Introduction to web development

Coding Course Curriculum

Week 7 - Course competition work, Plugins, Metrics, HTML DOM, APIs

What we'll cover this week

1. Recap of what we learnt last week (JavaScript & jQuery)

1. Making process on your websites

1. Optional content: Plugins, Metrics, HTML DOM, APIs

1. Homework

What we covered in week 6 - JavaScript & jQuery

What did we learn last week?

1. JavaScript & jQuery
2. So… how do I start using JavaScript?
3. And what is jQuery?
4. Getting jQuery into your website
5. Using jQuery to manipulate CSS

Task: Find a partner and together take a quick look through the notes from last week's session. If you're unclear on any of the concepts, work through them with your partner and an instructor

Course competition

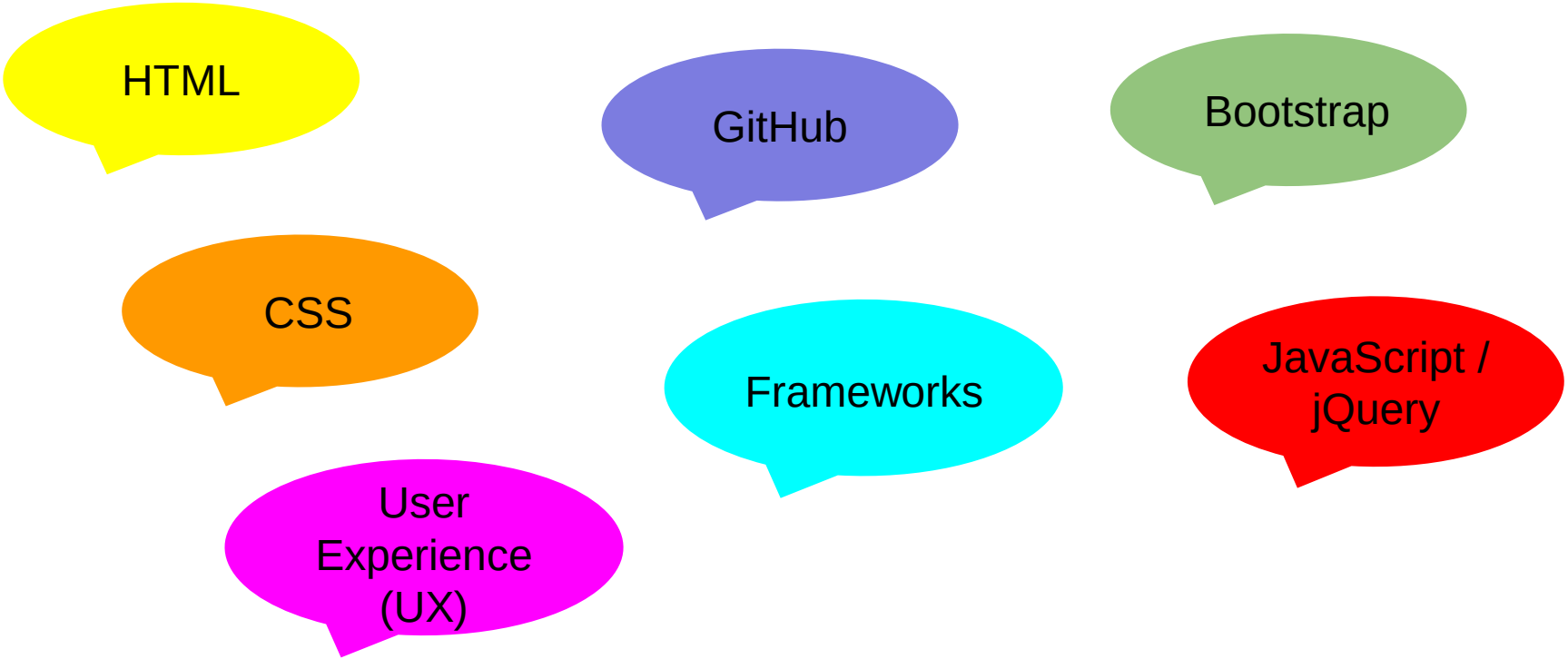
A reminder of the competition criteria: Must have

- A live website published on GitHub pages
- A minimum of two HTML files for:
  - 1 x landing page (Index.HTML) linked to a separate CSS file
  - 1 x ‘about’ page
- A minimum of one CSS file
- Good formatting
  - Code split into the appropriate files (separate HTML files & CSS files)
  - Files indented properly
- Good organisation
- Version control using git with sensible git commit messages

Nice to have

- A visually appealing design - good use of CSS and HTML elements, Twitter Bootstrap, Jquery & Javascript
- A contact form (for example name and email)
- Social buttons
- As many different HTML elements as you can manage
- A responsive site

Questions or queries?



