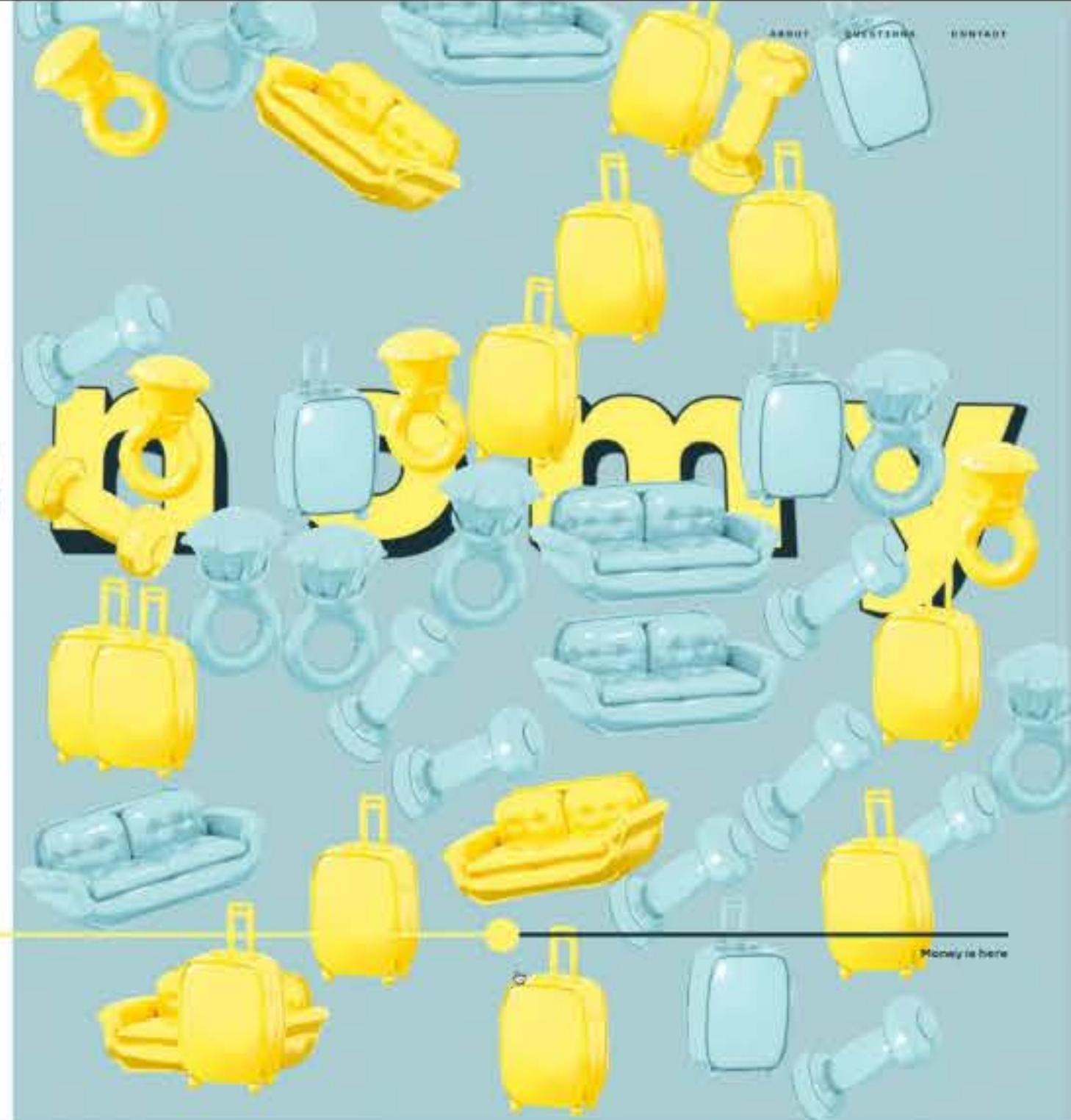
abcABC
1234567890

TT Travels font by TypeType is used exclusively in Regular, Bold and ExtraBold varieties.

=autonomy

TITLE LOANS AT A MONTHLY INTEREST RATE FROM 2.5%

Push and drag to the right



Left

An integrated loan calculator keeps application easy with an oversized text input field and two slider controls.

Above

The slider within the opening header fills half the viewport with floating objects that can be dragged and discarded.

How much money do you need?

\$ 500 000

For how long? 18 months

APPLY ➔

You will only need **3 documents** to get the money

1 • Your passport

2 • Vehicle license

3 • VRC



Left

3D typography uses CSS transforms to add movement, animating away from the position of the mouse pointer on the page.

Create a changing link cursor UI feature

A feature to change the appearance of the user cursor for navigation of content links

1. Document initiation

The first step requires the structure of the web page document to be defined. An HTML container declares the document to store sections for head and body. While the head section is used to load the external Javascript and CSS, the body is used to store the page content elements defined in step 2.

2. Link list

The example content consists of a list of links. Each link that is to have an icon uses the data-icon attribute to specify the name of the icon to display. This example sets the icon names as 1, 2 and 3 – which will be specified in the upcoming CSS presentation rules.

```
<ul>
<li><a href="#" data-icon="1">Link 1</a></li>
<li><a href="#" data-icon="2">Link 2 </a></li>
<li><a href="#" data-icon="3">Link 3</a></li>
</ul>
```

3. Cursor hover

Create a new file called "styles.css". The first set of CSS rules hides the browser cursor when hovered over a link using the data-icon attribute. Styling for the cursor element is also defined through the cursor class that will be applied to it. The cursor element will use fixed positioning to stay independent of the page scroll location.

```
a[data-icon]:hover{
  cursor: none;
}

.cursor{
  position: fixed;
  display: none;
  z-index: 999999;
  width: 5em;
  height: 5em;
}
```

4. Cursor icons

Each of the cursor icons will be made from a background image. Simple colours are used in this example, but you can load images by using url as demonstrated previously.

Each data-icon attribute corresponds to the attribute names defined in the HTML link – this feature will be activated in step 7.

```
.cursor[data-icon="1"]{
  background: orange;
}
.cursor[data-icon="2"]{
  background: blue;
}
.cursor[data-icon="3"]{
  background: red;
}
.cursor[data-icon="image"]){
  background: url(YOUR_IMAGE.png);
}
```

5. Create cursor

The JavaScript code is dependent on the document body being available, hence all code is placed into a load event listener attached to the browser window. This step focuses on creating the new cursor, made from a span element using the cursor class. The cursor element is attached to the document body after being created.

```
window.addEventListener("load", function(){
  var cursor = document.createElement("span");
  cursor.classList.add("cursor");
  document.body.appendChild(cursor);
  *** STEP 6 HERE
});
```

6. Mouse movement

A "mousemove" event listener is attached to the document for detect of any mouse cursor movement. This triggers a function to modify the left and top styles of the cursor element created in step 5, with a new position based on the real mouse

cursor location. A 5 pixel adjustment is required to make the CSS icon element appear where the real mouse cursor is focused.

```
document.addEventListener("mousemove", function(event){
  cursor.style.left = (event.clientX+5)+"px";
  cursor.style.top = (event.clientY+5)+"px";
});
```

*** STEP 7 HERE

7. Link events

A search is performed for each link using the data-icon attribute, with a forEach loop being used to reference each item found. All link items have a mouseover event listener applied to activate its visibility and copy the name of the icon from the link to the cursor's data-icon attribute. Similarly, link items also have a mouseout event listener to hide the cursor and reset the its data-icon attribute.

```
document.querySelectorAll("a[data-icon]").forEach(function(item){
  item.addEventListener("mouseover", function(){
    cursor.style.display = "block";
    cursor.setAttribute("data-icon", this.getAttribute("data-icon"));
  });
  item.addEventListener("mouseout", function(){
    cursor.style.display = "none";
    cursor.setAttribute("data-icon","");
  });
});
```



KINSTA: WORDPRESS HOSTING MADE EASY

Get the best speed, security and scalability for your website with this quality hosting solution



For artists and designers, having a strong online presence is vital in order to succeed in the ever-competitive creative industry. But while building a beautiful home in WordPress is easy, maintaining it online can be tricky for those unfamiliar with the ways of the web. Until now.

Introducing Kinsta, a WordPress hosting service that makes managing your website a cinch.

A highly affordable solution, equipped with all the latest web technologies, Kinsta not only helps WordPress websites deliver fast load times and have less downtime, it also takes tasks such as securing, caching and backing up your site off your hands, saving valuable time. And with Kinsta support staff being some of the most experienced WordPress developers in the industry, there's very little risk of anything going wrong.

So whether you're looking to set up your first website or are an experienced web developer without the time or resources to maintain your online space, Kinsta has you covered every step of the way.

A SERVICE LIKE NO OTHER

But why Kinsta? With the team having worked with WordPress for the last decade, not only are they experts

“Kinsta boasts state-of-the-art technology to make your websites load at lightning speed”

In the software, they have also developed new and improved ways of WordPress hosting.

One of the main things that separates Kinsta from other WordPress hosting providers is its use of the Google Cloud platform. This means its sites are hosted on and supported by the same state-of-the-art infrastructure created and maintained by Google (think Google Search, Google Maps and YouTube). On top of that rock-solid infrastructure, Kinsta boasts some impressive architecture, which includes state-of-the-art technology to make your websites load at lightning speed.

Not satisfied with the performance and usability of existing control panel solutions, Kinsta replaces the familiar control dashboard cPanel with its own MyKinsta dashboard. Designed with beginners in mind, the

MyKinsta portal is also packed with advanced features and tools that experienced developers will love. Easy to use and navigate, the portal allows tasks such as creating new WordPress websites, updating PHP version of your site and managing backups.

The WordPress hosting service Kinsta provides is second to none, evidence of which was seen in an independent test from Review Signal, which, for the fifth year in a row, rewarded Kinsta with the Top Tier WordPress Hosting Performance award.

So if you want to spend less time worrying about your website and more time being creative, Kinsta is the perfect hosting solution.

For more details visit kinsta.com

Kinsta is a managed WordPress hosting provider that helps take care of all your needs regarding your website. We run our services on cutting-edge technology and take support seriously.

[VIEW PLANS](#)

Features

These form the basis of our service.

- Fully managed
- Secure like Fort Knox
- Free migrations
- Ultimate speed
- Daily backups
- Google Cloud Platform

Features

Speed. Security. Migration.

MyKinsta is site management reimagined

Take MyKinsta For a Spin

Premium features on all plans

30 day money-back guarantee

In the news



MYSTIC MAKING

Celebrating an illustrious seventh birthday, could Italian agency illo prepares for its big Turin bash by building a Tarot reader capable of predicting the future?

Tarot-o-Bot

Project url illo.tv/tarotobot

Agency illo

Agency URL illo.tv

Follow @illotv

One of the toughest and most coveted skills within any design discipline is the ability to predict the future. Not for forecasting the lottery numbers, although that would be a helpful trick, but in order to predict and influence the hottest style trends. This is useful for communicating effectively to audiences, capturing attention and appealing to those "fashionable" sensibilities that maybe instill trust in a message. Like much of the tech trade, web design as an industry is fixated on foretelling the next big thing, and that's the premise behind Tarot-o-bot. An in-house project produced by Italian agency illo, the purpose was to celebrate their seven-year birthday and promote a big party being held in Turin. The theme was to be "Prediction", and with seven years regarded as quite a mystical number, it seemed appropriate to not only reflect illo's past but also into an exciting future. Boasting an international team of 12 staff and specialising in areas of motion design, illustration and set design, their portfolio includes clients such as Bloomberg, Airbnb, Ferrero and Snapchat. Having come a long way since being founded by creative partners Ilenia Notarangelo and Luca Gonnelli in 2012, it seemed like an ideal moment to speculate on what's to come for motion design and communication.

"That's how Tarot-o-bot came into being, creating an online experience that can predict the digital future," the team begins. "At the touch of a button, users can trigger the draw of three differently illustrated special tarots. Tarot-o-bot reads the data beyond the three Sacred Cards and generates a special and ironic prediction over design,





PROJECT STATS	TEAM
PROJECT DURATION 10 weeks	Ilenia Notarangelo <i>Partner & Creative Director</i>
NO. OF PEOPLE INVOLVED 4	Luca Gonnelli <i>Partner & Business Director</i>
	Giovanna Crise <i>Brand Developer</i>
	Nima Farzaneh <i>Data Scientist & Web Developer</i>



Tarot-o-bot

7

In the future, stock footage videos will be created by sophisticated AI software. They will still suck.



In the future you won't be able to secretly modify a deadline due to blockchain.

#fuck #bitcoin

Share it Another one
Link copied! (https://vimeo.com/14468888)



"Tarot-o-bot reads the data beyond the three picked tarots to generate an ironic prediction about the creative industry"



Tarot-o-Bot

7

In the future, a new law will legitimate designers to apply an additional preliminary fee to every client whose cousin is a graphic designer

#mycousincandoit

Share it Another one



art, tech and creative life." Only too happy to apply illo's characteristically minimal but colourful aesthetic and clarity in storytelling, all that remained was to shuffle the deck to see what's in store... And party!

Magnificent Seven

"Every other year we celebrate our birthday as a studio with a party, creating a moment of networking for the creative community in our city, Turin, and also to meet some of our internet friends in real life," says the team. "They call these events "Brillo" because in Italian this means both "I sparkle" and "drunk", which we're reliably told are two important features for an illo party! "This year we've celebrated our 7th birthday as a studio," explains Brand Developer Giovanna Crise. "Seven is a number with different and important meanings in most cultures and religions, including representing a 'totality in movement' like the end of a cycle, such as the week or the seven musical notes, then the start of a new one." Keen to give each party a theme, the team says it took some brainstorming and a random word generator to land on a prediction: "We wanted to create an experience for both the virtual and physical world in order to also involve the people who wouldn't be able to attend our party. So we started wondering what that experience could be, keeping in mind the chosen theme." Musing on the symbolism of the number seven, the illo team began researching the mystical world and familiar icons that might be useful. Were there any recurrent visuals, colours and styles that could be appropriated for the project? "One of the most emblematic elements of the prediction world is the tarot deck, that allows predicting the future," reveals Partner and Creative Director Ilenia Notarangelo. "Creating an illo version of a card deck got the whole team inspired immediately!"

Greetings Cards

Taking this tarot concept, the team wanted to rework the genuine card types and illustrations to better reflect technology and design predictions. In total they created 14 different card images spanning seven sacred roles and seven scared concepts, not necessarily intended initially for web use. "From the very beginning, we wanted a tarot reader at the party able to predict the future of the creative industry using our cards. The aim was to create a fun and ironic experience for guests, but we soon figured out that we wanted to extend it online, to involve our internet friends as well! So we decided to create Tarot-o-bot," says the team.

Just like the organisation of the party event, this was a studio project where illo were effectively fulfilling the roles of client and agency. No strangers to undertaking design work

Left to right, clockwise from back:
Cristina Pasquale (Art Director), Laurentiu Lunic (Motion Designer), Nima Farzaneh (Data Scientist), Luca Gonnelli (Partner & Business Director), Ani Karamanukyan (Studio Coordinator & Producer), Pinguino, David Cubitt (Motion Designer), Matteo Ruffinengo (Motion Engineer), Giovanna Crise (Brand Developer), Ilenia Notarangelo (Partner & Creative Director), Arianna Cristiano (Illustration Lead), Carla Giola (Visual/Set Designer), Sofia Buti (Illustrator).



without a direct commercial purpose, they produced a temporary pop-up shop in time for Christmas last year. The illoxmasminimarket (<https://illo.tv/xmas>) project allowed them to play with illo's most iconic colours and shapes to create their own versions of some typical festive ephemera like greetings cards, decorations, candles and recipes etc. "One of the main reasons we love doing these kind of projects is that they let us experiment with different fields of design, normally far from the motion and set design world we are used to," explains the team. "It also allows us to reinforce our studio culture and create other situations of team building which is very important for us."

Tarot Time

Leveraging the prediction theme and their own interpretation of the tarot deck, illo wanted to forge a simple experience where users could receive various predictions for every refresh of the webpage. Taking the Cards Against Humanity game as a reference, they went on to write around 100 different forecasts, starting from just as many card combinations. "On the homepage, users are introduced to Tarot-o-bot and are pushed to click on the black button shown on the screen. That action triggers the draw of three different tarots, presented face down," says the team. "The user can flip the cards by touching them, discovering which Sacred cards will help predict the future. Then Tarot-o-bot reads the data beyond the three picked tarots and generates an ironic prediction about the creative industry."

Developed internally alongside all the UX and UI work, the visual design centred around the classic tarot reader's table - albeit with a more colourful twist. With the cards illustrated before the front-end design work commenced, the decision was made to add extra thematic elements like stars, circles and all-seeing eyes. "The website is characterised by a high colour contrast that allows the animated cards and the sentences to emerge from the background," says the team. "The colour palette was chosen to be representative of the illo.tv style and to be easily recognisable, so we opted for saturated and vivid colours. The visual style is extremely minimal but full of little details, like the grain to give depth to the characters, and a customised font."

Disciplined Freedom

The hugely beneficial exercise of finding time for in-house projects is a common practice for good web design studios. In illo's case, the team describe a valuable feeling of being unthethered and not restricted in terms of creativity, a time to freely express their own personality. "An important feature that defines these works is that we have total freedom during our creation process," reveals Giovanna Crise, illo's Brand Developer. "Normally we need to cross-reference with the client, who will have, of course, some restrictions, such as a style, a colour palette or a budget etc. Usually, with the client, there is also a strict and determined timeline, one we have to stick to and the work is then developed thanks to regular feedback. Instead, in a studio project, we set internal deadlines, but normally we have to give priority to the clients' projects, so it's really important to have a really good organization of our work in order to deliver everything on time."



Tarot-o-Bot

illo.tv



The Tarots



Dear **Tarot-o-bot**, mother of tech updates and design trends, what do you foresee for our future?



Press Enter



Join Brillo7!



Favourite Predictions?

Talking to the illo team, you get an undeniable sense of the pleasure and pride they felt throughout the build of Tarot-o-Bot. A big part of this was of course the humour afforded to writing the various predictions that the cards would combine to produce. Here are three of their favourites, offering a playful look into what the future may hold..

"2021: the improved version of the Gmail auto-complete feature will write to your boss that you quit... because she knows what's best for you."

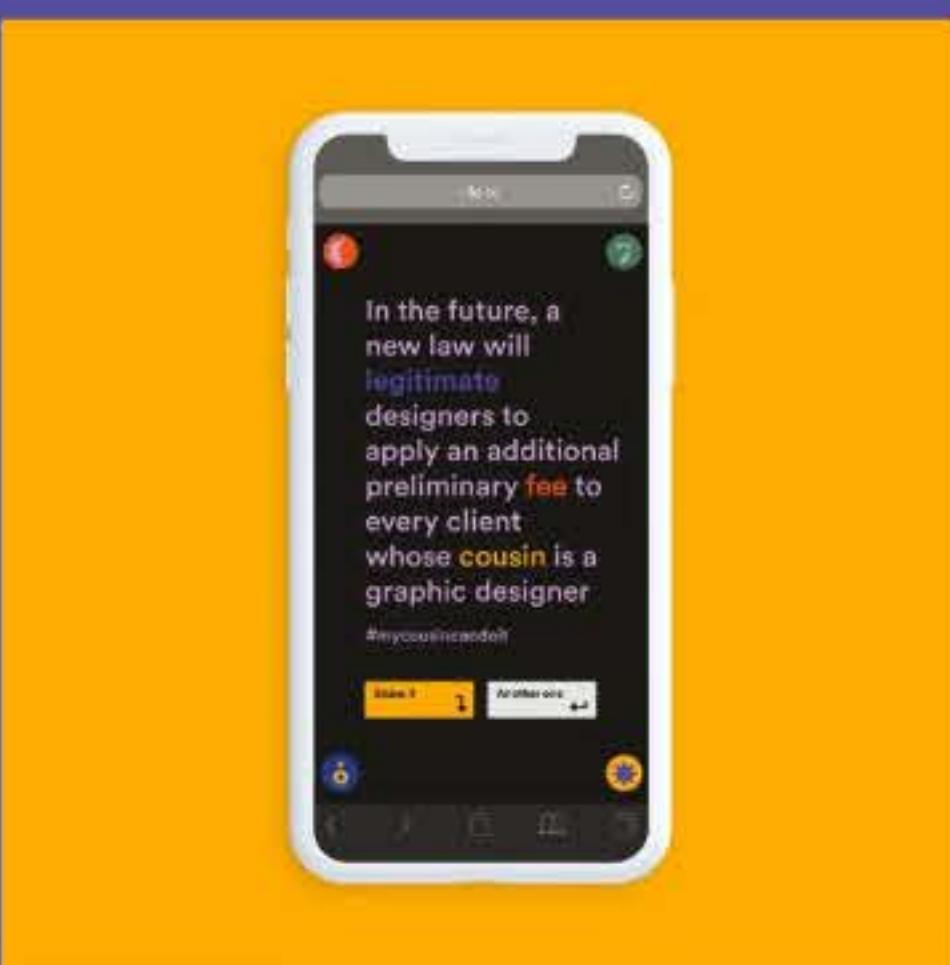
From The Update, The Artificial Intelligence and The Patron combination.

"In the future, a new law will legitimate designers to apply an additional preliminary fee to every client whose cousin is a graphic designer."

From the draw of The Patron, The Conspiracy and The Designer cards.

"The stars forecast that interns that will not meet deadlines will be publicly shamed on Instagram."

Combining from The Intern, The Deadline and The Visibility tarots.



SITE HIGHLIGHT

The illo team offer us a quick-fire honest response for which part of Tarot-o-bot they're most proud of when looking back on the project as a whole.

"One of the results we are most proud of is totally the predictions! We really enjoyed writing dozens predictions for Tarot-o-bot together – it was a beautiful occasion of team-building and to lough together wondering about impossible and funny Motion Design futures."

Animation Fixation

Being a studio focused on motion design, animation was also a really important element for the illo guys. They wanted those kinds of details worked in and embedded too, beginning with the cards themselves. Each tarot was animated within a loop to provide additional movement during site navigation, with users revealing extra little nuggets of interaction as they browse. "For example, to find out which tarots have been drawn, users shake each card with a mouse-over action and then flip them by checking on the back of the cards," says the team. "Other important visual elements are the floating particles presented on the black background of all the prediction-related pages. Users can easily interact with these elements simply moving their mouse around the screen or clicking on dark background letting other stars and dots appear!"

Opting to create their own hand-coded platform for the site's build, the team knew that much of the practical development challenges would be confined to the frontend finesse. "One of the most important aims was to be able to recreate a meaningful mobile experience that continued to be engaging, with different and fluent animations," says the team. "In fact, as mentioned before, we're mainly an animation-driven studio, so beyond the website's quite simple structure we experimented for the first time with four different types of animation techniques within the same experience. From classical GIFs and MP4 embedding straight from After Effects, to JavaScript for the background-particles interactions, to CSS to turn the cards and move them around the screen, to using Bodymovin."

Multiple Readings

One of the main technical issues they faced was combining these animation techniques into a smooth, unified experience across all browsers. "At times there are several web interactions presented in the same page, such as the shaking of the cards, their rotation and their movement, which could be tricky to accomplish alongside the interactions with other web elements such as the particles and other animated web elements," explains the team. "This was made more complex by the desire for Tarot-o-bot to be random and relatively

unique for each user that arrives. The solution was to effectively generate a whole new outcome each time using some imagination and ingenuity. Every time a user clicks on the main page button to foresee his or her future using our tarots, a new experience with randomly extracted card combinations and the corresponding text and hashtags is loaded. We managed that by pre-defining a dataset, including the phrases, the hashtags, the highlighted words and the card combinations. Each time a user loads the prediction page, a script will read that data set and randomly select a record from that data set, and hence showcases the corresponding cards, text and hashtags on the webpage."

Shared Success

As a digital experience, Tarot-o-bot as a project was really motivated by raising brand awareness for illo and, of course, its seventh birthday celebrations. However it was also promotional opportunity for beyond just preempting the party event, so a communication strategy was formulated to support the website's eventual launch. "We started posting on Twitter some predictions a few weeks before the party, without a real contest and without any reference to Tarot-o-bot or to our Prediction night," says the team. "In the meantime, we shared animated cards on our Instagram stories each day, revealing the characters and names of our tarots. This caused excitement among our followers and friends, who started to ask for more information about the project." This drive for audience engagement and curiosity was joined by a video teaser, still available on Vimeo, which unveiled Tarot-o-bot to the web world. Geared to social sharing, the experience itself had those functions built-in to let users copy prediction text and share it among friends. "In parallel, we started re-posting the Instagram stories in which people tagged our profile, sharing their own prediction. We wanted this interaction to be engaging, so we also commented ironically on the prediction in each story. Overall it commanded a huge time investment and a lot of hard work, but we've been rewarded by the many awards we received, such as the Mobile Experience, Developer Site and Site of the Day awards by Awwwards, and the several web magazines have written about us! The work done was totally worth it!"



From top: Giovanna Crise, Ilenia Notarangelo, Luca Gonnelli, Nima Farzaneh.



NEVER MISS AN ISSUE!



Issue 292

HTML in 2019, 17 pro CSS animation tips and techniques, speed up and optimise WordPress, create a circular carousel and 3D panoramas

SAVE UP TO 70%
ON A PRINT & DIGITAL
SUBSCRIPTION
[SEE PAGE 28](#)



Issue 291
The science of UI design, 7 pro SEO strategies, animate with mo.js, code web components with Stencil, speed up JS and fast builds with Grav CMS



Issue 286
35 Hot design tools, animate SVG with CSS, go web app with PWAs and performant design, voice-controlled UI and test with Jest

GOT AN APPLE DEVICE?

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

apple.co/1hsGYgl

CATCH UP ON ANY ISSUES YOU'VE MISSED BY DOWNLOADING
A DIGITAL EDITION OR PURCHASING IN PRINT bit.ly/2rh73Xk



Issue 290

State of the art CSS, 10 Pro CodePen secrets, say hello to Angular 8, create dynamic SVG art, animate with Matter.js and customise Bulma



Issue 289

50 Best JavaScript libraries, frameworks, APIs and tools, power up your social, tweak your SEO, and what's new in NodeJS 12



Issue 288

Create stunning 3D in the browser, 25 UI design secrets, rapid builds with Gulp, animate with react-spring and browse web pages with voice



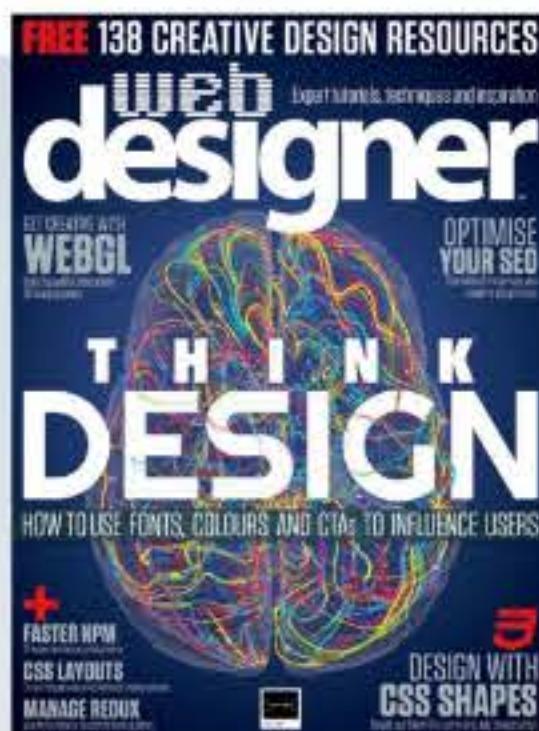
Issue 287

Build your own web components, 8 steps to inclusive design, custom WordPress, stop the bots with Google reCAPTCHA and scroll effects



Issue 285

Prototype with Adobe XD, 20 top UX tools & resources, how to sketch and storyboard, design UX-friendly forms and wireframe with Sketch



Issue 284

Think fonts, think colours, think design, and responsive layouts, manage Redux with Immutable.js and get productive with npm



Issue 283

JAVASCRIPT 2019 - 25 frameworks, tools and APIs, WordPress 5, CSS liquid effects, build a webVR scene and React & Redux



Issue 282

What you need to know to become a CSS pro: animation, Grid, Sass, optimisation, methodologies, hot new properties and APIs

PREFER TO READ ON ANDROID OR PC OR MAC?

Web Designer magazine is available on Google News and Zinio

tinyurl.com/yalm3wul



bit.ly/2xPbv4p





Information as Art

At **The Visual Agency** information becomes a beautiful, compelling visual feast. Applying technical prowess with a **design sensibility** born out of the age of data, this agency shapes our perception with striking images that **communicate** their messages with innovative style and clarity



THE VISUAL AGENCY

Who The Visual Agency

What Infographics, Interactive, Data Visualization, Static, and Motion Graphics

Where Via Panizza 7, Milano, 20144, Italy

Web <https://thevisualagency.com>

Key clients Brembo, Eni, Corriere della Sera, Barilla, Luxottica



Founded in 2011, The Visual Agency works in close collaboration with Density Design Lab. CEO Paolo Guadagni came up with the idea to create an agency that combines technical know-how and data-science with creative design. After getting a master's degree in physics, Paolo started his working career as a programmer in the 1980s. After 12 years at Microsoft, he decided to start digital PR - the first internet PR agency in Italy.

Paolo conceived the idea to create an agency specialising in infographics and data visualisation after meeting Professor Ciuccarelli, head of Density Design Lab at Politecnico di Milano.

In November 2011, a few months after his encounter with Professor Ciuccarelli, Paolo founded The Visual Agency: a team of designers specialised in data and information visualisation. Initially part of Alchimia Holding, the agency became independent in 2013. Paolo also became partner and co-founder of iCorporate, an Italian PR company.

"The first thing I did was name the agency," said Paolo. "The idea was to choose a name that instantly communicated what The Visual Agency could create for our clients. Back in 2011, it was much easier to get the domain name of your choice. Luckily, the .com that we wanted was available."

As the web is a visual medium, Paolo explained the importance of having a strong presence online. "The website is the business card of the modern era. We use our website to showcase the latest work we have created for our clients, which also illustrates the design and technical expertise we have. We update the site regularly to expand and extend the content to keep the site fresh. This is time-consuming and wouldn't be possible without the continuous work of a dedicated visual designer."

The Visual Agency has been able to build a diverse range of clients. Matteo Bonera, Creative Director, outlined their approach: "We do believe that you have to pitch for work continually. We pitch almost daily, with a strong accounts team who go into the marketplace and actively contact potential clients."

"Over the years, of course, we have built a reputation. This does mean we also gain a great deal of work from recommendations and repeat business from existing clients. As a specialist in visual communication, increasing numbers of businesses are realising they need professional help to develop and communicate their ideas and messages in a range of visual formats."

As an agency focused on visual communication in all its forms, is there a piece of work that encapsulates the ethos embedded in The Visual Agency? "On the occasion of the 500th anniversary of Leonardo da Vinci's death, we fully digitalised his greatest work, the *Codex Atlanticus*," Matteo explained to *Web Designer*. "We think this project truly represents our ethos."

"First of all, it allows us to globally communicate a cultural heritage that otherwise wouldn't be available," Matteo continued. "Other than that, it has great importance from a technical point of view too. The project requires a wide range of skills: it started with the creation of an extremely large database. Then, due to its complex and full-ranging nature, it involved the work of information designers, graphic designers, motion designers, programmers and marketing officers."

