COMP6080: Web Front-End Programming

Tutorial 3

Sidney Pham
`sidney.pham@unsw.edu.au`

UNSW

Week 3, Term 3 2020

Administrivia

A friendly reminder that Assignment 1 is due this Sunday at 8pm.

Administrivia

A friendly reminder that Assignment 1 is due this Sunday at 8pm.

Additionally, make sure you know how to get your labs marked — submit by the deadline, then turn up to the next lab and enter the lab marking queue.

Administrivia

A friendly reminder that Assignment 1 is due this Sunday at 8pm.

Additionally, make sure you know how to get your labs marked — submit by the deadline, then turn up to the next lab and enter the lab marking queue. If you didn't do this for lab01, I'm happy to (as an exceptional one-off) mark you quickly if you email me.

Differences

1. What's the difference between Node.js and NPM?

Differences

1. What's the difference between Node.js and NPM?

   Node.js is a JavaScript runtime environment. It lets developers use JavaScript to write command line tools and for server-side scripting. NPM is a package manager to help manage Node.js projects and their dependencies.

Differences

1. What's the difference between Node.js and NPM?

   Node.js is a JavaScript runtime environment. It lets developers use JavaScript to write command line tools and for server-side scripting. NPM is a package manager to help manage Node.js projects and their dependencies.
2. What's the difference between NPM and Yarn?

Differences

1. What's the difference between Node.js and NPM?

   Node.js is a JavaScript runtime environment. It lets developers use JavaScript to write command line tools and for server-side scripting. NPM is a package manager to help manage Node.js projects and their dependencies.

2. What's the difference between NPM and Yarn?

   Generally very little to a novice user. You can do your research to learn the differences between them, but on small scale applications the differences are negligible.

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS imports usually go in our <`head`> tag, because we want to import styles prior to the page loading, to prevent HTML rendering without styles.

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS imports usually go in our `<head>` tag, because we want to import styles prior to the page loading, to prevent HTML rendering without styles.

What are some advantages to internal CSS (compared to external CSS)?

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS imports usually go in our <`head`> tag, because we want to import styles prior to the page loading, to prevent HTML rendering without styles.

What are some advantages to internal CSS (compared to external CSS)? It's easy, it only requires one network request, etc.

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS imports usually go in our `<head>` tag, because we want to import styles prior to the page loading, to prevent HTML rendering without styles.

What are some advantages to internal CSS (compared to external CSS)? It's easy, it only requires one network request, etc. What are some disadvantages?

CSS Imports

Where do we generally include CSS imports in our HTML?

CSS imports usually go in our <`head`> tag, because we want to import styles prior to the page loading, to prevent HTML rendering without styles.

What are some advantages to internal CSS (compared to external CSS)? It's easy, it only requires one network request, etc. What are some disadvantages? I don't think markup and styling are the same concern, so they should be separated. If I'm trying to change how my page looks, I should only have to primarily check one place — the CSS file.

Aside: Separation of Concerns

The Web used to work by separating HTML, CSS and JS into different files, under the idea of Separation of Concerns. Every webpage would've had three source files.

Aside: Separation of Concerns

The Web used to work by separating HTML, CSS and JS into different files, under the idea of Separation of Concerns. Every webpage would've had three source files. With the advent of technologies like React, we began to consider our concerns by *components* not *markup / styling / dynamic behaviour*.

Aside: Separation of Concerns

The Web used to work by separating HTML, CSS and JS into different files, under the idea of Separation of Concerns. Every webpage would've had three source files. With the advent of technologies like React, we began to consider our concerns by *components* not *markup* / *styling* / *dynamic behaviour*.

That is, you'll commonly find HTML and JS found together — I think this is a *good* thing because the DOM (a JS concept) is so closely tied with HTML.

Aside: Separation of Concerns

The Web used to work by separating HTML, CSS and JS into different files, under the idea of Separation of Concerns. Every webpage would've had three source files. With the advent of technologies like React, we began to consider our concerns by *components* not *markup / styling / dynamic behaviour*.