Working on websites - troubleshooting

Now we will spend an hour working on our group projects

Optional content

(Google analytics, domain names for Github pages)

Google analytics

- Google Analytics is an analytics service for your website, provided for free by Google

- It allows you get an overview of how many people are visiting your site, where they come from, what they do on your site, and much more

- https://analytics.google.com/

How google analytics works

- How does Google analytics track web traffic to your website?
- By JS Plug-In, a snippet of JavaScript, which Google provide
- User visits page ⟹ the JavaScript sends a message to the Google Analytics site, logging the visit

Task:

1. Set up a Google Analytics account - You want to choose the default 'Universal Analytics' option
2. Go to the **Admin** section, create a new account for your personal site.
3. Click on the **Tracking Info** under the **Property** section, click on “**Tracking Code**” and install the analytics code on all the pages of your site.

Google forms

Google Forms uses the <iframe> tag to embed a mini-form document into your web page, where you want it

This can be a quick and easy way to collect information from users on your website via online forms.

Does this sound like a good addition to your website? Then follow the below instructions to embed your Google Form for the course competition!

Google forms - instructions

Task:

1. Create a Google account (if you don't already have one) so you can log directly into Google Forms (Or click on the Drive icon, then click "More", link for Google Forms)
2. There's a selection of forms you can use, either a blank form or a template form to add to
3. Using either of the forms, you can easily add in your own content to the form (e.g. changing the fields depending on the questions and data you'd like to collect, and change the appearance of the form). Full instructions can be found here.
4. Be sure to configure how your form functions and is accessed (e.g. enabling multiple answers responses from a single user, or customising the submission confirmation message).
5. Happy with the form? You can make it available by clicking on the "Send" button (top right corner of the screen).Choose how you share the form, via email, a link, or embedded on a web page. Click the "<>" icon to "Embed HTML".
6. Copy and paste the link provided by this icon, which will look something like this: <iframe src= "https://docs.google.com/forms/d/[...] </iframe>". Add this code into the relevant HTML code where you'd like your form to appear.

Domain names for github pages

Remember in Session 1 we introduced you to Domain Names?

Quick recap: in order to put up your own website at your own domain name you need two things:

- A web server to serve your site
- A domain name to point towards it

Domain registrars are companies like 123reg, godaddy and namecheap

As you've been working with Github and publishing your sites via GitHub pages, we wanted to give you some more guidance on setting up a custom domain using Github Pages.

How to create a domain name for Github pages

Task (if you have bought a personal domain name):

For Github pages there are three main stages to setting up a custom domain:

1. Pick a custom domain and register it with a DNS provider/Domain Registrar/DNS host (3 names for the same service). Some examples of these: 123-reg.co.uk, godaddy.com and namecheap.com.
2. Add your custom domain to your Github Pages site. Follow these instructions here.
3. Set up your custom domain with your DNS provider. For more information on how this should look find some examples from Github pages here and here.