The diverse nature of the projects The Visual Agency complete for their clients often means a highly flexible development schedule. Their Design Director Francesco Roveta explained their approach: "A large-scale project can take from six to ten weeks. The first step consists of content analysis, which includes a target audience and tone of voice

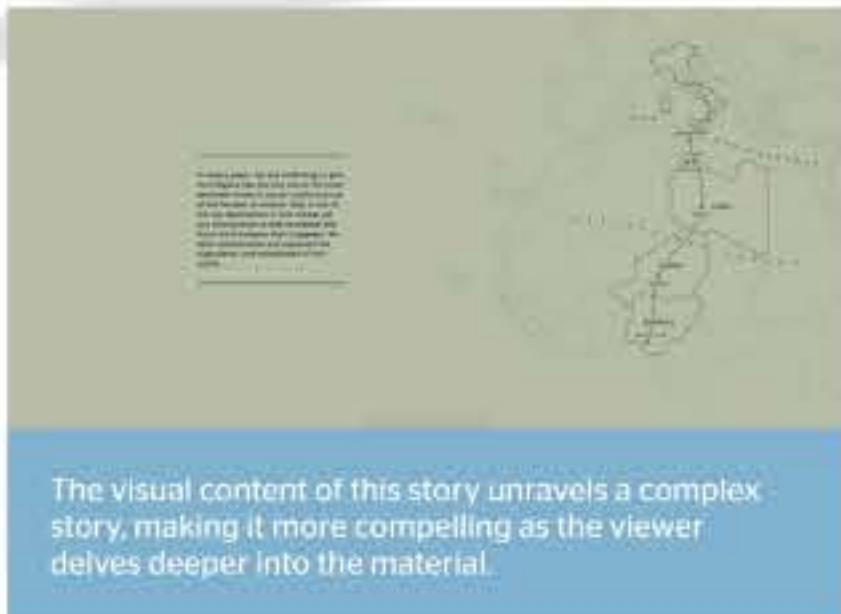


Paolo Guadagni

Co-Founder and CEO

“ Since our work is an intersection between communication, design and technology, it's essential for all our staff to stay up-to-date with all new developments to be competitive. We believe that excellence cannot be achieved through an exclusively top-down approach but rather through a top-down and bottom-up mix ”

Deconstructing the story with strong visuals and a striking limited colour palette ensures the information presented is always engaging.



The visual content of this story unravels a complex story, making it more compelling as the viewer delves deeper into the material.



Visualeyed

<http://visualeyed.com>

Visualeyed is an online magazine offering a new perspective and inspiring new forms of visual communication in journalism.

The magazine is conceived by Tommaso Guadagni, founder of Dalk (www.datatalk.it) – the first data-driven storytelling agency in Italy. Visualeyed hosts interactive articles designed by The Visual Agency, with a focus on environmental and social issues. Visualeyed periodically shares the most exciting data and visual journalism case studies published by the most relevant newspapers worldwide.

Recent examples of content The Visual Agency created include "Satellites". According to the Index of Objects Launched into Outer Space, there are more than 4,256 satellites currently orbiting our planet. Of those, only 1,419 are operative while the remaining 2,837 don't return data to the Earth any longer and are therefore counted as space debris.

The interactive visualisation displays all the operative satellites launched in orbit since 1989, updated to June 2016. The filters allow you to investigate a satellite's primary purpose, users, owning countries, and other specific information.

Another example is "Permette Signorina". In recent years, the sex trafficking of girls from Nigeria has become one of the most persistent flows of human traffic and one of the hardest to combat. Italy is one of the top destinations, yet this phenomenon is well concealed and much more complex than it appears. Visualeyed reconstructed and explained the organisation and mechanisms of this trade, with a visual explanation guiding you deep into the story.

Finally, there's "Russiagate", a long-form article that reconstructs the case on the possible interference of Russia in the US presidential elections. This is displayed through an in-depth overview of the events and the protagonists at the heart of the situation.

THE VISUAL AGENCY



The visual content that supports *Corriere della Sera's* La Lettura section ensures each feature is compelling.



Data visualization for Corriere della Sera

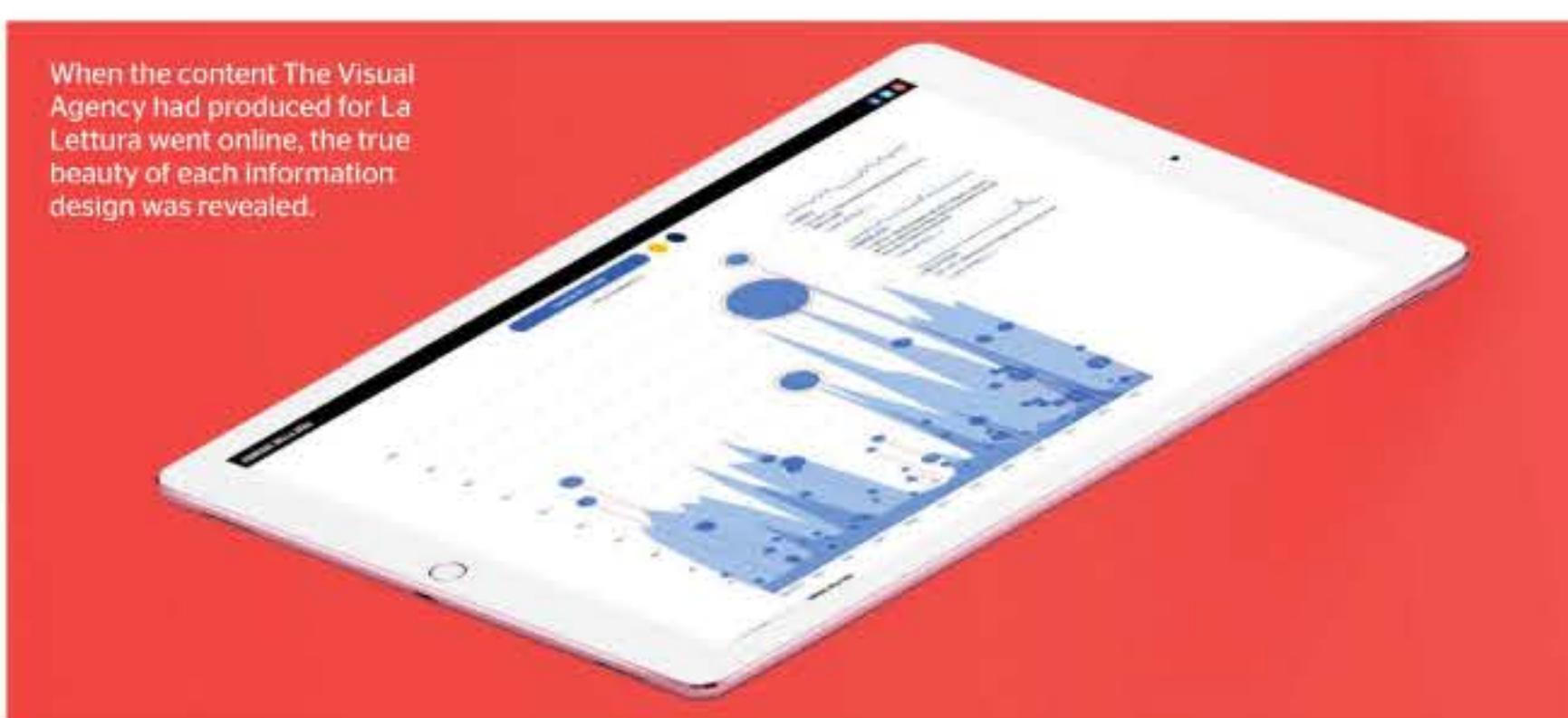
www.corriere.it

Since 2013 we have suggested content and designed cutting-edge visualisations on current affairs, politics and culture for one of Italy's leading newspapers, the *Corriere della Sera*.

The work includes the selection and suggestion of content to the editorial staff, the analysis of the data collected and the design of the visualisations on topical issues, politics and culture.

A series of numerical correlations become the protagonists of a new way of presenting information, which involves the reader. The most advanced graphic design resources meet journalism, and the coordinates of a new aesthetic of knowledge are drawn, with a new idea of creativity.

The visual data of La Lettura - a section of the Sunday insert of the *Corriere della Sera* - often includes infographics with each feature. In February of last year, this content appeared online for the first time, including an interactive visualization in collaboration with Google where we tracked and visualised what Italians searched on Google in 2017. The content won the award for Information is Beautiful at the Londra and ED Awards in Vienna.



definition, some copywriting activity and an early design of a storytelling concept."

The actual design developing then starts. This is the hardest and most complex part of the process. It begins with a benchmarking activity and ends with a proposal of a mock-up, navigable through a prototyping software.

Once adopted, the proposal turns into the real application through the backend and frontend development. A wide range of professionals are needed to carry out a project of this type: a backend and frontend development team, a design team composed of information design and UX / UI designers, and, if needed, a motion design team.

The Visual Agency has developed the tools they need to create world-class visual content. Matteo explained their approach: "Since the agency started, Adobe played an important role thanks to its wide range of products. Unfortunately, it wasn't enough to fulfil the needs of our designers to visualise data in the most cutting-edge ways.

"Our approach is to take the Adobe applications and integrate them with specific tools for data visualisation like RAWgraphs - a D3.js-based charts generator developed by Density Design Lab, and Gephi, a network graphs engine. To date, software like Sketch, Zeplin and InVision are fundamental to support a lean production flow, but we're still experimenting with new ways to help us with charts production with Open-Source and custom graphics libraries."

Daniele Berto, The Visual Agency's CTO, added: "It's true that HTML, CSS and JavaScript will continue to be the building blocks of the web, with technologies such as jQuery now being resigned to the past. The design scene is dominated by JavaScript and the related frameworks built onto it such as React, Angular and Vue; we can also add some special-purposes libraries as D3 for data visualisation and three.js for 3D animations."

Daniele continued: "Regarding the future, it's not easy to make predictions in this sector. Every year some web components appear on the scene and are defined as de facto standards. The web development scene is confused among independent projects and others supported by the big names in the industry. In the future, we hope that the chaos will reduce so that every single component does not need to be re-implemented whenever a new framework appears.

"The panorama of development tools and information technologies is continually changing. Every day we are looking for new frameworks and components that improve the solutions we create. Regarding frameworks, we recently used Svelte, realized by *The New York Times* programmer Richard Harris. This framework inspired us because it offers some great functions focused on interactive visualisations and long-form production.

"And an exciting library we ran into is Lottie-Web. We used it to develop the intro of our *Codex Atlanticus* website. It allows us to



Matteo Bonera

Creative Director

“ Communicating complexity is at the heart of our professional activity. We believe in making that information accessible. This process requires a wide range of competencies to be harmonically integrated: design, data and knowledge visualisation, art direction, and storytelling are only some of them ”



Agency breakdown





integrate a motion graphic video with a website and to customise the fruition of a vector animated video into a website."

As the content that The Visual Agency produces could be viewed on several devices, Matteo outlined how they handle these channels: "Depending on the nature of the project, several solutions for the different devices are studied. In cases where there is a lot of textual content and the visualisations are static or not that complex, such as long-form articles, the responsive design works properly and is the best solution.

"It is also necessary to design interface and web browsing solutions that ensure a satisfying experience, similar to that on the desktop. In other cases, especially for those projects that require complex or interactive visualisations, we design customised interfaces completely different from the desktop version for every device. In these cases, responsive design can't fit the needs, and the visualisations are designed based on the device features."

With so much content now consumed on social media networks, The Design Agency pays close attention to how their visualisations appear. The agency's Social Media Manager Nicola Petrus explained:

"As an information design agency, we don't provide social media and advertising related solutions in general. However, we recently decided to give more importance to our social media channels. We regularly monitor and publish content on Facebook, Twitter, LinkedIn, Instagram and Medium.

"Every channel is useful for different targets and goals, and not all are suitable for driving traffic to the main website. In any case, social media can be used in various ways, depending on the goals of a campaign or a publishing plan: increasing the followers, the engagement or the traffic to the website are some of them. We think that good social media management is an added value in terms of brand awareness. Furthermore, the social media profiles act as a company cover letter, as they are usually the first thing a client sees."

The specific nature of the work The Visual Agency produces has meant gathering an eclectic group of professionals. Paolo explained his approach to recruiting: "Since the agency's birth, we're always looking for dynamic, clever and resourceful employees, both in the production department and in marketing. At the moment, the team consists of 20 people, including employees, trainees and apprentices.

"We are lucky being based in Milan. This is also the location of the Density Design Lab and the research lab of the Politecnico di Milano, where a renowned communication design degree course is available. These institutions ensure a steady flow and generational turnover of talented young designers with excellent backgrounds in information design.

"Also, since our work is an intersection between communication, design and technology, it's essential for all our staff to stay up-to-date with all new developments to be competitive. We believe that excellence cannot be achieved through an exclusively top-down approach, but rather through a top-down and bottom-up mix. Therefore, we have implemented individual and collective learning initiatives such as sharing experiences and constant technological updating

"The data visualisation industry is constantly changing. There's space in the market with demand for these services growing. To anyone who wants to step into the industry, our advice is simple and clear: don't expect the business and the opportunities to come to you. You have to go out and find your own."

Codex Atlanticus

www.codex-atlanticus.it

In 2019, on the occasion of the 500-year anniversary of Leonardo da Vinci's death, The Visual Agency unveiled its collaboration with the Veneranda Biblioteca Ambrosiana - a renowned museum in Milan. The project was an innovative data-visualisation instrument that enables the general public to experience the Codex Atlanticus, one of da Vinci's most celebrated masterpieces, without boundaries and in a way that has never been possible before.

This self-initiated bilingual project in Italian and English offers a panoramic overview of the Codex Atlanticus combined with drill-down and analytical functions to explore this epic work in more detail.

Rearranging and filtering the pages of the Codex uncovers insights into Leonardo da Vinci's thought processes. Even though some of Leonardo's legacy

has been lost, the major part of his drawings and writings have survived thanks to Pompeo Leoni, an Italian sculptor from the 17th century, and Augusto Marinoni, a contemporary Italian historian.

Leoni first assembled all of the 1,119 loose pages and bound them into what we now know as the Codex Atlanticus. In 1970 the philological study by Augusto Marinoni provided - in addition to a critical analysis of the Codex - a list of 140 topics with page references and an estimated dating for each sheet. The Codex Atlanticus is not accessible to the public in its entirety, as the Biblioteca Ambrosiana can display only a small number of pages at a time.

We believe our design will open new frontiers in the in-depth study of this fundamental scientific volume and will allow users to explore first-hand the work of one of the greatest thinkers in history.



With a steady stream of high-profile clients and a drive to transform how information is presented, the future looks bright for The Visual Agency. Paolo said: "The agency is in continuous expansion. We have just restructured the office, and we are widening our team. For the coming years, the main goal is to expand our services abroad.

"First of all, we're entering the German, Austrian and Swiss market, thanks to the work of our German-speaking Client Director, who is establishing a contacts network in the United Kingdom and the USA too. Our production will not diversify, quite the contrary. This path will lead us to become data visualisation specialists."

As the world is awash with information, making sense of this vast array of data is at the heart of The Visual Agency. Built on a love of information and how its meaning can be communicated to an audience, this agency develops information spaces that are engaging and exciting to inhabit. Information becomes art in the hands of The Visual Agency.

**THE
V I S U A L
A G E N C Y**

Via Panizza 7, Milano - Italy
<https://thevisualagency.com>

Founder Paolo Guadagni
Year founded 2011
Current number of employees 20

Services Data Visualization, Infographics, Motion graphics, Interactive Dashboards, Data journalism



The Visual Agency used its skills to enable the *Codex Atlanticus* to be easily accessed by applying several filters to the data.

Accessible at the Veneranda Biblioteca Ambrosiana, the *Codex Atlanticus* is a tour de force of visual presentation.

Timeline

2011

November 17th, Paolo Guadagni founds The Visual Agency
Number of employees: 2

2012

The Visual Agency team Matteo Bonera (Creative Director), Giulia De Amicis (Information Designer), Paolo Guadagni (Co-Founder), Francesco Roveta (Design Director) and Mir Shahidul Islam (CTO) begin work on its first data visualisation projects
Number of employees: 4

2013

For the first time, a data visualisation is published in *Corriere della Sera*
Number of employees: 6

2014

The Visual Agency rebrands itself and wins its first award
Number of employees: 9

2015

The TVA Academy starts to teach training courses on data visualization
Number of employees: 10

2016

Tommaso Guadagni launches visualeyed.com and TVA partners with parcodiyellowstone, a brand identity and web design studio
Number of employees: 11

2017

The Visual Agency becomes Eni official supplier and carries out its second rebranding
Number of employees: 12

2018

Paolo Ciuccarelli partners with The Visual Agency
Number of employees: 14

2019

The Visual Agency open their brand new offices
Number of employees: 20

MISS

HOUDINI





“With the right tools to hand, developers can start creating innovative layout choices and themes without the heavy performance hit. Houdini will soon be helping spark fresh, creative approaches within web design.”

Matt Crouch

www.mattcrouch.net

Browsers are bringing down the barriers between CSS and JavaScript. See what that means for both designers and developers and how to get started using it today



DOWNLOAD FEATURE CODE
www.filesilo.co.uk/webdesigner

	Google Chrome	Mozilla Firefox	Opera	Microsoft Edge	Samsung Internet	Apple Safari	Spec
Layout API (Explainer Demos)	Partial support (Canary) Details	Intent to implement Details	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Paint API (Explainer Demos Article)	Shipped (Chrome 65) Details	Intent to implement (Servo) Details	Shipped (Opera 52) Details	No signal	Shipped (Internet 9.2) Details	In Development Details	Candidate Recommendation Spec
Parser API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec
Properties & Values API (Demos)	Partial support (Canary) Details	In Development Details	No signal	No signal	No signal	Partial support (Safari TP 67) Details	Working Draft Spec
AnimationWorklet (Explainer Demos Article)	Partial support (Canary) Details	No signal	No signal	No signal	No signal	No signal	First Public Working Draft Spec
Typed OM (Explainer Article)	Shipped (Chrome 66) Details	Intent to implement Details	Shipped (Opera 53) Details	No signal	Shipped (Internet 9.2) Details	In Development Details	Working Draft Spec
Font Metrics API (Explainer)	No signal	No signal	No signal	No signal	No signal	No signal	No spec

Maintained by Surma, source code & license in the [repository](#) on GitHub, last updated on Mon Jun 03 2019 22:47:49 GMT+0100 (British Summer Time)

The world-famous Harry Houdini was known for his daring acts of escape. It seems fitting to use his name to describe a collection of APIs that do the same thing for CSS.

The Houdini task force consists of representatives from organisations such as Apple, Google and Microsoft and other members of the CSS Working Group. They have been working on a set of APIs that give developers more direct access to the way the CSS engine works.

Browsers are very good at abstracting complex styling tasks away from developers. It will split paragraphs of text onto separate lines without needing to be told. Elements can be sized and placed next to each other automatically by using a couple of properties and letting the rendering engine handle the rest. Each time the page updates, the browser will take the HTML, CSS and JavaScript and convert them into pixels on screen in a process known as the “rendering pipeline”.

Firstly, the browser reads through the content and builds a structure known as a “render tree”, which is then used to calculate where things should appear on the page in a “layout” step. From there, it turns those into pixels in a step called “painting”. With all the elements painted, it sticks them together into one page in a process called “compositing”. To improve the performance of a website, we should always focus on optimising the critical render path.

If we want a visual effect on a site that the browser does not support natively we instead need to add JavaScript and HTML with polyfills. This makes repetitive changes near the start of the pipeline, which results in poor performance.

The aim of Houdini is to open up CSS and allow developers to apply these polyfills further along the pipeline and speed things up. They also open up new opportunities to create effects not previously possible. While not every website will use these new APIs directly, they allow frameworks and libraries the opportunity to level out browser inconsistencies.

What can I use now?

The APIs have been worked on for the past few years, with each one being jointly developed by all members of the Houdini task force. All the APIs follow the strict W3C standardisation process.

If enough of a consensus is reached, an initial draft specification known as a “working draft” is created. From there, it gets refined further before reaching “candidate recommendation” level.

A specification marked as a candidate recommendation can start to gather feedback from implementors – in this case browser vendors. This is

where we start to see wider browser support. From here, it goes to “proposed recommendation” and then “W3C recommendation”, where it starts to achieve full browser support.

Currently, the forerunner is the Paint API which is at candidate recommendation level. The Chromium-based browsers Chrome, Opera and Samsung Internet all support it, with Firefox and Safari working on their implementation. Typed OM is closely related and as a result these same browsers support this as well.

Chrome is leading the way with the other APIs. To play around with the Layout API, animation worklets or the Properties and Values API, you need to use Chrome Canary with “Experimental Web Platform features” flag enabled. These are still under active development and can change at any point. Check out <https://ishoudinireadyyet.com> to see the current state of Houdini.

The aim of Houdini is to open up CSS and allow devs to speed things up

Worklets

Small processes with a big impact

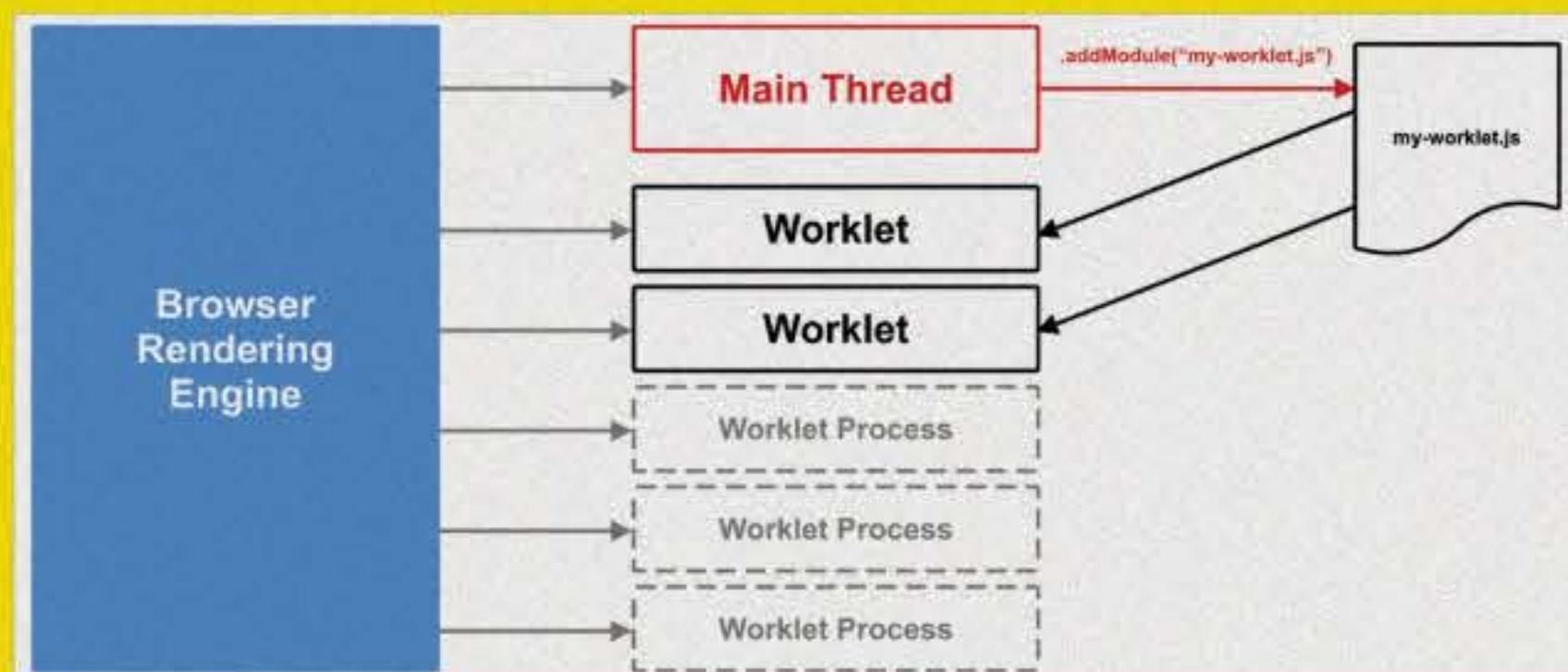
By default, all JavaScript running on a page will operate on the main thread. This means that it shares resources with the rendering of the user interface. Long-running code can end up blocking the main thread and slowing down the interface. Doing this too often makes for a bad user experience.

In order to combat this, the concept of a web worker was introduced. Web workers operate on a separate thread – freeing up the main thread. If we know some JavaScript could take a long time to run, we can add it to a worker and keep the interface responsive.

However, web workers are not cheap to create and are not designed to be used in large numbers. While the idea is a good one they would not be suitable for all jobs – particularly ones that could potentially run multiple times a second.

Worklets operate much like a pared down web worker. They are purely class-based, have no access to the global scope and only work from the data they are passed. They are limited in what they can do, but this means they are cheap to create. Browsers can spin up multiple versions of the same worklet in parallel to have them perform their work as quickly as possible.

Once the browser's rendering engine starts up on first load, it creates multiple worklet processes alongside the main thread. These are placeholders that are ready to be used when the page requires a worklet. When one is requested, it fetches the file and uses a couple of these processes to create new copies of this worklet.



Worklets are the underpinning for all of Houdini's major APIs. Each have their own implementation

When a worklet is called upon, the browser then delegates the work to one of the copies it created. As a result, worklets cannot rely on any external state outside of their class, as there is no guarantee that same data will be used on the next turn.

Worklets are the underpinning for all of Houdini's major APIs. Each have their own implementation that are optimised for the role they fulfil.

The Layout and Paint API have specific methods that are automatically called by the browser as required. Animation worklets have their "animate" method that is called on every frame of the animation. It's important these methods stay as light as possible to avoid any slow renders.

Worklets are crucial for creating highly optimised code for repetitive tasks. It's possible we could see more worklets in the future specialised for either graphics and audio processing.

READ MORE:
WEB WORKERS VS SERVICE WORKERS VS WORKLETS
<https://bit.ly/2mkE9IA>
OFFICIAL W3C SPEC
<https://drafts.css-houdini.org/css-animationworklet/>

Properties and Values API

Give CSS variables the types they need

CSS custom properties – also known as "CSS Variables" – allow us to define a value in one place and use it elsewhere. For example, we could create custom properties to hold theme colours that elements further down the page could make use of. The problem with custom properties becomes apparent when we try to animate between them. As the property could refer to anything, the browser falls back to treating it as a string. As there is no way it knows how to animate between strings, it ends up jumping from one to the other.

The Properties and Values API helps by providing types to custom properties. We can use JavaScript to register a property with the browser, which then takes care of the rest.

```
CSS.registerProperty({  
  name: "--main-color",  
  initialValue: "#ecf0f1",  
  inherits: true,  
  syntax: "<color>"  
});
```

The Properties and Values API helps by providing types to custom properties

```
initialValue: "#ecf0f1",  
inherits: true,  
syntax: "<color>"  
};  
});
```

The "registerProperty" method is a new property on the CSS global object. Here we define a few characteristics about the property such as its name, an initial value if it is not specified and whether or not it inherits from elements higher up.

The important attribute is "syntax", which describes the type of the property. CSS already knows how to deal with different types of data including numbers, angles and URLs. By specifying the type, we can save the browser work and let it know what our values are.

USE CASE: SMOOTH ANIMATIONS
Transition between two numbers to provide a timeline for the Paint API
<https://drafts.css-houdini.org/css-animationworklet/>

Typed OM

Read and write style values on elements

The CSS Object Model (CSSOM) has been part of JavaScript for a long time. It allows us to extract the computed style of an element reading its "style" property or by using "getComputedStyle". The former returns styles applied directly to that element, while the latter includes all styles including inherited ones.

```
document.body.style.fontSize // "2rem"
window.getComputedStyle(document.body).fontSize
// "32px"
```

The value they return is always a string regardless of what was supplied originally. This makes adjusting

READ MORE:

WORKING WITH THE NEW CSS

TYPED OBJECT MODEL

developers.google.com/web/updates/2018/03/cssom

OFFICIAL W3C SPEC

<https://drafts.css-houdini.org/css-typed-om/>

these values buggy and inefficient for us as well as the browser as it needs to constantly convert back and forth between a number and a string.

```
document.body.fontSize += "1rem"; // "2rem1rem", not "3rem"
```

The Typed OM works similar to the CSSOM but also provides types for the values. There are two new ways of accessing them – the "attributeStyleMap" works like the "style" property, with "computedStyleMap" like "getComputedStyle".

```
document.body.attributeStyleMap.get("font-size")
document.body.computedStyleMap().get("font-size")
```

These both return a CSSUnitValue object, which contains the value and the unit as separate properties ready to be read and updated.

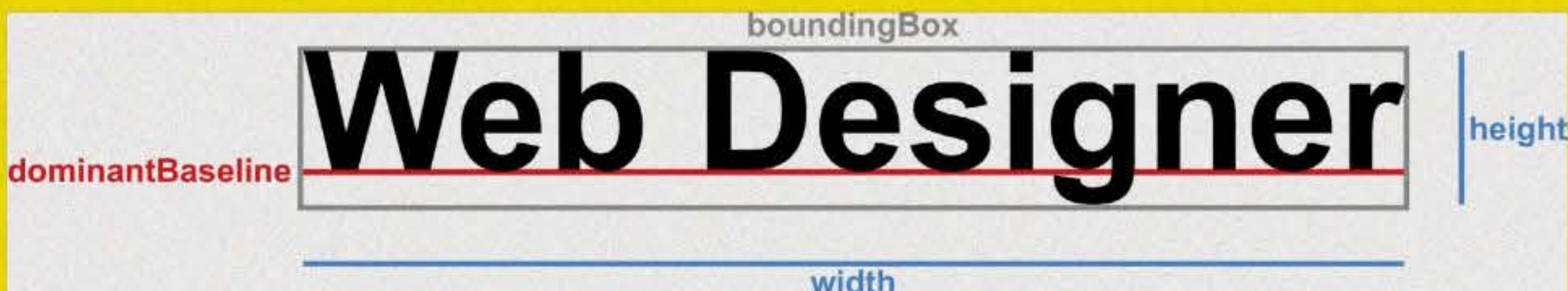
These objects can also be created and used directly to calculate values, much like the CSS "calc()" function does.

The Typed OM is capable of plenty more, such as type conversion and value clamping. Check out the links in the READ MORE info box opposite.

```
document.body.computedStyleMap()
▼ StylePropertyMapReadOnly {size: 291}
  size: 291
  ▶ __proto__: StylePropertyMapReadOnly
document.body.computedStyleMap().get('op
▼ CSSUnitValue {value: 1, unit: "number"}
  unit: "number"
  value: 1
  ▶ __proto__: CSSUnitValue
document.body.attributeStyleMap.set('opa
undefined
document.body.computedStyleMap().get('op
▼ CSSUnitValue {value: 0.5, unit: "numbe
  unit: "number"
  value: 0.5
  ▶ __proto__: CSSUnitValue
```

Font Metrics API

Understand how text is positioned on screen



It can be important to know exactly how much space some text will take up on screen. Perhaps it needs to fit inside a container or, for complex text like mathematical notation, where other elements need to go.

This can, in part, be achieved through a <canvas> element. The "measureText" method gets the text width and not much else. It has inconsistent support, requires a <canvas> tag and ultimately may not accurately reflect the end result on screen.

```
const div = document.querySelector("div");
const element = document.measureElement(div);
const text = document.measureText("Web
Designer", div.computedStyleMap);
```

The proposed solution is document based and introduces two new methods for measuring text.

