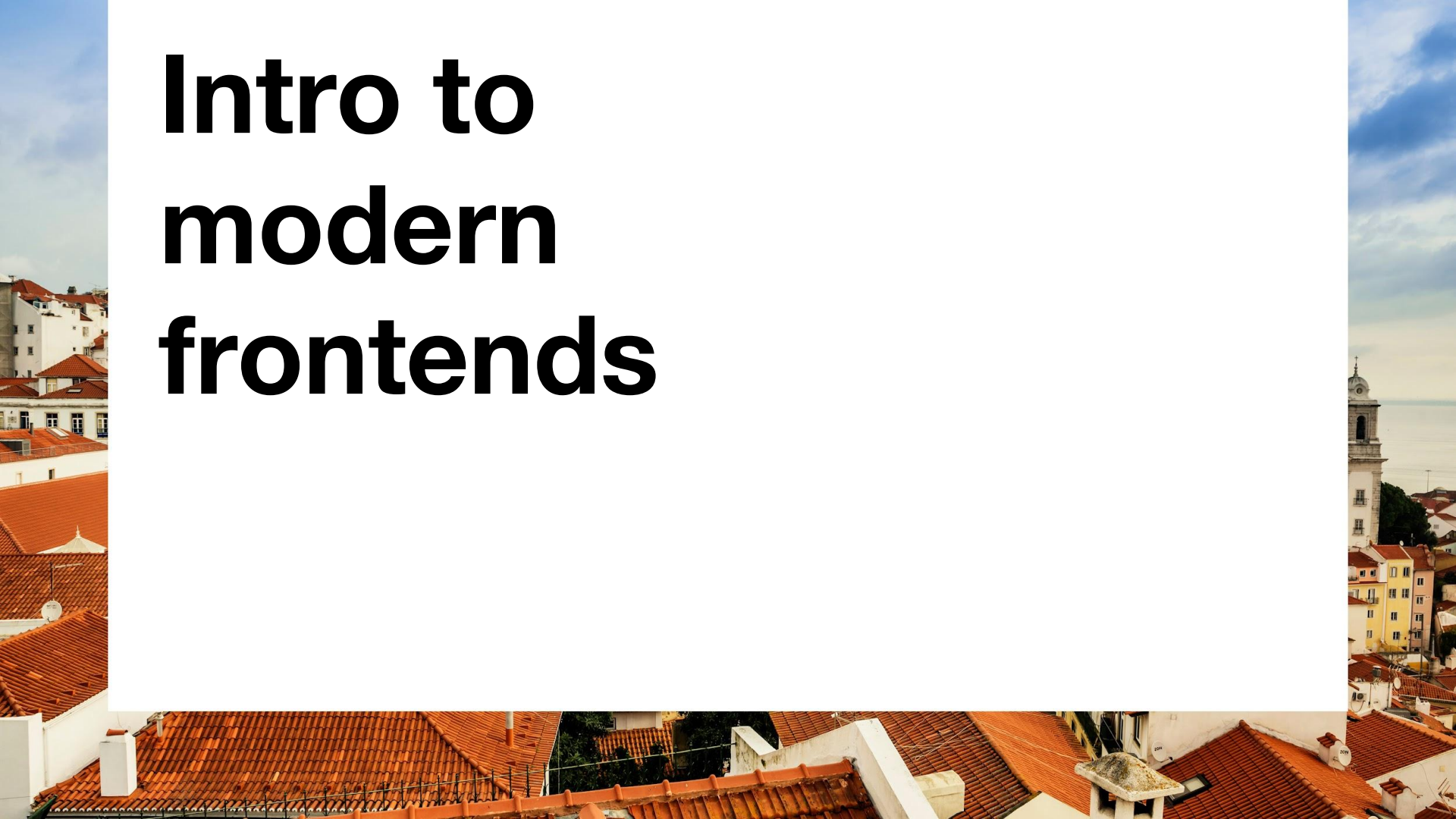
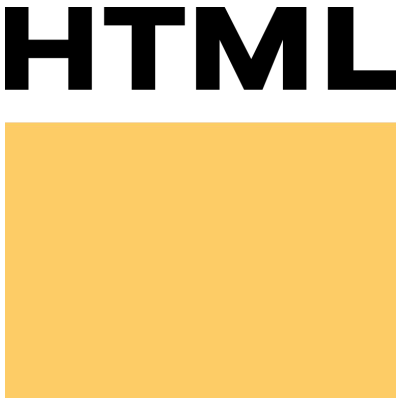
Intro to modern frontends