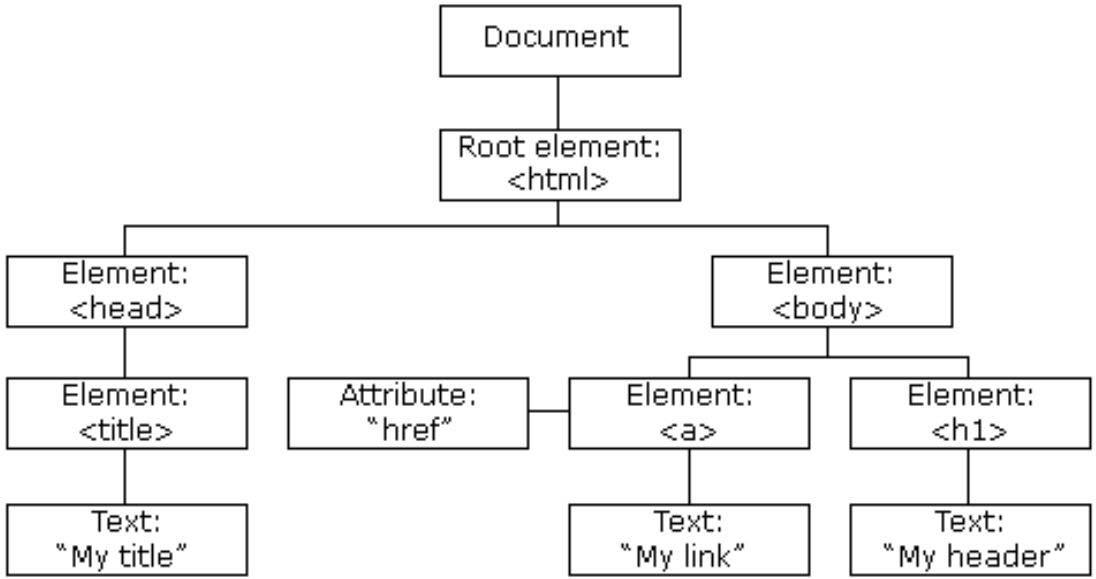
The HTML DOM

- HTML, CSS and JavaScript are the three main components that make almost every website. How do browsers comprehend our code?

- When they receive code, browsers build websites by converting the HTML they receive into a JavaScript object called the Document Object Model (DOM)

- In fact, from the Developer Tools console, you can examine the DOM for any website

