

# LTG Dev Tools Workshop

Viv Richards

[www.vivrichards.co.uk](http://www.vivrichards.co.uk)

@11v1r

Alan Richardson

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

@eviltester

# Introductions and Overview

# Session overview

- Chrome DevTools
  - What about other browsers?
- A look at tools hidden within the Chrome web browser
- Hopefully share a few tips and tricks which may be new to some of you
- Looking at DevTools from both a coder and testers point of view and how it can be of use
- Slides: <https://utm.io/uhmC>

# Session Guidance

- Ask questions
- Attention during slides and demo sections - so as to not distract others
- Feel free to test whatever you want
  - the Experiments and Exercises in the slides are suggestions,
  - The applications listed are suggestions
- Follow your own path, you might want to...
  - Jump ahead through the tools and slides,
  - Stay and explore a tool section,
  - follow the slides and use suggested exercises.
- Take breaks whenever you need to
  - Formal break about 11:00 - you might need to remind us

# What do you want out of this?

To help guide our adhoc speaking sections...

# Overview of the main suggested apps under test

- Sweetshop
  - <https://sweetshop.netlify.com/>
- Playground
  - <http://playground.epizy.com/>
- And to experiment with...
  - The Pulper (CRUD app)
    - <https://thepulper.herokuapp.com/>
  - Todo MVC
    - <http://todomvc.com>
  - Buggy Games
    - <https://www.compendiumdev.co.uk/games/buggygames>
  - Test Pages
    - <https://testpages.eviltester.com>

# Dev Tools?

# DevTools

*DevTools is a set of web developer tools built directly into the web browser.*

*DevTools can help you edit pages on-the-fly and diagnose problems quickly, which ultimately can help you build better websites, faster.*



# Open Sesame

- Right Click Inspect
- Hamburger \ More Tools \ Developer Tools
- Chrome Keyboard Shortcuts
  - <https://developers.google.com/web/tools/chrome-devtools/shortcuts>

# Device Toolbar, Docking, Run Command

# Device Toolbar

Device Mode allows you to approximate how your page looks and performs on a mobile device. Simulate mobile viewport.

*With Device Mode you don't actually run your code on a mobile device. You simulate the mobile user experience from your laptop or desktop.*

# Device ToolBar Functionality

- Responsive vs Device List - headers
- Rotate Device
- Throttling
- Explore Hamburger
  - Show/Hide Media Queries
  - Rulers,
  - Screenshots,
  - etc.

# Docking Dev Tools

- Dock to Left, Right, Separate Window
- From Hamburger “Customize...” menu
- Ctrl+Shift+P - dock

# Run Command

Inspect -> Ctrl+Shift+P

Inspect -> cmd+Shift+P

- Command Completion - partial text
- e.g.
  - Screenshot
  - Reload
  - Media queries
  - Show
  - JavaScript

# Experiments & Exercises

- <https://sweetshop.netlify.com/>
  - Does the site work well on mobile?
  - Is it responsive?
  - Explore the site with different devices: home page, about page, sweets page
  - Do the media queries work well for the site?
  - Does it work well with a throttled connection?
  - Explore the site with rulers on, does that help?
  - Did you take a screenshot?
  - Try docking the console tools in different ways, what works best for your screen?
  - Explore the command menu: switch javascript on and off, what else can you do?
- Try other sites and apps that you use
- more on next page...

# Experiments & Exercises

- Try twitter in mobile device mode (refresh page) - notice anything different?
- Spectator magazine site has a limit on articles you can read in a month e.g. Spectator.co.uk . Does this check depend on JavaScript? What happens if you reach that limit and use Run Command to switch JavaScript off?
- Testers Playground (<http://playground.epizy.com/devtools.php>)
  - Open Sesame
  - Be mobile



# Elements Panel, Console

# Elements Panel

- What is it?
- What do we see?
  - Dom, Applied CSS
- What if we don't understand HTML?
  - Look it up on Mozilla Developer Network
  - <https://developer.mozilla.org/en-US/>
- What do we use it for?
  - Learning, Manipulation, Automating
- How is this different from the “View Source” option?
  - Rendered vs Sent, Fixed vs Raw

# How Do We Use It?

- DOM Inspection, Manipulation (Create, Amend, Delete, Move)
  - How? (Right click, ..., double click amend)
- Ctrl+F
  - Find by XPath, CSS Selector, text
- CSS state triggers
  - Style view, Right click

# How Do We Use It?

- Break On DOM Manipulation
  - Subtree modifications - can be useful when attempting to create automated execution sync points. Triggers JavaScript debug breakpoint.
- Explore Context Menu:
  - Copy as Xpath, CSS ; Edit as HTML
- Element Screenshots - Ctrl+Shift+P \ screenshot
- Remote Devices

# Console

*The Console has 2 easy uses: viewing logged messages and running JavaScript.*

# Some Simple JavaScript

- 2+2
- Make Page Editable
  - `document.designMode="on"`
  - `document.designMode="off"`
- Show JavaScript variables
  - `VariableName`
  - `console.table(variable)`
    - Also JSON objects