"measureElement" measures the content of an element on the page. The "measureText" measures the text using the styles defined without an element. It won't be affected by other styles cascading into it.

The returned FontMetrics object contains lots of information about the contents. This includes basic

The "measureText" method gets the text width

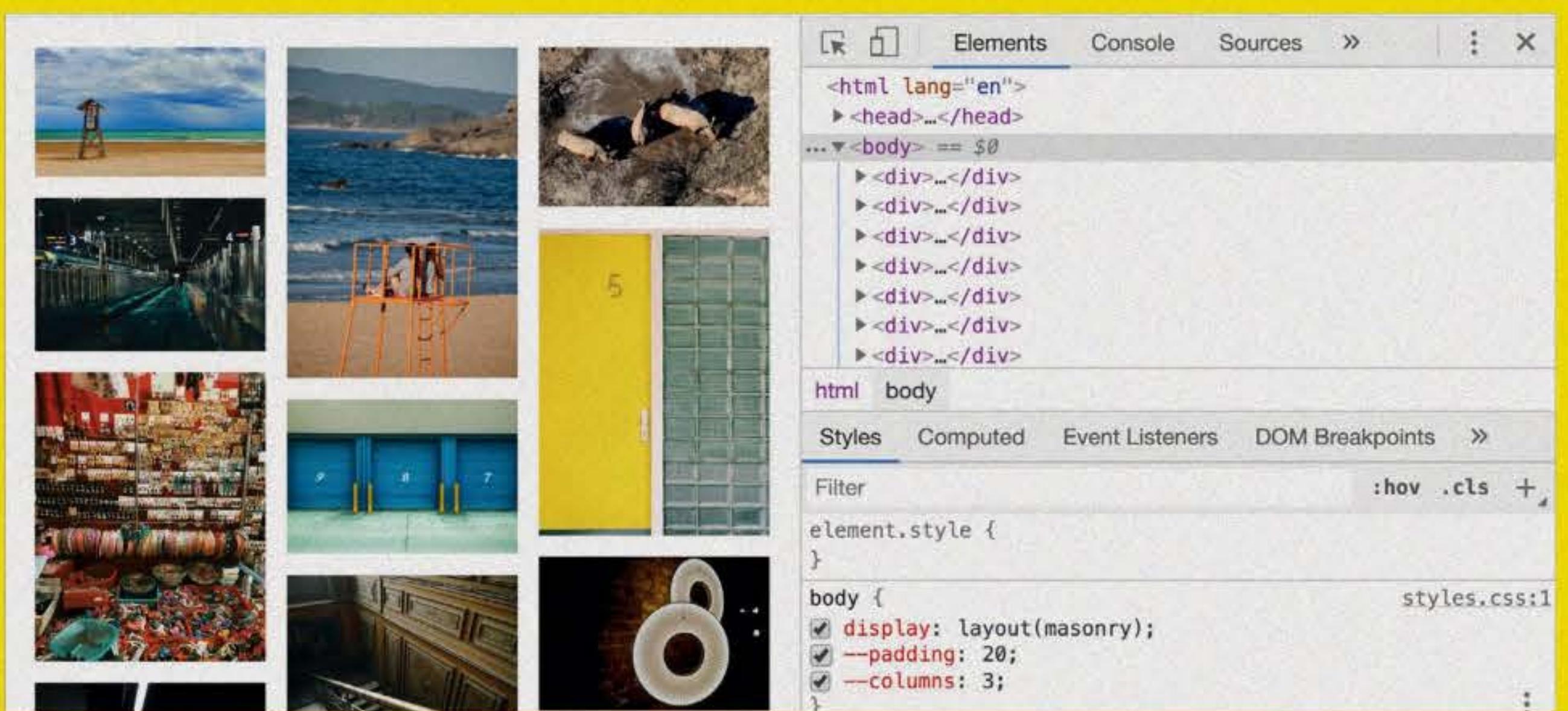
information such as the width and height as well as more in-depth values such as the baseline and spacing. It is still early days for this API.

It has some edge-cases to work out before a full set of methods can be created and a specification built. There are also security concerns around tracking users based on how certain fonts render.

It may be a few years, however, until we see this in any browsers.

USECASE: CALCULATE LINE BREAKS

Know exactly where a paragraph will break to avoid orphans.



Elements Console Sources > X

```
<html lang="en">
  > <head>...</head>
  ... > <body> == $0
    > <div>...</div>
    > <div>...</div>
    > <div>...</div>
    > <div>...</div>
    > <div>...</div>
    > <div>...</div>
    > <div>...</div>
```

html body

Styles Computed Event Listeners DOM Breakpoints >

Filter :hov .cls +

```
element.style { }
body {
  display: layout(masonry);
  --padding: 20;
  --columns: 3;
}
```

styles.css:1

Layout API

Create unique layout algorithms

All browsers have layout algorithms in place to help position content. By default, all elements will be in flow layout. In Latin scripts such as English, any elements set as "inline" will flow left to right in the inline direction and any elements set as "block" will flow top to bottom in the block direction.

While they work well, they make most websites look the same. To make a site stand out, we would need to use properties such as "position: absolute" and calculate offsets manually. A site such as Pinterest uses this approach to make its masonry-style layout, but it can cause performance issues on larger pages.

The Layout API aims to avoid this problem by handing this logic to a worklet. It knows the desired dimensions of the parent and its children and can instruct the renderer exactly where it wants them.

```
class MyLayout {
  static get inputProperties() { return [] }
  async layout(children, edges, constraints,
  styleMap) {}
  registerLayout("my-layout", MyLayout);
}
```

Each worklet requires a "layout" method, which runs each time it needs to recalculate the layout. It also needs to be asynchronous as laying out the content inside can be paused or moved to a separate thread at any time. The first argument is an array of children with the styles that are applied to them. The second contains border, padding and scrollbar size known as the element's "edges". The third defines the rest of the usable space called the "constraints". The

final argument details the properties being requested from "inputProperties" similar to the Paint API.

```
child.layoutNextFragment({ fixedInlineSize:
  200 })
```

All this information is used to generate positioning instructions called "fragments". The "layoutNextFragment" method takes information about the child, such as its desired inline and block size and the renderer takes care of the rest. The result is a set of fragments ready for the browser to paint.

```
body { display: layout(my-layout); }
```

On the CSS side, the layout is defined like any other. The layout function takes the name supplied when registering the worklet. While the option is open to everyone, it's unlikely most would need to create layout worklets for every site. Worklets can be shared and included on a site without knowing about the underlying algorithm. This API is more likely to be used as a way to polyfill future layout systems.

READ MORE:

HOUDINI LAYOUT API

houdini.alitch.me/layout

OFFICIAL W3C SPEC

<https://drafts.css-houdini.org/css-layout-api/>

5 to follow

UNA KRAVETS

twitter.com/Una

Una is the creator of extra.css – a library of small flourishes powered by the Paint API.

SAM RICHARD

twitter.com/snuqua

Sam regularly gives in-depth talks about Houdini and created the demo site houdini.glitch.me

IAN KILPATRICK

twitter.com/bfgeek

Ian is an engineer working on Houdini in Blink. He welcomes feedback on the Chrome implementation.

SURMA

twitter.com/dassurma

Surma is a developer advocate at Google and can be found discussing the performance benefits of Houdini.

ROSSEN ATANASSOV

twitter.com/cssrossen

Rossen is an Edge developer and co-chairman of the CSS Working Group building Houdini.

In the future, the Paint API will be very useful for polyfilling CSS functions that are not yet in all browsers, such as "conic-gradient()".



```
.el {
  background: conic-gradient(red, magenta, blue, cyan, lime, yellow, red);
  width: 75%;
}

@supports (background: paint(id, foo)) {
  .el.is-loaded {
    background: paint(conic-gradient, red magenta blue cyan lime yellow red);
  }
}
```

LIVE EDIT

Paint API

Create dynamic images for CSS on the fly

The penultimate step in the rendering pipeline is the painting phase. At this point, the browser knows exactly the content to show but not how it appears. The rendering engine will look at the styles applied to each element and adjust its instructions accordingly.

While some styles are fairly straightforward, others allow functions to run that decide their appearance. For example "background" can take many functions, which include "url()" for images, "rgb()" for colours and "linear-gradient()" for a gradient effect.

```
#target {background: paint(my-effect);}
```

The Paint API allows us to define our own painting function, which works in much the same way. All functions create an image that the engine can make use of depending on the property it's used against.

```
class MyWorklet {
  paint(ctx, size, style) {} }
```

The only requirement inside a paint worklet is a single "paint" method. Here we provide a set of instructions that a browser can follow whenever it needs to repaint the element. It is called with a few arguments that give the method some useful info.

The first argument is a context, which provides a space we can draw upon. It works similar to the

context used when drawing onto <canvas> elements, by using instructions such as "moveTo" and "fillRect" to start building up the image that CSS can make use of.

There are some differences between this context and that used for <canvas> elements. For example, there are no image data or text methods available for security and performance reasons, but it is possible they will appear in later revisions to the specification.

The second argument contains the dimensions of the paintable area it needs to create. Typically, this is the width and height of the target element including padding. We can use this information to make sure we are drawing onto the context in the right place.

We can also ask for a set of other style properties. For example, we could want to change the colour of the background to complement the text colour. This is done through a getter within the worklet.

```
static get inputProperties() {
  return ["color", "--custom-property"]; }
```

The "inputProperties" value is an array of all properties the worklet is interested in. This can include custom properties that supply some further customisation. These values are passed in as the third argument to the paint method as style objects from the Properties and Values API.

There is a fourth argument that can be used to

access arguments supplied to the paint function in CSS. This allows for configuration at the point of definition, such as the colours to use in a gradient. This is working its way through the specification process and is not ready for use just yet.

```
registerPaint("my-effect", MyWorklet);
```

The worklet needs to be registered with the browser for it to be picked up in CSS. The "registerPaint" method takes the name we use to reference it and the worklet itself.

```
CSS.paintWorklet.addModule("my-worklet.js");
```

Finally, in order to link the JavaScript to the CSS, the browser needs to download the worklet. By supplying the path, the browser handles the rest of the process for us.

READ MORE:

USING THE CSS PAINTING API

developer.mozilla.org/en-US/docs/Web/API/CSS_Painting_API/Guide

OFFICIAL W3C SPEC

<https://drafts.css-houdini.org/css-paint-api/>

Tutorial: Create a striped background

Combine a selection of Houdini's features to generate a versatile and reusable effect

1. Set up the right environment

The Paint API will not work when viewing files directly on our local machine. Create a directory and use npm's package runner "npx" to generate a local server for the files instead. Once that's up and running, we need to open the provided address in a browser that supports both the Paint API, typed OM and the Properties and Values API, which is currently only Chrome Canary with the experimental flag on.

```
> npx http-server
```

2. Create a worklet

All of the work for the Paint API happens inside a worklet. This will live in a separate file and is managed by the browser. We will also need to register the worklet with a name to make sure the CSS parser knows about it. We can do that in the same file.

Create "stripe-worklet.js" in the root of the new directory and add an empty worklet. We will import it shortly.

```
class StripeWorklet {  
  paint(ctx, size, style) {}  
}  
registerPaint("stripe", StripeWorklet);
```

3. Serve some HTML

Before we go any further, we need to create some HTML to serve to the browser. This will include the element we will style as well as some imported CSS and JavaScript to supply the effect.

The styled element has the ID of "target". In this case it's a `<div>` but any element will work.

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <link rel="stylesheet" href="style.css" />  
    <script src="script.js"></script>  
  </head>  
  <body>  
    <div id="target"></div>  
  </body>  
</html>
```

4. Apply worklet to target

Our page is not styled yet. By making the target element as big as the screen, we can see our effect a bit easier. Our worklet will work independent of the size of the target.

Create "style.css" with some base styles. Apply the newly created paint worklet to the target element and supply the colours of the two stripes, which we will make use of later.

```
body {  
  margin: 0;  
}  
#target {  
  height: 100vh;  
  background: paint(stripe);  
  --stripe-color-1: #e22f1c;  
  --stripe-color-2: #ee8276;}
```

5. Add module to page

With everything else set up we now need to add the worklet. This will download the file from step 2, which registers itself and makes the worklet ready to use.

The "paintWorklet" property is new for the Paint API. By checking for it, we can make sure browsers without support don't throw an error when trying to run it. Create a "script.js" file and add the module within it.

```
if ("paintWorklet" in CSS) {  
  CSS.paintWorklet.addModule("stripe-worklet.  
js");  
}
```

6. Calculate stripe values

The "paint" method is called each time the element needs to repaint. The second and third arguments supply information about the element that will receive this background. Create some constants from this information inside the empty "paint" method created earlier. We need enough stripes to cover the element once it's been rotated diagonally.

```
paint(ctx, size, style) {  
  const thickness = 100;  
  const color1 = "black";  
  const color2 = "white";  
  const longestSide = Math.max(size.width,  
size.height);  
  const diagonalLength = Math.sqrt(  
    Math.pow(longestSide, 2) + Math.  
pow(longestSide, 2) );  
  const noOfStripes = diagonalLength /  
thickness;  
}
```

7. Paint to the context

The first argument "ctx" is the context we will paint our background to, and works very similar to `<canvas>` contexts.

Add in a loop directly after the constants from the previous step. This will alternate the stripe colour as they are drawn. To make the maths easier, we draw the lines straight down and rotate the context as a whole.

```
ctx.translate(size.width / 2, size.height / 2);  
ctx.rotate(-Math.PI / 4);  
ctx.translate(-diagonalLength / 2,  
-diagonalLength / 2);  
for (let i = 0; i < noOfStripes; i++) {  
  ctx.beginPath();  
  ctx.strokeStyle = i % 2 ? color1 : color2;  
  ctx.lineWidth = thickness;  
  ctx.moveTo(i * thickness + thickness / 2,  
0);  
  ctx.lineTo(i * thickness + thickness / 2,  
diagonalLength);  
  ctx.stroke();
```

8. Define colour properties

Refreshing the page will show some black and white stripes filling the screen. This is the Paint API in action! To get the colours we supplied, however, we need some extra set up using the Properties and Values API.

Register some new properties with the browser inside script.js. The important part is the "syntax" property, which tells the browser to treat these as colours.

```
if ("registerProperty" in CSS) {  
  CSS.registerProperty({  
    inherits: false,  
    initialValue: "black",  
    name: "--stripe-color-1",  
    syntax: "<color>" });  
  CSS.registerProperty({  
    inherits: false,  
    initialValue: "white",  
    name: "--stripe-color-2",  
    syntax: "<color>" });  
}
```

9. Apply custom colours

At the top of the worklet, we need to provide an array of custom properties the paint method should know about. If any of these values change the method will run again. With the "`<color>`" syntax, this now allows us to use CSS to animate these values like any others.

Finally, update the "color1" and "color2" constants to use these new values instead.

```
static get inputProperties() {  
  return ["--stripe-color-1", "--stripe-  
color-2"];  
}  
[...]  
const color1 = style.get("--stripe-color-1");  
const color2 = style.get("--stripe-color-2");
```

Animation Worklet

A new way to create smooth animations

There are already plenty of ways to create animations. CSS transitions and keyframe animations can move between two values and the Web Animations API in JavaScript that can help sync animations together using a single timeline.

While these all have their own use-cases, by using an animation worklet we're able to perform some extra tricks.

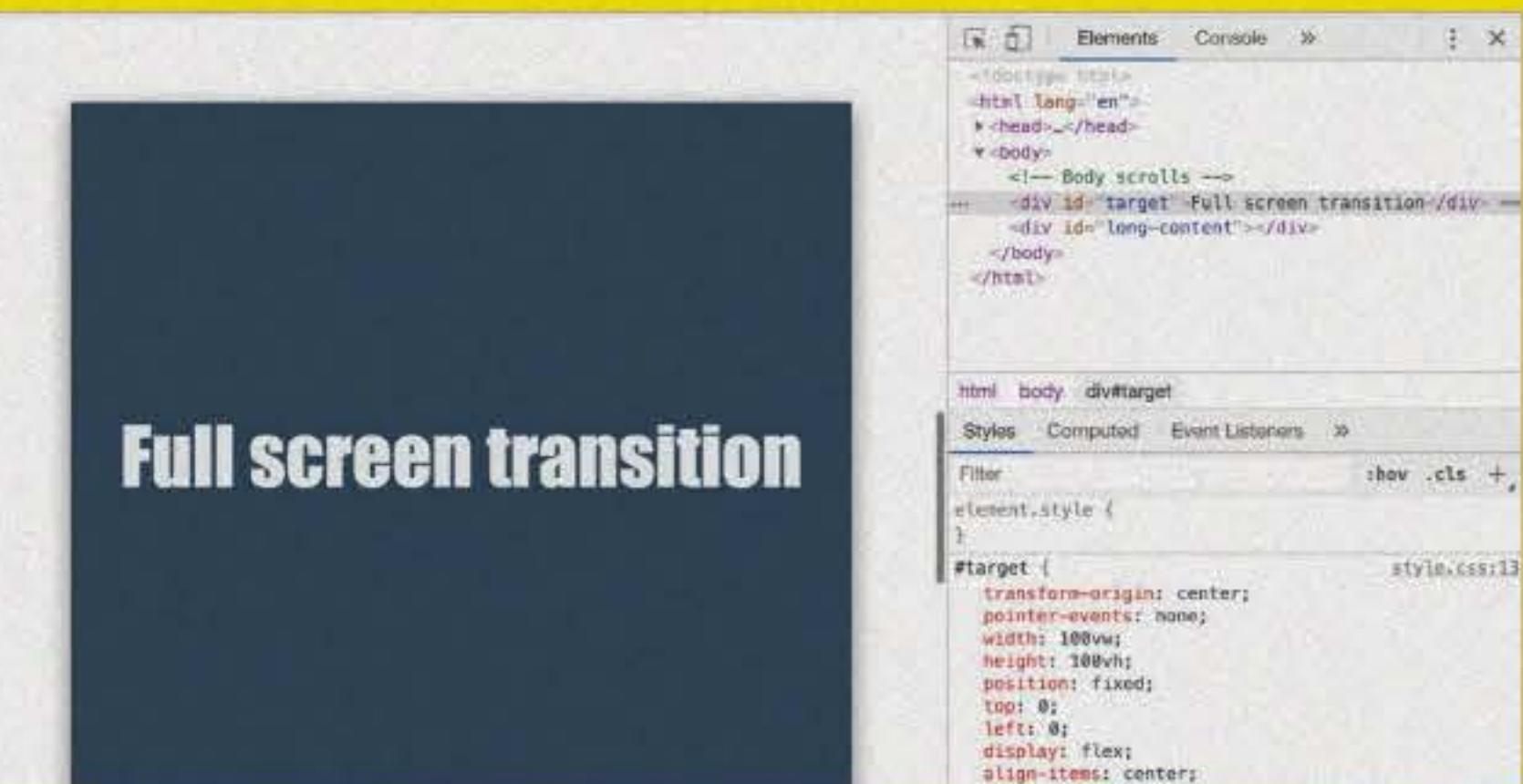
Unlike the Paint and Layout APIs, the animation worklet is not applied through CSS, but instead through JavaScript.

```
await CSS.animationWorklet;
addModule("animation-worklet.js");
const target = document;
getElementById("target");
const effect = new KeyframeEffect(
  target,
  [
    { transform: ["scale(0)", "scale(1)"] },
    { duration: 1000, fill: "forwards" }
  ],
  const animation = new WorkletAnimation(
    "zoom-to-fill",
    effect,
    document.timeline
  );
  animation.play();
```

The WorkletAnimation interface works similar to the Web Animations API. Both take a target to animate and a set of keyframes to apply. In this case, we're scaling an element to full size over 1 second.

The first argument is the name of the animation worklet loaded in on the first line. The final argument "document.timeline" means this animation will track the main timeline of the page and gets passed to the worklet.

```
class ZoomToFillWorklet {
  animate(currentTime, effect) {
    effect.localTime = currentTime / 2;
  }
}
```



The WorkletAnimation interface works similar to the Web Animations API

```
registerAnimator("zoom-to-fill",
  ZoomToFillWorklet);
```

The job of an animation worklet is to control the timeline of the animation. It has a required "animate" method, which is called on every frame. This can contain any logic necessary to control that timeline. In this example, the local time is half of the current time, which causes the animation to run at half speed. It could also contain an easing function to make transitions ease in or out.

Timelines do not have to be a measure of time either. By using a WorkletAnimation, we can define other sources of progression that can be mapped to a timeline.

```
new ScrollTimeline({
  scrollSource: document,
  querySelector("body"),
  orientation: "vertical",
  timeRange: 1000
});
```

A ScrollTimeline creates a timeline based on the scroll position of an element. We can use it to have animations play based on a scrollable element, which is useful for parallax effects.

All of this logic happens during the composition phase of the rendering pipeline. This means that the animation is less likely to be affected when the page is busy with other calculations.

The animation worklet looks to be an interesting addition to the platform. Adjusting the local time of animations opens up the door to many more creative, reusable effects for interfaces.

READ MORE

HOUDINI'S ANIMATION WORKLET

developers.google.com/web/updates/2018/10/animation-worklet

4 ESSENTIAL RESOURCES TO CHECK OUT TODAY

IS HOUDINI READY YET?

ishoudinireadyyet.com

Keep up with support of all APIs across different browsers with this handy grid.

CSS HOUDINI EXPERIMENTS

css-houdini.rocks

A collection of examples with interactive demos showing what Houdini is capable of.

HOUDINI RELOADED

spec.fm/podcasts/toolsday/215273

The team behind the Toolsday podcast discuss their experiences working with Houdini.

CSS HOUDINI & THE FUTURE OF STYLING

youtu.be/GhRE3rML9t4

Una shows how the APIs operate at a browser level in this talk from JSConf EU.

GET NET FROM £2.10 PER ISSUE!

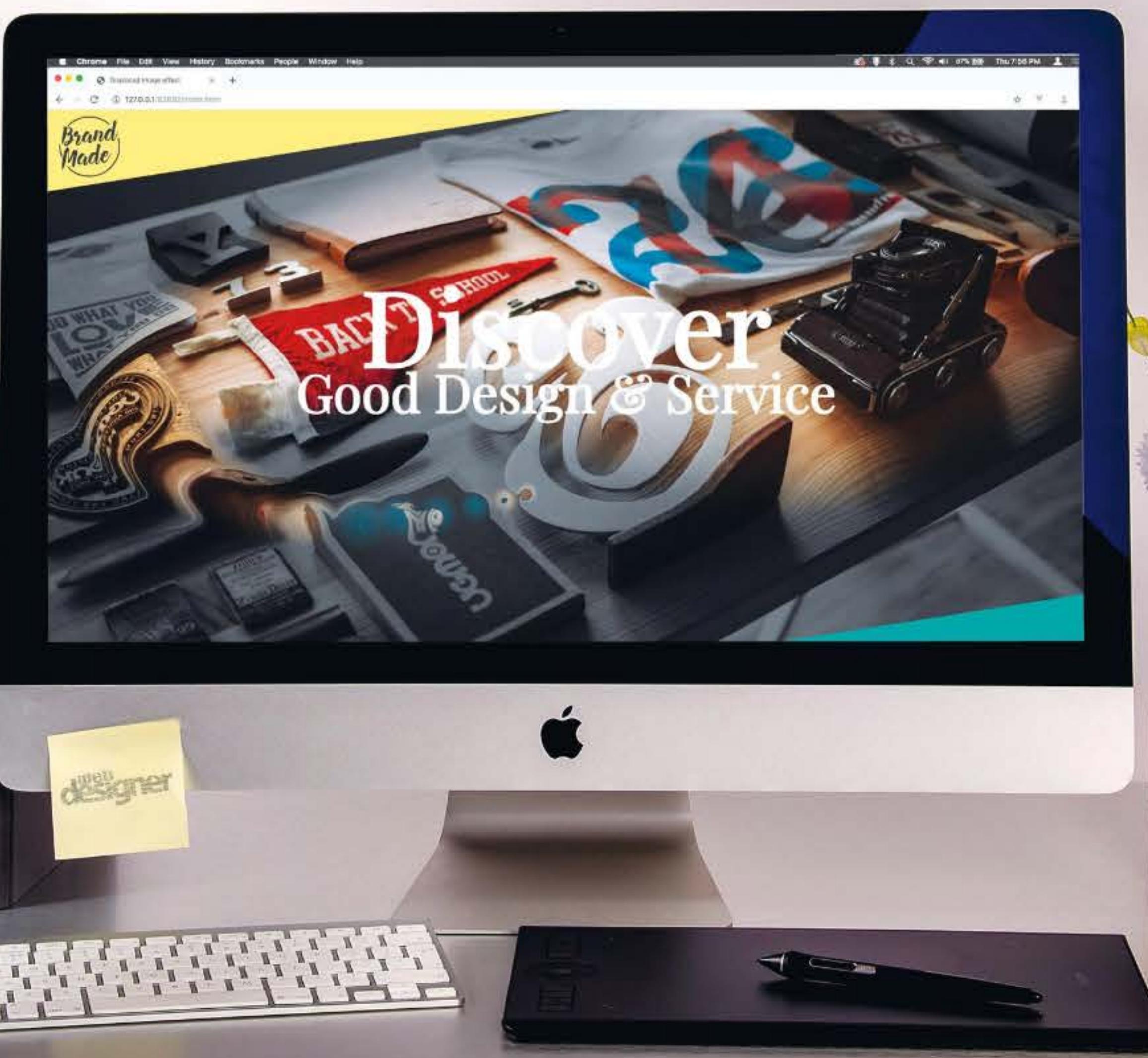


This offer is ONLY available via our online shop
www.myfavouritemagazines.co.uk/digital

Available on
iOS or Android

Create an interactive WebGL header

Use curtains.js and WebGL filters to create an interactive displacement image that transitions from black and white to colour as the user moves over it.



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



sing WebGL's filters has allowed many effects found in the special effects industry and games to come to the web.

The power of these effects is that the filters themselves are small programs known as shaders using GLSL, the OpenGL Shader Language. These programs are based on the C programming language and are incredibly fast because it runs very low level on the graphics card. This tutorial will not focus on the famously difficult GLSL language, but rather on using Curtains.js to take a pre-built shader and interactive code, then apply it to a page design so that you can get all the benefit without spending months learning GLSL!

This tutorial will focus on a web designer's primary job of designing the page, creating a great user experience by laying out the content and fitting the interactions into that so that it works seamlessly together. This is all thanks to curtains.js, a prebuilt library specifically designed for placing your WebGL textured planes exactly where you as the designer lay out the image that you want to apply the effect to. So you position these using your existing skillset of CSS, then the library converts that image into the dynamic WebGL content that the shader runs on. This means that post-processing effects can be added to your pages by simply designing where your images are to go!

1. Follow the script

To get started with the project, open the folder 'start' from your project files folder in your code editor. Open up the 'index.html' page for editing and add the CSS and script links as shown in the head. You will notice that there is shader code on the page, it's a good idea to turn off any 'auto formatting' options in your editor, as this code will not work if it's modified onto different lines.

```
<link rel="stylesheet" href="css/design.css">
<script src="js/curtains.js"></script>

<link href="https://fonts.googleapis.com/css?family=Playfair+Display&display=swap" rel="stylesheet">
```

2. Build up the page

The next code is for the body layout. The canvas div will hold the WebGL context while the image placed in 'fullwidth-image' is grabbed and used for the effect. Following this is a tag line that will be placed over the image in the centre of the screen.

```
<div id="page-wrap">
  <div id="canvas"></div>
  <div id="fullwidth-image">
    
  </div>
  <div id="tag"><h1>Discover</h1>
    <h2>Good Design & Service
```



```
</h2>
</div>
```

3. On brand

The logo is going to be placed in the top left corner so the code for that is added here along with some navigation. This will be placed in the bottom right hand corner of the image effect on the screen and will use angled shapes to enhance the design.

```
<div id="logo"></div>
<nav>
  <ul>
    <li><a href="#about.html">About</a></li>
    <li><a href="#portfolio.html">Portfolio</a></li>
    <li><a href="#contact.html">Contact</a></li>
  </ul>
</nav>
```

4. Make introductions

The next section of code defines the main text and content on the page and this will take place below the image effect. The content wrapper for this is the article, with a CSS grid defined inside here to place the content, which starts with an introduction.

```
<article class="main">
  <div class="grid">
    <div class="col1 panel">
      <h2>Elevating digital experiences<br>for your customer</h2>
      <p class="intro">Intro text here</p>
    </div>
    <div class="col3">
      
      <h3>Our Clients</h3>
      <p>Text here</p>
    </div>
```

5. Define columns

The next content continues the columns that have been started in the previous step. A three column grid will allow content to sit neatly inside of this below the introduction. There are paragraphs of placeholder text in the final project files, you can add your own paragraphs here.

```
<div class="col3">
  
  <h3>Unique Approach</h3>
  <p>Text here</p>
</div>

<div class="col3">
  
  <h3>Unrivaled Service</h3>
  <p>Text here</p>
</div>
```

6. Finish the page off

The article is closed up and the footer allows the page to be finished off. At this stage there is no design in place so we will create that shortly. Below this content should be your scripts for the shaders, leave those as they are but just scroll down below that.

```
</div>
</article>
<footer>
  <p>2019 Web Designer Magazine</p>
</footer>
```

Placeholder elements

Any image that you place in your design that will be used by curtains.js will just be a placeholder, and this will be hidden before being replaced by a WebGL mesh.

Tutorials

Create an interactive WebGL header



7. The main effect

After the shaders, and before the closing body tag, add the script tag here that points out to the finished effect code. This will take the user interaction and the image, then create the colour displaced effect on top of the image. Save the page and switch over to the 'design.css' file from the CSS folder

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

8. Design the page

Add the following CSS to the desing.css file, which should be an empty file. This set the body to be at least the full height of the page. All elements will include the padding and border in their sizes. The body typography is set up with a sans serif typeface, the size is set and the line height increased to aid legibility.

```
min-height: 100%;  
}  
*{  
    box-sizing: border-box;  
}  
body {  
    margin: 0;  
    font-size: 1.1em;  
    font-family: sans-serif;  
    background: #1f4e5a;  
    line-height: 1.6em;  
}
```

9. Positioning elements

The wrapper of the page for the image is given a full width size and 95% of the viewport height, just to allow the page content below to be shown slightly. The canvas is positioned inside of this, so that it fills the page-wrap.

```
#page-wrap {  
    width: 100%;  
    min-height: 95vh;  
    position: relative;  
    overflow: hidden;  
}  
  
#logo img{  
    padding: 0 0 0 20px;  
    width: 140px;  
    height: 140px;  
}
```

10. Add the image

The main image is set to fill the space left inside the page-wrap. The image's visibility is turned off, however if there is no WebGL supported by the browser, this will dynamically be made visible again so that something will appear in the screen.

```
#fullwidth-image {  
    position: absolute;  
    top: 0;  
    left: 0;  
    height: 95vh;  
    width: 100vw;  
}  
  
#fullwidth-image img, #fullwidth-image  
canvas {  
    display: none;  
}
```

11. Something for everyone

Here the class of 'no-webgl' is going to be added if the browser doesn't support WebGL and therefore the effect won't work. The image content is set to fill that space that's left behind completing the content when the image effect can't be supported.

```
.no-webgl #fullwidth-image {  
    display: grid;  
    justify-content: center;  
    align-items: center;  
}  
  
.no-webgl #fullwidth-image img {  
    display: block;  
    object-fit: cover;  
    min-width: 100%;  
    min-height: 95%;  
}
```

12. Top corner

Over the top of the existing page content in the top left hand corner the logo will be placed. A slightly yellow background image will be placed behind this to give the design an angled effect. The background size property is used to make this fill the appropriate size for this to work.