Part I: A brief history of frontend technologies

HTML/CSS/JS/AJAX

Pre-1999



HTML (1993)

- Standard bringing structure to data
- Very basic functionality
- HTML elements can have classes, properties and IDs
- Updated a lot with HTML5 specification (2008)

CSS (1996)

- Adds styling declaration to HTML
- Works by targeting HTML elements, mainly by their classes or IDs
- Updated with CSS3

JavaScript (1995)

- Brings interactive functionality to HTML (and CSS)
- Runs in browser (and more)
- Regulated and updated by the ECMAScript specification

Example

HTML

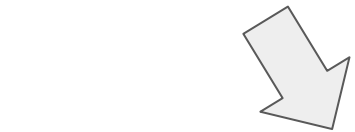
```html
<div class="alert">
  <span class="closebtn">&times;</span>
  <strong>Danger!</strong> Indicates a dangerous or potentially negative action.
</div>

<div class="alert success" aria-label="success-alert" >
  <span class="closebtn">&times;</span>
  <strong id="successId">Success!</strong> Indicates a successful or positive action.
</div>
```

CSS

```css
.alert {
  padding: 20px;
  background-color: #f44336;
  color: white;
  opacity: 1;
  transition: opacity 0.6s;
  margin-bottom: 15px;
}

.alert.success {
  background-color: #04AA6D;
}

#successId {
  background-color: #FFFF00;
  color: black;
}
```

JavaScript

```javascript
<script>
var close = document.getElementsByClassName("closebtn");
var i;

for (i = 0; i < close.length; i++) {
  close[i].onclick = function(){
    var div = this.parentElement;
    div.style.opacity = "0";
    setTimeout(function(){ div.style.display = "none"; }, 600);
  }
}
</script>
```



Danger! Indicates a dangerous or potentially negative action. ×

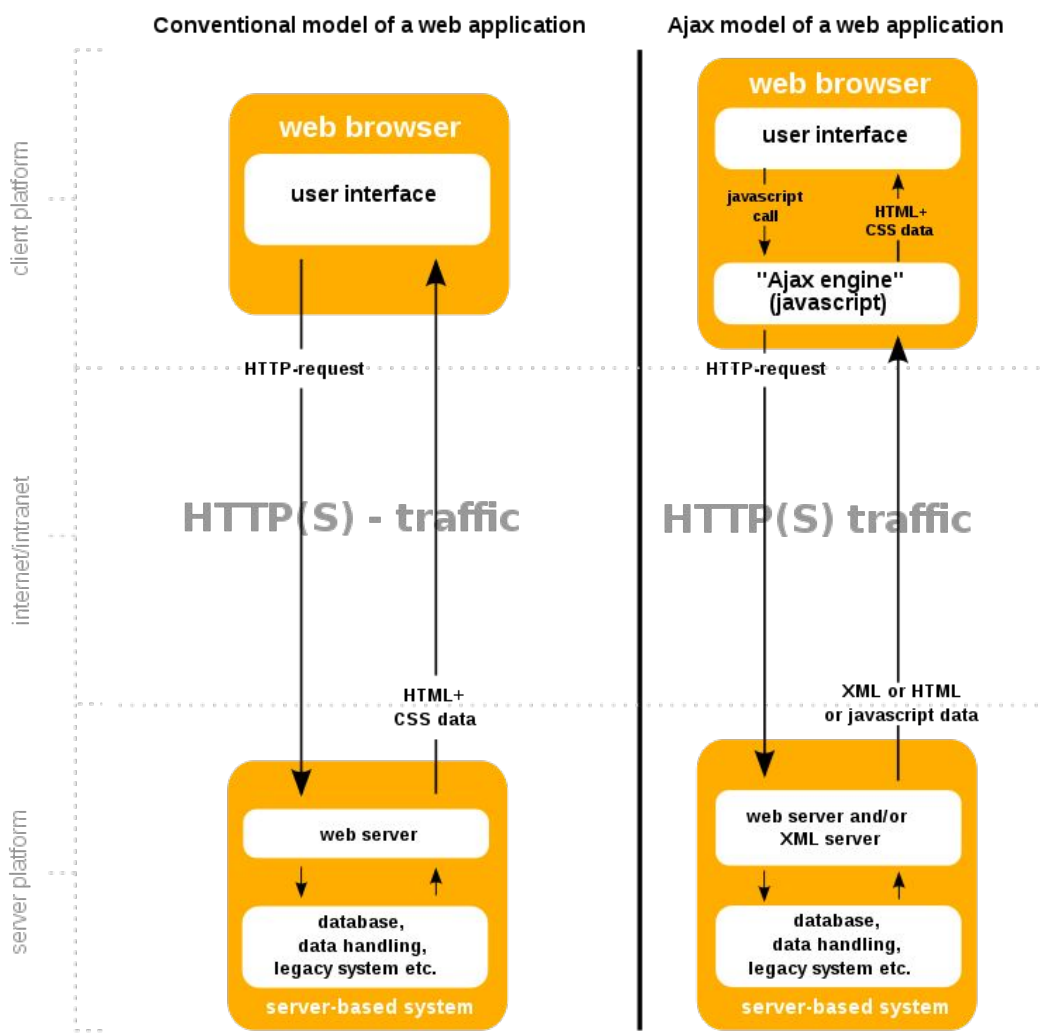
Success! Indicates a successful or positive action. ×