# Useful Basic JavaScript

- Base 64 decoding and encoding
  - `atob("aGVsbG8=")`
  - `btoa("test")`
- JSON conversion - pretty print
  - `JSON.stringify( anObject , null, 2)`
  - `JSON.stringify( JSON.parse('{ "message" : { "title": "hello" } }') , null, 2)`

# Just For Fun?

- `speechSynthesis.speak(new SpeechSynthesisUtterance(2+2))`



# Experiments & Exercises

- <https://sweetshop.netlify.com/>
  - Inspect that pages and look at the HTML. Do you understand it? Look up the elements on developer.mozilla.org
  - Drag and drop the elements and change the look of the page.
  - Edit the names of some of the Sweets
  - Try amending some of the CSS, switching parts on and off
  - Does the page sweets.html have any image issues? Can you fix it in the DOM and show the image so that your defect report mentions the correct image to use?
  - If your city is not listed in the checkout page, can you add it?
- more on next page...

# Experiments & Exercises

- <https://sweetshop.netlify.com/>
  - Can you add any tasty sweets to your basket for £0.00? Or even better -ve number (make them pay you to shop on their site)
  - Next release we will change the “Add to Cart” button to read “I want to buy this. Add it to my cart!” is this going to cause any issues? Can you investigate this requirement early?
  - Can you change the maxlength for the password on the login page?
- Try other sites and apps that you use
- more on next page...

# Experiments & Exercises & Challenges

- Experiment with some of the JavaScript on the slides
- Visit some web apps and sites you are familiar with,
  - compare the DOM in Elements view to the View Source e.g. twitter.com
  - Explore with the dev tools
  - any JavaScript errors or output in the console? E.g. vivrichards.co.uk, bbc.co.uk
- bbc.co.uk and instagram.com embed a lot of JSON in their HTML
  - can you format any of it using the code listed on the slides?
- Have you played/tested Buggy Games?
  - [https://www.compendiumdev.co.uk/games/buggygames/driving\\_game/console\\_driver.html](https://www.compendiumdev.co.uk/games/buggygames/driving_game/console_driver.html)
  - [https://www.compendiumdev.co.uk/games/buggygames/the\\_coloured\\_square\\_game/colouredsquare.html](https://www.compendiumdev.co.uk/games/buggygames/the_coloured_square_game/colouredsquare.html)
  - [https://www.compendiumdev.co.uk/games/buggygames/the\\_coloured\\_square\\_game/colouredsquarechanger.html](https://www.compendiumdev.co.uk/games/buggygames/the_coloured_square_game/colouredsquarechanger.html)
  - [https://www.compendiumdev.co.uk/games/buggygames/the\\_one\\_to\\_nine\\_calculator/calc1to9.html](https://www.compendiumdev.co.uk/games/buggygames/the_one_to_nine_calculator/calc1to9.html)

# Challenges

- Mozilla.org - does Elements view show anything interesting in the head?
- Amazon.co.uk - use the Elements view to see what is in the source at the bottom of the page
- Can looking at the HTTP headers of a request to Paypal.com get you a job?
- Twitter: Can you use a combination of Element view and Run Command to take a screenshot image of a tweet?
- Instagram:
  - If you look at an Instagram feed e.g. <https://www.instagram.com/ministryoftesting/> How could you test the footer, since the page keeps auto loading new content?
  - I can't right click on an instagram image in the browser and save the picture. How could you save the picture of an instagram post from the feed?

# Challenges

- Testers Playground (<http://playground.epizy.com/devtools.php>)
  - Inspector gadget
  - Style is everything
  - 404 (Not Found)

# Sources & Audits

# Sources Panel

- View files
- Edit CSS and JavaScript
- Create and save Snippets of JavaScript, which you can run on any page.  
(Snippets are similar to bookmarklets)
  - Ctrl+P can run these if you delete the “>” and type “!”
- Overrides, changes you make in DevTools get saved to the code on your file system.

# Audits Panel

*The Audits panel is now powered by Lighthouse. Lighthouse provides a comprehensive set of tests for measuring the quality of your web pages.*

*Demo: pick a site and audit it*



# Experiments & Exercises

- Choose some websites you are familiar with and audit them for different aspects.
  - Do you understand the error messages?
  - Look them up?
  - Are any serious?
- Testers Playground (<http://playground.epizy.com/devtools.php>)
  - Sourcerer

# Network & Application

# Network Panel - Basics

*In general, use the Network panel when you need to make sure that resources are being downloaded or uploaded as expected.*

- Record/ Stop Recording
- Preserve Log
- Offline - useful for testing JS Heavy Sites
- Disable Cache
- Request View as List
- Request View - Inspect Request Headers, Response Preview, Response, etc.