```
#logo{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 150px;  
    background: url(..../img/left-y.png) top  
    left no-repeat;  
    background-size: 50% 150px;  
}
```

13. Navigation placement

The navigation is going to be placed in the bottom right of the main image effect. Again an angled background image will be placed over the top of the effect. The navigation is placed inside of this and aligned to the right edge of the screen.

```
nav{  
    position: absolute;  
    bottom: 0;  
    right: 0;  
    width: 100%;  
    height: 150px;  
    background: url(..../img/right-c.png)  
    bottom right no-repeat;  
    background-size: 50% 150px;  
    text-align: right;  
}  
  
nav li {  
    text-align: center;  
    display: inline;  
}
```

14. The menu elements

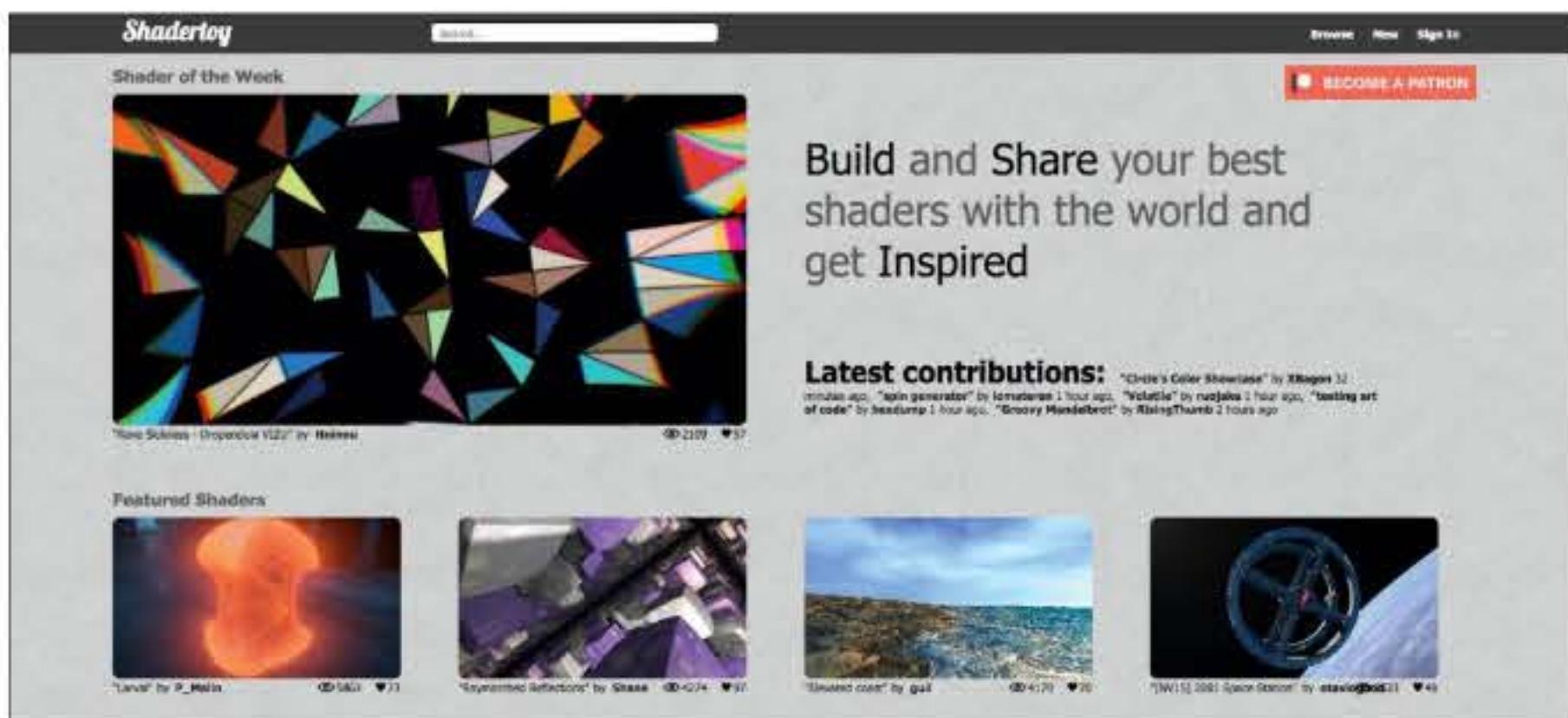
Each menu element in the navigation is styled to remove the usual unordered list bullet and the padding that goes with this. The link itself is then styled to be inline so that the menu appears along one line with the appropriate colour formatting.

```
nav ul {  
    list-style-type: none;  
    padding: 0;  
    margin: 0 10px 0 0;  
    padding-top: 100px;  
}  
  
a {  
    text-decoration: none;  
    padding: 10px;  
    color: #93eae0;  
    display: inline-block;  
    font-size: 0.8em;  
    transition: 0.5s all;  
    text-transform: uppercase;  
}
```

15. Setting the tag line

Over the centre of the image will be a tag line, this is set up here and all headings are going to be set in an appropriate typeface so that is also added. As a final part of the menu, the hover colour of the rollover is added into the navigation.

```
a:hover {  
    color: #222;  
}
```



Going down the rabbit hole

If for any reason you want to get into shaders, then a good place to start is somewhere like Shader Toy (www.shadertoy.com). The site has contributions of shaders that can be anything from full on 3D scenes to filters that effect images. This at least allows you to see what is possible. If this piques your interest then you might want to learn a little about how to make your own. A great interactive instructional guide is the Book of Shaders (<https://thebookofshaders.com>). This will take you through a self-paced guide, that allows you to write all the code within the site, giving you instant feedback on the results. This allows for experimentation without any set up.



```
}

h1, h2, h3{
  font-family: 'Playfair Display', serif;
}

#tag{
  color: #fff;
  position: absolute;
  text-align: center;
  padding-top: 20%;
  width: 100%;
}
```

16. Finalising the tag

The final part of completing the text tag is to specify the headings size, which is made dynamically from the width of the browser screen. The margin is set smaller on the heading 1 text at the bottom so that the heading 2 tag sits closer to it below.

```
#tag h1{
  font-size: 10vw;
  margin: 0 0 5vw 0;
}

#tag h2{
  font-size: 5vw;
}
```

17. Setting the content

The rest of the page content will reside inside the 'main' article class. Here the colour is set to stand out against the body of the page and an angled background image will be placed in both the top left and bottom right of the section to continue the theme set with the main image above.



Background images

Each background image you have on an element can be resized and positioned to build up complex layouts just from within one div.

Tutorials

Create an interactive WebGL header



Elevating digital experiences
for your customer

Shared value capacity building radical activate youth revolutionary, NGO rubric collaborative cities.

```
.main{  
    color: #cde5eb;  
    background-image: url(..../img/left-dc.  
png), url(..../img/right-c.png);  
    background-position: top left, bottom  
right;  
    background-repeat: no-repeat;  
    background-size: 50% 150px;  
    padding-bottom: 100px;  
}
```

18. Making the grid

The grid of the page is defined with a 12 column grid that allows for a variety of placements. The layout added in this design will simply create a standard three column layout that makes the design of information work well.

```
.grid {  
    width: 80%;  
    margin: 0 auto;  
    display: grid;  
    grid-template-columns: repeat(12, 1fr);  
    grid-gap: 80px;  
    padding: 100px 0;  
}  
  
.col3 {  
    grid-column-end: span 4;  
}
```

19. Other sizes

The introduction text will be given a full width column, so this column spans over 12 of the sections for that.



19

The images in the three columns will be set to fill the width of that column and will be given an 8 pixel border with a 9 pixel corner radius on each of the corners.

```
.col1 {  
    grid-column-end: span 12;  
}  
  
.col3 img{  
    max-width: 100%;  
    border: 8px solid #cde5eb;  
    border-radius: 9px;  
}
```

20. Introduction panel

The main introduction is held inside a class called panel. This simply reduces the width of the content at the edges by 20% on the left and right. It's also pushed down below the angled background image. The heading for this is set in here.

```
.panel{  
    padding: 100px 20% 20px;  
    text-align: center;  
}
```

21

21. Finishing off

The footer is the final element of the page to be created, this again follows the familiar styling with an angled background image to keep consistency within the design. Save the page and view the content in your browser to see the rippling interactive colour effect.

```
footer {  
    width: 100%;  
    height: 250px;  
    background: #c95940 url(..../img/left-r.  
png) top left no-repeat;  
    background-size: 50% 150px;  
    color: #eee;  
    text-align: center;  
    padding-top: 120px;  
}
```

Elevating digital experiences
for your customer

21

Shared value capacity building radical activate youth revolutionary, NGO rubric collaborative cities.



ents

testes; effective when silo, relief agile. Then resources preliminary thinking inclusion do-impact investing, disrupt, and effective social entrepreneurship thought leadership.



Unique Approach

Shared value capacity building radical activate youth revolutionary, NGO rubric collaborative cities. Radical targeted changemaker rubric mobilize. Resilient, inspiring expose the truth natural resources agile.



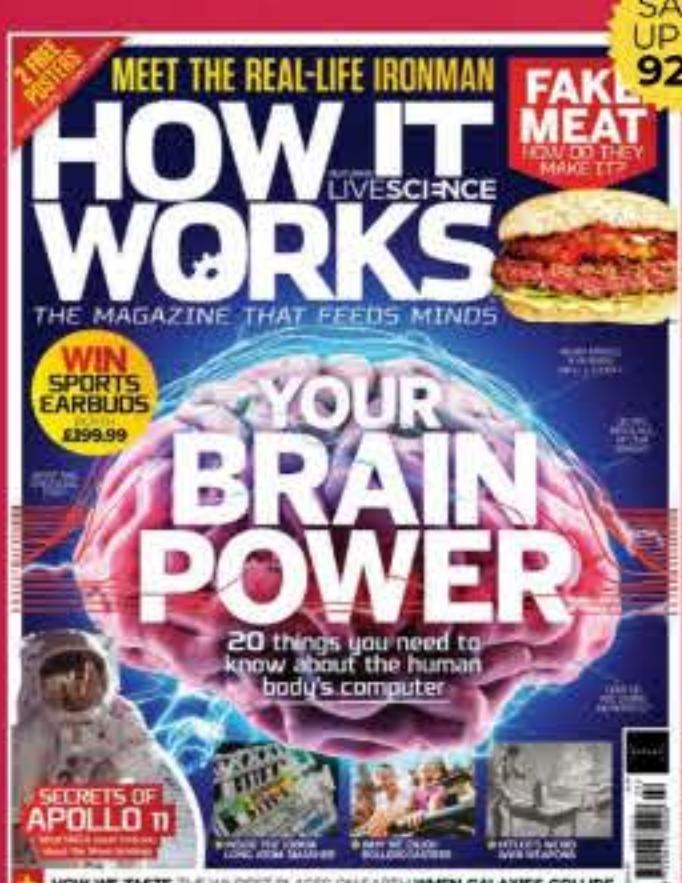
Unrivaled Service

Milestones shared unit of analysis initiative do-gooder commitment energize equal opportunity. B-corp optimism NGO; targeted program area; social return on investment our work do-gooder game-changer.

Autumn Sale

5 ISSUES FOR £5/\$5/€5*

BIG SAVINGS ON OUR BEST-SELLING MAGAZINES



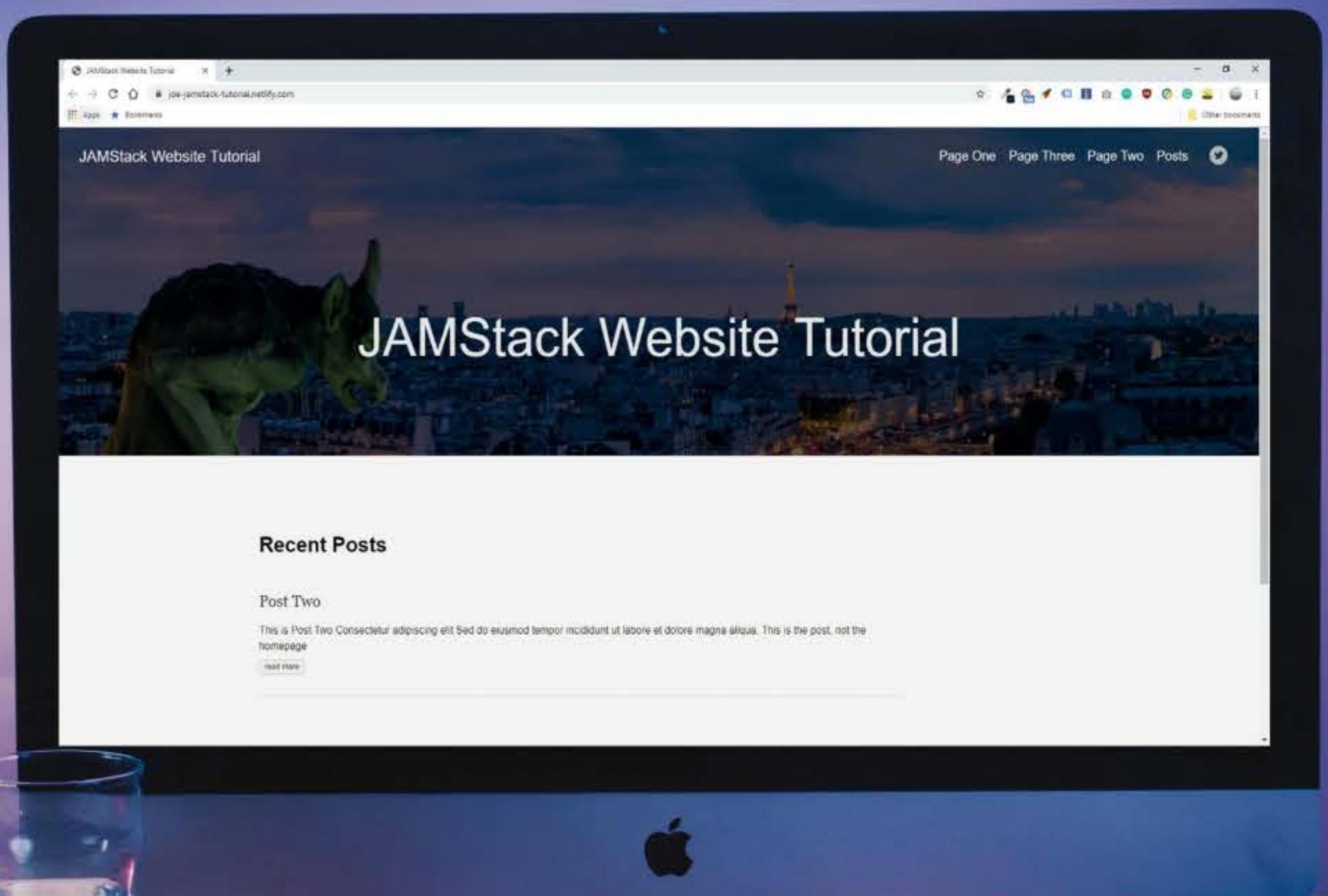
For great savings on all of our magazines, see the entire range online
myfavouritemagazines.co.uk/autumn192

ORDER HOTLINE 0344 848 2852

***TERMS AND CONDITIONS:** This offer entitles new subscribers to receive their first 5 issues for £5 for UK readers, 5 issues for €5 and 5 issues for \$5 for overseas readers. After these issues, standard subscription pricing will apply – please see online for details. The biggest savings are compared with the overseas RRP. For all savings details see online offer page. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14-day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: bit.ly/maqtandc. Offer ends 11 October 2019.

Build a website with JAMstack

The future of the web, JAMstack is now a viable method of building dynamic websites, and here's how



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



Traditionally when a user clicks onto a web page, a series of actions take place. The user's browser sends a request to the website's server. The server then runs through the backend code and generates the correct page for the user with any location based customisation, the latest news posts or any personal login data all ready to be shown (Welcome, you are logged in as). This data is sent back to the user's browser and rendered as a web page.

All this happens in less than a few seconds, assuming the server is configured correctly and not under heavy load, the front end pages have been optimised properly, none of the backend queries to fetch those news posts aren't running slow.

Quite a lot that could go wrong, and even when it goes well, it's slightly slower than a user is comfortable with in the lightning fast digital world we now live in.

Enter the JAMstack, a method of creating and serving websites with minimal load required by the server. JAM stands for Javascript, API and Markup. Pages are built in markup language ahead of time and served as static HTML files as a user requests them. This concept is not new, in fact it is a flashback to the original non-dynamic web where entire websites were a directory of HTML files.

This time there's one giant difference – the sites in the JAMstack aren't static. Using an API any database or server side functions are replaced with APIs.

This tutorial will cover how to set up a blog website in the JAMstack to serve as an introduction to the concepts. The tools used will be Hugo as a static site generator, Netlify as the build tool and GitHub as free hosting for the files.

1. Download the required tools

Download a copy of Victor Hugo – this is one of several starter kits to get things off the ground quickly. Also make sure you have node installed on your development environment.

<https://github.com/netlify-templates/victor-hugo>

```
cmd
cd cmd
D:\laragon\www\laracasts2
cd D:\Users\joefo\Documents\Writing\Tutorials\JAMstack
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack - (victor-hugo@1.0.0)
cd site
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack\site
hugo new page-one.md
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack\site\content\page-one.md created
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack\site
hugo new post/post-one.md
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack\site\content\post\post-one.md created
D:\Users\joefo\Documents\Writing\Tutorials\JAMstack\site
```



2. Create a working directory and install Hugo

Create a directory and name it "JAMstack". This is where we will work on our development machine. Extract the downloaded files for Hugo into this directory and open it in command prompt. To install all dependencies, open command prompt or a terminal in the JAMstack folder and run npm install

npm install

3. Start the server

Once the dependencies have finished, run npm start. The development server is now running a local copy of Victor Hugo, accessible by default at localhost:3000 – opening that link should show the welcome screen if everything was successful. Once tested, press Ctrl + C to stop the server.

4. Add a page and a post

Change directory to the site folder, then, using the "hugo new" command, add a page-one.md and a post-one.md. Developers working in Windows will need to download the hugo.exe and add a path to get this to work but easy to follow documentation exists on the hugo website.

hugo new page-one.md
hugo new post/post-one.md

5. Add content to the page and post

For testing purposes, some content needs to be added to the new post and page. Open the directory for the project in a file browser and navigate to JAMstack/Site/Content. Within this folder the file page-one.md should now exist. There is also a folder named "post" which contains the post-one.md. Open both of these files and add some content in markdown below the --- (or in some cases +++).

```
# Lorem ipsum dolor sit amet
## Consectetur adipiscing *elit*
Sed do eiusmod tempor incididunt ut labore
et dolore magna aliqua.
```



6. Add the theme as a sub module

The text entered in the previous files can't be viewed without a theme. As an example, the Ananke theme will be used. Delete the contents of the current site/layouts folder, leaving it empty. Make a new directory in the site folder called themes, then change to it and

Markdown – A quick “Gotcha”

If you're new to using markdown for content, make sure to include a space between the "#" and the content. # Represents a header (h1), whereas a ## will show as an h2. If the content is touching the symbol it will show as plain text.

Tutorials

Build a website with JAMstack

JAMStack Website Tutorial

JAMStack Website Tutorial

Recent Posts

Post One - The first test post

This is the first test post! Consectetur adipiscing elit Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. This is the post, not the homepage.

Read more

run the following code to import the theme as a git submodule. Normal cloning is not compatible with Netlify.

```
mkdir themes  
cd themes  
git submodule add https://github.com/<THEMECREATOR>/<THEMENAME>  
https://github.com/budparr/gohugo-theme-ananke.git
```

7. Configure the theme and run the server

Copy the contents of site/themes/gohugo-theme-ananke/exampleSite/config.toml to over the one in the site folder. At the top of the page, replace the base url with "/" and delete the line themesDir = "../". Save the config file, open a terminal and run npm start command. You can also change the name of the website (if you want to) by replacing the "title" value.

```
cd site  
npm start
```

8. Test in a browser

Open <http://localhost:3000/page-one/> to see the theme rendering the markdown of the page as a fully styled page. Opening the homepage, the first post will now be visible. This means that the static site generator is now functional.

9. Configure the menu

When viewing the website, it is noticeable that the page made earlier is not showing in the navigation.

Returning to the content, add a line in the front matter configuration in order to tell Hugo which menu to render the page in.

```
TOML  
+++  
menu = "main"  
+++  
YAML  
---  
menu: "main"  
---
```

JAMStack Website Tutorial

Page Two

This is Page Two

Consectetur adipiscing elit

Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. This is the post, not the homepage.

What's the benefit to this extra work?

Relying on checklists can facilitate new learning. The extra effort in setting up a JAMstack website results in a lighting fast load time. Testing a WordPress website against its identical version running on the JAMStack loads in 0.77 seconds vs 6.83s

Docs Tutorials Plugins Features Blog Showcase Contributing

Search

Fast in every way that matters

Gatsby is a free and open source framework based on React that helps developers build blazing fast **websites** and **apps**

Get Started →

How Gatsby works

Pull data from anywhere

DATA SOURCES

What to do when a client prefers editing with WordPress

WordPress is still extremely popular, and many non technical content editors have been trained in the process of creating pages and posts using the WordPress interface. For that reason, there may be instances where a client would still want to use WordPress. Thanks to something called "Headless" WordPress, the ever popular content management system can be transformed into a headless CMS. This means that WordPress is still used as the content management system, in place of Forestry.

To get started converting a WordPress website, the front end needs to be removed by replacing the theme with a blank theme containing only an empty index, and the minimum style.css showing information about the theme. An API is then used to feed a static site generator such as Gatsby.js. Netlify is connected to the project, and whenever a page or post is updated in WordPress it automatically builds the static site with the new data from the API.

10. Push the files to a github

Next, animate the circle to respond to user clicks. Now the static site is up and running, it will need to be accessible on a development server. The first step is to push the code to github. Create a new repository on github, and then either use the command line to push the code from the project folder or the github desktop app.

```
git remote add origin https://github.com/
[githubusername]/
jamstacktutorial.git
git push -u origin master
```

11. Connect to Netlify

Netlify will bring everything together, build the site and serve it on a temporary domain. Start off by creating an account at Netlify ([netlify.com](#)) and link it with a github account. Once everything is set up click "New site from Git".

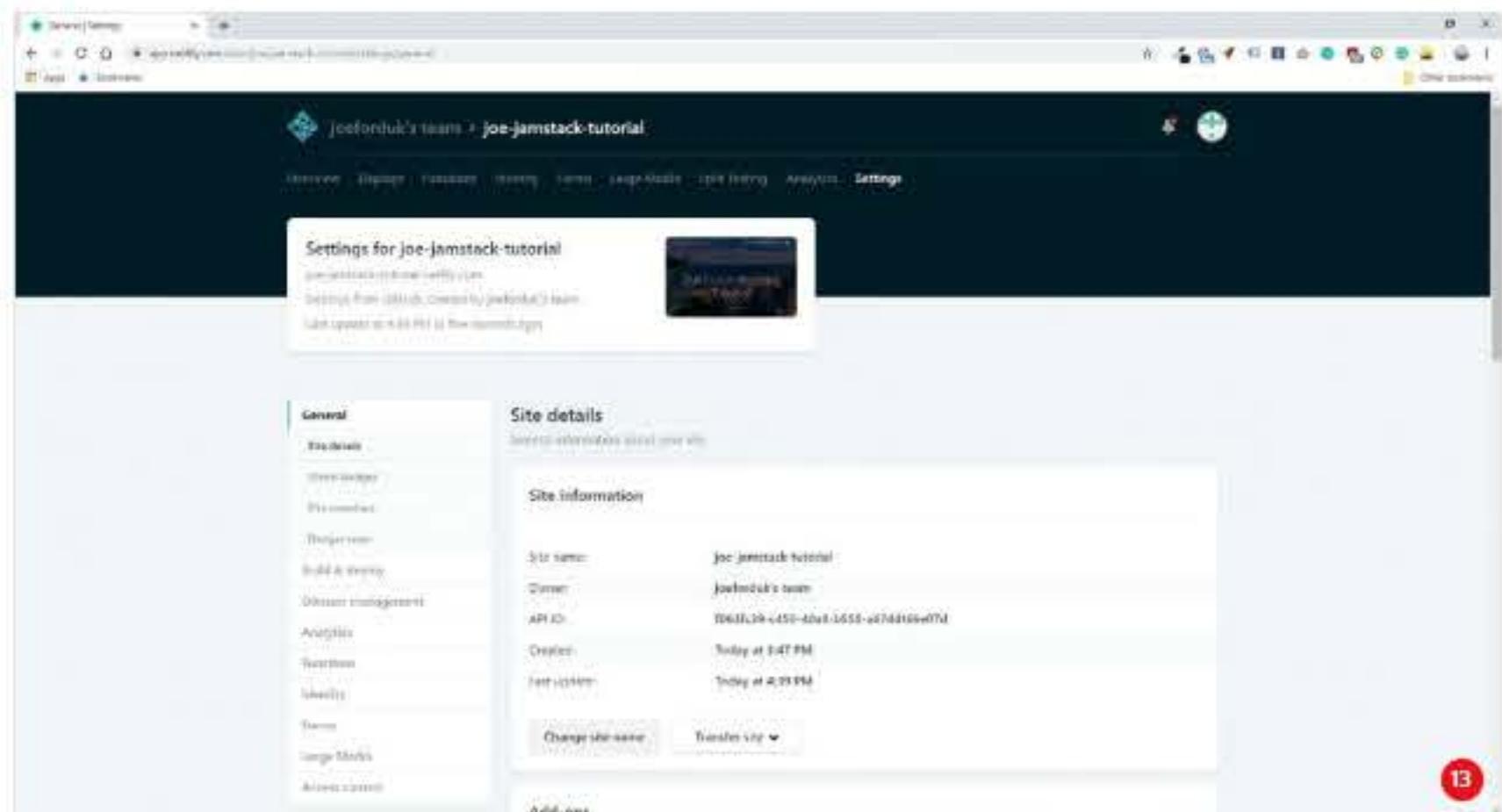
[netlify.com](#)

12. Connect to Github

For continuous deployment, click "Github", then select the repo that was made earlier in the tutorial. Netlify should automatically detect the best build options for the project. It should read "npm run build". If all is well, then click deploy.

13. Build site and view on Netlify link

Once the site has built, a message will appear "deployed". Netlify has given the website a temporary



domain name which may not make sense – this can be changed simply by clicking site settings. Change the name, then click the link to see the site loaded through the JAMstack.

14. Make a change to test deployment

The website now loads through the JAMStack. It's lightning fast, and it automatically updates when a commit is made to github.

To test this, return to the local development environment and run hugo new page-three.md from the site directory. Then open the created file, add some content, click save and commit the file to the repo. Within moments you will notice the update is live on the netlify link.

15. Add a CMS to the static site

Working in plain markup and using the command line to create pages will not sit well with most clients.

To make the website more friendly, install a content management system. [Forestry.io](#) is a perfect fit for the current setup. Go to the website and create an account using github.

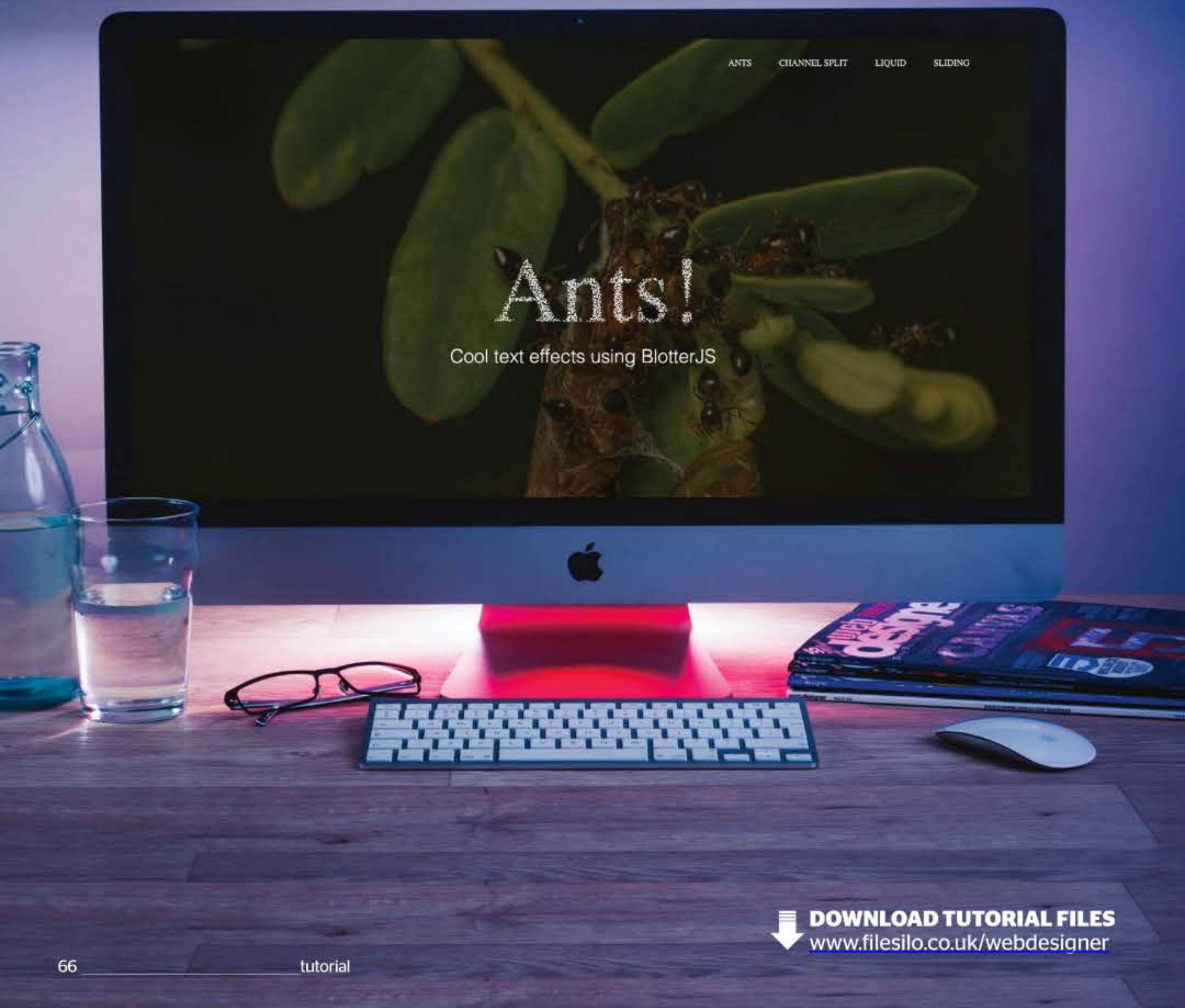
16. Configure Forestry

Click on add new site then select Hugo as the static site generator, git as the provider and fill in the information in the following forms.

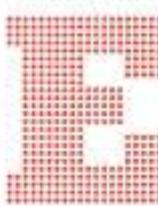
Click submit, and the new CMS will load, ready for content changes. Now, pages can simply be edited from the sidebar, as well as posts and a huge array of other options.

Create cool text effects with Blotter.js

In this tutorial, you will learn how to use Blotter.js to create four interesting text effects, and also how to add mouse-based animations



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

 every element of design - typography, colours, images, shapes and patterns etc, carries a visual weight. Some elements are heavy and draw the eye, while other elements are lighter. Typography is one of the heavier elements that can be manipulated to evoke different moods and feelings in web users, particularly when type is distorted in some way. For instance, users are likely to be drawn towards squeezed, arched or warped text as compared to ordinary text. The implication is that, as a designer, you might be drawn to using cool text effects on some of your site's elements such as headlines in order make your site unique, while in other cases, you might be interested in directing user attention to some parts of the site. The conventional way to achieve cool text effects was using CSS and gifs to combine different images and generate playful text on screen. However, the approach has its constraints because the effects it develops have already become too common. A novel alternative to achieve the same effects is using Blotter.js, a JavaScript API for drawing unconventional text effects on the web. The advantage of using the library is that it enables a designer to build and manipulate text effects that utilise GLSL (OpenGL Shading Language) shaders without requiring that the designer write GLSL. In this tutorial, four text effects are developed using the library. Then animations based on mouse movements are added to one of the effects.

1. Get started

Begin by creating a folder, Blotter.js, on your desktop to store the tutorial files. Since the tutorial files are minimal, store all resources in the root folder. Files required in the tutorial include CSS styling, JS script file, background images and the root html file.

2. Create the page structure

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

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title> Blotter JS </title>
  </head>
  <body> </body>
</html>
```

3. Adding HTML

The tutorial aims to develop four different text effects where each runs on its own separate page. As such, we create four different html files and place a navigation menu to easily browse through the different pages. Second, an extra div is also added to contain the canvas of the JS library. The script is thereafter added in the body section. The basic structure of the page is laid out as follows...

```
<body>
  <div id="container">
    <nav>
      </nav>

      <ul class="main-nav">
        <li><a href="index.html">Ants</a></li>
        <li><a href="index2.html">Channel Split</a></li>
        <li><a href="index3.html">Liquid</a></li>
        <li><a href="index4.html">Sliding</a></li>
      </ul>

      </nav>

      <div id = "Ants" > </div>

      <h2 > Cool text effects using BlotterJS </h2>
  </div>
</body>
```

4. Link the CSS

We create a separate stylesheet to style each of the HTML elements. First, open your code editor and create main.css. Create a link to this file in your html document by adding the following code in the head section. Add a link to the font we will use. Since no styles have been added, the page renders with basic styling.

```
<link rel="stylesheet" href ="/styles.css"
>
<link href='http://fonts.googleapis.com/
css?family=Lato:100,300,400,300italic' >
```

5. Style the background

In order to make each page unique, a separate background image is used. The images used in the tutorial are all downloaded from [Pexels.com](#) and are free to use: Ants by Jimmy Chan - [www.pexels.com/photo/selective-focus-photography-of-ants-on-leaves-2454796/](#); Water by Matthias Cooper - [www.pexels.com/photo/blue-water-1147124/](#); CRT by Huynh Dat [www.pexels.com/photo/black-crt-tv-2251206/](#); and doors by Chelsea Cook [www.pexels.com/photo/photo-of-blue-](#)

[doorway-2929910/](#). Download these and save them in the root folder. To style the background, we use in-style CSS for each of the pages in order to make their backgrounds unique. The style appends a background image to the container div. Render the page to notice the changes to the background. A margin is on all sides.

```
<div id = "container" style ="background-
image:
linear-gradient (rgba(0, 0, 0, 0.6), rgba(0,
0, 0, 0.6)),
url(ants2.jpg);
background-size: cover; background-position:
center;
height: 100vh; background-attachment:
fixed;">
```

6. Style background and text

In order to eliminate the margin observed on all sides of the image, open the main.css file and add the code below. This code here sets the margin and padding to 0 and also ensures that the whole page background is used.

```
body, html {
  padding: 0;
  margin: 0;
  width: 100%;
  height: 100%;
  overflow: hidden;
}
```

Next, we style the text by adding the code below.

```
h2 {
  font-family: 'Lato', sans-serif;
  font-size: 30px;
  font-weight: 100;
  text-decoration: none;
  color: #fff;
  text-align: center;
  position: absolute;
  left: 10%;
  right: 10%;
  top: 60%;
}
```

7. Styling the web page - styling the menu

Next, we style the menu as it's currently out of place. Essentially, we need the navigation menu to be at the top right side so as to ease browsing. Simply copy the code on the next page to the stylesheet. This code removes underlines from the menu items, positions

Fun fact - what Blotter isn't!

