CODEFIRST: Professionals

Web Fundamentals - Session 6

Chapter 0 – Preamble

So far, we've learnt how the Web works (in brief), to display content online in a structured way (HTML), style and design our content (CSS), work collaboratively on a website (GitHub & Git), and discovered some shared best practices, mindsets and tools that developers use to make their coding lives easier.

In today's session, we're going to be doing a deep dive into programming for the web.

How JavaScript Came About

Back in the beginning of the the Web, when the first browsers were being created (Netscape vs Internet Explorer), Marc Andreessen (the co-creator of the Mosaic/Netscape Navigator browser, whose engine is now used to power Firefox, and co-founder of Netscape) realised the Web needed to become more dynamic - It was all HTML and CSS at the time, and there was limited interaction. So they experimented using different languages that had already existed to allow data to be dynamically manipulated through websites, and were not too happy with the results. In short, they recruited Brendan Eich to try embedding one of these languages, but in the end, he decided to create a language that would be prototype-based, and JavaScript was born.

But wait! What even is Programming?

Hold on! What even is programming? There are so many different programming languages available - why do they all exist, and where do I start? And OMG I'M SO CONFUSED.

To be concise:

1. **Programming** is the way humans tell computers to do logical things for them in a systematic fashion. It lets humans create ways for other people to interact using computers.
2. There are a wide variety of programming languages, BUT to decide which to use, we need to decide what we want to use it FOR, and then **pick the language best suited for the use**.
3. Programming languages are organised into **paradigms**, that is - ways of thinking and communicating with the computer.
4. JavaScript is powerful because it is:
    a. **multi-paradigm** (it can accommodate a range of thinking/programming styles),

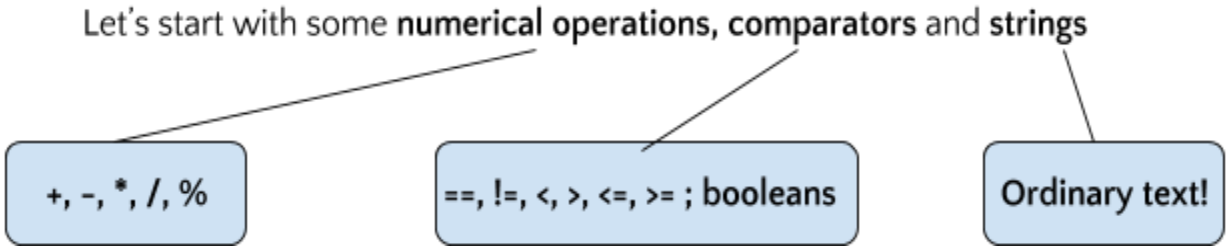
b. **prototype-based** (things that you have defined can be changed easily #datMVPlife),
c. **designed to be used for the Web** (whereas other languages have been designed for scientific computing, e.g. Python - although Python is a very flexible language too, or native application development, e.g. Swift / Objective C for iOS, and .NET for Windows),
d. and finally, it is a **full-stack language** (it can be used **both** on the **client** and **server** - other languages, such as Python or Ruby, are server-side languages).

In essence, it is **super flexible**, and therefore super awesome □□□

Chapter 1 – JavaScript & jQuery

So... how do I start using JavaScript?

Good question! Because JavaScript is a standard Web technology, you can write it directly in your browser! Let's open our Developer Tools and start writing some JavaScript!



All three of these can be used to form an **expression**; which is an **operation** performed on a **data type**.
A **function** is a **group of expressions** which **come together** to **perform a particular task** when it is **called upon** (more formally, invoked) by an an **action** (more formally, an event). This can be an action made by the user, or by the server, or some sort of external event which your program is listening to.
A **variable** is what JavaScript uses to **store, organise and manage raw data** which is being handled by functions & expressions. Variables are therefore manipulated by functions and expressions. A function takes a variable (or sometimes exact / raw data), digests it and does something to it, and returns a changed value.

JavaScript uses semicolons, like CSS does, to decide when to stop executing a particular task.

If you want to write JavaScript in your webpages, there are a couple of ways of doing it:

- You can link it using the same way we have been linking our CSS files (***recommended)***
- Or you can write it within <script> tags directly in your HTML file.

And what is jQuery?

JavaScript can be quite complicated to learn, and tedious for basic functionalities. jQuery is a **JavaScript library** that is useful for building interactive webpages.

