Web Frameworks

1: Define what a framework is

> *The framework controls how you can create an application or website, a library is something you control*
>
> Inversion of Control is a key part of what makes a framework different to a library. A library is essentially a set of functions that you can call, these days usually organized into classes. Each call does some work and returns control to the client.
>
> A framework embodies some abstract design, with more behavior built in. In order to use it you need to insert your behavior into various places in the framework either by subclassing or by plugging in your own classes. The framework's code then calls your code at these points. (https://martinfowler.com/bliki/InversionOfControl.html)

2: Why do we use frameworks and libraries?

We use frameworks to make our life easier. It allows to us to create applications with a rapid development approach. When we use frameworks like **bootstrap** a lot of the work is already done for us, but unlike a library - it also makes our websites look and conform to certain way.

hover.css (css) and lodash (javascript) are examples of librries where they help you out by making your coding life easier, but they do not control what your application **should** look like - you still have that freedom.

Jquery is a javascript library that came out in 2008; This supported many of the features that are now supported in vanilla javascript since the release of es6 (es2015)

So as any piece of software, frameworks and libraries need to be maintained and updated. Currently Bootstrap is at today is at version 4.1.3 (August 2018)

3: Choosing the right framework

Even though frameworks (and libraries) do a lot of the work for us, you still need to be able to read and write code.

In this paper we focus mainly on the design of a site, not the functionality or the data transfer. So our focus lies at HTML and CSS. Most of the frameworks do have javascript components in them, but we do not need them for prototyping a website, which is what we would be doing at this stage. Although knowledge of Javascript wouldn't hurt, it is not strictly required.

So what is the right framework to use and why?

There is no right answer to this question. The framework (if one is needed) might be different for each job that you tackle. Just because you know of a framework doesn't mean it is the right one or the go to framework for each project.

Your project design (framework) shouldn't be limited by what you know, your the design should be evaluated from scratch every single time

There are many frameworks that do the same or similar things:

- Bootrap
- Material Design (Google look)
- Zurb
- Find more here

4: Installing a framework

Most frameworks have a CDN (Content Delivery Network) setup and you can use the framework using that option. This is great if you want to create quick mockups and play around with a framework to try it out.

Pros	Cons
Quick to use	Location is outside of your project
Nothing to install	Can be slow to load
	Your browser will priorities it with a list of resources that is matched with the other resources it needs to load

Another option and better for apps / websites that end up in prodcution is to download the libraries into your project, so that they are bundled with your application and not relying on a remote resource.

Pros	Cons
Library / Framework is bundled with your application	Takes a little longer to set up
Control the version that you can use	Need to install Node to use a package manager (most developers do this)

For example let's look at bootstrap:

Bootstrap has CSS links and Javascript links:

EXAMPLE 1: Links from external source

Add this code to the `<head>` tag

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min
.css" integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
```

Add this code just above the closing body tag (`</body>`)

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.mi
n.js" integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.j
s" integrity="sha384-
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
crossorigin="anonymous"></script>
```

EXAMPLE 2: Links from internal source

Assuming you have nodejs installed do this:

1: Create a file called `package.json` and add 2 curly braces in the file and save it. Then type this command in a terminal or git bash (on windows) after you have navigated to your working folder.

```
npm i bootstrap popper.js jquery
```

This will create a folder called `node_modules` where these files will be saved.

Next add these lines to your html file:

Add this code to the `<head>` tag

```
<link rel="stylesheet"
href="node_modules/bootstrap/dist/css/bootstrap.min.css">
```

Add this code just above the closing body tag (`</body>`)

```
<script src="node_modules/jquery/dist/jquery.min.js"></script>
<script src="node_modules/popper.js/dist/umd/popper.min.js"></script>
<script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
```

Example 1 is much quicker to set up, but you would be in trouble if bootstrap (or anything you are using) would go down. Now note that with bootstrap this is highly unlikely, but don't ever rely on another server when your website or app is in production. In the end it is your product and therefore your reputation 😃

5: Using a framework

So now that a framework is installed we need to know how to use it. Frameworks generally have some (hopefully good) documentation associated with it. So you will need to know how to read it, if you need to find information.

There will be code examples with that documentation and so therefore you need to know how to read the code.

A lot of these frameworks a lot of CSS rules setup, and so we can reference to those in our HTML code. We add the names of the CSS rules into the `class=""` attribute in the HTML code.

For this example we refer again to the use of bootstrap.

A simple example is a button:

The documentation for buttons are:
https://getbootstrap.com/docs/4.1/components/buttons/

There you will see this code block:

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

We can see that the HTML code for the button is the same as normal, but in the **class** attribute there are the references to the bootstrap class => `class="btn btn-primary"`

We are referencing 2 css rules here, the first one is for generic button stuff and the second one is specific for the color.

There is no limit to the amount of css rules you can reference to as, but you need to separate them by a single space.

Every setup works like this, which makes it easy - your job as the developer is to know which class is needed when.

6: Customizing the given code

Customising is possible, but it is not always easy. It is recommended that when you want to change anything you create your own CSS Rule in your own stylesheet. So for example you might end up with something like this:

```
class="btn btn-primary my-button"
```

The reason for this is that most libraries are compiled from other languages (SASS, LESS and Javascript) and to find the source code for that isn't always easy or logical. On top of that if the version changes and your code is dependant on it - it may cost you a lot of time to bug fix the changes.