Blotter is great for elements like titles, headings and texts used for graphic purposes. It's not recommended that Blotter be used for lengthy bodies of text, and should in most cases be applied to words individually.

Tutorials

Create cool text effects with Blotter.js

them at the top right and sets properties for the visited and active links. A horizontal border is added to indicate when the links are hovered. Now render the page. Observe that the menu is now well placed at the top right side. Repeat steps 3 - 7 for the other index.html files, being careful to add a unique background in each page. When you navigate each of the links, they should now change to each of the different page backgrounds.

8. Set up JavaScript

In setting up the JavaScript environment, three JavaScript files are referenced: the blotter.min.js, which is a minified version of Blotter.js; script.js, which contains the custom JavaScript functionality that we will set up in the tutorial; and a material JavaScript file which sets up the required text effect. As you will observe, Blotter.js refers to its effects as materials and each type of text effect requires its own material JS file. The simplest approach to install blotter.js, which we will use, is accessing it directly from a CDN resource. Add the code below in the body section:

```
<script src='https://s3-us-west-2.  
amazonaws.com/s.  
cdn.io/2621168/blotter.min.js'></script>  
  
<script src='https://cdnjs.cloudflare.  
com/ajax/libs/  
Blotter/0.1.0/materials/fliesMaterial.min.  
.js'> </script>
```

Alternatively, Blotter.js can also be downloaded directly from GitHub here: <https://raw.githubusercontent.com/bradley/Blotter/master/build/blotter.min.js>. Once downloaded, it can be referenced directly as follows:

```
<script src="js/blotter.min.js" type="text/  
javascript"></script>
```

The fliesMaterial.min.js contains the JS functionality used in creating the first text effect which recreates text as a cloud of points. It's available in the CDN resource and can be downloaded for local use. Finally, create a new script.js file and save it in the root folder. The anticipated JavaScript functionality will be added to this file. Link it in the body section.

However, ensure to place it before the <section> containing the text.

Using Blotter.js with ease

Using Blotter.js involves only four steps. This is a simple checklist to help you adapt to the library once you have installed it in your applications...
1) Create the text to be rendered. 2) Create the material for the effect. 3) Pass the blotter text and material to the scope. 4) Append to the specified document element to render.

9. Create the Blotter text

The first text effect is created using the flies material file - it recreates text as a cloud of points which can be easily manipulated. Open script.js file and begin by creating a Blotter text object as shown in the code below. The name Ants! will be rendered as points in our case. Specify the font to be used, its size and fill colour.

```
var text = new Blotter.Text("Ants!", {  
    family: "serif",  
    size: 150,  
    fill: "white",});
```

10. Create the Blotter material

Next, the material is created. Begin by creating a Blotter material object and specify various attributes of the material in order to create a unique feel to the animated objects. The code assigns values to the cell width, its radius and speed.

```
var material = new Blotter.FliesMaterial();  
material.uniforms.uPointCellWidth.value =  
0.01;  
material.uniforms.uPointRadius.value = .8;  
material.uniforms.uSpeed.value = 5;
```

11. Rendering

Before the flies effect can be rendered, we need to pass the material object and the Blotter text to the canvas. Copy the code below to achieve the effect. Note that the previously defined div in step 3 is referenced as the location where the effect renders.

```
var blotter = new Blotter(material, {  
    texts : text,  
});  
var scope = blotter.makeText(text);  
let elem = document.getElementById('Ants');  
scope.appendTo(elem);
```

12. Positioning the flies effect

Since the effect renders at the top left corner, we next position it at the center of the page by adding CSS. Copy the code below which specifies absolute positioning for the element.

```
#Ants {  
    position: absolute;  
    left: 40%;  
    right: 30%;  
    top: 40%;  
}
```

13. Manipulating the flies effect

Now that the effect has rendered, try to manipulate some elements such as the actual text, width, radius and speed of the particles. As an example, the text is changed to "Bees", speed set to 8, and radius 0.6. Render the page, you should observe the text has changed to Bees and the particles are now moving much faster.



14. Channel split effect

Before we move on to the next effects we are assuming that that you have already created the four individual html files (index1.html to index4.html). It's also assumed that four different script files have also been created, each containing a unique cool text effect. Now, open index2.html, which is similar in structure to index.html. We need to make only two significant changes: renaming the div which holds the canvas to ChannelSplit <div id = "ChannelSplit" ></div>, And renaming the script file to the new JS file with the channel split effect <script src="split.js"></script>. Next, open split.js and begin by creating the Blotter text and Blotter material. Observe, though, that unlike in step 10, new properties are added for the effect.

```
var text = new Blotter.Text("Signals", {  
    family: 'Montserrat',  
    size: 150,  
    fill: "#000",  
    paddingLeft: 10,  
    paddingRight: 50,  
});  
var material = new Blotter.  
ChannelSplitMaterial();  
material.uniforms.uOffset.value =  
0.05;  
material.uniforms.uRotation.value =  
50;  
material.uniforms.uApplyBlur.value  
= 1; // 0 false,  
1 true  
material.uniforms.uAnimateNoise.  
value = .3;
```

15. Render the channel split effect

Next, we pass the text and material to the scope and render the effect similar to step 11. As earlier noted, the materials use a similar structure with only their properties changing.

16. Liquid distortion effect

Open index3.html. We also make only two significant changes, firstly renaming the div which holds the canvas to Liquid <div id = "Liquid" ></div>. Secondly, we rename the script file to the new JS file with the liquid distortion effect <script src="liquid.js"></script>. Next, open liquid.js and begin by creating the Blotter text and Blotter material. Simply copy the code.

[← FULL BLOTTER DOCUMENTATION](#)



FliesMaterial

Vecting in a tireless swarm, the `FliesMaterial` recreates your text as a cloud of animated points.

`Blotter.FliesMaterial` inherits from `Blotter.Material`, and therefore has the same properties and functions as `Blotter.Material`.

```
construction var material = new Blotter.FliesMaterial();

uniforms material.uniforms;
An object that holds the uniforms that describe how the material's effects should
be rendered. You should not manipulate this property directly.

uPointCellWidth material.uniforms.uPointCellWidth
```

Blotter materials - a shortcut to their application



Using the library for the first time can be quite daunting, especially if you're not used to a lot of coding. It turns out there is a shortcut to easily test out the actual settings you require for any effect you're exploring. Navigate to <https://blotter.js.org/#/materials> to view the five different types of Blotter materials. Select the Flies Material which we demonstrated in the previous steps. Observe that at the top right corner, a set of controls is provided to easily adjust the animation on the left side. Similarly, documentation is also provided for all properties that can be used with the material. It's advisable to first interact with the various controls and thereafter, paste the values to your application as you improve your understanding of the different properties.

Source

Define text style

```
const text = new Blotter.Text('Liquid', {
  family: "'EB Garamond', serif",
  size: 150,
  paddingLeft: 50,
  paddingRight: 50,
  fill: '#fff'
});
```

Use a material

```
// https://blotter.js.org/#/materials
let material = new Blotter.
LiquidDistortMaterial();
```

Set material opts

```
material.uniforms.uSpeed.value = 0.3;
material.uniforms.uVolatility.value = 0.10;
material.uniforms.uSeed.value = 0.1;
```

17. Create the liquid distortion effect

Pass the text and material to the scope and render the page afterwards. Ensure to add the styling for the render.

```
#Liquid {
  position: absolute;
  left: 30%;
  right: 30%;
  top: 40%;
}
let blotter = new Blotter(material, {
  texts: text
});
```

```
// Apply to element
let scope = blotter.forText(text);
let elem = document.
getElementById('Liquid');
scope.appendTo(elem);
```

18. Sliding door effect

Open index4.html. We also make only two significant changes: renaming the div which holds the canvas to Sliding `<div id = "Sliding"></div>`, and renaming the script file to the new JS file with the sliding doors effect `<script src='slide.js'></script>`. Next, open slide.js and begin by creating the Blotter text and Blotter material. Simply copy the code below.

```
var text = new Blotter.Text("Doors", {
  family: 'Montserrat',
  size: 150,
  fill: "#fff",
  paddingLeft: 50,
  paddingRight: 50,
})
var material = new Blotter.
SlidingDoorMaterial();
material.uniforms.uDivisions.value= 10;
material.uniforms.uDivisionWidth.value =m2;
material.uniforms.
uAnimateHorizontal.value = 1;
material.uniforms.
uFlipAnimationDirection.value = 1;
material.uniforms.uSpeed.value = .3
```

19. Render sliding door

Pass the text and material to the scope and afterwards, render the page. Ensure to add the styling for the render.

```
#Sliding {
  position: absolute;
  left: 30%;
  right: 30%;
  top: 40%;
}
var blotter = new Blotter(material, {
  texts: text
})
let scope = blotter.forText(text);
let elem = document.
getElementById('Sliding');
scope.appendTo(elem);
```

20. Animating the flies effect

As a conclusion, we demonstrate that the rendered text can also be animated in response to mouse movements. Simply open the split.js file and paste the code below. The code animates the rotation and offset values of the rendered text. Render the same to see the interesting distortions.

```
document.onmousemove = moveIt;
function moveIt(event) {
  material.uniforms.uRotation.
  value = (event.clientX * .1);
  material.uniforms.uOffset.value
  = (event.clientX * .0001);
}
```

SUBSCRIBE TODAY SAVE 70%

ON YOUR ANNUAL BUNDLE SUBSCRIPTION
WITH OUR EXCLUSIVE CODE*

ADD PHYSICS TO THREE.JS
Bring collisions, gravity and forces to your 3D sites.

TYPESCRIPT WORKFLOW
Standardise code contributions and reduce reviews with Husky

UTILISE FRAMER X
Add interactions to create more intuitive prototypes



Use voucher
code at checkout
DB70CRV
(online use only)



myfavouritemagazines.co.uk/net/code19

* Offer does not apply to digital only subscriptions and print only subscriptions

GREAT REASONS TO SUBSCRIBE

- Print edition delivered to your door
- 13 issues in a one-year subscription
- iPad and iPhone edition download
- Android edition download
- Money-back guarantee



CHOOSE A PACKAGE

PRINT EDITION ONLY

**SAVE UP TO
28%**

12-month subscription (13 Issues)

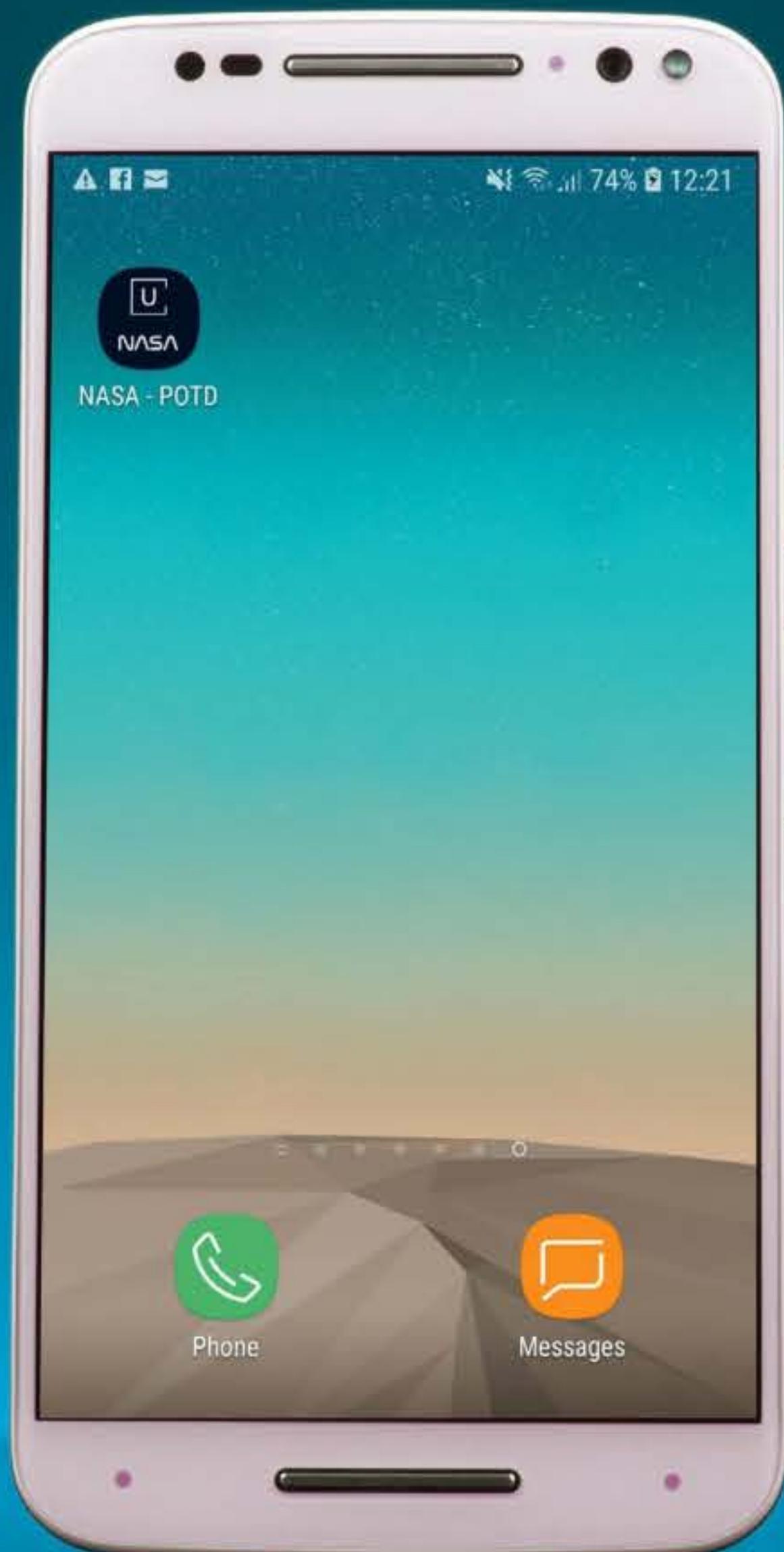
DIGITAL ONLY

**SAVE UP TO
67%**

12-month subscription (13 Issues)

TERMS AND CONDITIONS: This offer is available to all new subscribers. Applies to print + digital bundle subscriptions only for Imagine FX, Computer Arts, 3D Artist, 3D World, Net and Web Designer. Voucher code db70crv provides a 70% discount on print + digital bundle subscriptions only. The code can only be redeemed online via the web link stated. Prices and savings quoted are compared to buying full-priced print + digital bundles. You will receive 13 issues in a subscription year. You can contact us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. For full terms and conditions please visit: www.bit.ly/magterms. Offer ends 30/11/19.

Build an API-driven app - pt2



In the second part of this introduction to Ionic, learn how to connect NASA's apod API to an app, then build and run on iOS and Android devices



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



This is the second part of a two-part series on how to use your HTML, CSS and JavaScript skills to create a real mobile app for both Android and iOS using the Ionic framework.

If you missed the first article you can still follow along because the first step will get you to the right starting point.

In the first part we installed Ionic, created a blank template app, built up a basic mobile app that talked to a public API, and ran the app for testing purposes in your local browser. In this part we go through the process of building and deploying the app on your own Android or iOS device.

Building and deploying the app involves using a number of tools that need to be downloaded and configured, and can be a little tricky (one of them is also over 7Gb, so you also need to make sure you have plenty of space). In this article we aim to simplify it down to something as clear and straightforward as possible to get you up and running. To keep it simple, we will focus on building on Mac. If you're using Windows then the steps are the same but some of the command line instructions will be different; please refer back to the Ionic documentation.

If you built an app in part 1 of this tutorial (see Issue 292) you can jump straight to step 11 to build and deploy. Alternatively, with this article you can switch to another example app which displays the picture of the day from a NASA API. This API gets round issues with CORS (cross-origin security restrictions). We think it looks a bit cooler, but it makes no difference which app you choose.

1. Start where part 1 left off

If you didn't do part 1 of this tutorial from last month, start by downloading the finished project for part 1 either from the link supplied or from our public repository. Use "npm install", then "ionic serve" in "star-wars-info", to preview the project. Ensure you have Ionic and Cordova installed.

```
git clone https://github.com
UntappedSolutions/web-designer-ionic
tutorial.git
ionic -v
cordova -v
npm install
ionic serve
```

2. Change app to use real http.get()

In `home.page.ts` replace the `apiService` in the constructor with private `http$: HttpClient`. This gives you the real `http$.get()` used to make API calls. You will also need to add `HttpClientModule` to the imports in `app.module.ts`. Finally, remove the line calling `searchForPeople`.

```
home.page.ts
import {HttpClient} from '@angular/common/
http';
constructor(private http$: HttpClient) {}
```

```
ngOnInit() {
  this.people$ = this.http$
  get('https://swapi.co/api/people');
}
App.module.ts
imports: [BrowserModule, IonicModule
forRoot(), AppRoutingModule,
HttpClientModule],
```

3. Change HTML

Add an `*ngIf` to the `<ion-list>` tag which uses the `async` pipe. It's also important to use `as` as it gives you a variable to use when the observable returns data. Then change the `*ngFor` to loop through `people.results`. You'll now see 10 results from our Star Wars API on the screen.

```
<ion-list *ngIf="people$ | async as people">
  <ion-item
    *ngFor="let person of people
    results"
    no-padding
    [routerLink]="/detail/"
    person.id"
    routerDirection="forward">
    <ion-label>{{person.name}}</ion-label>
    <ion-icon name="planet" slot="end"
    color="tertiary"></ion-icon>
  </ion-item>
</ion-list>
```

4. Get API key from NASA:

Before we implement the NASA API for the picture of the day app, we will need an API key. Head to the NASA public API page and sign up for a free API key: api.nasa.gov. This is a common practice with all APIs and worth getting familiar with.

5. Generate array with 100 dates to list

In `home.page.ts` use JavaScript's `Date()` to get today's

date. Then create function to set up the last 100 days' array. It will take a start date and a count. Create a new instance variable initialised to an empty array then loop count number of times and each time push 2 items onto the array in an object, a date formatted specifically for the NASA API call, and one formatted for viewing.

```
public days = [];
const today = new Date();
this.setDaysArray(today, 100);
}
setDaysArray(date, noOfDays) {
  for (let n = 0; n < noOfDays; n++) {
    this.days.push({
      apiDate: this
      getApiDate(date),
      displayDate: this
      createDisplayDate(date)
    });
    date.setDate(date.getDate()
    - 1);
  }
}
```

6. Angular date formatting

Use the angular function `formatDate` to take a JavaScript date and format it into any desired string. It takes 3 parameters: the date object, a desired format and a time zone. We will use "yyyy-MM-dd" for the api and "MMM d yyyy" for on screen display.

```
import {formatDate} from '@angular/common';
getApiDate(date)
{
  return formatDate(date, 'yyyy-MM-dd',
  'en_GB');
}
```

Tutorials

Build an API driven app

The screenshot shows a browser window titled "Ionic App" displaying a list of NASA APOD images. The list is ordered by date from Sep 6 to Sep 19, 2019. Each item in the list has a small icon and a right-pointing arrow. The DevTools panel is open, showing the DOM structure, styles, and console logs. The DOM panel shows the HTML code for the page, including scripts for runtime, polyfills, styles, vendor, and main. The styles panel shows the CSS for the body element, including font smoothing and margin/padding rules. The console panel shows Angular startup messages and a warning about StatusBar.styleDefault not being available.

```
createDisplayDate(date) {
  return formatDate(date, 'MMM d yyyy',
  'en_GB');
}
```

7. Change home HTML to use new days array

Change the ngFor to loop through days, as well as changing the detail from id to day.apiDate. You will also need to use day.displayDate within the label. Finally, change the title of the page and remove *ngIf from the <ion-list> component. In app-routing.module.ts change "id" to be ":date". Our first screen is now complete.

```
<ion-list>
  <ion-item
    *ngFor="let day of days"
    no-padding
    [routerLink]="/detail/" + day
    apiDate"
    routerDirection="forward">
    <ion-label>{{day.displayDate}}<
    ion-label>
    <ion-icon name="planet" slot="end">
```

Refactor tool in IDE

It's useful to delete files using a refactoring tool within your IDE. This will check through all your files for any uses and alert you before deleting the file.

```
color="tertiary"></ion-icon>
</ion-item>
</ion-list>
```

8. API call on detail page

In detail.page.ts, change detail to be date. Then using the new planetary/apod NASA API with your api_key added as a parameter, call http\$.get(). This can be copied in from [home.page.ts](#) and then removed from there. Be sure to copy/remove the http\$ initialiser in the constructor as well.

```
export class DetailPage implements OnInit {
  private date: string;
  public detailData$;
  constructor(
    private activatedRoute: ActivatedRoute,
    private http$: HttpClient) { }

  ngOnInit() {
    this.date = this.activatedRoute
    snapshot.paramMap.get('date');
    this.detailData$ = this.http$
    get('https://api.nasa.gov/planetary
    apod?api_key=Fg8TtYfOzIZvqh0wIx83N9TVo4jWsji
    PT8pz7rA&date=' + this.date);
  }
}
```

9. Detail.page.html changes

We already have detail.page.html set up using an async pipe with an API call. That means now all we need to do is change the properties shown to match

the new API data structure. Use url, title, date, explanation and copyright to fill out the detail on the page.

```
<ion-content *ngIf="detailData$ | async as
detailData" class="content-container">
  <ion-img src="{{detailData.url}}" alt="{{detailData.title}}" center></ion-img>
  <h1>{{detailData.title}}</h1>
  <p>{{detailData.date}}</p>
  <p>{{detailData.explanation}}</p>
  <p>{{detailData.copyright}}</p>
  <ion-content>
```

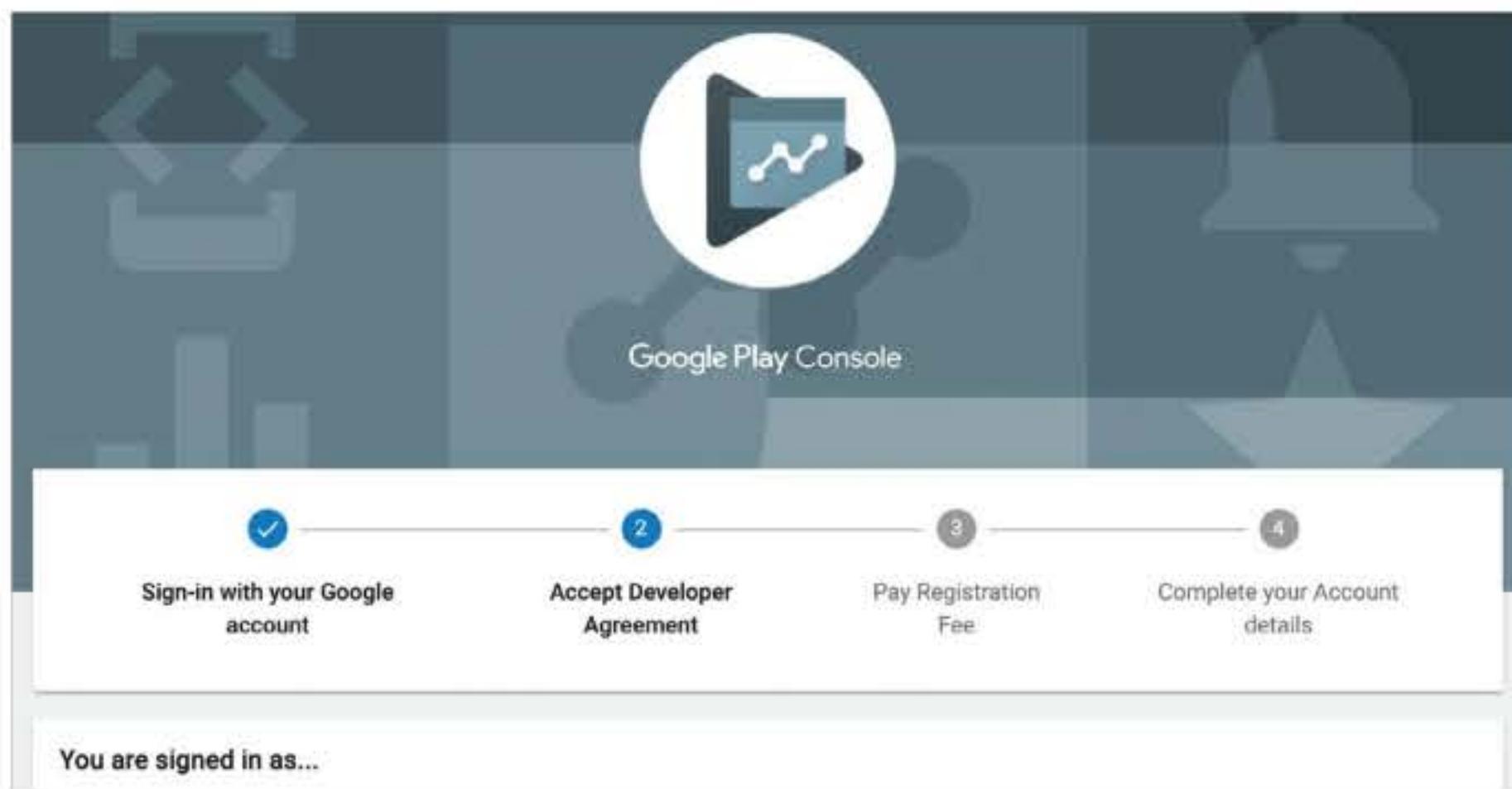
Back NASA - Picture of the day



A Harvest Moon

Date taken: 2019-09-13

Famed in festival, story, and song the best known full moon is the Harvest Moon. For



10. Delete dummy service and clean code

Now we have a basic working app making a real API call, it's a good time to go through the code and remove anything that isn't needed. We have replaced the use of a service with a real `http$get()` call from within the detail page, so our services directory can be removed now. In addition, check that there is no redundant code left in home or detail pages.

11. Make splash screen and icon resources

Ionic is a JavaScript framework that is wrapped around Cordova. The Cordova CLI has many useful functions: one is making the right size app icon and splash screens for all phone sizes. Save an `icon.png` file in `/resources` measuring `1024x1024px`, and a `splash.png` measuring `2732x2732px`. Run Ionic Cordova resources and it will generate all the assets required for iOS and Android.

```
npm i -g cordova-res
ionic cordova resources
```



12. Edit configuration to show new name

We have chosen to use Cordova as our build process for this article (the alternative Capacitor is a relatively new build process). You should now run Cordova to prepare your project for build. Open newly created

```
<?xml version="1.0" encoding="utf-8"?>
<widget id="io.ionic.startert" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
  <name>NASA - POTO</name>
  <description>NASA - Picture of the Day</description>
  <author email="contact@untapped.com" href="http://thisisuntapped.com/">Ionic Framework Team</author>
  <content src="index.html" />
  <access origin="*"/>
  <allow-intent href="http://*/*" />
  <allow-intent href="https://*/*" />
  <allow-intent href="tel:*" />
  <allow-intent href="mailto:*" />
  <allow-intent href="file:///>
  <allow-intent href="geo:*" />
  <preference name="ScrollEnabled" value="false" />
  <preference name="android-minSdkVersion" value="10" />
  <preference name="BackupWebStorage" value="none" />
  <preference name="SplashMaintainAspectRatio" value="true" />
  <preference name="FadeSplashScreenDuration" value="300" />
  <preference name="SplashShowOnlyFirstTime" value="false" />
  <preference name="SplashScreen" value="screen" />
  <preference name="SplashScreenDelay" value="0000" />
  <preference name="ShowSplashScreenSpinner" value="true" />
  <preference name="SplashScreenSpinnerColor" value="white" />
```

`config.xml` and then change the app's name, description and author.

```
<name>NASA Image Viewer</name>
<description>An app built using NASA's picture of the day API</description>
<author email="contact@thisisuntapped.com" href="http://thisisuntapped.com">Untapped</author>
```

13. Set up Android build environment - Install JDK

To build your webapp for an Android device, set up JDK (Java Development Kit). More details on the installation instructions can be found on the Ionic set up page. When installing JDK8, make sure you have version 8 selected – Cordova will only build with this version. The lines below help roll back to version 8.

```
Java -version
/usr/libexec/java_home -V // shows
installed JDK versions
export JAVA_HOME=$(/usr/libexec/java_home -v
1.8) //Selected version 8 as chosen dev
environment - mac specific
Java -version // check final result
```

Android Play Store release

The release process for an Ionic app is a discipline in itself. Often handled by product managers in bigger organisations, it requires specific knowledge of the Apple App Store and Google's Play Store. We've covered the basics here to get you started.

