COURSEWARE

Introduction

Contents

Overview

React is the most popular front-end JavaScript library for building web applications, with the aim for creating *reactive* user interfaces.

It is developed by **Facebook**, and written entirely using **JavaScript**.

Amongst its competitors are:

- **Google**'s front-end web development framework, **Angular**, which is written in a combination of **JavaScript** and the extensionist language **TypeScript**.
- **Vue.js**, created by **Evan You**, which focuses on expanding HTML through specialised attributes called *directives*.
- There are others with a far smaller usage share, such as **Backbone.js**, **Ember** and **Svelte**.

Advantages

Modularity

React aims to make responsive web applications which can quickly change and adapt to the data provided to them.

It does this through utilising *states* and *reusable components*, allowing the developer to blueprint certain aspects of their application to use and re-use again:

- **states** keep track of what is happening within a certain component at any given time
- **components** split entire web pages (e.g. a home page) into different sections according to their functionality (e.g. a carousel)

By following this modular setup, React components can effectively correspond to the *microservices* utilised within the backend of a web development stack.

The Virtual DOM

React uses a virtual version of the **Document Object Model (DOM)** used in **HTML**, which represents the layout of the web page.

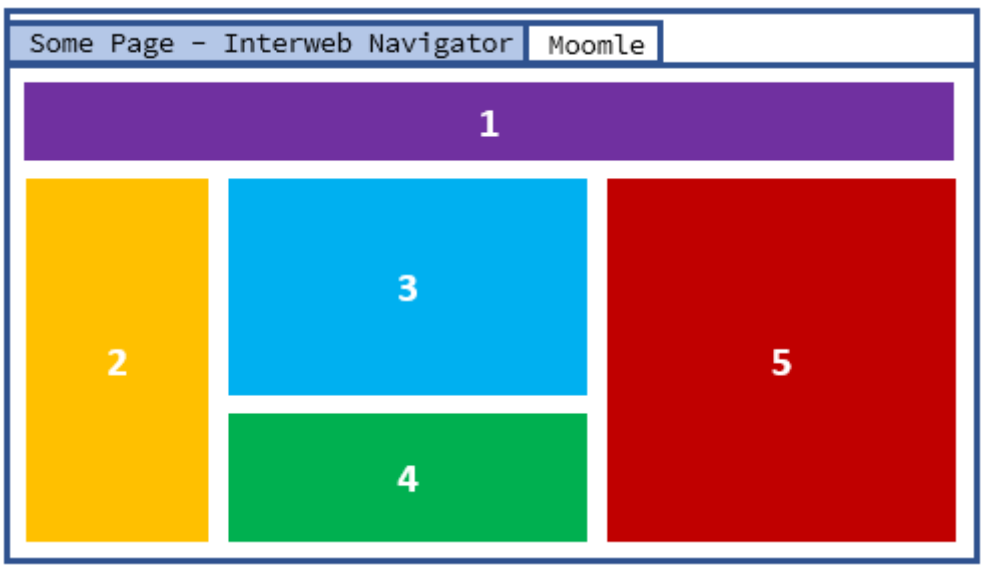
Traditionally, web pages would update everything at once, usually refreshing the page, if something changed within the page's DOM, which is incredibly inefficient, particularly on larger pages.

React's solution to this issue is to only update the DOM when the *state* of a *component* changes.

Instead of reloading the entire page, React adopts a *lightweight* approach: the React virtual DOM on the *server-side* (the developer's end) figures out the smallest number of changes necessary to update the DOM on the *client-side* (the user's end) to match it, then updates only those specific areas.

React UI

As mentioned above, the React UI is split into *components*:



Each number represents a different *child* component of the entire Web page, which is a *parent* component.

This allows for far greater modularity when creating web pages.

React, and therefore each component, is written in JavaScript, which might also contain:

- JSX - an extension to JavaScript that defines what HTML the component contains
- CSS - any relevant styling for the HTML

An example component might look like this:

```
import { Component } from "react";

const Styles = styled.div`
  .p {
    background-color: #222;
    font-size: 15px;
  }
`;

class SomeComponent extends Component {
  render() {
    return(
      <div>
      <h2>Henlo frens</h2>
      <p>It me</p>
      </div>
    );
  }
}

export default SomeComponent;
```

The entire page is written in JavaScript, which then defines two things within it:

- the `Styles` constant is supplying the CSS for the paragraph (`p`) element
- the `return()` function is supplying the JSX to the `render()` function, which converts it into HTML

React Tooling

React requires two external tools to work:

- **Node Package Manager (NPM)**, the repository which both React and other JavaScript libraries are stored in
- **Node Package Executor (NPX)**, the package executor for NPM which allows us to run a React project

Tutorial

Create a basic React Application

*Ensure that a recent version of **Node.js** is installed first.*

Open a command line on the Desktop, then run the following command:

```
npx create-react-app <SOME_NAME_HERE>
```

This should run the `create-react-app` tool, which will set up a new React project for you with no configuration needed.

Next, enter the following command:

```
npm start
```

This will open up a development server at `localhost:3000` with the sample React application you just created running on it:

Exercises

Edit App.js

In this Exercise, you should edit your App.js page, using the general layout from the example in the module, to render some HTML on the page with some CSS styling.