DYNAMIC WEB PAGES

CI435: Introduction to Web Development

Semester Two

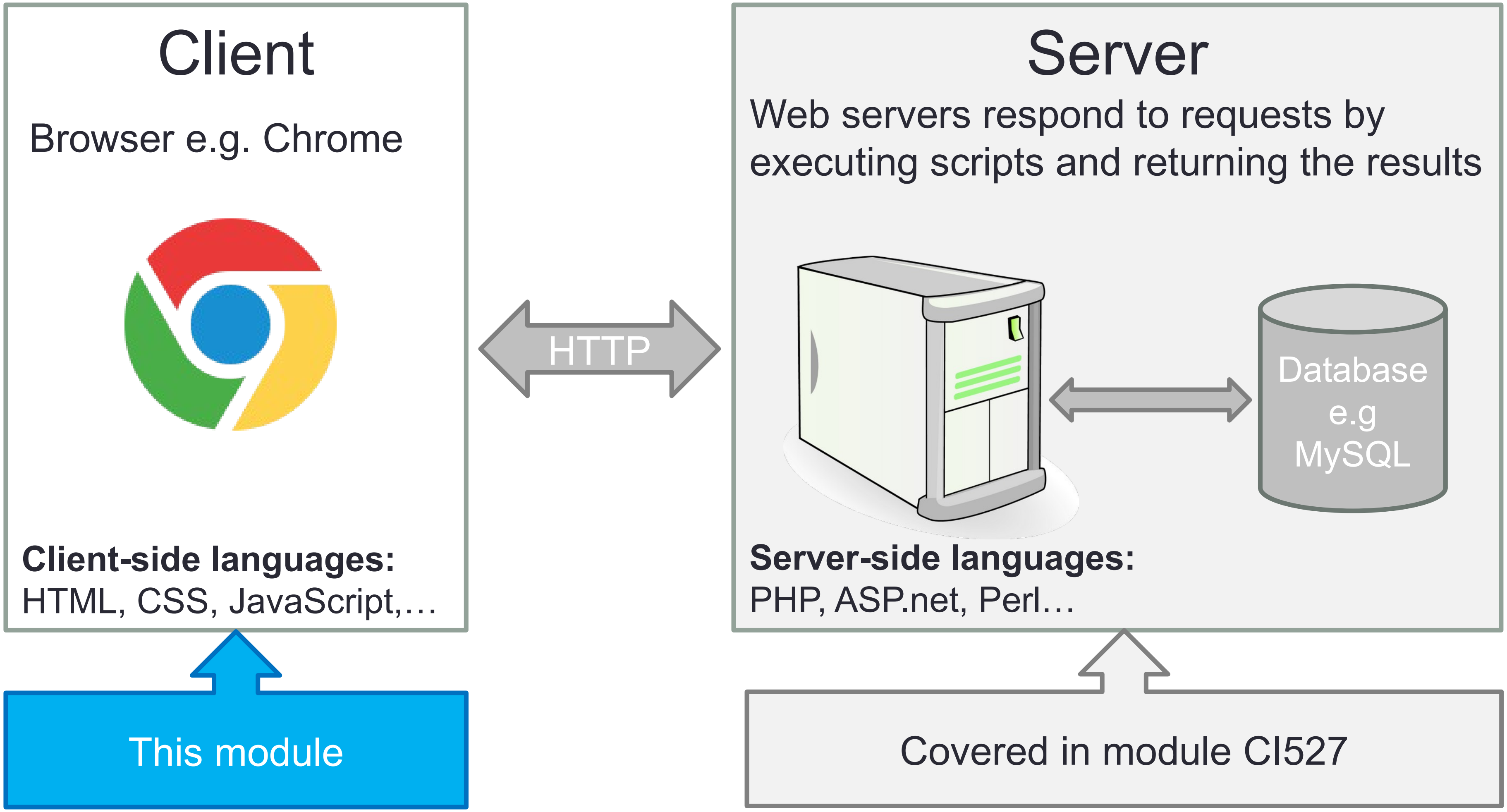
Robin Heath (with thanks to Marcus Winter)

Session overview

- This week we will look at:
  - Content and assessment in semester 2
  - A brief history of JavaScript
  - Tools we use
  - Examples we use
- Next week we'll look at JavaScript basics: statements, variables, data types and operators

SEMESTER OVERVIEW

Client and server technologies



Client

Browser e.g. Chrome

Client-side languages:
HTML, CSS, JavaScript,…

Server

Web servers respond to requests by executing scripts and returning the results

Server-side languages:
PHP, ASP.net, Perl…

This module

Covered in module CI527

Semester overview

- Semester 1: you created a **static website**
  - In a static website the content never changes

- Semester 2: you learn to develop **dynamic websites**
  - In a dynamic website the content may change due to user interaction or when the page is reloaded
  - This is done with scripting on the client and/or server
  - This module focuses on a client-side scripting with **JavaScript**

Content

- JavaScript overview
- Developer tools
- Variables and datatypes
- Control structures
- Functions and Events
- Document Object Model
- Working with forms

Indicative schedule available on studentcentral under CI435 > Module Information

Assessment

- This semester is assessed by an **exam**

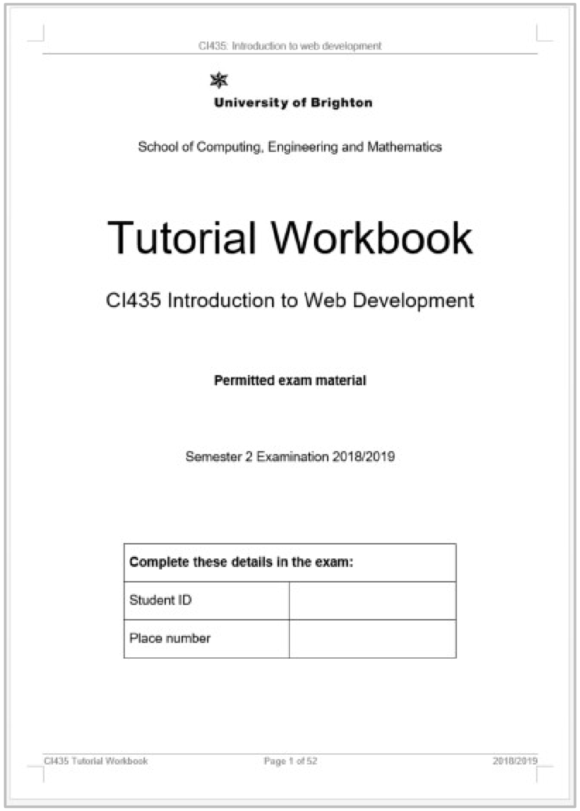
- The exam is worth **50%** of the final module mark

- You will be given a **tutorial workbook**, which is the **only material** you can bring into the exam

Tutorial workbook

The tutorial workbook:

- contains all weekly tutorial worksheets with questions and coding exercises to complete
- each tutorial is followed by a few blank A4 pages
- use the pages to **document your programs and any other notes** which you have made from that week's lecture or tutorial



CI435: Introduction to web development

University of Brighton

School of Computing, Engineering and Mathematics

Tutorial Workbook

CI435 Introduction to Web Development

Permitted exam material

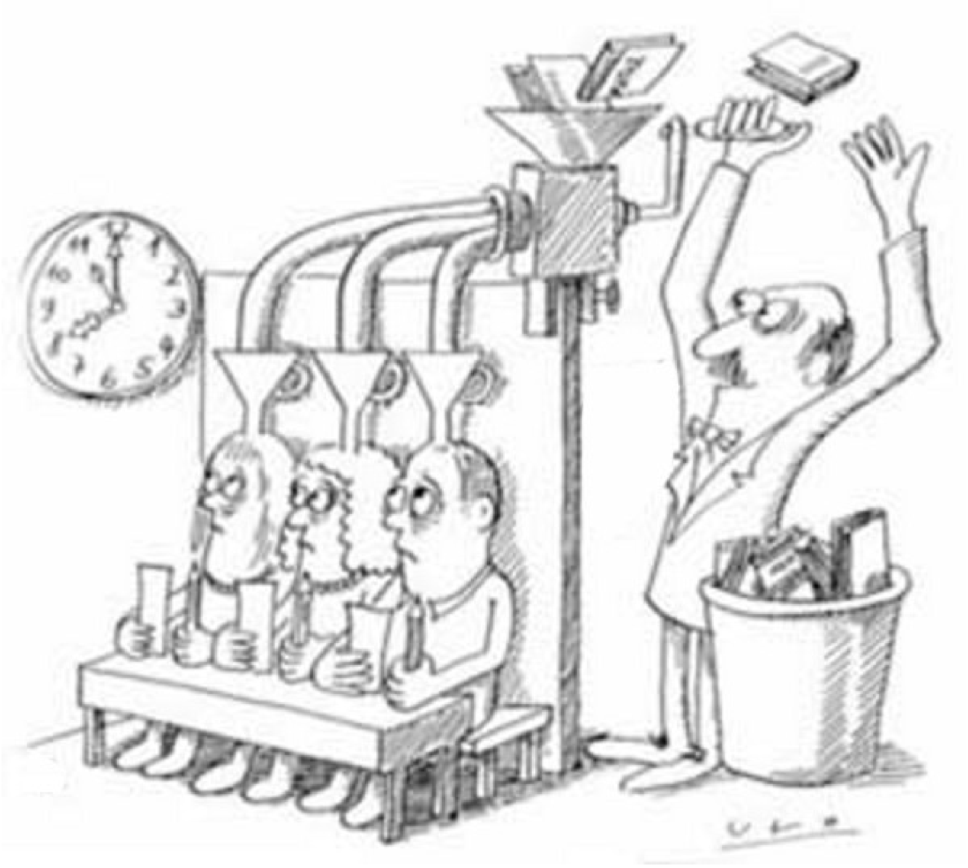
Semester 2 Examination 2018/2019

Complete these details in the exam:	
Student ID	
Place number	

CI435 Tutorial Workbook Page 1 of 52 2018/2019

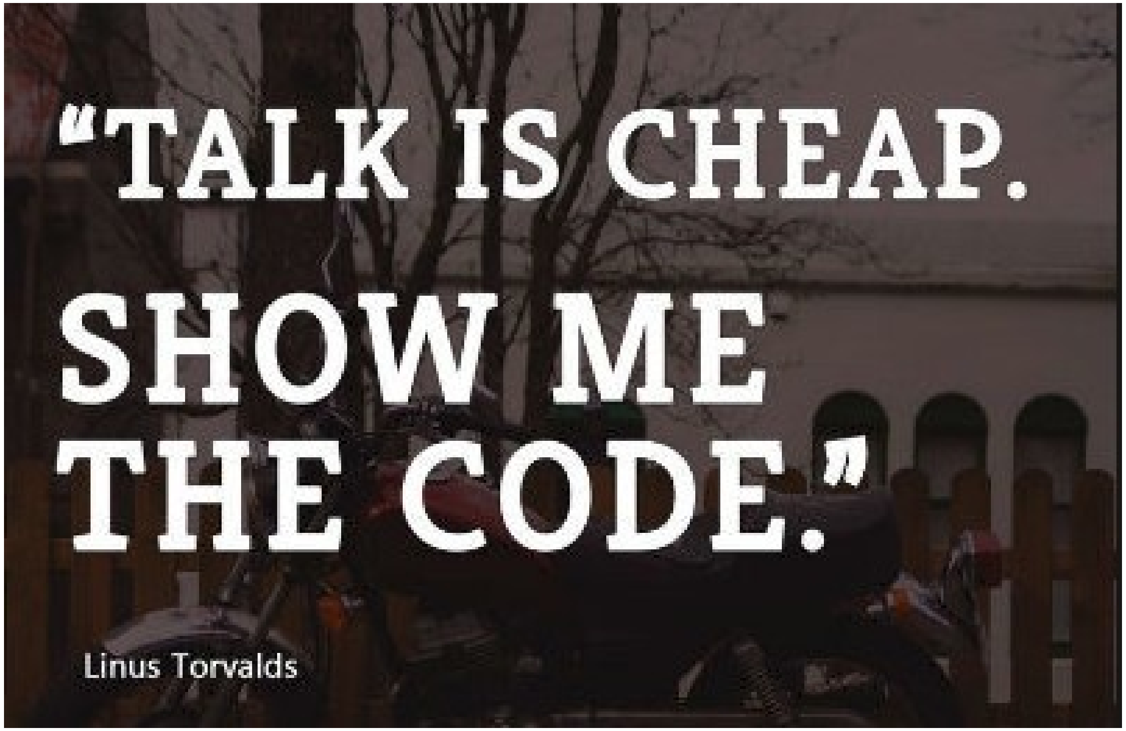
1. Download the Tutorial Workbook from studentcentral:
   CI435 > Study Materials > Semester 2 > Tutorial Workbook
2. Print the Tutorial Workbook before the exam
   Only a printed version is permitted in the exam

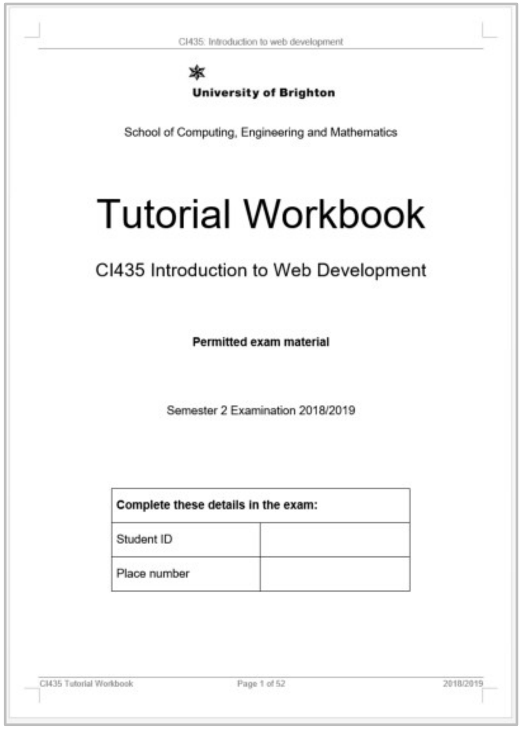
How to succeed in this module?