Releasing an app through the Android Play Store is a little easier than iOS. They don't have a beta test platform – the expectation is that you pass around the app's .apk file for test installing, and then once ready only use the platform to release to the public. Within the Play Store console, go to Releases. Here you can upload your APK file and set release notes and information. Once the release is set up, you can push it live. Note that Android do not approve every build and release, you have to be 100% sure your app is bug-free and ready before releasing.

14. Gradle & Studio

Gradle is a build tool and compiles the project from JavaScript to Java/Swift. Details can be found on their website but we advise using a package manager like Homebrew to install. Next, follow the instructions on Android Studio's website to install.

Installing gradle

```
brew install gradle // will auto update
homebrew if installed already
export PATH=$PATH:/opt/gradle/gradle-5.6.2
bin
```

Installing Android Studio

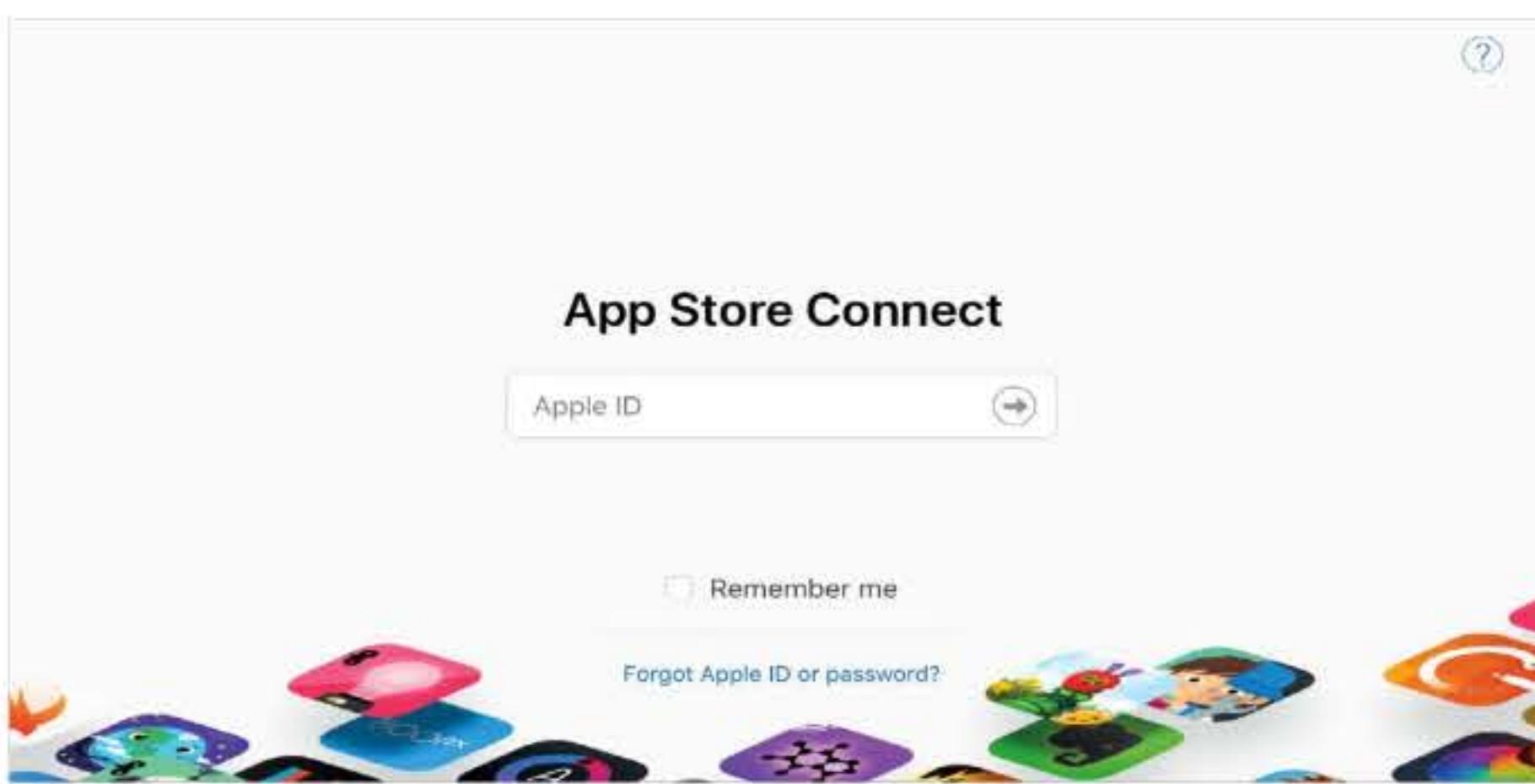
```
export ANDROID_SDK_ROOT=$HOME/Library
Android/sdk
export PATH=$PATH:$ANDROID_SDK_ROOT
platform-tools - install adb
// Ensure these have been added to .profile
bash file so can be used by other
directories
```

15. Ensure right SDK for target phone

In SDK Manager, once installed, ensure you have the target SDK for the Android device that you wish to run on. Your version can be found in the About

Tutorials

Build an API driven app



section of Settings on your phone. Then in SDK Manager, download and install the correct version for your device.

16. Build Android .apk file

You're now ready to build the Android app! Run the Ionic Cordova build command to build. This will create a file in /platforms/android/app/build/outputs/apk/debug/. You can use flags to do production, debug or release builds. Debug is useful if you want to use Inspect element to see what's happening while running on device.

```
Ionic cordova build android  
Ionic cordova build android --prod  
Ionic cordova build android --prod --release  
Ionic cordova build android --debug
```

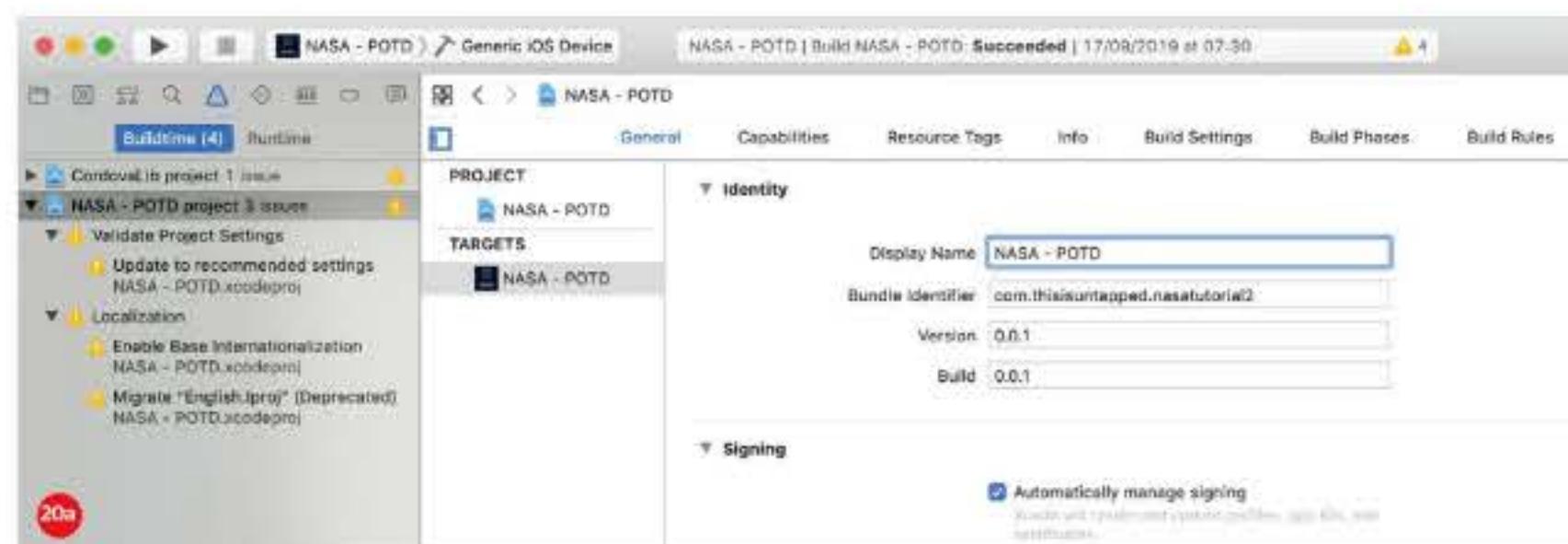
17. Install on Android device

Enable developer settings on your Android device and ensure stay-awake and USB-debugging are on, then connect device accepting any pop-ups. Open a terminal window and enter the below commands to connect over ADB. Open directory/platforms/android/app/build/outputs/apk/debug in terminal and run ADB install -d app-debug.apk to install the app on your device!

```
adb devices  
// may need to run adb devices a few times  
after re-connecting device  
adb kill-server // useful command
```

Splash screen options

You can edit some splash screen properties such as duration open and fade-out length. You will need to delete platforms and re-run Ionic Cordova resources



```
adb start-server // useful commands  
adb install -d app-debug.apk
```

18. Set up iOS build environment

Download and install x-code from Apple, this requires an app ID. You need to open and then let the wizard finish installation. Once installed run xcode-select --install to install command line tools.

```
xcode-select --install  
npm install -g ios-deploy
```

19. Setting up development team

All iOS apps must be code signed, even for development. Luckily, Xcode makes this easy with automatic code signing. The only prerequisite is an Apple ID. Go to Xcode > preferences > Accounts and click plus to add account and sign in with Apple ID. Once logged in, a Personal Team will appear in the team list of the Apple ID.

20. Build for iOS

Run Ionic Cordova build iOS and then open the generated xcode file .xcodeproj. From here you can either run on a simulated device or run/install on your real device. Click the blue icon on the left at the root of the tree and from here you can change your app's name and also ensure that under Signing, your Team is selected in the drop down. Click device in the top

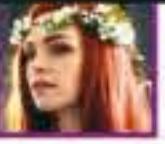
Apple App Store release process

Once you have built and tested your app locally, you will want to upload it to Apple's App Store to either send it out for beta testing or full release. Apple have the Test Flight product, which is a very useful tool that gives clients the ability to test your app without installing direct files. From within Xcode, use the Archive option to start the upload process. We recommend using the auto-signing option to simplify the process. Once your archive is created it will give you the option to upload it to your Apple account. Sign in via Apple Connect to either release the app to individuals via Test Flight or into the App Store itself. The Apple staff are very helpful in resolving any app approval issues. Quick tip: it pays to be kind and helpful in comms with the approval team!

toolbar and select your connected iPhone from the list. Hit run build and install on device!

```
ionic cordova prepare ios  
ionic cordova build ios
```



FREE! 113 MINUTES OF PRO VIDEO TUITION + 60 FANTASY FIGURE REFERENCE PHOTOS  + CUSTOM BRUSHES

1 NO.1 FOR DIGITAL ARTISTS **ImagineFX**



THE ART OF COMMISSIONS

The do's and don'ts of painting for clients

WIZARD ADVICE

GET WORK AT MAGIC!

Create a portfolio for Magic: The Gathering

FRESH SKILLS IN

PORTRAIT ART

Make bold choices and try out unique techniques with our brilliant step-by-step workshops!



PAINTING DRAGONS

HOW TO EMBRACE THE CHAOS OF WATERCOLOUR ART!

TAP FOR
TEXT-FREE
COVER!



ISSUE 180

INSIDE!

QUICKLY BUILD UP CONTRAST IN PENCIL

THE LATEST KRITA ART TECHNIQUES

ANDREW HEM ON ART CONNECTIONS

ESSENTIAL ART RESOURCES

Exclusive videos and custom brushes are available with your digital editions!

38

SKETCHES TO INSPIRE YOU!

WATCH & LISTEN

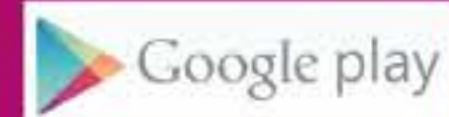
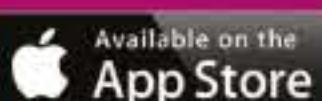


FREE
67-MINUTE LESSON

App Store is a trademark of Apple Inc. registered in the U.S. and other countries. App Store is a service mark of Apple Inc.

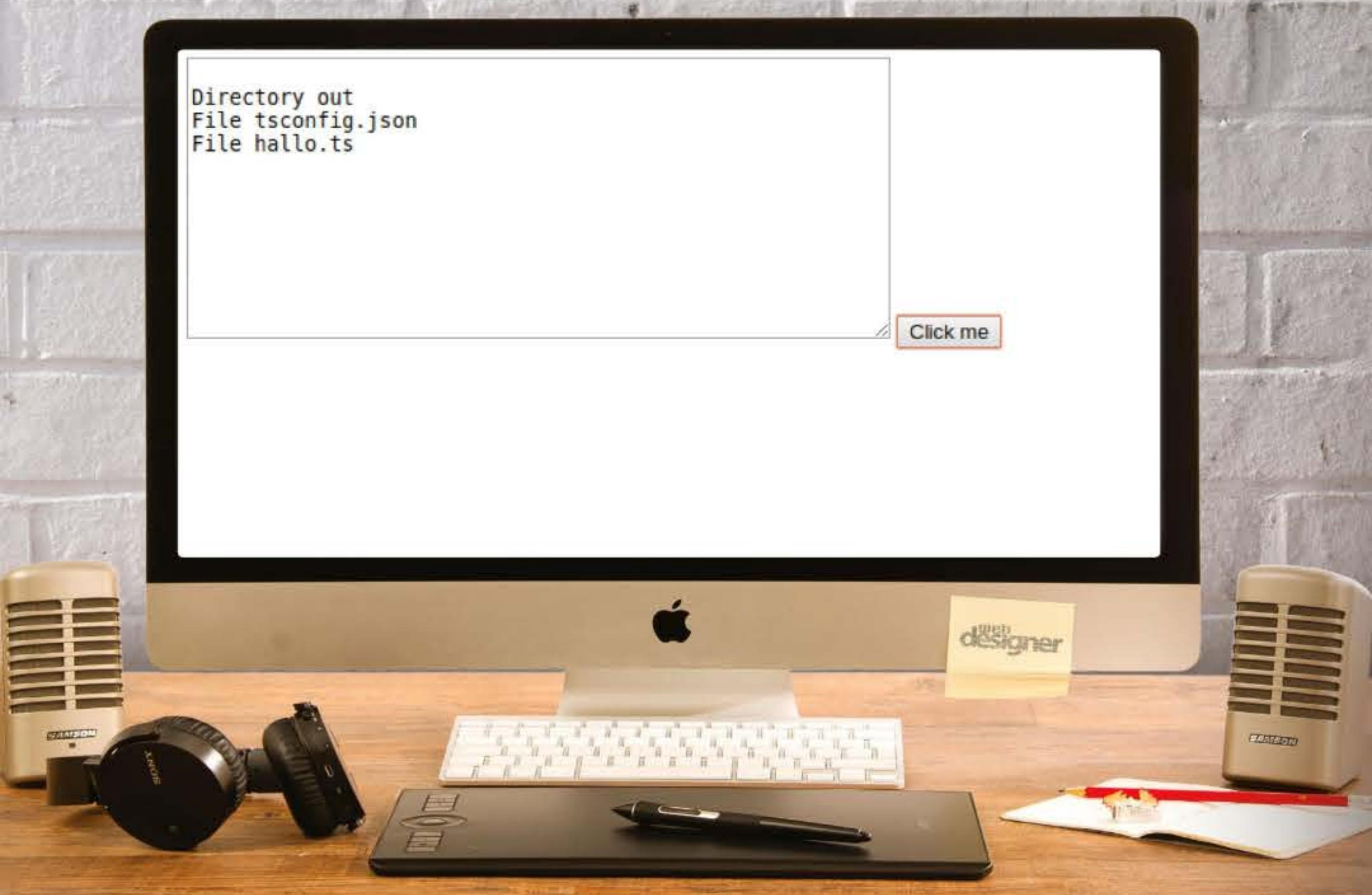
Try out a digital edition for **FREE** today!

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



Say hello to Google's Native File System API

A recently-introduced Web API concept intends to let websites access the host file system directly



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





iven that the web traditionally is a place crawling with all kinds of miscreants, more than one web designer may well suffer from **hypertonia** while reading the title of this tutorial. Letting JavaScript code loose on a client's workstation sounds like an incredibly stupid idea. But fret not. Google has taken measures to ensure that the native file system API cannot be abused excessively. Due to that, developers working on browser-based applications will be delighted to hear about the new-found possibilities. For the first time in the history of the web, JavaScript code running outside of Node.js will be able to access the remanent storage of its host. This, of course, leads to a variety of fascinating options - for example, large artificial intelligence programs could store gigabytes' worth of data which is then processed in ASM.js. This ties in nicely to various other recent developments.

However, at the time of writing, the API has significant limitations and is only available inside of Google Chrome. Read on to find out more about the current state of the native filesystem API and about the current issues faced by early adopters.

1. Chrome ahoy!

Access to the native filesystem API is limited to beta versions of Chrome at the time of writing. Sadly, the Ubuntu package sources only provide stable builds. Fortunately, installing a different version is not difficult. Simply enter the command shown accompanying this step.

```
tamhan@TAMHAN18:~$ sudo apt-get install  
google-chrome-beta  
[sudo] password for tamhan:
```

2. Fire up the beta

Chrome betas can install along the normal version. Invoking it is best done via the command line using the command sequence shown. When the browser starts up for the first time, it will ask you to perform another configuration run - the profile storage of the beta of the normal version are not shared.

```
tamhan@TAMHAN18:~$ google-chrome-beta  
[11505:11505:0921/211209.225952:ERROR:sandb  
ox_linux.cc(369)] InitializeSandbox() called  
with multiple threads in process gpu-  
process
```

3. Ensure versioning

3. Beta Version
Beta versions of Chrome have a different toolbar icon. Should you feel unsafe, simply open the About Chrome window. It will show version information looking similar to the one in the figure accompanying this step.

4. For other operating systems

Chrome betas are not exclusive to Ubuntu Linux

About Chrome

Google Chrome

Version 78.0.3904.21 (Official Build) beta (64-bit)

Get help with Chrome

Report an issue

3

file-system

Reset all to default

Experiments 78.0.3904.21

Available Unavailable

Native File System API
Enables the experimental Native File System API, giving websites access to the native file system – Mac, Windows, Linux, Chrome OS, Android
#native_file-system api

Enabled ▾

6

Should you find yourself on a Windows or Mac OS machine, open the URL <https://www.google.com/intl/gb/chrome/beta> in a browser of your choice. This will open the page - click the blue button to get platform-specific update instructions.

5. Enable for Localhost

While the native filesystem API can be enabled on a per server basis, we shall limit ourselves to activating the function for localhost. Open **chrome://flags/** in order to access the flag configuration facility. Next, look for the string "file-system" and proceed to clicking "Available".

6. Enable, part deux

After that, click Default to open a menu in which you select the enabled option leading to the result shown in the figure. Chrome will display a bar at the bottom of the screen - use its button to perform a restart to enable the feature.

7. Open a file picker

Putting the pedal to the metal requires us to get a file handle. It is a data structure which describes a relationship between your browser and a file or folder on the host. This, fortunately, can easily be handled by invoking the method `chooseFileSystemEntries`. It proceeds to displaying the file chooser dialog of the host operating system.

```
<html>
<body>
    <script >
        async function myFunction(){
            fileHandle = await window.
chooseFileSystemEntries();
        }
    </script>

    <button onclick="myFunction()">Click
me</button>
</body>
</html>
```

8. Beware of the secure context

Common dialogs can be used to overflow a user's browser. Google thus lets you invoke chooseFileSystemEntries only from a user's event handler.

Not on Android!

NOT ON ANDROID: Google tries to make new Chrome features available on Android. This currently is not the case here - the ever-tighter permission model around the file system seems to pose UX challenges.

Tutorials

Say hello to Google's Native File System API

Abstract

This specification provides an API for representing file objects in web applications, as well as programmatically selecting them and accessing their data. This includes:

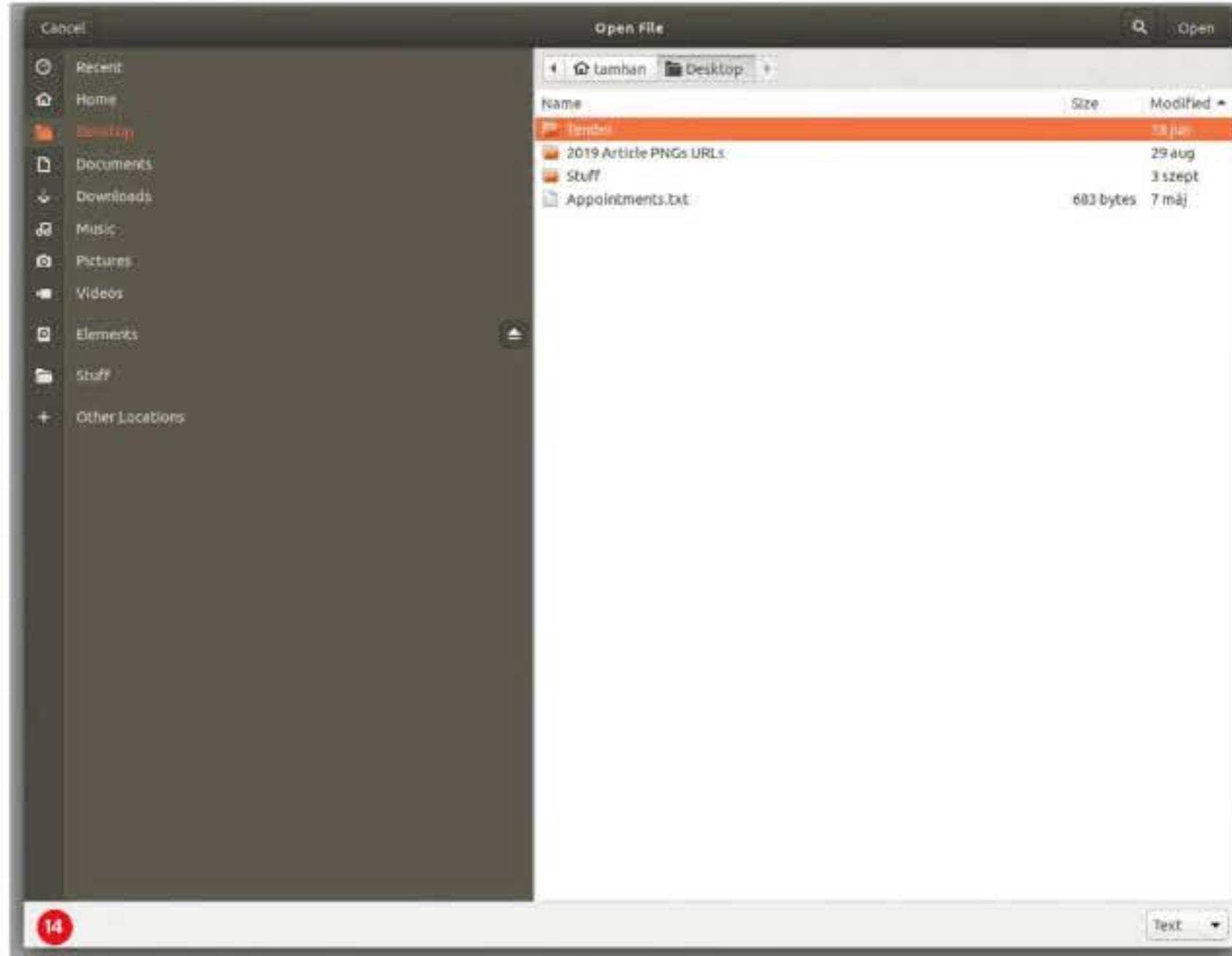
- A [FileList](#) interface, which represents an array of individually selected files from the underlying system. The user interface for selection can be invoked via `<input type="file">`, i.e. when the input element is in the [File Upload state](#).
- A [Blob](#) interface, which represents immutable raw binary data, and allows access to ranges of bytes within the [Blob](#) object as a separate [Blob](#).
- A [File](#) interface, which includes readonly informational attributes about a file such as its name and the date of the last modification (on disk) of the file.
- A [FileReader](#) interface, which provides methods to read a [File](#) or a [Blob](#), and an event model to obtain the results of these reads.
- A [URL scheme](#) for use with binary data such as files, so that they can be referenced within web applications.

Additionally, this specification defines objects to be used within threaded web applications for the synchronous reading of files.

[§ 10 Requirements and Use Cases](#) covers the motivation behind this specification.

This API is designed to be used in conjunction with other APIs and elements on the web platform, notably: [XMLHttpRequest](#) (e.g. with an overloaded `send()` method for [File](#) or [Blob](#) arguments), [postMessage\(\)](#), [DataTransfer](#) (part of the drag and drop API defined in [\[HTML\]](#)) and [Web Workers](#). Additionally, it should be possible to programmatically obtain a list of files from the `<input>` element when it is in the [File Upload state](#). These kinds of behaviors are defined in the appropriate affiliated specifications.

10



The code must furthermore be served in a secure context. In practice, this usually means that https is required per <https://w3c.github.io/webappsec-secure-contexts/#localhost>. Critically, the W3C specification does not provide a final decision for the localhost server.

Given that uncertainty, user agents *may* treat localhost names as having potentially trustworthy origins if and only if they also adhere to the localhost name resolution rules spelled out in [let-localhost-be-localhost] which boil down to ensuring that localhost never resolves to a non-loopback address.

9. Testing for functionality

Invoke the file using a Local Web server of choice. Next, proceed to clicking the button in order to trigger our event handler. If everything is set up correctly, a CommonDialog will pop up. The most common problem at this point in the procedure is forgetting to restart the browser after enabling the native API.

10. File API, reloaded

Getting a file handle is part of the solution. Interacting with the contents of the file in storage requires the use of the W3C file API, which is documented at <https://w3c.github.io/FileAPI/>. The figure accompanying this step provides an overview of the most important functions.

11. We need ROT13!

ROT13 might not be the safest encryption algorithm out there at the moment, given that it is only a simple substitution process. But for now, it will suffice. Start out by adding the ROT13 function from <https://gist.github.com/dsoares/c65f1bacac0acd46fd9f> to our project.

```
<script >
    function rot13(str) { // LBH QVQ
        var re = new RegExp("[a-z]", "i");
        var min = 'A'.charCodeAt(0);
        var max = 'Z'.charCodeAt(0);
        var factor = 13;
        var result = "";
        str = str.toUpperCase();

        for (var i=0; i<str.length; i++) {
            result += (re.test(str[i]) ?
                String.fromCharCode((str.charCodeAt(i) -
                    min + factor) % (max-min+1) + min) :
                str[i]);
        }

        return result;
    }
```

12. Add a workspace!

With that, the next step in the process is creating a text area which is to be populated during program

Active Trials	My Registrations	Completed Trials
Badging	Makes it possible to display a small numeric badge in some operating system specific context (e.g. the dock on macOS or task...	REGISTER
Badging V2	Makes it possible to display a small numeric badge in some operating system specific context (e.g. the dock on macOS or task...	REGISTER
Contact Picker	The Contact Picker allows websites to request the names, e-mail addresses and/or phone numbers of the contacts available...	REGISTER
Dynamic delegation of autoplay	Dynamic delegation of autoplay by transfer of user activation	REGISTER
getInstalledRelatedApps (For Chrome 73+)	Allow web sites to determine whether the user has installed their corresponding app.	REGISTER
hrefTranslate	Supporting the hrefTranslate attribute on the HTML.Anchor prototype	REGISTER
is-input-pending	In order to enable developers to complete their work as fast as possible if the user isn't interacting, but respond to user input s...	REGISTER
MediaCapabilities: encrypted (EME) decodingInfo()	New encryption configuration inputs have been added to the decodingInfo() API. This will expose these so sites using enc...	REGISTER
Native File System API	The new Native File System API enables developers to build powerful web apps that interact with files on the user's local devic...	REGISTER
Per-method quota for canMakePayment()	This feature allows a website to query whether the user has a Google Pay wallet or cards in a user's database in separate coin...	REGISTER

On the art of API design

Getting an API design right on the first go is almost impossible. Problems usually only crop up once developers start to exercise the functionality in danger. As changing an already-released API (often called the binary break) leads to all kinds of bad blood, the facility of the Origin Trial has recently been introduced. It's a new-fashioned way to test experimental APIs. In principle, a developer must sign his website up for testing the API, and is then provided with a unique ID. This ID is embedded in the HTML markup of the application served on this domain. From that point in time, the application can use the new-fashioned APIs. Information on this topic, as always, is available on GitHub via <https://bit.ly/2mDYKra>.

execution. Simply modify the sample as per the code outlined in the step.

```
</script>
<textarea id="textEditor" aria-
label="Text Editor"></textarea>

<button onclick="myFunction()">Click
me</button>
</body>
```

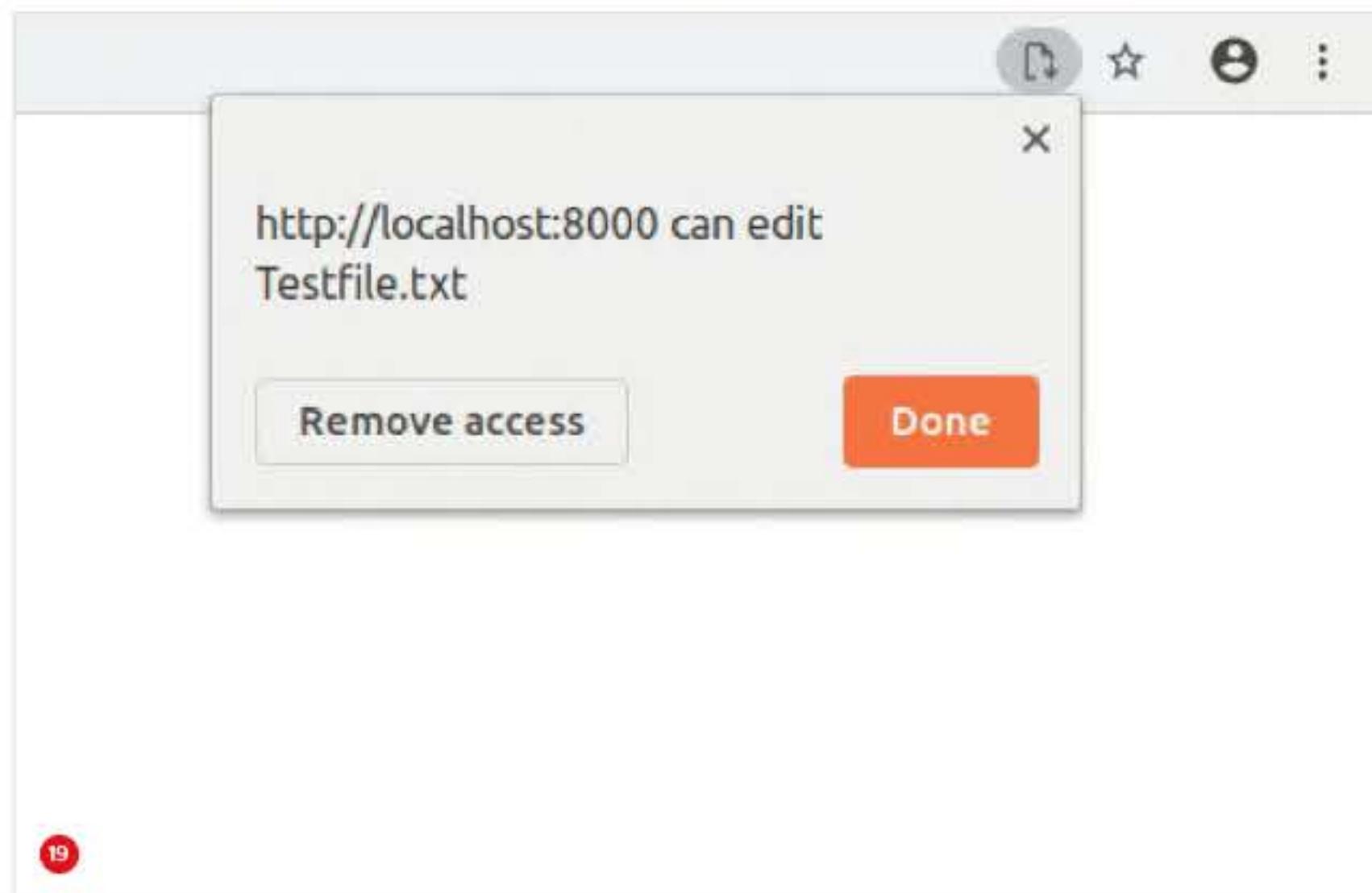
13. Understand what's happening

The file method simply returns a file reference. Our code must, however, access the text which is stored in the file. This is made more complex by modern chapter encodings which use more than one byte per character. Fortunately, invoking the text method takes care of this problem.

```
async function myFunction(){
    var textArea = document.
getElementById("textEditor");
    fileHandle = await window.
chooseFileSystemEntries();
    const file = await fileHandle.
getFile();
    const contents = await file.
text();
    textArea.value =
rot13(contents);
}
```

14. Restrict the choices

By default, a common dialog lets users select any file they want. Given that our code works with text files, we can pass in a parameter to restrict the types. The parameters shown in the code restrict the dialog to .txt files, leading to the result displayed in the figure.