# Network Panel - Advanced

- Filtering - All, XHR, Img, etc.
- FilmStrip - Capture Screenshots for loading
- Throttling
- Right Click Menu
  - Save As HAR
  - Copy - as curl etc.
  - Block - domain or specific requests
  - Clear Cache/Cookies (All)

# Application Panel

*Inspect all resources that are loaded, including IndexedDB or Web SQL databases, local and session storage, cookies, Application Cache, images, fonts, and stylesheets.*

# Experiments & Exercises

- Sweetshop - <https://sweetshop.netlify.com>
  - Check out the storage usage of Sweetshop
    - Any issues?
  - Can you create/amend/delete the data?
- TodoMVC - <http://todomvc.com>
  - Use one of the versions
    - Is any data sent to the server? (Check network tab?)
    - Is any data stored in the Application Panel?
    - Can you create/amend/delete the data?
- Twitter:
  - Network panel: View the traffic that twitter sends using XHR
- More on next page..

# Experiments & Exercises

- Testers Playground (<http://playground.epizy.com/devtools.php>)
  - Hidden within
  - Slow down
  - Image is everything
  - I'm hungry
  - Storage hunter

# Experiments & Exercises

- Any websites that you know of that only let you read some content per month e.g. medium.com, spectator.co.uk - what application storage are they using to track you? Can you amend it?
- What are other sites storing locally on your machine? What happens if you change the values?



Security, Extensions, Features, Console

# Security

- Show any Security Error or warnings
- Demo
  - Eviltester.com
  - VivRichards.co.uk

# Chrome Extensions

- Bug Magnet
  - <https://bugmagnet.org/>
- Useful Snippets
  - <https://github.com/eviltester/usefuljsnippetextension>
- Open In Incognito
  - <https://chrome.google.com/webstore/detail/open-in-incognito/peacnpgnojjkmbepjndigjojeoipkflb>

# Chrome Extensions

- Check My Links
  - <https://chrome.google.com/webstore/detail/check-my-links/ojkcdipcgfaekbeaelaapakgnjflfglf>
- Foxy Proxy Standard
  - <https://getfoxyproxy.org/>
- JavaScript Switcher
  - <https://github.com/maximelebreton/quick-javascript-switcher>
- CounterString Generator
  - <https://chrome.google.com/webstore/detail/counterstring/keklpkmoikeicakpclclkdmlhgkklmbd>

# Other Chrome Features

- Incognito Mode
- People / Profiles
- Task Manager
  - Good for spotting memory leaks and potential issues
  - Ebay.co.uk, channel4.com, twitter.com

# Console Panel - Advanced

- Use useful snippets - to show JavaScript

# Experiments & Exercises

- <https://sweetshop.netlify.com/basket.html>
  - The checkout process in sweetshop does not calculate basket total correctly, can you use Elements > Break on Subtree modifications to create a breakpoint when the total is calculated to see what is going wrong?
- Useful Snippets will generate JavaScript to the console you could try
- Experiment with some gist code from <https://gist.github.com/eviltester/>
  - You could run as snippets...
  - A Simple ZType bot <https://gist.github.com/eviltester/076b8e52a9b3c0bb3868a0ec6150c860> and run it against <https://zty.pe/>
  - The todo mvc snippets
    - Create data in todomvc app <https://gist.github.com/eviltester/f4b01d0b900cd4cbf64426f86c48a291>
- More on next page...

# Experiments & Exercises

- Install and Try some of the Chrome Extensions
- Use incognito mode to bypass the 'read too many articles' functionality on medium.com or spectator.co.uk
- Test a web app with the Task Manager visible and see if Memory or CPU gives you any concerns
- Play the Memory Footprint vs CPU race game
  - Pick a few websites - pick sites you think might be JavaScript heavy (or different parts of the same app)
  - Open them in different tabs
  - Open the Task Manager
  - See which one races to the top first (it wins the leaky memory and over-use of CPU prize)
- More on next page...



# Experiments & Exercises

- if you want some documentation and tutorials for JavaScript
  - <http://jsforcats.com/>
  - <https://www.learn-js.org/>
  - <https://www.eviltester.com/jsauto>
  - [testautomationu.applitools.com/automating-in-the-browser-using-javascript](https://testautomationu.applitools.com/automating-in-the-browser-using-javascript)
  - Alan's JavaScript and Games workshop slides <https://utm.io/uhjP>

# Experiment

# Wrap Up, Questions

# Lessons Learned & Questions

# Resources - Websites

Sweet Shop - <https://sweetshop.netlify.com/>

Chrome Keyboard Shortcuts

<https://developers.google.com/web/tools/chrome-devtools/shortcuts>

MDN – Introduction to HTML

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Getting\\_started](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started)

Challenges - (<http://playground.epizy.com/devtools.php>)

Practice Sites - <https://testpages.eviltester.com/styled/page?app=testpages&t=Others>

# BONUS: Simple JavaScript Game Bots

Galaxians game <http://michal.biniek.pl/games/galaxian.html>

```
var infiniteLivesBot = setInterval(function(){if(oLives.iLives<3){oLives.add();}},1000);
```

```
var autofireBot = setInterval(function(){MyShip.shoot();},200);
```

```
var machineGunner = setInterval(function(){new bullet({x: MyShip.x + aParams.width/2, y: MyShip.y - aParamsBullet.height, direction: -1});},200);
```

Stop bots with clearInterval e.g. `clearInterval(machineGunner);`

See <https://gist.github.com/eviltester/7262f4d0e8226cd2fa57d750e3faff95>