Demo 1

HTML + CSS + JS

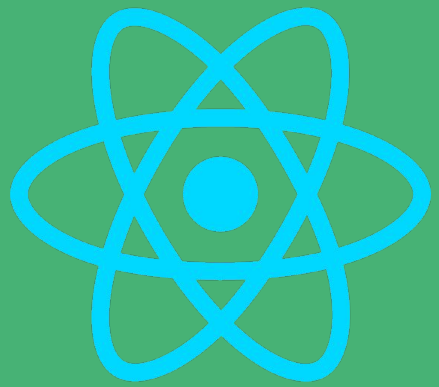
AJAX

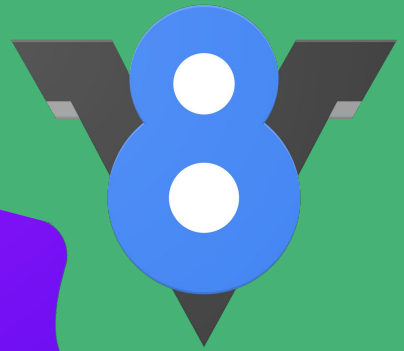
- Asynchronous JavaScript and XML
- Asynchronously update parts of a web page, without reloading the whole page.
- Non-blocking callback-based behaviour
- Can be used to transport plaintext, JSON etc, not only XML
- Nowadays Fetch API being used instead - same idea, better technology
- Idea of asynchronous operations still widely used



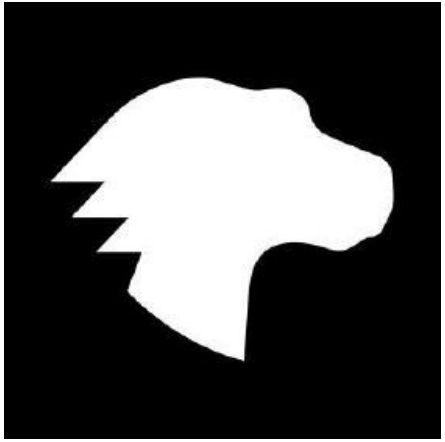
Part II: The modern days

Browser APIs/ES6/Node.js/React





Modern Browser APIs



DOM

- Represents a structure of a document (eg. an HTML file) in memory
- It is represented in a tree structure and allows for programmatic access to it
- Backbone of the web

Page Visibility API

- User has left the current tab, either by switching to another tab, window or application
- Can also detect when user returns to page