THE HTML DOM

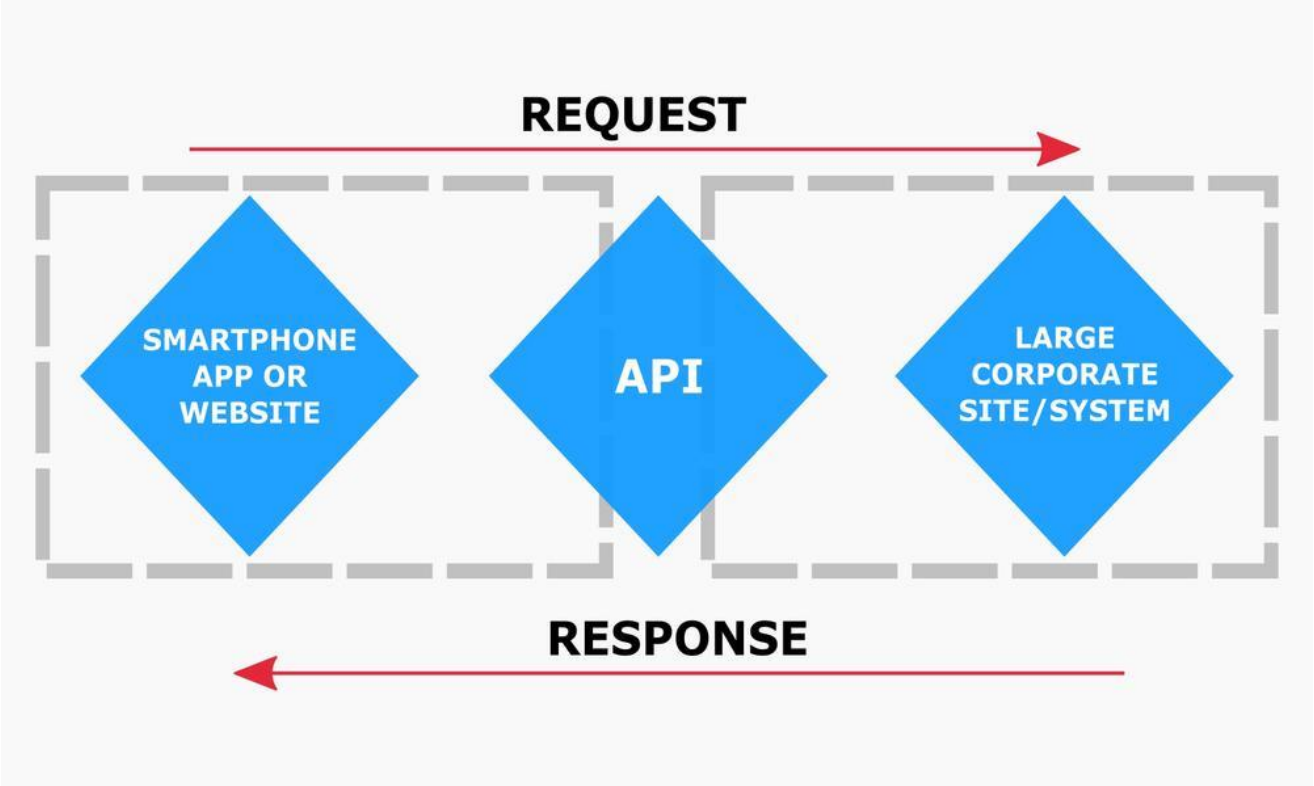


The DOM is demonstrated by the node-tree visual representation in your developer tools, identifying relevant attributes (and their respective values), CSS styles, JavaScript event listeners, breakpoints, and properties of each HTML element.

What is an Application program interface (API)?

- The web is made up of a large network of servers, all connected together. Every web page on the internet is stored on a remote server. A remote server is part of a remotely located computer that is “optimized” to process requests
- When you type www.facebook.com into your browser, a request goes to Facebook’s remote server. Once your browser receives the response, it interprets the code and displays the page
- To the browser, also known as the *client*, Facebook’s server is an API. This means that every time you visit a page on the Web, you interact with some remote server’s API
- An API isn’t the same as the remote server—rather **it is the part of the server that receives requests and sends responses**.

Or to put it another way...



Working with company APIs

What are some good APIs for us to try out on our websites?

Let's go with the well-established, public facing APIs of some big tech cos:

- Twitter for Websites
- Google Maps
- Facebook Social Sharing Plugins
- Facebook Graph API

As Developers, we love tools that make our lives easier, so let's get a useful tool for interacting and testing interactions with APIs. Postman is a Chrome extension that does just what we need it to.

Some companies provide their own consoles, which allow us to try out HTTP requests in browser too; pretty handy!

Interacting with APIs

- The DOM is of the utmost importance, next to the ability to make requests to a server (XMLHTTPRequest*) when we're thinking about interacting with APIs, because they come together to set the standards for **AJAX (Asynchronous JavaScript and XML*)**

- AJAX is a technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script

- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

What are REST & AJAX? How are they different?

- To fully understand AJAX, we need to understand how information on most websites are communicated over the standard web protocol, HTTP (HTTPS is similar with a layer of security) - using **REST (Representational State Transfer)**
- AJAX is not the same as REST

- AJAX is a programming interface that allows us to implement a set of client-side data handling techniques to **retrieve and access data** from a server, whereas

- REST is an architectural style - it is a **standard way** of handling, sending and responding to **HTTP (Hyper-Text Transfer Protocol)** requests

JSON

- So now we know (roughly)
    - how clients fetch data (AJAX), and
    - how applications are built on the server-side to listen for our requests for data (REST architectures), so…

how is the data received?

JSON request example (from Facebook)

- This Facebook example demonstrates how their Graph API is structured: This GET request:

```
GET graph.facebook.com
/facebook/picture?
redirect=false
```

Returns this JSON:

```
{
  data:
  {
    is_silhouette: false,
    url: "https://scontent.xx.fbcdn.net/v/t1.0-1/p100x100/12006203_10154088276211729_2432197377106462187_n.png?oh=d1b6b18e1846c1adefd157a45c4d384d&oe=58226457"
  }
}
```

Facebook's Graph API was built to handle requests with specific parameters, in order to return their logo to us in JSON form

Homework

Finishing off

Task:

Add plugins and metrics to your project.

Watch this video and this video to consolidate your knowledge of RESTful web APIs and how to use them.

Refresh your memory on AJAX and how it is used in websites and web applications to interact with APIs with this video.

Homework continued...

Group Project

Task:

Meet outside of class to work on your project!

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

(link to document including suggesting for working with Bootstrap, JavaScript and jQuery, drop-down menus, pop-up boxes, image slideshows / carousels, a form to send an email, hover effects, and more!)

Further resources...

- The HTML Document Object Model video by Udacity
- What is the DOM? By Chris Coyier on CSS Tricks
- IBM's more in-depth guide to AJAX and REST
- StackOverflow: AJAX IS NOT REST Ep.1, Ep.2
- jQuery AJAX functions!
- Learn about AJAX in Vanilla (pure) JavaScript from W3 Schools
- Facebook's Graph API Explorer
- Instagram's API - it's more advanced