Recap: A **library** is an implementation of an API; it is a set of functions that a developer can call, usually organised into classes. It contains the compiled code that implements the functions and protocols (maintains usage state).

jQuery is so common in webpages that, for beginners, 'learning JavaScript' has in many cases become 'learning jQuery'. This is the approach that we're going to take in this course.

Getting jQuery into your website

jQuery is just a JavaScript file that can be **downloaded** from the jQuery downloads page. There are a couple of different versions - I'd go for the latest. **Alternatively,** you can use the **JQuery CDN,** and **link directly** to their hosted online version.

For each version you can either get the *normal* code, which is useful for your own development, or the ***minified*** (jquery.min.js) code, which has had all the space (and some other unnecessary stuff) taken out to make it as small as possible, so it downloads quicker.

Once you've downloaded the file and saved it in your site folder, or chosen the CDN version you want to use, you need to **link** to it in your page.

For the time being we'll do this in the head (you can improve page-load times by moving your javascript to the bottom of the page, but we won't bother with this at the moment).

```
<!DOCTYPE html>

<head>

  <script src="jquery.min.js"></script>

  <!-- any other stuff e.g. stylesheet links -->
```

Using jQuery to manipulate CSS

You can do a lot of stuff with jQuery. Here we'll just look at the basics: selecting elements on the page and doing stuff with them.

Read the API documentation here, and give it a shot

You can **experiment** with jQuery using the Console section of the **Chrome developer tools**. You will need to be on a page where jQuery is loaded (e.g. these course notes or the demo page you will download in the exercise).

One of the nice things about jQuery is its ability to select elements via their CSS selectors. To select elements jQuery uses the $(' ') function. For example:

```
$('li')              // selects all the li elements on the page

$('li .important')    // selects all the li with class="important"

$('#main-title')     // selects the element with id="main-title"
```

jQuery then has several ways of manipulating those elements.

In Class Demo:

1. Use this demo to see how an element can be manipulated by un-commenting the lines of jQuery in the JS section
2. Can you make the list items go green?
3. Let's try another demo: See how you can input text onto a page using jQuery

More CodeFirst: Girls' jQuery notes on jQuery here - Spend some time reading this and implementing it in your groups in class.

The almighty AJAX

So we've talked about front-end / client-side and back-end, server-side, but how do we connect the two and get them talking to each other?

We have to perform an Ajax request over HTTP/HTTPS (transfer protocol).

Using AJAX, we can:

- Update webpages without having to reload them
- Request data from servers after the page has loaded
- Receive data from servers after the page has loaded

- Send data to a server during the page session, in the background

Once you have tried using jQuery on CSS and in the further notes linked above, give it a shot using the jQuery documentation here.

Chapter 2 – jQuery resources

Obviously, we've only just scratched the surface of what's possible with jQuery. Things get a lot more interesting when you can create bits of JavaScript to be run in response to a user action.

This allows you to build up interactions like "when the user clicks the submit button, check that their email is a valid email, if it isn't make the field go red and add the words 'email is invalid' at the bottom of the form".

Learning more jQuery

We won't be spending any more time on jQuery this course, and will be moving onto backend stuff next term. If you want to learn more about jQuery you might want to try some of the following resources:

- The Codecademy jQuery Course
- The jQuery Learning Center
- The CodeSchool jQuery Course
- W3 Schools Tutorial

CodePen/jsFiddle

CodePen and jsFiddle are sites which allows you to try out small bits of HTML, CSS and Javascript. It's a really useful tool for getting good help with JavaScript (and HTML/CSS) online: If you're having a problem:

1. Create a jsFiddle or CodePen showing what you've tried.
2. Post on StackOverflow describing the problem, with a link to the jsFiddle/CodePen.

People will be able to help you better if they can see the code themselves. Often they will respond with a working jsFiddle. (as in this example)

CodePen is used by many front-end devs to showcase their portfolios - it even has a "hire me" button.

Homework

Finishing off

Task:

Add JS to your website. Continue to work on your first_site until it's something you can be proud of!

Preparation

Find out about the command line:

- What is the Command Line?
- How does it allow me to interact with my computer?
- Read this.
- (**Optional**) Try it on Codeacademy

Group Project

Task:

Meet outside of class to work on your project!

Additional suggestions you could think about for your project are here.

Further Resources

Bootstrap provides some JS functionalities as well, built on jQuery too.

What's the difference between a Framework and a Library?

Ben Howdle talks about different JS Frameworks

The Mozilla Web Developer Guide