Graphics APIs

- Canvas API - 2D graphics (animation, games, data visualization, drawing, etc) rendering
- WebGL API - 2D and 3D graphics rendering
- Image capture API - capturing images from photographic device and editing them

Fetch API

- Replaces XMLHttpRequest object for asynchronous requests
- Better and uses modern paradigms, compared to previous methods

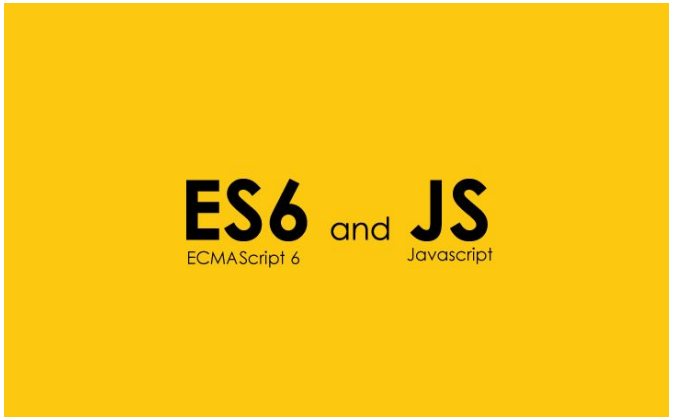
Service Workers API

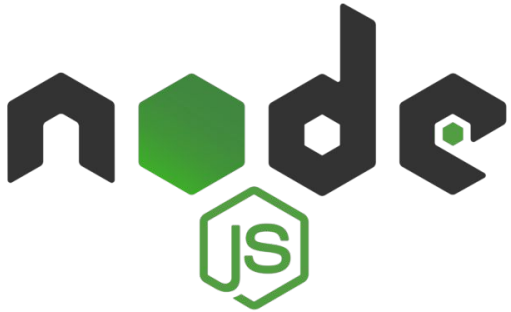
- Proxies that sit between web applications, the browser and/or the network
- Can intercept network requests and enable offline experiences
- Also implement push notifications and background sync functionality

Device APIs

- Geolocation API
- Sensor APIs - Accelerometer, AmbientLightSensor, Gyroscope, Magnetometer
- Gamepad API
- Bluetooth API

ECMAScript JavaScript & Node.js





ECMAScript JavaScript

- JavaScript standard introducing new features in the language
- Ensures compatibility and interoperability of web applications across all web browsers
- (Mostly) requires transpiling - source code rewriting to “dumb” JS
- ES6 (ECMAScript 2015) a major version: Introduces constants and variable scoping, arrow functions, intuitive OOP-style classes, module import/export mechanism and more
- ES7, ES8, … , ES12 (2021)

Node.js

- Server-side (backend) JavaScript
- Based on Chrome’s V8 JavaScript engine
- Aims to optimise for throughput and scalability of web applications
- Architected for asynchronous programming

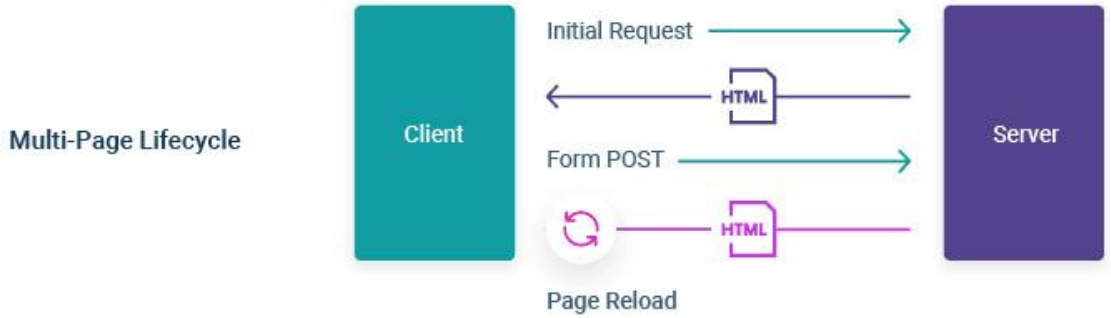
Single-page Applications (SPAs)