Transfer learning will not work!

How to succeed in this module?





CI435: Introduction to web development

University of Brighton

School of Computing, Engineering and Mathematics

Tutorial Workbook

CI435 Introduction to Web Development

Permitted exam material

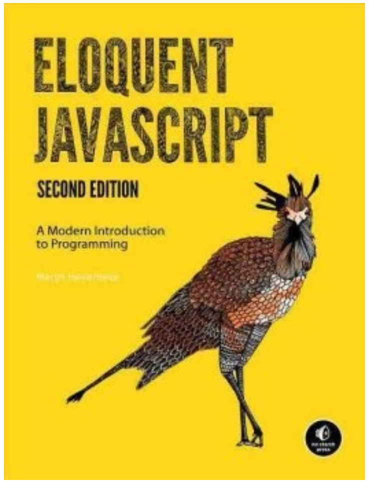
Semester 2 Examination 2018/2019

Complete these details in the exam:	
Student ID	
Place number	

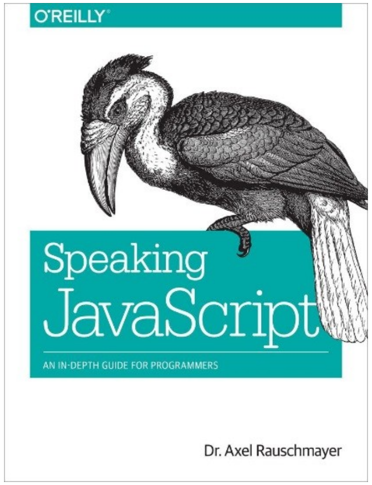
CI435 Tutorial Workbook Page 1 of 52 2018/2019

Learn by writing code and fixing mistakes!

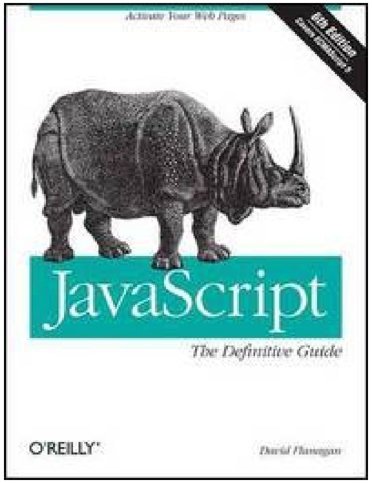
Recommended books



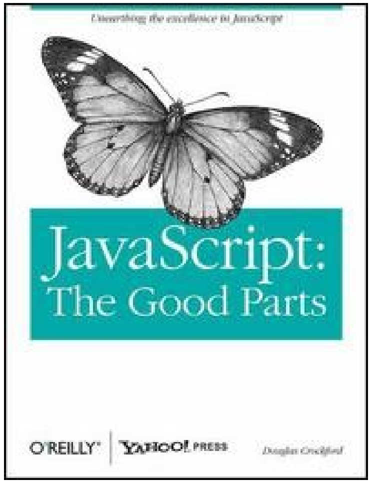
Eloquent JavaScript
Marijn Haverbeke



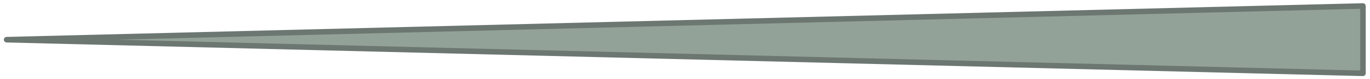
Speaking JavaScript [online]
Axel Rauschmayer



JavaScript: the definitive guide
David Flanagan


Unearthing the excellence in JavaScript
JavaScript: The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

JavaScript: the good parts
Douglas Crockford



Beginner

Advanced

Recommended websites

- **The Mozilla Developers Network (MDN)**
  - https://developer.mozilla.org/en-US/
  - A very complete reference for all sorts of web technologies
- **Code Academy (JavaScript track)**
  - http://www.codecademy.com/tracks/javascript
  - An online tutorial on JavaScript that starts from the very basics and gradually builds up to more advanced concepts

Note: There are **many** JavaScript resources available online. Some are useful, others are old, inaccurate and/or demonstrate bad practices.

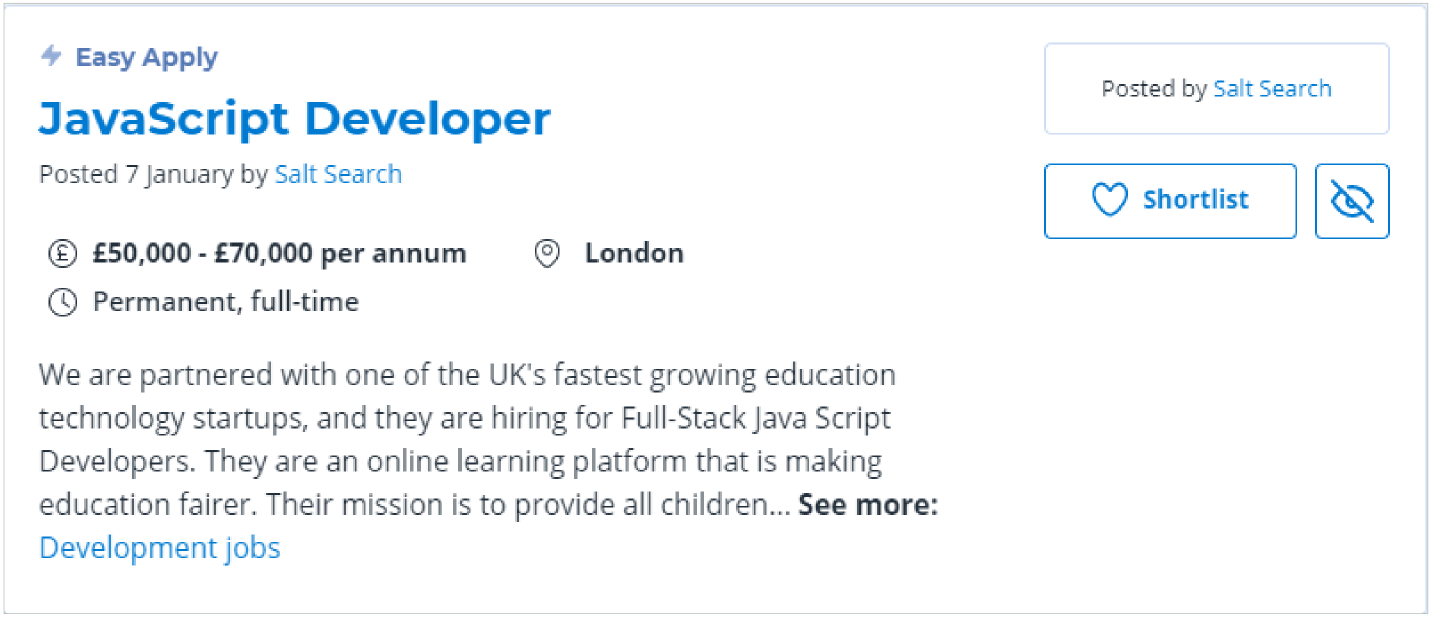
JAVASCRIPT

JavaScript jobs

A quick look at job advertising sites (reed.co.uk) Feb 2021:

