JavaScript Rd2

“[JavaScript] is simultaneously a simple, easy-to-use language that has broad appeal, *and* a complex and nuanced collection of language mechanics which without careful study will elude the true understanding of even the most seasoned JavaScript developers.”

-Kyle Simpson, *You Don’t Know JS*

Overview

1. ES6 quality of life features
    a. Modules
    b. Rest/Spread
    c. Template literals
2. Functional Programming in JS
3. A brief overview of the Document Object Model (DOM)
    a. HTML parsing and script loading
    b. Basic DOM manipulation
4. Assignment #2

ES6 or ECMAScript 2015

- Introduced a bunch of language features. Comprehensive list and demo is here: http://es6-features.org/
- Most useful to get used to in this course
  a. Templates using `` syntax
  b. Modules - Import/Export
  c. Rest, Destructuring, Spread Syntax, Object rename
  d. Arrow functions

ES6 Templates

Uses backticks to make string manipulation simpler

```
`Hello World` // As per normal ‘Hello World’

Say we’ve set name here as js variable as ‘Alex’

`Hello ${name}` // ‘Hello Alex’

// Can go a step further and even do evaluation

`Hello ${20 * 8}` // `Hello 160`
```

ES6 Modules

Importantly solve some pretty classic headaches for JS beginners.

- Script load timing agnostic
- Avoids namespace pollution
- Makes code modular! yay.

ES6 Modules

Introduced the keywords import, export.

```
import { namedVar } from ‘./a-relative-path’;

import defaultExport from ‘./a-relative-path’;
```

And the otherside of the equation:

```
export { namedVar };

export default someDefaultExport;
```

ES6 Destructuring, Rest, Spread Syntax

- Timesaving, intuitive syntax
- Saves repetitive renaming
- Allows much cleaner Object and Array manipulation

ES6 Rest/Spread/Destructure (Demo)

```
// destructuring. Can be done in function args too!
const { x, y } = obj // extracts key-values x, y

// Array destructure pretty much the same
const [ x, y ] = list // extracts key-values x, y

Spread syntax (rest in Demo)

const someList = [1, 2, 3, 4, 5]
const evenLongerList = [...someList, 6, 7]
```

Thinking Functionally in JS

Functions in JavaScript are ‘first class’ objects; that is they can be more or less treated in the same way as variables, passed around, and even returned by other functions. This isn't a unique trait of JavaScript but it does allow you to think and program differently than you might otherwise be used to.

Thinking Functionally in JS

Imperative

Like a recipe. An instruction set for achieving a result focused on the process.

Declarative (Functional)

Description of the logic of a program rather than the way the logic is implemented.

.map, .filter, .reduce and arrows.

Key things:

- Get rid of ‘global’ variables, state.
- Get rid of side effects
- Aim for function purity - we can reason easily about what a function does because it always does the same thing with the same input.
- Much more ‘expressive’ way to code.

Functional Example.

Browser Object Model (BOM)

- Abstraction over the browser.
- Browser provides a normalised API which allows JS to do I/O calls like writing, reading to the document from the interface (we'll get to this).
- Window is also populated with:
  - Browser specific API globals (location, navigation, fetch, localStorage...).
  - JavaScript globals
  - Provides the context environment for your scripts to run in.

Document Object Model (DOM)

- Abstraction over the document.
- Provides the page interface that we can update our html with and create dynamic pages.
- DOM is a tree-like structure. There is a root node which has children HTML elements/nodes.
- More on this in future weeks.

Document Object Model (DOM)

EXAMPLE!

The Assignment - ‘Instacram’

AIM: Build an image driven social media ‘Single Page Application’ (SPA) using all your sweet frontend web development skills.

We don’t expect you to know how to do that by any means yet.

- We’ll provide a bit of code and hopefully enough guidance to get through it okay.
- There will be progressively more difficult milestones, but the base project will be very achievable
- We will provide an API which you’ll need to interact with; more info on this soon. This is very ‘real world’.

The Assignment

Things to consider:

- Think about how to separate your API logic from your DOM manipulation/render logic
- Think about how to manage the state of a user (simple and more complex solutions)
- Read up on how the API works (when it’s released), and think about what data you’ll need where, and how to combine relevant bits of data together.

The Assignment

Finally, some rules...

- This is very much an individual assignment!
- You must use nothing but vanilla JavaScript, HTML and CSS. Please do not submit bundles or precompiled JavaScript.
- Do not use a framework like React, Vue, Angular.. etc!
- Despite us providing a little bit of node.js knowledge in this course we don’t want you using npm packages to supplement your code.
- You can if you like you snippets of code as long as attributed correctly.

Mostly though, try and have fun.