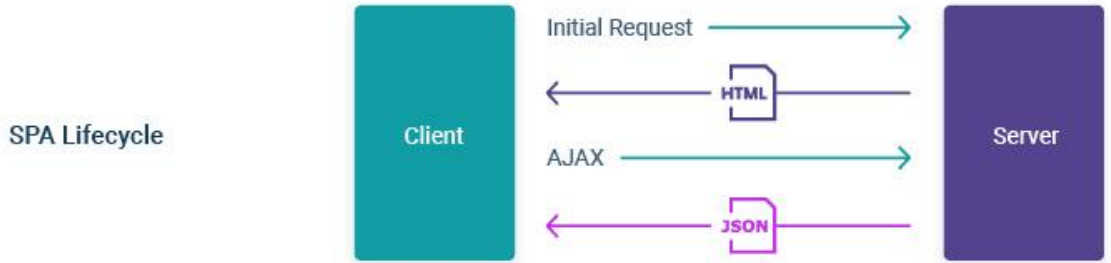
Multi-page applications:

- Web server responds with full “brand-new” HTML pages
- A browser refresh occurs when a new page is loaded
- Good for search engine optimization (SEO) - crawlers can index individual pages

Single-page applications:

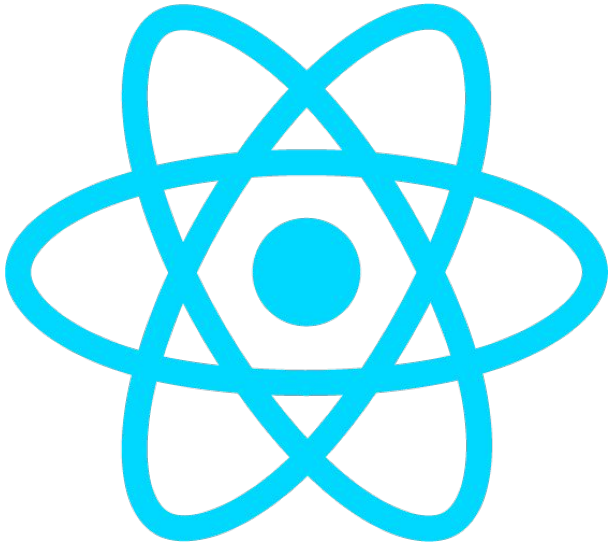
- Dynamically rewrite the current web page based on new data coming from server
- A browser refresh is not triggered
- Difficult for SEO
- ...but apps are faster, more responsive and feel native (and code can be reused for mobile apps)





React

- SPA framework by Facebook
- Efficient use of a virtual DOM
- Web applications consist of JSX components with:
  - Hierarchical structure
  - One-way data flow - top down, using props
  - State or lack of state



```
class Main extends Component {
 constructor() {
   super()
   this.state = {
     books: []
   }
 }
 render() {
   <BooksList books={this.state.books} />
 }
}
```

```
const BooksList = ({books}) => {
 return (
   <ul>
     {books.map(book => {
       return <li>book</li>
     })}
   </ul>
 )
}
```

UI CSS frameworks, component libraries and toolkits

- Writing CSS from scratch is tedious
- That's why there are a lot of frameworks that offer pre-defined styles which form components
- Component libraries help enforce uniform design and quick prototyping
- Component toolkits also sometimes implement the design language of a company: Base Web (Uber), Material Design (Google), Carbon Components (IBM)
- Bootstrap is the most famous CSS components and templates library in the world

Demo 2

Cats and Dogs with Bootstrap

Useful links

Thinking in React:
https://reactjs.org/docs/thinking-in-react.html

CSS Selectors reference:

https://www.w3schools.com/cssref/css_selectors.asp

ES6 Features:

http://es6-features.org/

Fetch API docs:

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

React-Bootstrap:

https://react-bootstrap.github.io/

Demos:

https://codesandbox.io/s/table-html-css-js-vr7iw

https://codesandbox.io/s/table-bootstrap-3vyvk

“Homework”

Re-introduction to JavaScript (general syntax info):
https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript

Intro to React (all Main Concepts sections recommended): https://reactjs.org/docs/hello-world.html

ES6 features (at least Extended Parameter Handling, Scoping, Template Literals, Destructuring Assignment):

http://es6-features.org/