- Java: 2,400
- JavaScript:4,200

Currently amongst the most in-demand languages



Easy Apply

JavaScript Developer

Posted 7 January by Salt Search

£50,000 - £70,000 per annum   London

Permanent, full-time

We are partnered with one of the UK's fastest growing education technology startups, and they are hiring for Full-Stack Java Script Developers. They are an online learning platform that is making education fairer. Their mission is to provide all children... **See more:** Development jobs

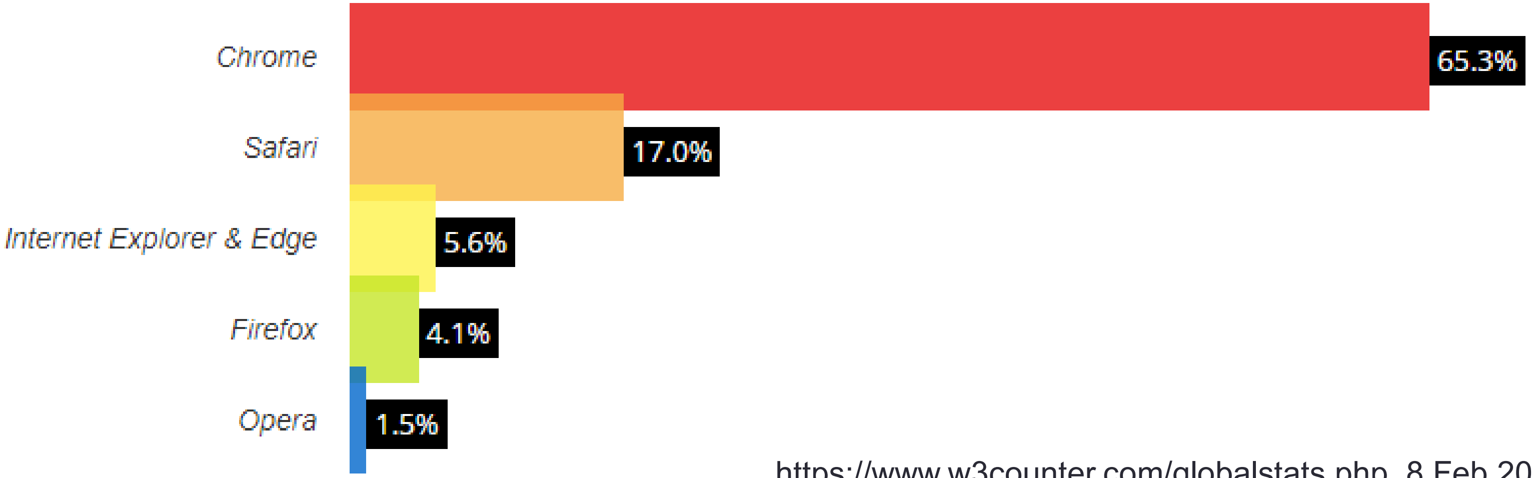
Posted by Salt Search

Browser compatibility

There are many browsers out there, all of which support different features and versions of JavaScript.

Supporting all browsers is a big challenge in a professional environment but outside the scope of this module.

We'll focus on recent, **standards-compliant**, browsers, e.g. Chrome, Firefox (Safari, Opera , IE10, Edge)



https://www.w3counter.com/globalstats.php, 8 Feb 2021

Background

Source: wikipedia.com

JavaScript was created in 1995 by **Brendan Eich**
- in only 10 days!

1996 - 1997 JavaScript was standardised by **ECMA***
- it's official name is **ECMA Script**.

Read: A brief history of JavaScript:

- https://auth0.com/blog/a-brief-history-of-javascript/

* ECMA = European Computer Manufacturers Association (now “ECMA International”)

What is JavaScript?

- Lightweight scripting language

- Originally designed to add interactivity to HTML pages
  - HTML ☾ content
  - CSS ☾ presentation
  - **JavaScript ☾ behaviour**

- All major websites use JavaScript
- Some use only JavaScript (!)

What is it used for in the browser?

JavaScript Is primarily used to make pages interactive.

Common uses:

- Check form values
- Change appearance of HTML elements (e.g. show/hide)
- Modify the HTML document structure
- Communicate with the server via AJAX
- Web APIs, e.g. Google Maps, Twitter, etc.
- Browser APIs, e.g. graphics, storage, etc.

Declarative vs. Procedural

Unlike HTML and CSS, which are declarative languages, JavaScript is a **procedural** language:

- declarative: describe how things should be
- procedural: describe how things should be done

Many features similar to other procedural languages:

- variables, data types, operators
- statements, conditional statements, loops
- functions, namespaces

Compiled vs. Interpreted

Unlike compiled languages (e.g. Java, C, C++), JavaScript is an **interpreted** language:

- compiled: read, parse and translate source code once into an executable file, then executed many times

  workflow: code > compile > run > code > compile > run > ...

- interpreted: read, parse and translate source code each time upon execution

  workflow: code > run > code > run > ...

Strongly vs. Weakly typed

Unlike strongly typed languages (e.g. Java, C, C++), JavaScript is a **weakly typed language** language:

- strongly typed: variables are of a specific data type and only can hold data of that type:

```
Java:      int x = 10;
           x = "hello";  // error
```

- weakly typed: variables can hold any type of data:

```
JavaScript:      var x = 10;
           x = "hello";  // not a problem
```

A serious programming language?

Being an interpreted, weakly typed and having some unique quirks*, JavaScript has for a long time not been taken seriously by "real" programmers.

Today, JavaScript is not only the *lingua franca* on the Web, but has developed into a general purpose language that increasingly breaks into other domains such as:

- server-side and desktop computing
- embedded computing
- Internet of Things

(* We'll learn about these as we go along...)