That is, you'll commonly find HTML and JS found together — I think this is a *good* thing because the DOM (a JS concept) is so closely tied with HTML. Additionally, structuring files by component enables easy reuability.

Aside: Separation of Concerns

The Web used to work by separating HTML, CSS and JS into different files, under the idea of Separation of Concerns. Every webpage would've had three source files. With the advent of technologies like React, we began to consider our concerns by *components* not *markup* / *styling* / *dynamic behaviour*.

That is, you'll commonly find HTML and JS found together — I think this is a *good* thing because the DOM (a JS concept) is so closely tied with HTML. Additionally, structuring files by component enables easy reuability.

However, some people have had particularly zealous opinions for and against: see this thread and this talk.

JavaScript Imports

Where do we generally include JS imports in our HTML?

JavaScript Imports

Where do we generally include JS imports in our HTML?

JavaScript imports usually go at the bottom of our <`body`> tag. This is to ensure that the default elements in the DOM fully load before our JavaScript (which generally interacts with the DOM) is run.

JavaScript Imports

Where do we generally include JS imports in our HTML?

JavaScript imports usually go at the bottom of our `<body>` tag. This is to ensure that the default elements in the DOM fully load before our JavaScript (which generally interacts with the DOM) is run.

However, another solution is to use the `defer` attribute on the `<script>` tag (see: this article).

Question

Why do we use `className` in React, but `class` in vanilla JavaScript?

Question

Why do we use `className` in React, but `class` in vanilla JavaScript?

> *`className` is the JSX component property that equates to `class`. It's transpiled to `class`.*

Question

Why do we use `className` in React, but `class` in vanilla JavaScript?

> *`className` is the JSX component property that equates to `class`. It's transpiled to `class`.*

Really, it's because `class` is a reserved keyword in JavaScript, and JSX is an extension upon JavaScript.

Exercise: Interactive Building Art

The file `building.html` contains some HTML and CSS, which we are going to use to create some interactive art. However, we will not be editing `building.html` directly — instead we will use JavaScript and the DOM API in the script `building.js` to manipulate the DOM.

Task 1: Looking at the Code

Firstly, before rendering the HTML file, what can you tell about the page? What sorts of elements does the HTML body contain? What do you notice about the CSS?

Task 1: Looking at the Code

Firstly, before rendering the HTML file, what can you tell about the page? What sorts of elements does the HTML body contain? What do you notice about the CSS?

> *The body only contains two elements: one is a `div` tag with the `class` building, the other is a `script` tag. When the HTML renders it will execute whatever code is in building.js. The CSS not only contains the regular styling for body, elements with `ID` building and elements with `class` window, but it also seems to define styling for these elements in some 'night mode'.*

CSS Selectors

The night mode styling works using more advanced CSS selectors (specifically, *attribute selectors*).

CSS Selectors

The night mode styling works using more advanced CSS selectors (specifically, *attribute selectors*).

They're pretty easy to understand: see MDN or a good cheatsheet.

CSS Selectors

The night mode styling works using more advanced CSS selectors (specifically, *attribute selectors*).

They're pretty easy to understand: see MDN or a good cheatsheet.

BTW, the Emmet abbreviations are based on CSS selectors, so learning them will help you write your HTML more quickly!

Task 2: Adding Windows

Using only JavaScript and the DOM API, add 9 square windows (with `class window`) to the building. The windows should be 50×50px, with a margin of 25px.

Task 3: Dynamic Windows

Now, add a keyboard shortcut that will add a window when the up (`ArrowUp`) button is pressed, and remove a window when the down (`ArrowDown`) button is pressed.

Task 3: Dynamic Windows

Now, add a keyboard shortcut that will add a window when the up (`ArrowUp`) button is pressed, and remove a window when the down (`ArrowDown`) button is pressed.

There are three KeyboardEvent events: `keydown`, `keyup` and `keypress` (deprecated!). I suggest defaulting to `keydown`.

Task 4: Dynamic Building

Add another keyboard shortcut that will move the building left/right by 50px when the left/right buttons are pressed.

Task 5: Toggle-able Night Mode

Add an event handler that will toggle on/off night mode when the user clicks anywhere on the screen.