19

```
async function myFunction(){
    var textArea = document.
getElementById("textEditor");
    fileHandle = await window.
chooseFileSystemEntries({
        accepts: [
            {
                description: 'Text',
                extensions: ['txt']
            }
        ]
});
```

15. Store a file!

Performing ROT13 encryption is just half of the experiment. Our code would be much more useful if

it could save the results of its work. For this, we need to return to the common dialog to get another file handle. It will point to the output file...

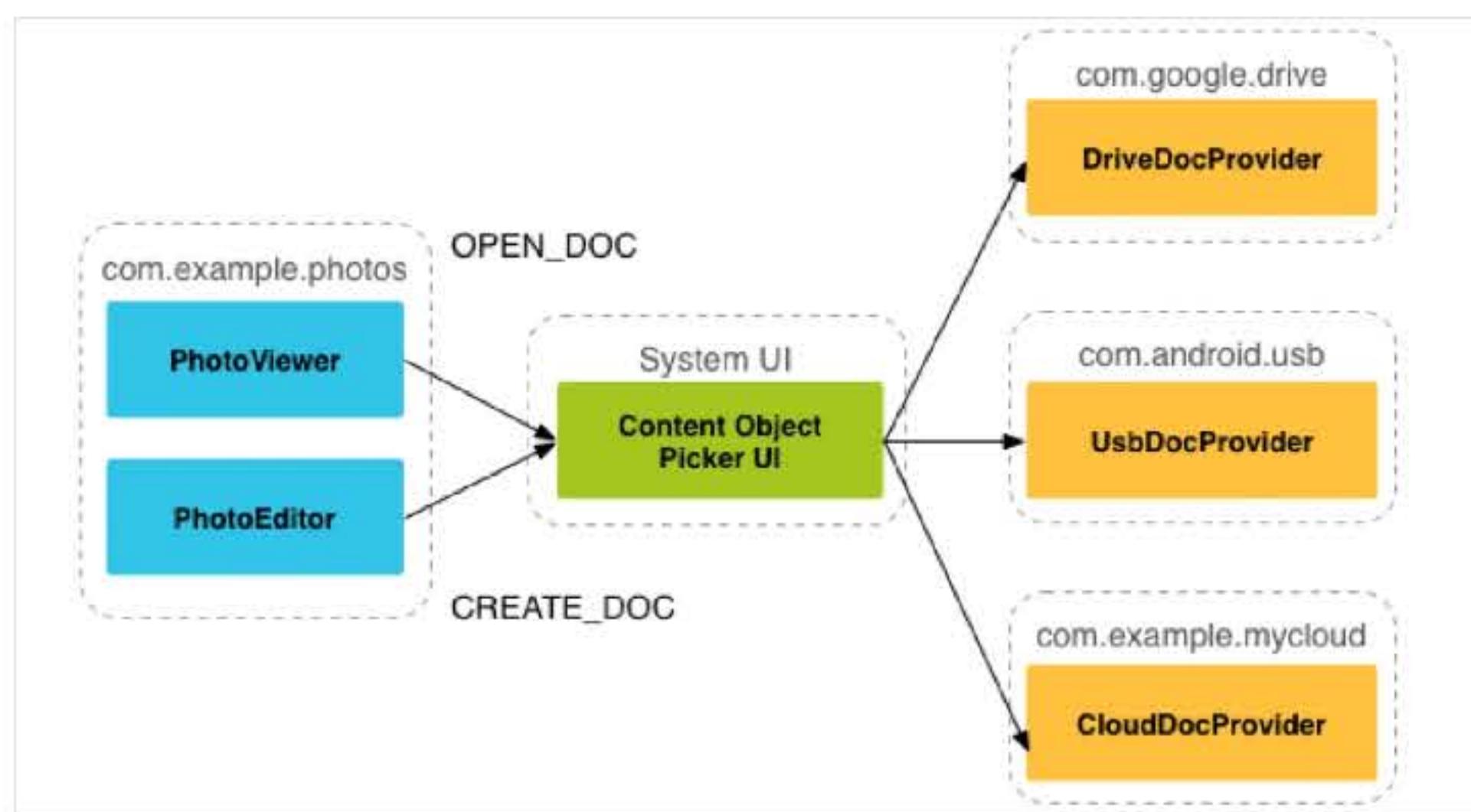
```
function getNewFileHandle() {
    const opts = {
```

CommonDialog?

CommonDialog frequently sees use as a generic name for control libraries providing file pickers and similar niceties. The term comes from a control which Microsoft introduced in early versions of its Visual Basic RAD suite.

Tutorials

Say hello to Google's Native File System API



Why Android is problematic

Developing file managers and similar programs was simple in early versions of Android. As long as the program declared the relevant capability, the operating system got out of the way. Sadly, Google changed all of that by introducing the storage access framework (SAF for short) which brought along all kinds of restrictions. It also requires the use of the relatively complex class structure shown in the image accompanying this boxout. As if this weren't enough, the user interface is also extremely confusing - the Chrome team even admit this openly in the web site of the native file system API. Yours truly sees these problems with pleasure. Developers have been complaining about the SAF, only to be ignored.

```
        type: 'saveFile',
        accepts: [
            description: 'Text file',
            extensions: ['txt'],
            mimeTypes: ['text/plain'],
        ],
    },
    const handle = window.
        chooseFileSystemEntries(opts);
    return handle;
```

16. Writing, part one

In principle, the code for writing is not too different from its reading colleague. We invoke the above-mentioned method to get our hands on a file handle, and continue to create a WriteStream.

```
async function mySFunction(){
    var fileHandle = await getNewFileHandle();
    const writer = await fileHandle.createWriter();
```

17. Writing, part two

Files should be truncated before being written to. Keep in mind to invoke close after the file has been saved. Most operating systems come with relatively complex caching algorithms, which wait for an opportunity to make a developer's life difficult by delaying writes.

```
    await writer.truncate(0);
    await writer.write(0, "This is a
sample!");
}

await writer.close();
```

18. Run a test

Save the file and refresh its contents in Chrome. After that, a click on the save button is enough to trigger the saving process. When working on a practical encryption tool, you would – of course – replace our constant string with the content of the TextView.

19. Mind the prompt

Google's main venue of protecting the native file system API is the use of prompts. Clicking the save button leads to a new symbol in the toolbar (see figure). It informs the user that the currently-loaded web application has write access - for now. The permission expires the moment the page is reloaded.

20. Directory management

Google does not limit the native filesystem API to interacting with files. We can also enumerate an entire directory if the user selects it. For this, the `chooseFileSystemEntries` method must be invoked with the `openDirectory` parameter.

```
async function myFunction(){
    var textArea = document.
getElementById("textEditor");

    const opts = {type:
'openDirectory'};
    const handle = await window.
chooseFileSystemEntries(opts);
```

21. On Cancel?

`chooseFileSystemEntries` is purely promise-based. Should the user decide to click the Cancel button of the `CommonDialog`, the promise will be "failed" by the JavaScript engine. This usually also aborts the execution of the rest of the routine.

22. Directory management, part deux

Selecting a folder leads to the display of another permission dialog. After the confirmation dialog is affirmed, the results of the transaction will show up in the output window (see figure). The actual output code isn't that difficult - entries is but an iterable containing one entry for each element in the filesystem.

```
const entries = await handle.getEntries();
for await (const entry of entries) {
    const kind = entry.isFile ? 'File' :
'Directory';
```

23. What happens next?

Google actively pushes forward development - a general release is intended in Chrome 81. Check the Chrome Platform Status site (www.chromestatus.com/features/schedule) for release dates. It is worth noting that neither Microsoft nor Mozilla have yet committed to the API.

Chromie Platform Status		All Features · Releases · Samples · Beta
Release Timeline		
Please note: chrome beta addresses are subject to change.		
STABLE		
 Chrome 77	NEXT UP	 Chrome 79
Beta was Aug 8 - Aug 15 Stable 21 days ago (Sep 10)	Beta between Sep 19 - Sep 26 Stable in 21 days (Oct 22)	Beta coming Oct 31 - Nov 7 Stable in 70 days (Dec 10)
FEATURES IN THIS RELEASE		
Tab	FEATURES PLANNED IN THIS RELEASE	
Error Page API	Blink	Performance , Web Camera , Video
Polymer 2.0	Chromium API	Performance , Web Camera , Video
Document Reference Request Response	Blink	Performance , Web Camera , Video
Performance Budget Monitoring in CSS	CSS	Performance , Web Camera , Video
(Revised) Shared Worker API	CSS	Performance , Web Camera , Video
HTTP/2 Client Connection Establishment	HTML>CSS	Performance , Web Camera , Video
Service Worker	HTML>CSS	Performance , Web Camera , Video
Websocket Upgrade Functionality in Service Workers	Blink>Fonts	Performance , Web Camera , Video
HTTP/2 Server Pushes	Blink>Fonts	Performance , Web Camera , Video
WebRTC stereo microphone API	Blink>HTML	Performance , Web Camera , Video
WebRTC stereo microphone API	Blink>HTML>Focus	Performance , Web Camera , Video
Ttk-Accessibility	Blink>JavaScript>Internationalization	Performance , Web Camera , Video
Annotations That Is CSS3 Compliant	Blink>JavaScript>Internationalization	Performance , Web Camera , Video
<small>Google is an open source project. The contents of this page under CC-BY-SA license. Code on GitHub: https://github.com/GoogleChrome/chrome-status</small>		

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



Get Creative Bloq **direct to your inbox** with our weekly web design newsletter



Graphic design

Web design

3D

Digital art

www.creativebloq.com



WEBXR

UNIFYING REALITY TECHNOLOGIES

WEBXR IS THE LATEST EVOLUTION IN THE EXPLORATION OF VIRTUAL AND AUGMENTED REALITIES. IN THIS ARTICLE, YOU'LL LEARN WHAT WEBXR IS AND INVALUABLE RESOURCES TO GET STARTED USING IT.



“

As the pace of technology races along, it has become critical to unify approaches and standards, before fragmentation inhibits further growth. WebXR taps the power of the web along with the unification of these realities, under one philosophical umbrella.

”

Richard Mattka

Creative Director | Designer | Developer

twitter.com/synerayseeker



WEBXR CHALLENGES

PERFORMANCE AND SECURITY ARE CRUCIAL TO SUCCESS

Convincing XR experiences depend on low latency, high precision and large amounts of data processed rapidly to render out scenes, animations and much more. Latency, which refers to the delay in communications between a user's movement and what they see, has to be as low as possible. Ideally this should be in the range of 20ms or better to create the illusion of reality. If there is even a slight glitch or alignment issue, it can disrupt the experience or even cause nausea.

The web also has added specific challenges because traditional methods of communications, such as Web Sockets, are not fast enough. Browsers can introduce other incremental drops in performance. High amounts of data to visualize scenes can push the limits of WebGL rendering as well.

Finally, web environments introduce security concerns, because of the need for such direct communication between devices. The new WebXR Device API addresses these challenges, along with providing a method of standardised interfacing with XR devices. This will become increasingly valuable as the range and volume of hardware evolves in coming years.

The WebXR Device API provides the following key capabilities:

- Identify and locate (detect) VR or AR output devices
- Translate the movement vectors (direction and magnitude) from input devices and sensors into useable vectors for application
- Render 3D scenes to devices at an appropriate frame rate.

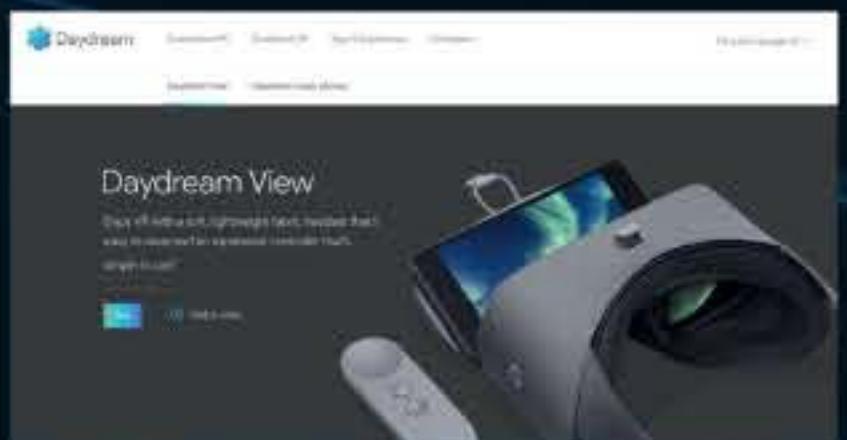
WHAT ABOUT BROWSER SUPPORT?

Browser support is always important when it comes to emerging technologies. Limited support can slow adoption or stop new API development all together if major browsers don't integrate features into their ecosystem.

Chrome is currently leading the charge with support for WebXR added to the latest browser versions. Firefox is also making it available as an experimental technology, which means that attributes and methods are not yet stable and subject to change. As the technology continues to evolve, support will as well. You can keep up with the latest support for the WebXR Device API here: <https://caniuse.com/#search=WebXR>

WEBXR DEVICES

DEVICES SUPPORTING XR CONTINUE TO GROW YEAR OVER YEAR



Google Daydream

<https://vr.google.com/daydream>

Google's DayDream VR line includes Daydream ready phones that work with their headset and controllers. They are also developing a standalone VR device that will use their WorldSense technology, eliminating external motion sensors.



HTC Vive

www.vive.com/us

HTC Vive is an immersive VR product line, integrating sound and touch with visuals. They also feature room scale tracking technology that allows users to interact with rooms and objects around them.



Magic Leap One

www.magicleap.com/magic-leap-one

Magic Leap One uses a unique technology that emulates the physiology of how human sight works. Using light fields and virtual light rays, Leap creates a unique XR experience, on their own custom tech and OS.



Microsoft HoloLens

www.microsoft.com/en-us/hololens

Microsoft HoloLens offers a mixed reality (augmented and virtual) experience on untethered (free to roam) devices. It combines new hardware advancements, artificial intelligence and XR solutions. They also offer integration with their cloud and AI services.



Facebook's Oculus Rift

www.oculus.com

Oculus comes in three main forms: Quest, Go and Rift. Quest offers an all-in-one gaming solution, with no PC required. The Rift is a more powerful gaming VR device that works with your PC. The Go is another all-in-one device intended for entertainment.

The Go is another all-in-one device intended for entertainment

8 TO FOLLOW

Mozilla MR

@mozillareality
The Mozilla Mixed Reality team. An industry leader in leading edge tech dev, working on high-performance XR for the web.

Three.js

@threejs_org
Official account for the world's leading 3D engine for the web. Exploring WebGL optimized experiences beyond the screen into VR and XR.

A-Frame

@aframevr
The official account for A-Frame, which is a web framework for building XR experiences, using Three.js and latest APIs.

Exokit

@exokitXR
The Exokit Browser is a "WebXR meta-browser" that runs inside your web browser. Researching WebXR and solutions for developers.

Microsoft HoloLens

@HoloLens
The official account of Microsoft's HoloLens device. Follow along to keep up to date with latest news and offerings on their gear.

Nell Waliczek

@NellWaliczek
Nell is a principal engineer at Amazon working on the Sumerian project, a WebXR initiative enabling development across a wide range of devices.

Oculus

@oculus
Oculus' official account for their line of XR devices. Learn about where device development is headed and what one of the top makers is offering.

8th Wall

@the8thwall
8th Wall is developing solutions in Web AR that run across multiple platforms and devices. Should be exciting to follow their progress.

WHERE CAN WEBXR EXCEL?

INDUSTRY REPORT

SOME KEY TAKEAWAYS FROM THE XR INDUSTRY REPORT BY ALEX HADWICK

According to the research, 93% of enterprises using VR have seen a "positive effect on their business". More important for those looking ahead, the report states 98.6% of users were "considering investing more into XR in the next few years." The report also outlines the most popular use cases for XR within companies using it. These include:

- Product design and prototyping (over 96%)
- Workforce/ project collaboration (93%)
- Educational learning (90%)
- Training/worker guidance (90%)
- Sales & marketing/external comms (84%)
- Manufacturing (81%)

The report also reveals sectors that are most likely to incorporate XR technologies. These include:

- Education (56%)
- Architecture/Engineering (44 %)
- Healthcare (42%)
- Manufacturing (40%)
- Automotive (36%)

Interestingly, although gaming is where XR was thought to have its greatest impact, research indicates this already in decline. The report states, "Gaming is now viewed with far less interest by companies", down to 33% from 50% in 2018.



IMMERSIVE ARTWORK

You need to look no further than the cutting-edge experiences being created by artists like Julius Horsthuis, fractal artist and filmmaker, to understand the possibilities VR/XR are beginning to create. Using VR and projected experiences, Julius is pushing the way we think about art and visual experiences forward, and into the realm of altered reality. If XR is to extend beyond the conventions of the screen or headset displays, it may be artists like this that find new ways to push the boundaries of how we experience art. Be sure to check out his work here <http://www.julius-horsthuis.com>

360 VIDEO / IMMERSIVE ENTERTAINMENT

Entertainment and marketing both have been exploring the potential of XR for years. VR films and 360 photography have both been growing in popularity. Films are being created that break the convention of a single camera view. This allows audiences to look around at the environment around them while watching a story unfold. It opens up a totally new world of possibilities for content creators. Putting the audience in the midst of the action can unlock completely new ways of telling stories, and even new stories to tell. It is potentially one of the most exciting uses of this new technology.

MODEL VISUALIZATIONS

Arguably one of the most practical use cases for XR is viewing 3D models. This covers a broad range of applications. Imagine, stepping through an architectural model of a building before it's built, to explore or troubleshoot its design. You can explore new products before buying them, or as a designer improve them in real-time, before production.

Models could be products, buildings, new vehicles, but also environments. Exploration of locations that are unsuited to humans and derived from data collected could add great insight and understanding, unmatched by any other means. Seeing something, seeing data visualized can provide a completely new perspective to enhance learning.

WEBXR SHOWCASE

CHECK OUT THESE EXCITING AND INSPIRING EXAMPLES



Amazon Sumerian Express

<https://bit.ly/2riVzED>
The Sumerian Express is an online WebXR experience rendered using WebGL and runs in Chrome browsers and VR devices. Amazon Sumerian is at the leading edge of WebXR tools and development, and well worth checking out.

Rome's Invisible City

<https://bit.ly/2mp7yBQ>
Rome's Invisible City VR Experience, created by the BBC and ScanLAB Projects, is a WebXR experience intended to educate visitors on various locations as they explore 360 images in a virtual world.



Learnbrite

www2.learnbrite.com

LearnBrite is a virtual classroom or customizable meeting space that you can interact with on mobile, desktop and devices. It supports WebXR and can host 1000s of people at the same time!

“ Sumerian Express is an online WebXR experience ”



Rome's Invisible City VR

Alexander Armstrong takes you on a 3D Virtual Reality tour of Rome's magnificent Pantheon and the underground quarry the Romans mined to build it.

• 1,142 ratings | 1,453 shares

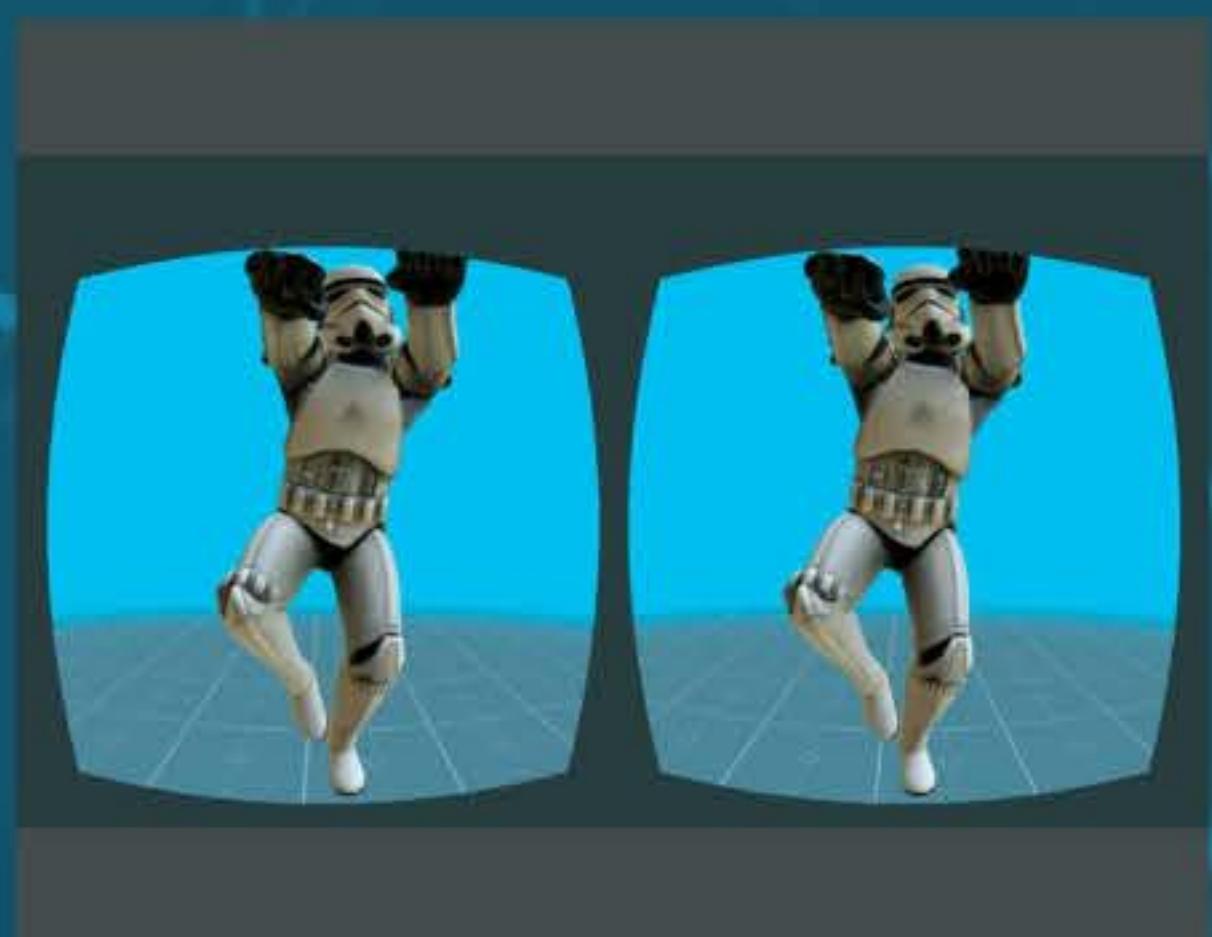


Immersive Web - 360 Photos

<https://bit.ly/2o6D7AN>
A real-time 360 photo demo hosted by Immersive Web. This is just one of numerous examples found here, <https://bit.ly/2snZukm>. This is a great collection of examples of how WebXR can be used.

Sketchfab

<https://sketchfab.com>
Sketchfab is an invaluable resource for 3D models and assets. Not only can you download excellent models to use, Sketchfab is actually an excellent example of WebXR, allowing you to navigate and explore models in VR!



TOP APIs & TOOLS

WebXR Device API

W3C Working Draft, 21 May 2019

This version:
<https://www.w3.org/TR/2019/WD-webxr-20190521/>

Latest published version:
<https://www.w3.org/TR/webxr/>

Editor's Draft:
<https://immersive-web.github.io/webxr/>

Previous Versions:
<https://www.w3.org/TR/2019/WD-webxr-20190205/>

Issue Tracking:
[GitHub](#)
[Inline In Spec](#)



W3C WORKING DRAFT - WEBXR DEVICE API

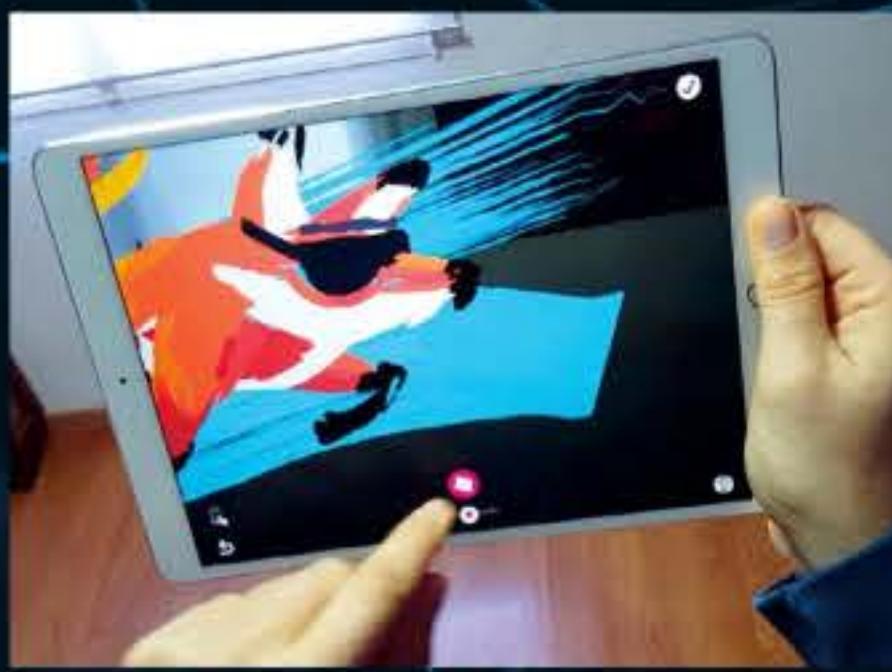
<https://www.w3.org/TR/webxr>

The official working draft online paper for the WebXR Device API. The world wide web consortium is the most official resource for the status of this emerging technology.

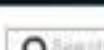
MOZILLA WEBXR EXPERIMENT: A-PAINTER XR

<https://blog.mozvr.com/responsive-webxr-a-painter-xr>

A-Painter XR – An experiment in extending a WebVR paint program called “A-Painter” to support WebXR, and giving it AR capabilities. A great read with loads of links to learn more.



MDN web docs



Technologies ▾ References & Guides ▾ Feedback ▾

MOZILLA WEBXR DEVICE API

https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API

Mozilla WebXR Device API article outlines the current status of the API and details on implementation. It outlines goals of WebXR, diagrams on how it works and application lifecycles.

WebXR Device API

Jump to: WebXR Device API Concepts and usage

WebXR Interfaces

Specifications

Browser compatibility

See also

View technology for this page

Reports

Web APIs

WebXR Device API

Related Topics

RESOURCES

A-Frame Web Framework

<https://aframe.io>

A-frame is a popular framework for building virtual reality experiences on the web. Built on Three.js for 3D rendering, is supported on a wide range of devices and platforms.

Immersive Web - WebXR

<https://immersive-web.github.io/webxr-samples>

WebXR samples that have been designed to demonstrate the new WebXR Device API. The goal is to give developers a “set of simple, fun, readable apps that demonstrate various aspects of using the API”.

ARCore

<https://developers.google.com/ar>

ARCore is an SDK created by Google to implement of augmented reality applications on Android devices. While not exclusively WebXR focused, it does offer great XR models to follow as you develop.

AR.js

<https://github.com/jeromeetienne/AR.js>

Ar.js is a target/marker Augmented Reality library for the web. It is open source and very fast. It utilizes Three.js and the jsartoolkit5. It can also be used with A-frame.

Three.js

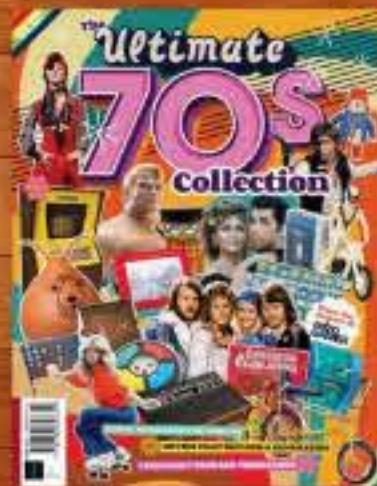
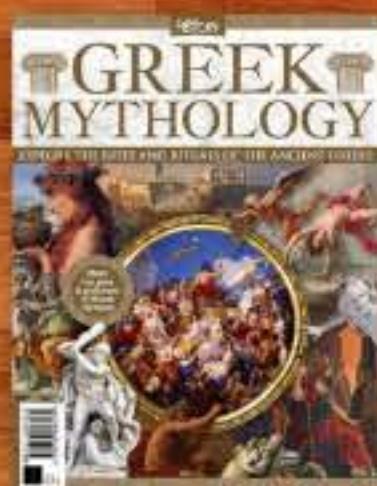
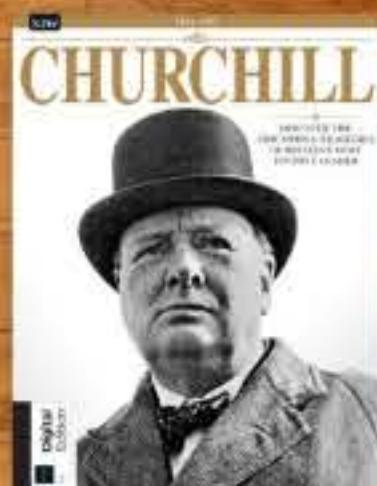
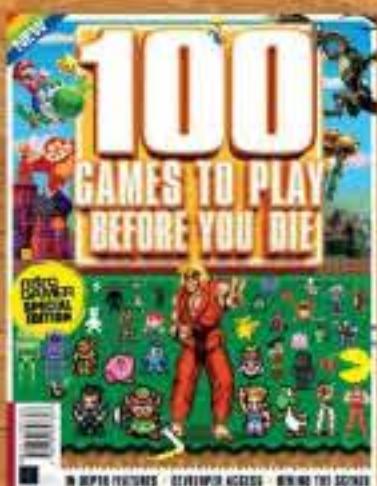
<https://threejs.org>

Three.js is a cross-browser JavaScript library/API used to create and display animated 3D computer graphics in a web browser. Three.js uses WebGL.