TOOLS

Tools

A **text editor**, **web browser** and **web space** is all we need:

Recommended text editor: **Notepad++**

Alternatives: VS Code, Brackets,, Atom, Sublime

Recommended browser: **Google Chrome**

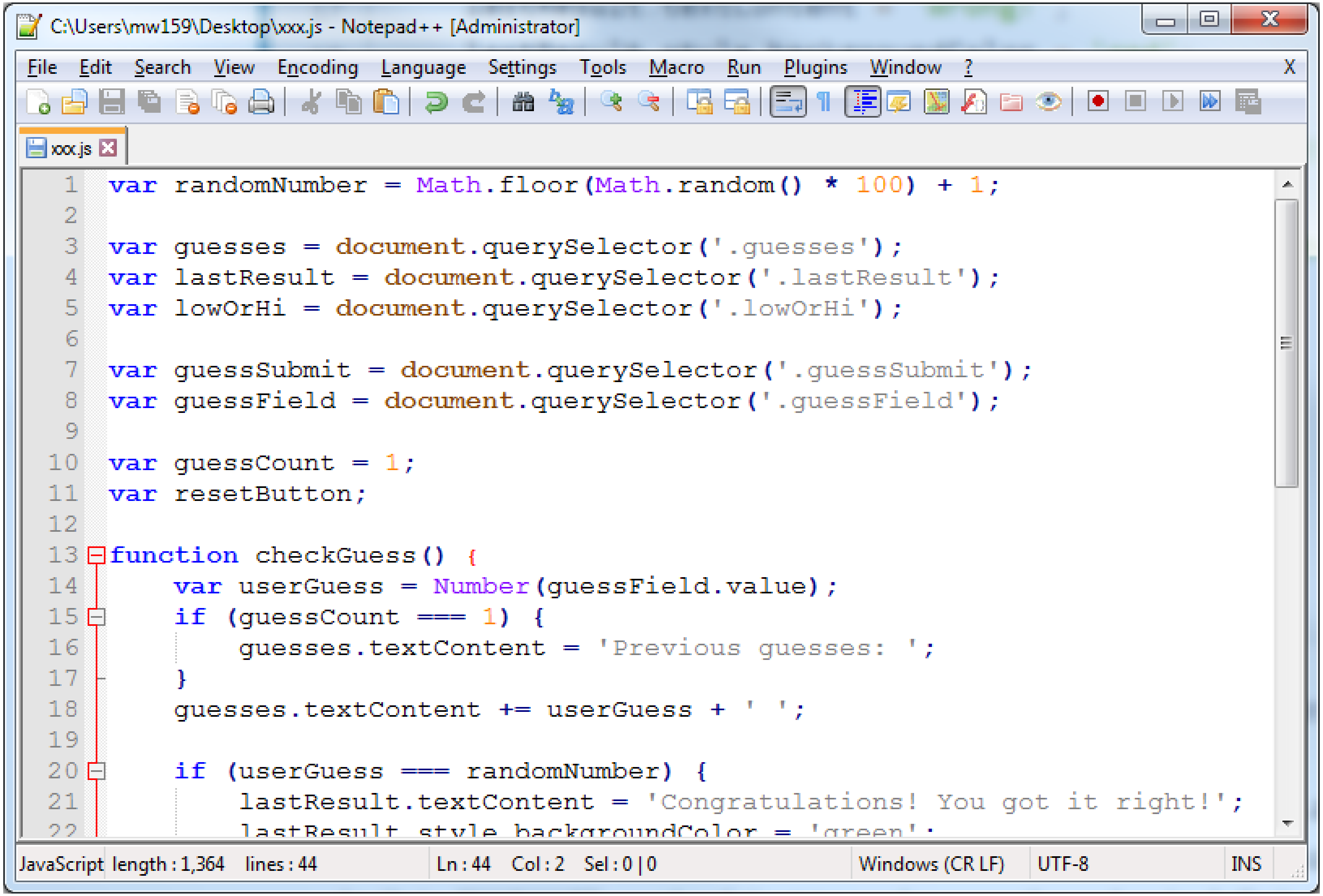
Alternatives: Mozilla Firefox

Your brighton.domains Web space:

`http://<username>.brighton.domains/`

FileZilla to transfer files (local computer / web space)

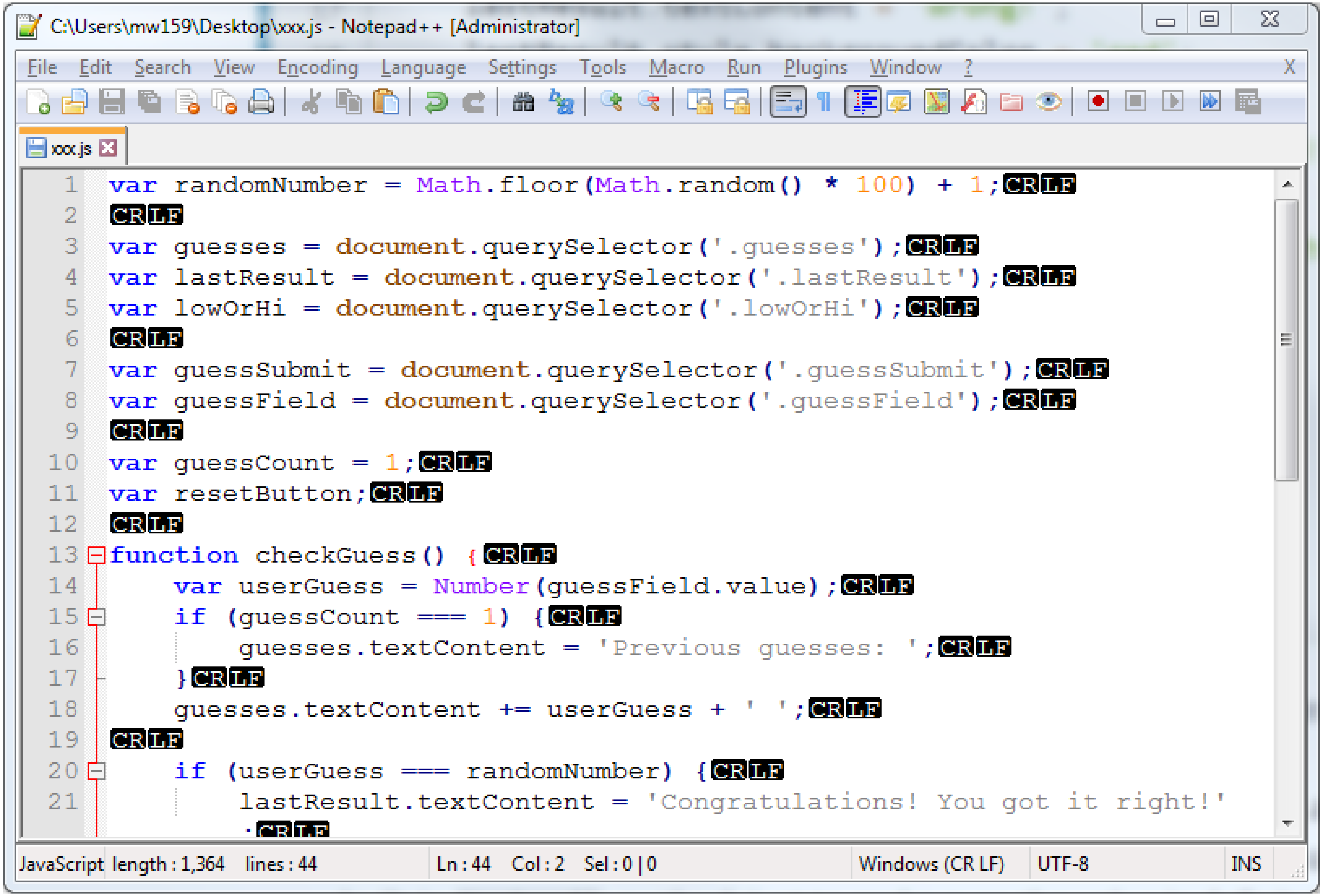
Notepad++



```
var randomNumber = Math.floor(Math.random() * 100) + 1;

var guesses = document.querySelector('.guesses');
var lastResult = document.querySelector('.lastResult');
var lowOrHi = document.querySelector('.lowOrHi');

var guessSubmit = document.querySelector('.guessSubmit');
var guessField = document.querySelector('.guessField');

var guessCount = 1;
var resetButton;

function checkGuess() {
    var userGuess = Number(guessField.value);
    if (guessCount === 1) {
        guesses.textContent = 'Previous guesses: ';
    }
    guesses.textContent += userGuess + ' ';

    if (userGuess === randomNumber) {
        lastResult.textContent = 'Congratulations! You got it right!';
        lastResult.style.backgroundColor = 'green';
```

JavaScript length : 1,364 lines : 44 Ln : 44 Col : 2 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

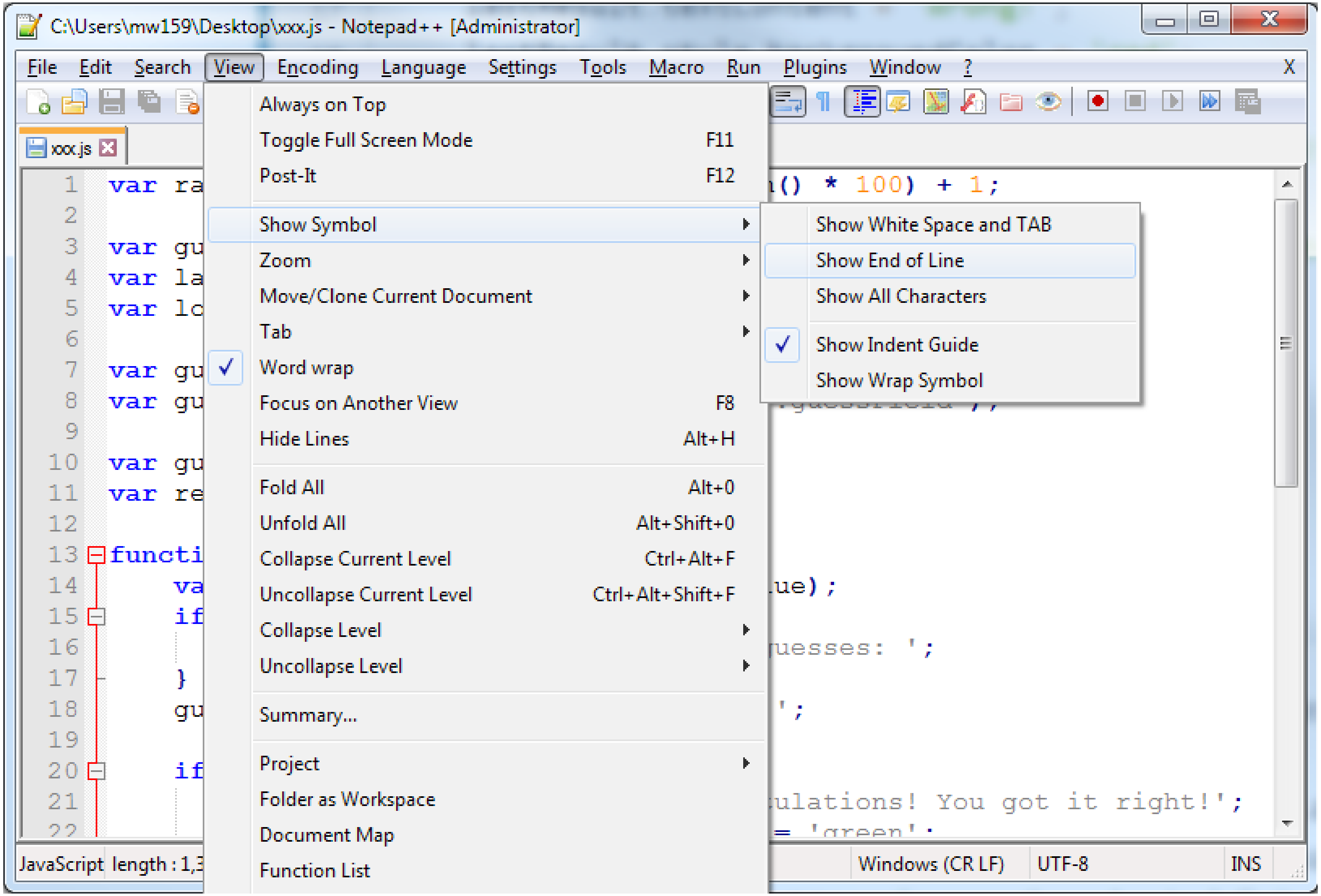
Notepad++



```
var randomNumber = Math.floor(Math.random() * 100) + 1;

var guesses = document.querySelector('.guesses');
var lastResult = document.querySelector('.lastResult');
var lowOrHi = document.querySelector('.lowOrHi');

var guessSubmit = document.querySelector('.guessSubmit');
var guessField = document.querySelector('.guessField');

var guessCount = 1;
var resetButton;

function checkGuess() {
    var userGuess = Number(guessField.value);
    if (guessCount === 1) {
        guesses.textContent = 'Previous guesses: ';
    }
    guesses.textContent += userGuess + ' ';

    if (userGuess === randomNumber) {
        lastResult.textContent = 'Congratulations! You got it right!'
        ;
```

JavaScript length : 1,364 lines : 44 Ln : 44 Col : 2 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notepad++ > View > Show Symbol > Show end of line (untick)