WebXR Explainer

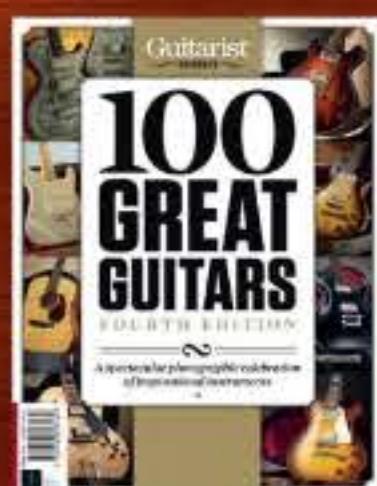
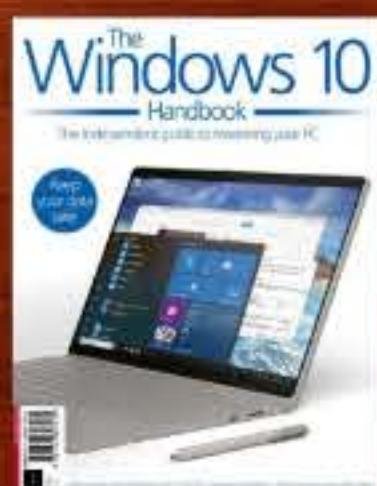
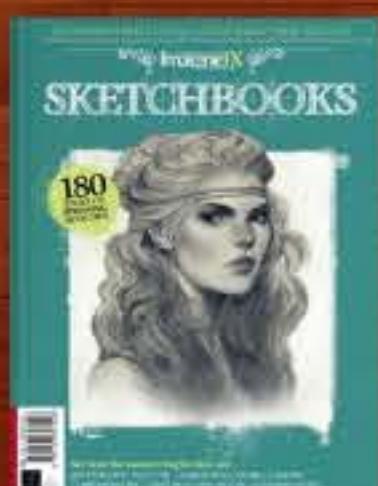
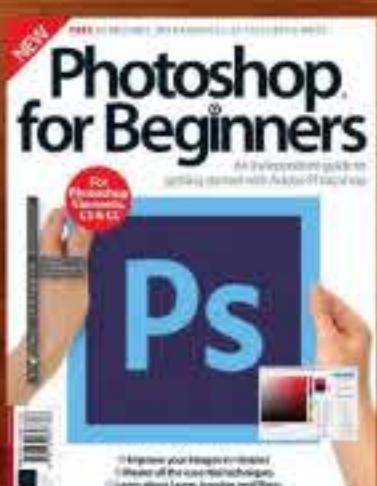
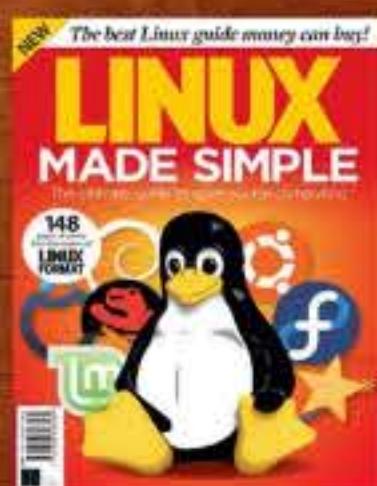
<https://github.com/immersive-web/webxr/blob/master/explainer.md>

A great overview on everything WebXR. From explaining the concepts to status on where progress on the API is. Also includes a deep dive into the API with code examples.



Discover another of our great bookazines

From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes



Get great savings when you buy direct from us



1000s of great titles, many not available anywhere else



World-wide delivery and super-safe ordering

F U T U R E

www.myfavouritemagazines.co.uk
Magazines, back issues & bookazines.

Get your listing in our directory
To advertise here call the ads team on 01225 442244

HOSTING LISTINGS



Featured host: Netcetera

netcetera.co.uk
03330 439780

About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide.

As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services designed to more effectively manage IT

infrastructures. A state-of-the-art data centre environment enables Netcetera to offer your business enterprise-level colocation and hosted solutions.

Providing an unmatched value for your budget is the driving force behind our customer and managed infrastructure services. From single server to fully customised data centre suites, we focus on the IT solutions you need.

What we offer

- **Managed hosting** - A full range of solutions for a cost-effective, reliable, secure host.
- **Cloud hosting** - Linux, Windows, Hybrid and Private Cloud Solutions with support and scalability features.

- **Data centre colocation** - Single server through to full racks with FREE setup and a generous bandwidth.
- **Dedicated servers** - From QuadCore up to Smart Servers with quick setup and fully customisable.

5 tips from the pros

1. Reliability, trust & support

Reliability is a major factor when it comes to choosing a hosting partner. Netcetera guarantees 100 per cent uptime, multiple internet routes with the ability to handle DDOS attacks, ensuring your site doesn't go down when you need it.

2. Secure and dependable

Netcetera prides itself on offering its clients a secure environment. It is accredited with ISO 27001 for security along with the options of configurable secure rackspace available in various configurations.

3. 24/7 technical support

Netcetera has a committed team of

knowledgeable staff available 24/7 to provide you with assistance when you need it most. Our people make sure you are happy and your problems are resolved as quickly as possible.

4. Value for money

We do not claim to be the cheapest service available, but we do claim to offer excellent value for money. We also provide a price match on a like-for-like basis, as well as a price guarantee for your length of service.

5. Eco-friendly

Netcetera's environmental commitment is backed by use of eco-cooling and hydroelectric power. This makes Netcetera one of the greenest data centres in Europe.



Testimonials

Roy T

"I have always had great service from Netcetera. Their technical support is second to none. My issues have always been resolved very quickly."

Suzie B

"We have several servers from Netcetera and their network connectivity is top-notch, with great uptime and speed is never an issue. Tech support is knowledgeable and quick in replying. We would highly recommend Netcetera."

Steve B

"We put several racks into Netcetera, basically a complete corporate backend. They could not have been more professional, helpful, responsive or friendly. All the team were an absolute pleasure to deal with, and nothing was too much trouble, so they matched our requirements 100 per cent."

Supreme hosting



cwcs.co.uk
08001777000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and are passionate about what they do.

- Colocation hosting
- VPS
- 100 per cent network uptime

UK-based hosting



cyberhostpro.com
08455279345

Cyber Host Pro are committed to providing the best cloud server hosting in the UK; they are obsessed with automation. If you're looking for a hosting provider who will provide you with the quality you need to help your business grow, then look no further than Cyber Host Pro.

- Cloud VPS servers
- Reseller hosting
- Dedicated servers

Cluster web hosting



fasthosts.co.uk
08081686777

UK-based and operating 24/7 from dedicated UK data centres. Fasthosts keep over 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.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
03333365509

1&1 Internet is a leading hosting provider that enables businesses, developers and IT pros to succeed online. Established in 1988, 1&1 now

operates across ten countries. With a comprehensive range of high-performance and affordable products, 1&1 offers everything from simple domain registration to award-winning website building tools, eCommerce packages and powerful cloud servers.

- Easy domain registration
- Professional eShops
- High-performance servers

SSD web hosting



bargainhost.co.uk
08432892681

Since 2001, Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database, which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Agency hosting specialist



nimbushosting.co.uk
02031266781

Nimbus Hosting have partnered with agencies to develop our revolutionary platform STORM. With a team dedicated to outstanding support, our 5-star Google reviews truly speak for themselves. Join the thousands of agencies and freelancers who are benefitting from a control panel that speeds up your website development as well as your client's websites. Supercharge your digital projects today with STORM.

- 30-second WordPress install
- Deploy directly from GitHub
- Easy team management

Flexible cloud servers

elastic hosts

elastic hosts.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



Featured: **Northcoders**

northcoders.com
Twitter: @northcoders
Facebook: Northcoders

About us

Northcoders is the coding bootcamp for the north, based in the heart of Manchester and built upon northern values of grit, determination and community spirit. No matter what your background, you can fast-track your career and become a web or software developer in 12 weeks at their

full-time bootcamp, or fit their course around your life with their 24-week part-time bootcamp. Their internal career support team will help find you work as a developer, setting up interviews with your choices of Northcoders Hiring Partners across the north of England.

What we offer

- **Full-time:**
Fast-track your career in just 12 weeks

- **Part-time:**
Fit our curriculum around your life in 24 weeks

5 tips from the pros

1. Get started with coding

The best way to know if coding is for you is to just try it! We recommend the free, online JavaScript track of Codecademy to get you started with the basics.

for you, set aside a few evenings each week to really start making progress! If coding is for you, this should be fun.

4. Be prepared

We'll be with you every step of the way when you apply. Make sure you go through all the materials we recommend and ask for help if you're stuck.

5. Get social

With Northcoders, you're not just on a course, you're part of a community that will stay with you long after you graduate. Make the most of it!

2. Do your research

Make sure you read plenty of student reviews to make sure you're applying somewhere reputable. Read their blog and have a look at their social channels.

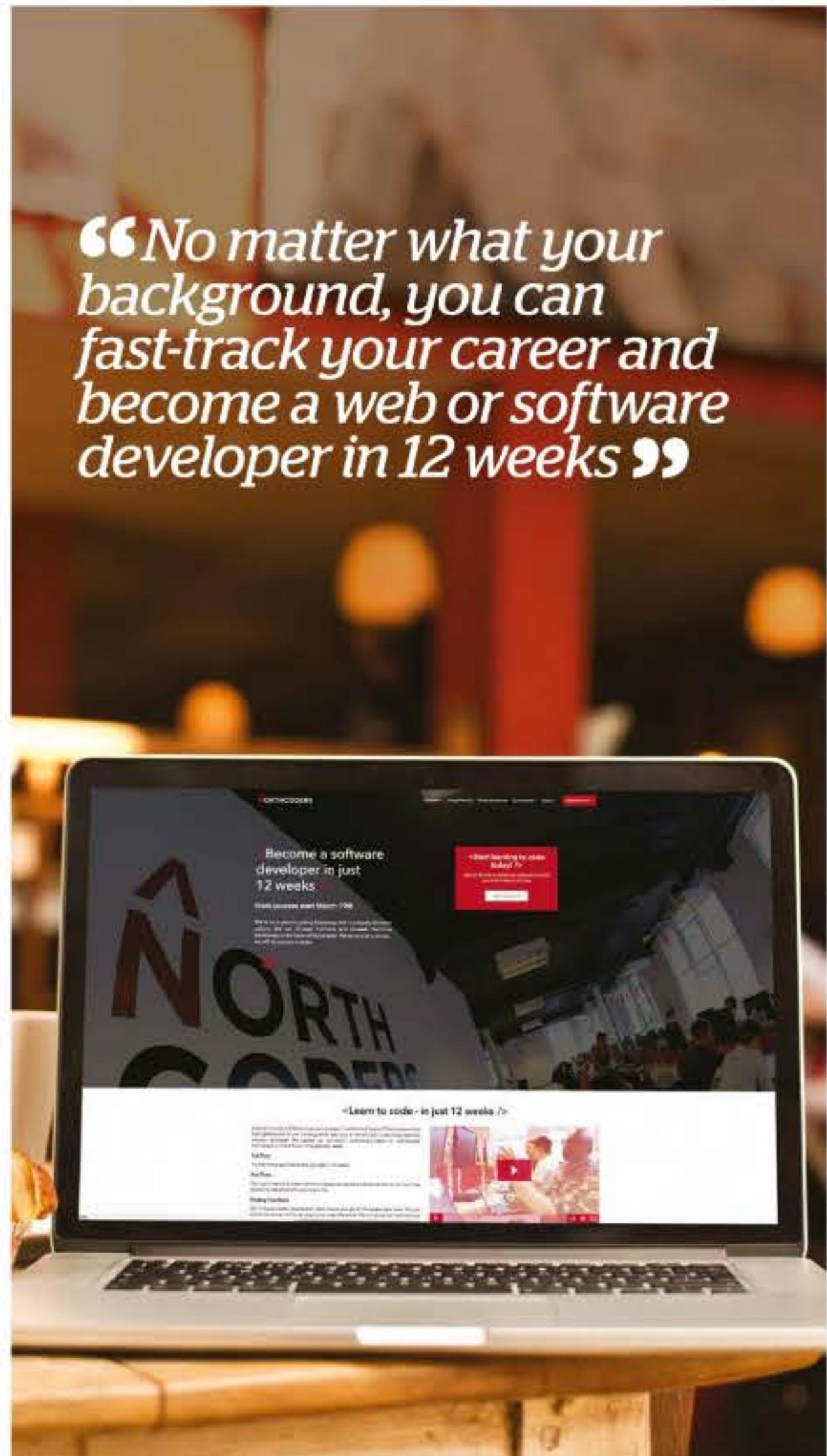
3. Throw yourself in

Once you've decided it's right

“ Becoming part of this vibrant, caring community was something I hadn't expected before the course, but now I couldn't be without it. To be a Northcoder is to be enlightened, inspired and supported

Joanne Imlay

Primary school teacher to software developer at Careicon



“ Northcoders delivered their part of the bargain in spades. They provided tremendous assistance in turning me into the full product - a well-rounded, capable future tech employee - and they have the contacts to deliver the opportunities for such people.

Joe Mulvey

Maths teacher to software developer at Auto Trader

udemy

UDEMY

udemy.com

Twitter: [@udemy](https://twitter.com/udemy)
Facebook: [udemy](https://facebook.com/udemy)

The inspiration for Udemy began in a small village in Turkey, where founder Eren Bali grew up frustrated by the limitations of being taught in a one-room school house. Realising the potential of learning on the internet, he set out to make quality education more accessible. Udemy is now a global marketplace for learning and teaching online. Students can master new skills by choosing from an extensive library of over 40,000 courses, including HTML, CSS, UX, JavaScript and web development.

40,000+ courses: There is a course for every designer and dev.

Self-paced learning: Learn how to code at your own pace.



THE IRON YARD

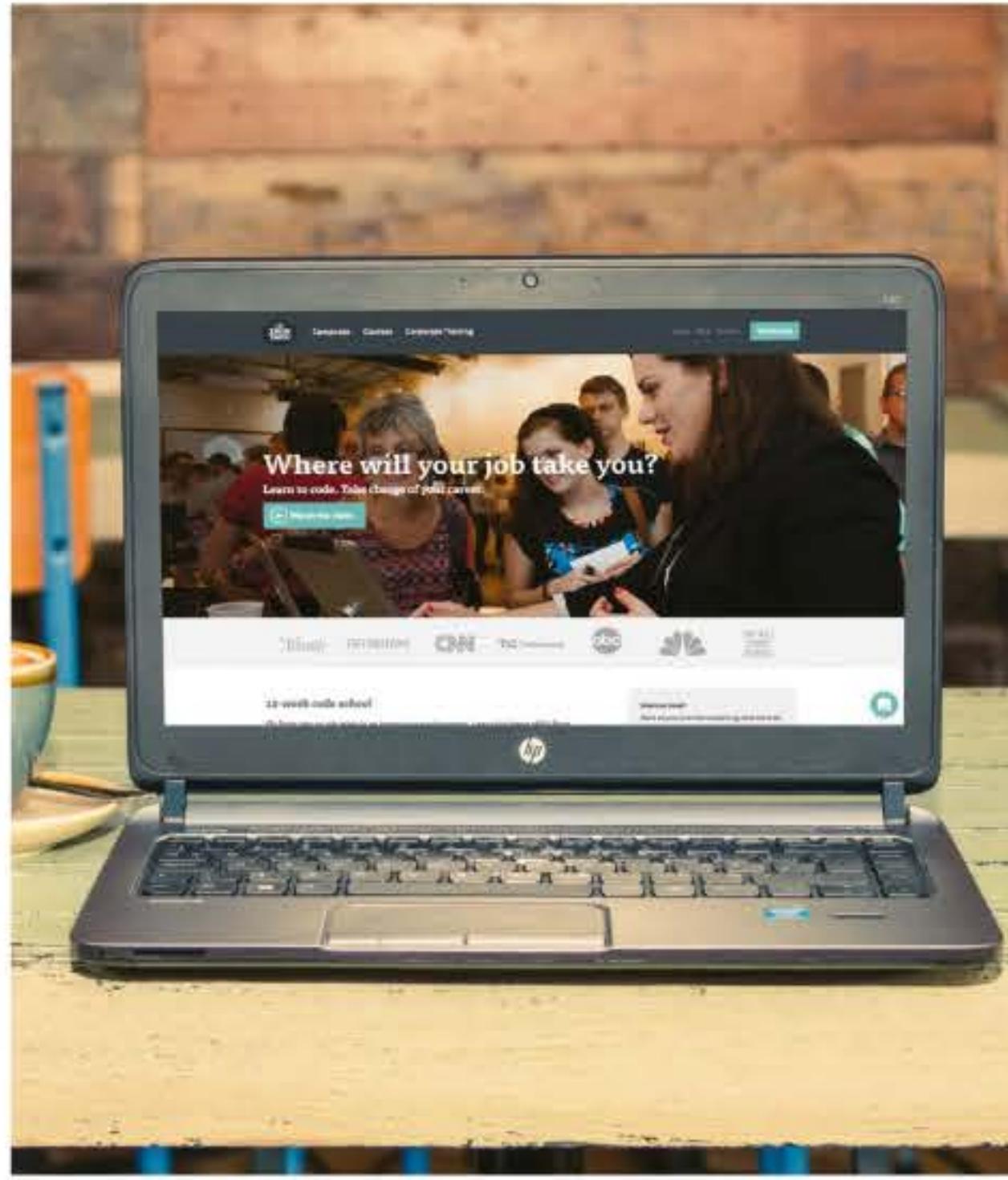
theironyard.com

Twitter: [@TheIronYard](https://twitter.com/TheIronYard)
Facebook: [TheIronYard](https://facebook.com/TheIronYard)

The Iron Yard is one of the world's largest and fastest growing in-person **code schools**. It offers full-time and part-time programs in 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

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



futurelearn.com

feedback@futurelearn.com

Choose from hundreds of free online courses, from Language & Culture to Business & Management, Science & Technology to Health & Psychology. Learn from the experts. Meet educators from top universities who'll share their experience through videos, articles, quizzes and discussions.

- Learn from experts
- Free courses
- All-device access

GYMNASIUM



thegymnasium.com

help@thegymnasium.com

Gymnasium offers free online courses, designed to teach creative professionals in-demand skills.

Courses are all self-paced and taught by experienced practitioners with a passion for sharing practical lessons from the design trenches.

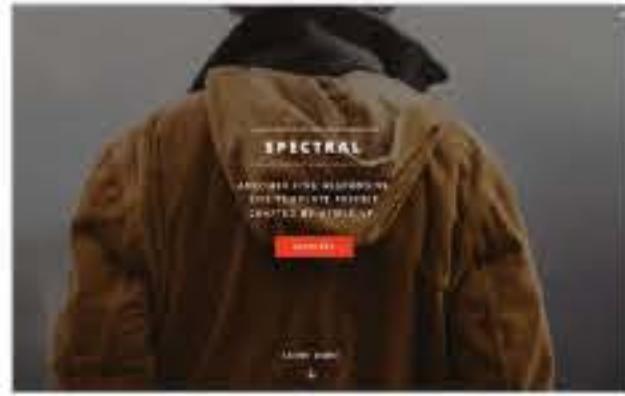
- Gain real-world skills
- Get expert instruction
- Career opportunities

Free with your magazine

Instant access to these creative resources...

Essential assets and resources

Get textures, fonts,
backgrounds and more



Exclusive video tutorials

Learn to code/create
with HTML, CSS & JS



Tutorial project files

All the assets you'll need
to follow our tutorials



Plus all of this is yours too...

- All-new tutorial files to help you master this issue's HTML, CSS and JavaScript techniques
- 50 Editorial Lightroom presets and LUTs from Sparklestock (worth \$19)
- 50 Retro films Lightroom presets from Sparklestock (worth \$19)

→ Log in to www.filesilo.co.uk/webdesigner

Register to get **instant access**
to this pack of must-have
creative resources, how-to
videos and tutorial assets

Free
for digital
readers too!

Read on your
tablet, download on
your computer





The home of great downloads – exclusive to your favourite magazines from Future!

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, in one place

Get started

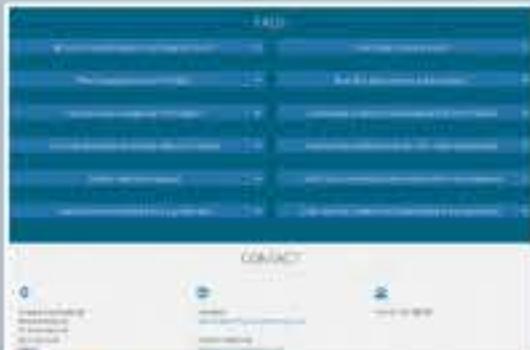
Everything you need to know about accessing your FileSilo account



- 01** Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.



- 02** You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.



- 03** If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at the address below:

filesilohelp@futurenet.com

An incredible gift for subscribers



Subscribe today & unlock the free gifts from more than 70 issues

Access our entire library of resources with a money-saving subscription to the magazine – that's more than 900 free resources

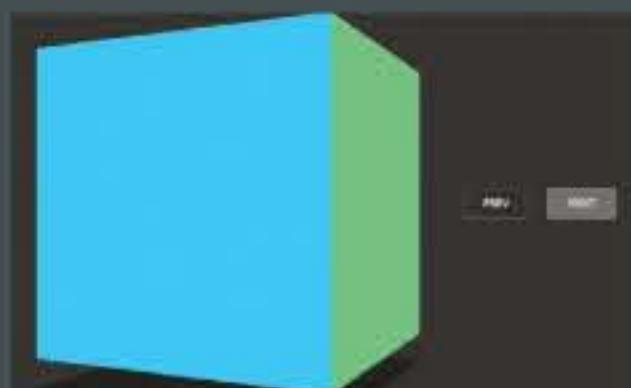
Over 70 hours of video guides

Let the experts teach you to create and code



More than 400 tutorials

Get the code you need to get creative



Over 2500 creative assets

Templates, fonts, textures and backgrounds



Head to page 28 to subscribe now



Already a print subscriber?

Here's how to unlock FileSilo today...

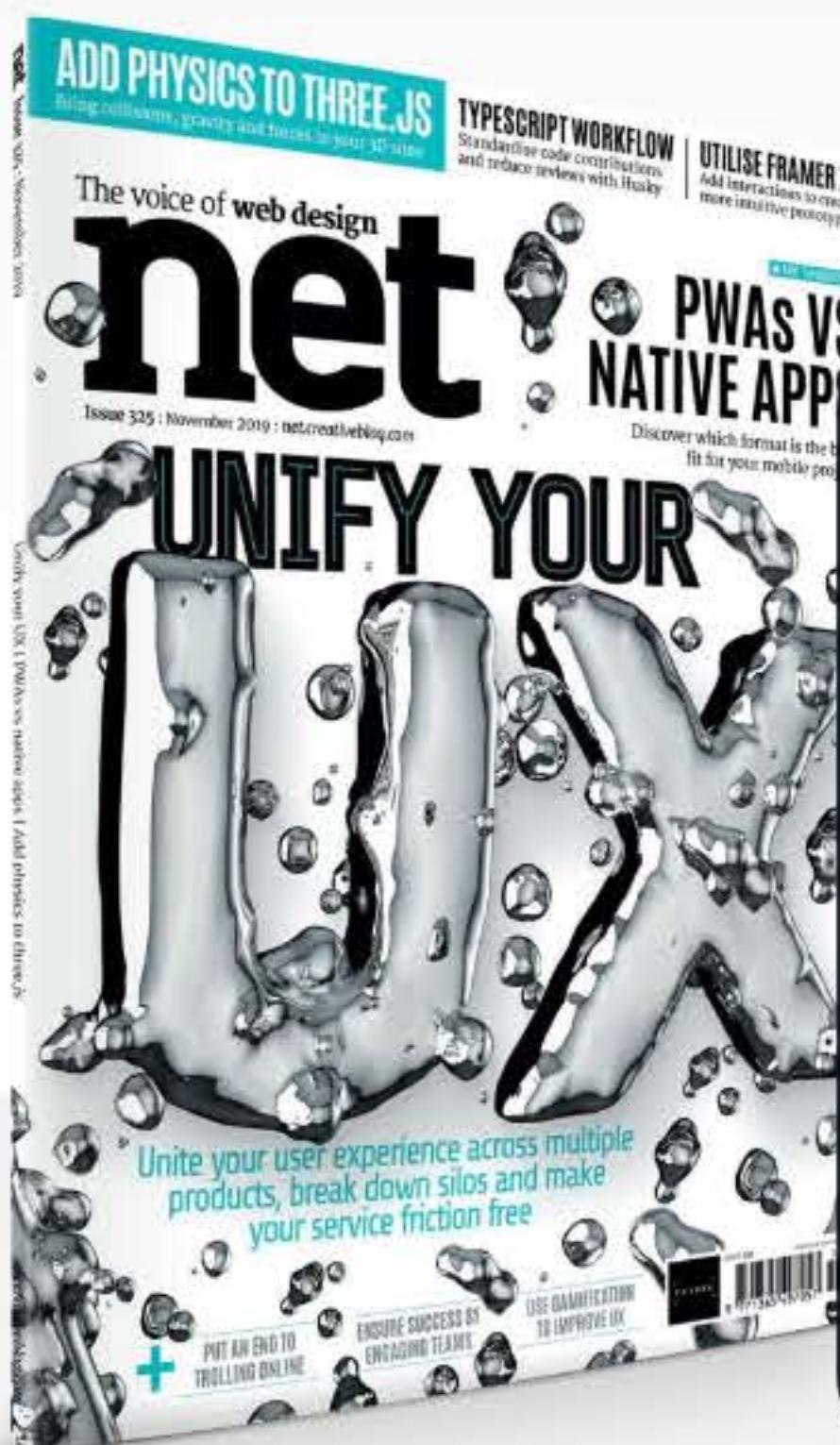
Unlock the entire Web Designer FileSilo library with your unique Web ID – the ten-digit alphanumeric code printed above your address details on the mailing label of your subscription copies – also found on any renewal letters.

More than 900 reasons to subscribe

+
More added every issue

net

PRINT & DIGITAL EDITION



TRY OUR
DIGITAL
MAG FOR
FREE!

Discover the no.1 choice for web designers and developers. Each issue is packed with the latest trends, technologies and techniques, plus exclusive video tutorials. Don't miss it!

PRINT: myfavouritemagazines.co.uk/NETMAG16
iPAD: netm.ag/net-ipad-uk OR netm.ag/net-ipad-us



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.



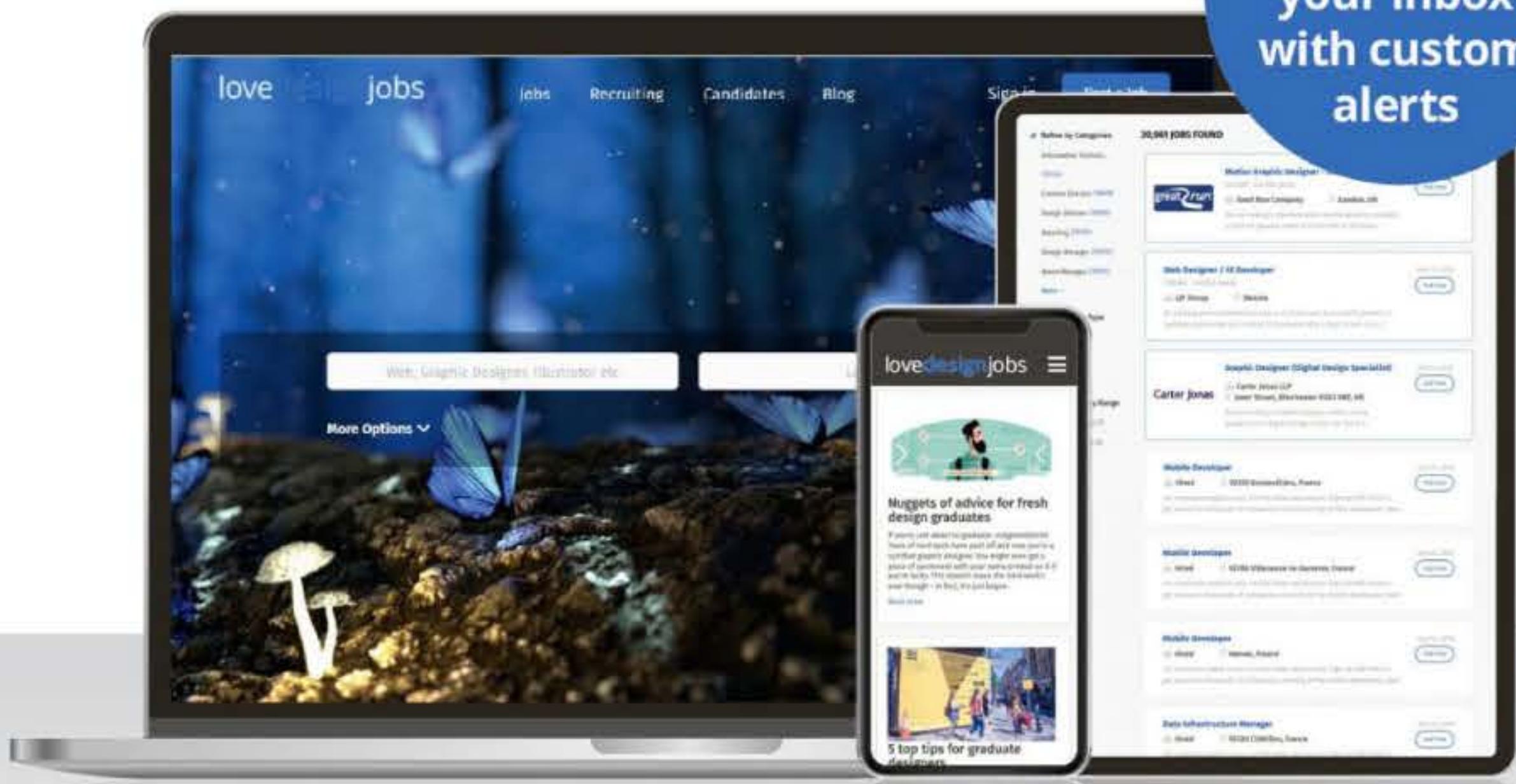
PRESS PAUSE
ENJOY A MAGAZINE MOMENT

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

Looking for work?

Thousands of active global job vacancies

Get latest
jobs direct to
your inbox
with custom
alerts



lovedesignjobs

powered by CREATIVE BLOQ

Web Developers

Web Design

CG Artists

Digital Artists

www.lovedesignjobs.com

Looking for Staff? Call James on his bat-phone TODAY on 07973 290 109 or email him on james@lovedesignjobs.com