Notepad++

Find	CTRL + F
Replace	CTRL + H
Undo	CTRL + Z
Redo	CTRL + Y
Save	CTRL + S

Increase Indent: [select] Tab

Decrease Indent: [select] Shift + Tab

Chrome Developer Tools

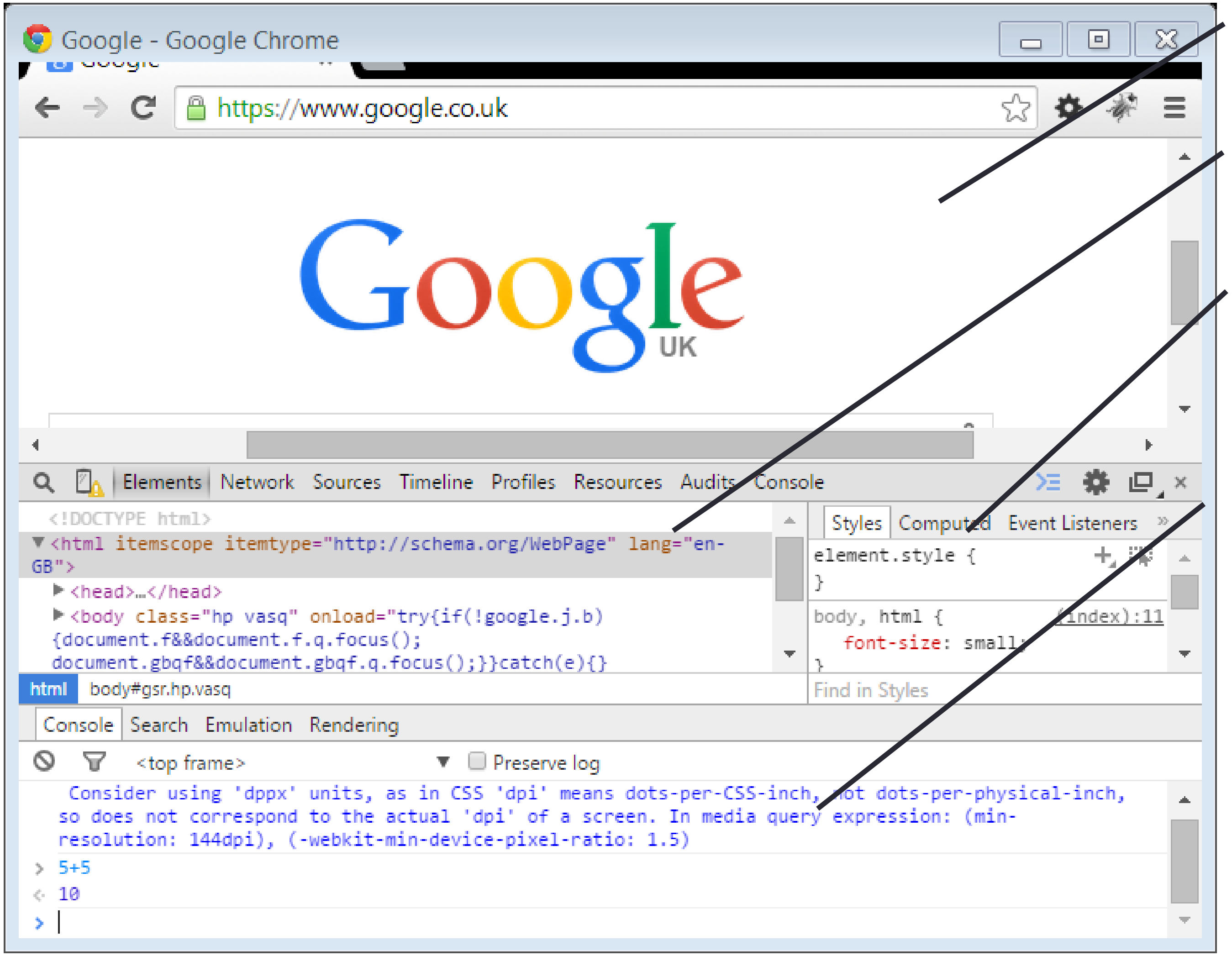
Powerful tools built into the Chrome browser:

- Inspect HTML, CSS, JavaScript
- JavaScript console
- Debug JavaScript
- Analyse and optimise page loading
- etc.

Activate developer tools:

- CTRL + SHIFT + I (CMD + ALT + I on Mac)
- or right-click page and select Inspect from context menu

Chrome Developer Tools



```
<!DOCTYPE html>
▼<html itemscope itemtype="http://schema.org/WebPage" lang="en-
GB">
  ▶<head>…</head>
  ▶<body class="hp vasq" onload="try{if(!google.j.b)
  {document.f&&document.f.q.focus();
  document.gbqf&&document.gbqf.q.focus();}}catch(e){}
```

```
element.style {
}
body, html {                    (index):11
  font-size: small;
}
```

```
 Consider using 'dppx' units, as in CSS 'dpi' means dots-per-CSS-inch, not dots-per-physical-inch,
so does not correspond to the actual 'dpi' of a screen. In media query expression: (min-
resolution: 144dpi), (-webkit-min-device-pixel-ratio: 1.5)
> 5+5
< 10
> |
```

Viewport

HTML

CSS

JavaScript Console

Errors in your code will typically appear here.

JavaScript statements can be directly entered, e.g. 5+5 evaluates to 10

Chrome Developer Tools

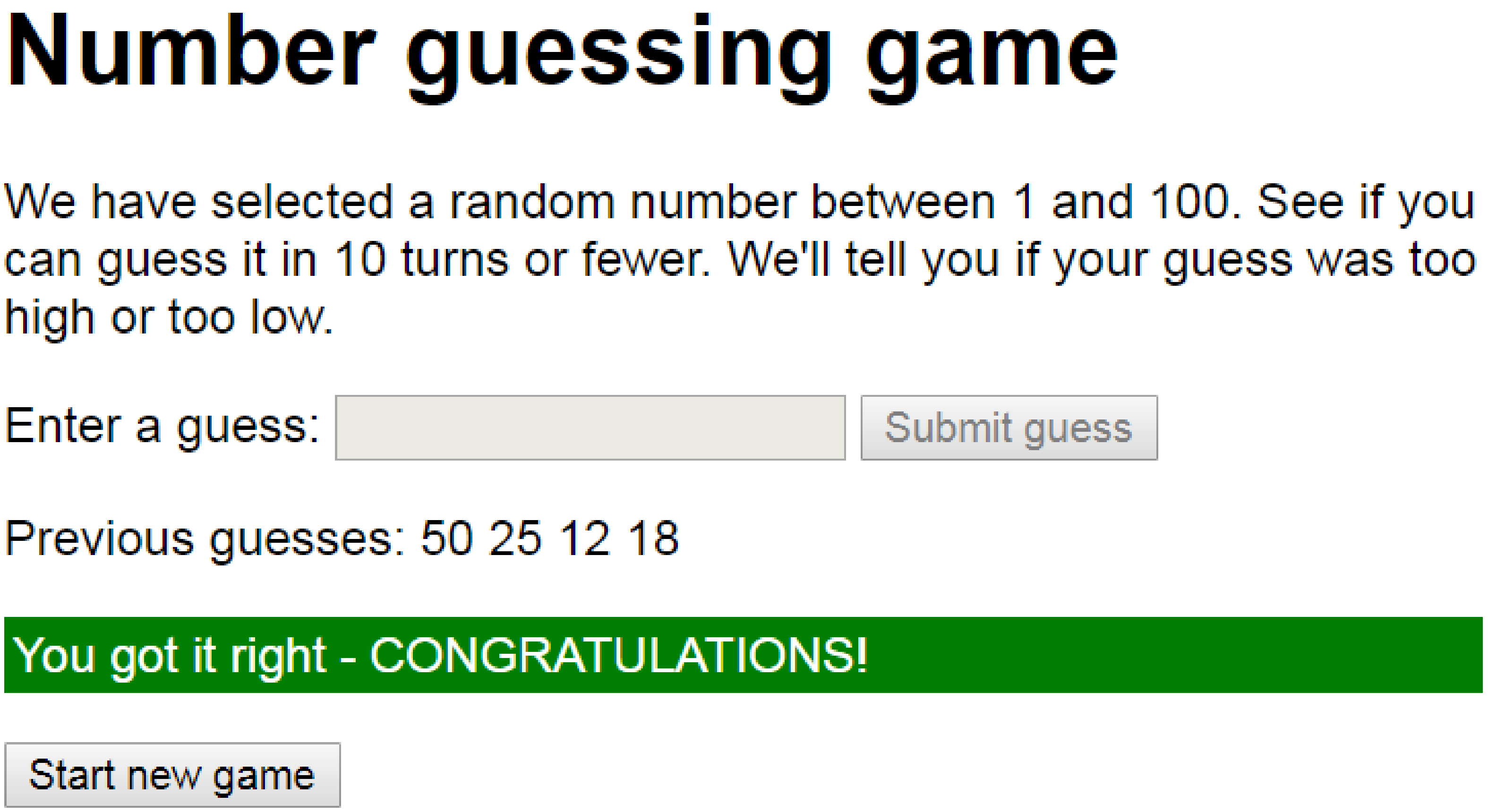
Inspect and debug JavaScript code in Chrome:

https://developers.google.com/web/tools/chrome-devtools/javascript/

We'll explore the Chrome Developer Tools in more detail in the coming sessions...

DEMO CODE

Example 1



Number guessing game

We have selected a random number between 1 and 100. See if you can guess it in 10 turns or fewer. We'll tell you if your guess was too high or too low.

Enter a guess:

Previous guesses: 50 25 12 18

You got it right - CONGRATULATIONS!

Example 2

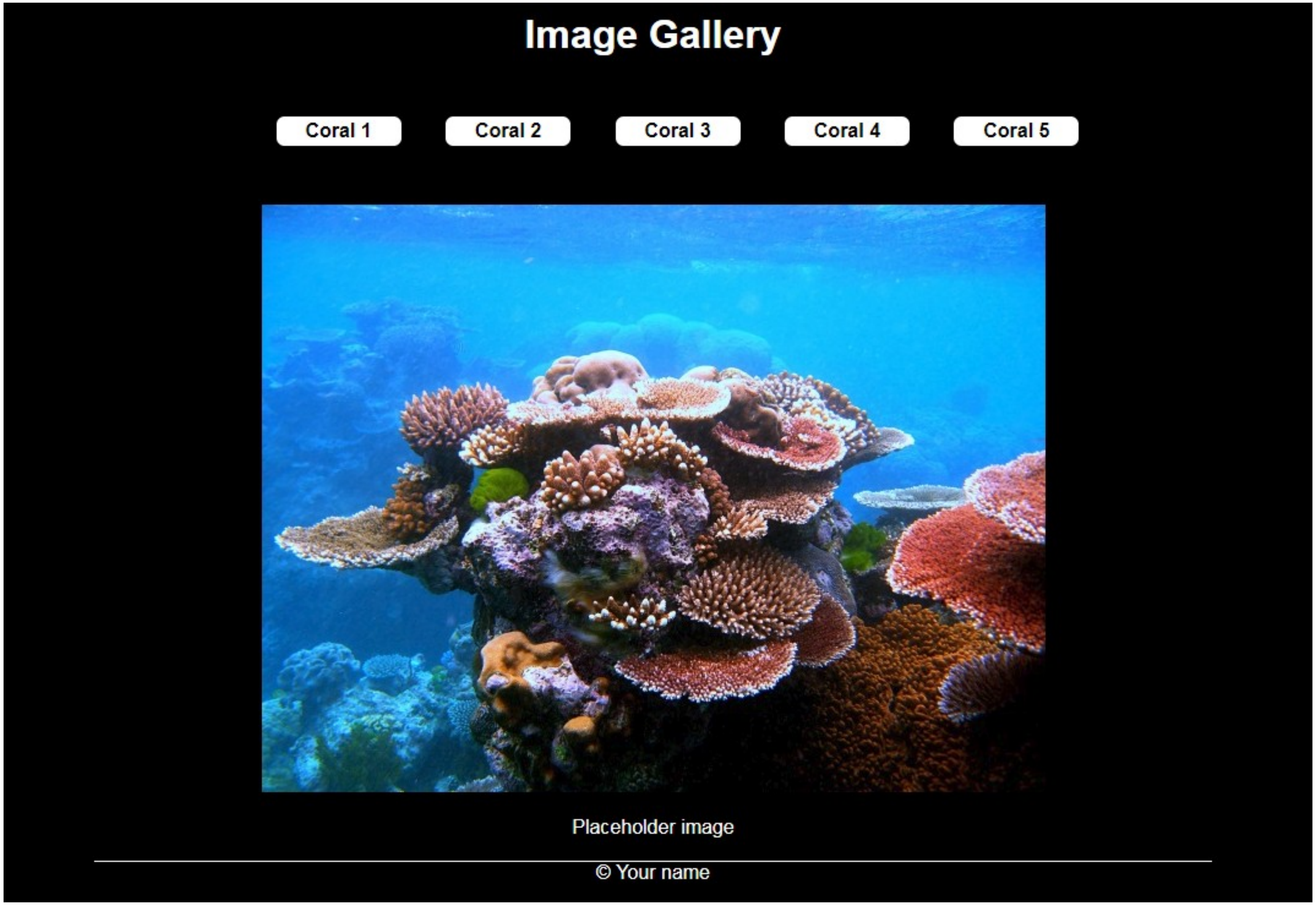


Image Gallery

Placeholder image

© Your name

Based on these examples we'll cover:

- Variables
- Control structures and loops
- Strings, arrays and objects
- Functions
- Events
- Document Object Model
- Dynamic styling
- Handling forms

By the end of the semester you will be able to build dynamic web pages like these examples

Recommended reading

Chrome developer tools: Using the console
https://developers.google.com/web/tools/chrome-devtools/console/

Chrome developer tools: Inspect and debug JavaScript code
https://developers.google.com/web/tools/chrome-devtools/javascript/

A brief history of JavaScript:
https://auth0.com/blog/a-brief-history-of-javascript/

Any Questions?

KEEP
CALM
AND
KEEP
CODING