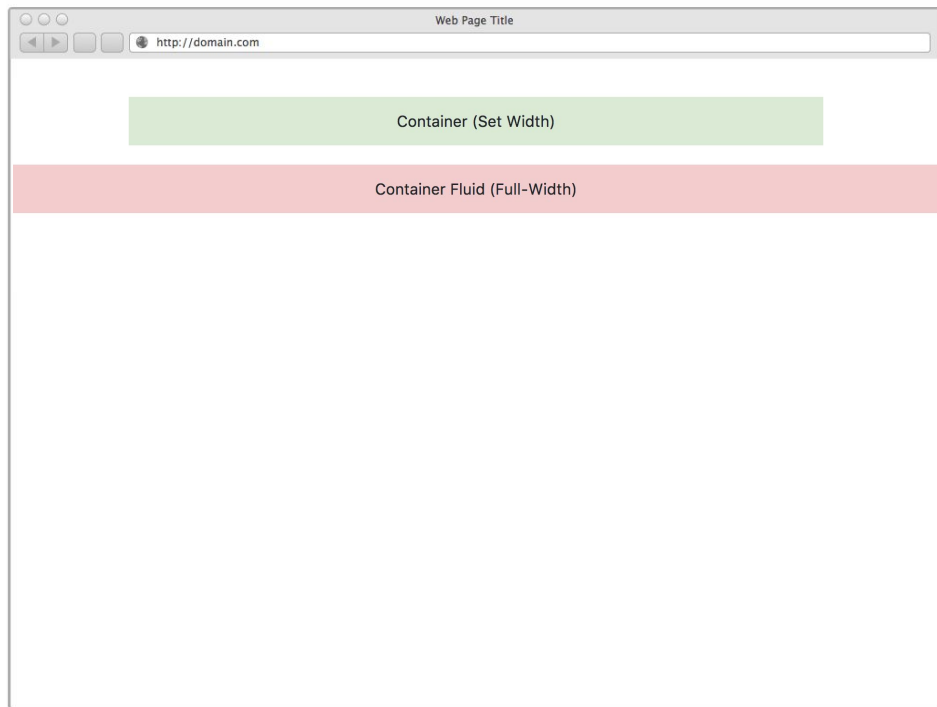


Lecture 06: Bootstrap continued, Intro to JavaScript

ITP 303 Full-Stack Web Development

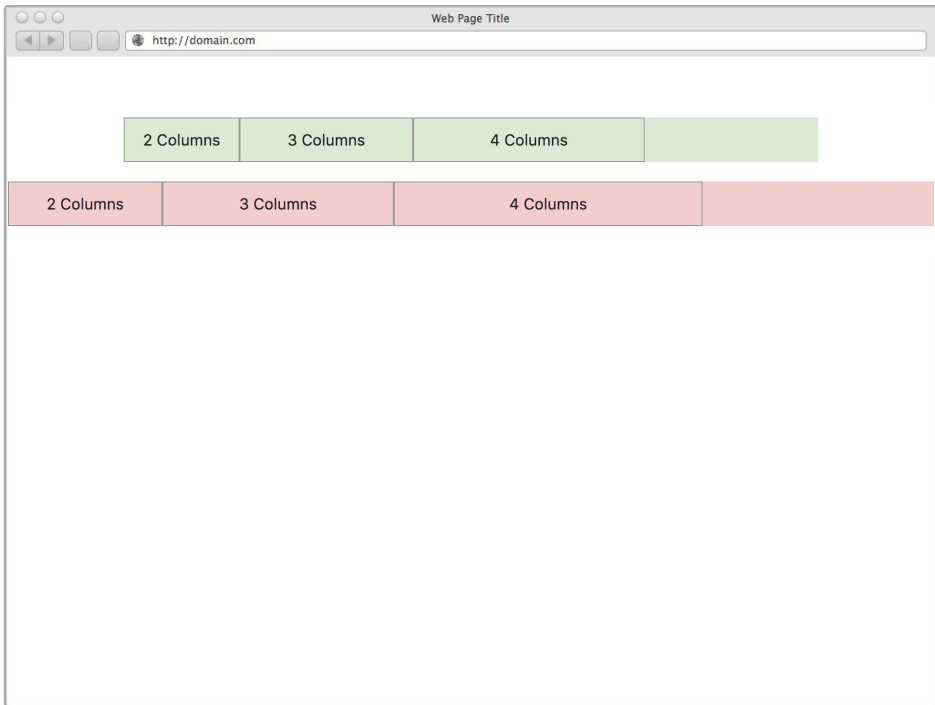
Bootstrap: Grid Structure



```
<div class="container">  
  Container (Set Width)  
</div> <!-- .container -->
```

```
<div class="container-fluid">  
  Container Fluid (Full-Width)  
</div> <!-- .container-fluid -->
```

Bootstrap: Grid Structure



```
<div class="container">
  <div class="row">
    <div class="col-2">2 Columns</div>
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
  </div> <!-- .row -->
</div> <!-- .container -->
```

```
<div class="container-fluid">
  <div class="row">
    <div class="col-2">2 Columns</div>
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
  </div> <!-- .row -->
</div> <!-- .container-fluid -->
```

Bootstrap: Grid Structure



```
<div class="container">
  <div class="row">
    <div class="col-2">2 Columns</div>
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
  </div> <!-- .row -->
  <div class="row">
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
    <div class="col-5">5 Columns</div>
  </div> <!-- .row -->
</div> <!-- .container -->
```

```
<div class="container-fluid">
  <div class="row">
    <div class="col-2">2 Columns</div>
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
  </div> <!-- .row -->
  <div class="row">
    <div class="col-3">3 Columns</div>
    <div class="col-4">4 Columns</div>
    <div class="col-5">5 Columns</div>
  </div> <!-- .row -->
</div> <!-- .container-fluid -->
```

Bootstrap: Breakpoints

Mobile-First Responsive Design.

Name	Size	Devices
Extra Small	$\leq 575\text{px}$	Portrait Phones
Small	576px - 767px	Landscape Phones
Medium	768px - 991px	Tablets
Large	992px - 1199px	Desktops
Extra Large	$\geq 1200\text{px}$	Large Desktops

```
/* Extra small devices (portrait phones, less than 576px) */  
/* No media query since this is the default in Bootstrap */
```

```
/* Small devices (landscape phones, 576px and up) */  
@media (min-width: 576px) { ... }
```

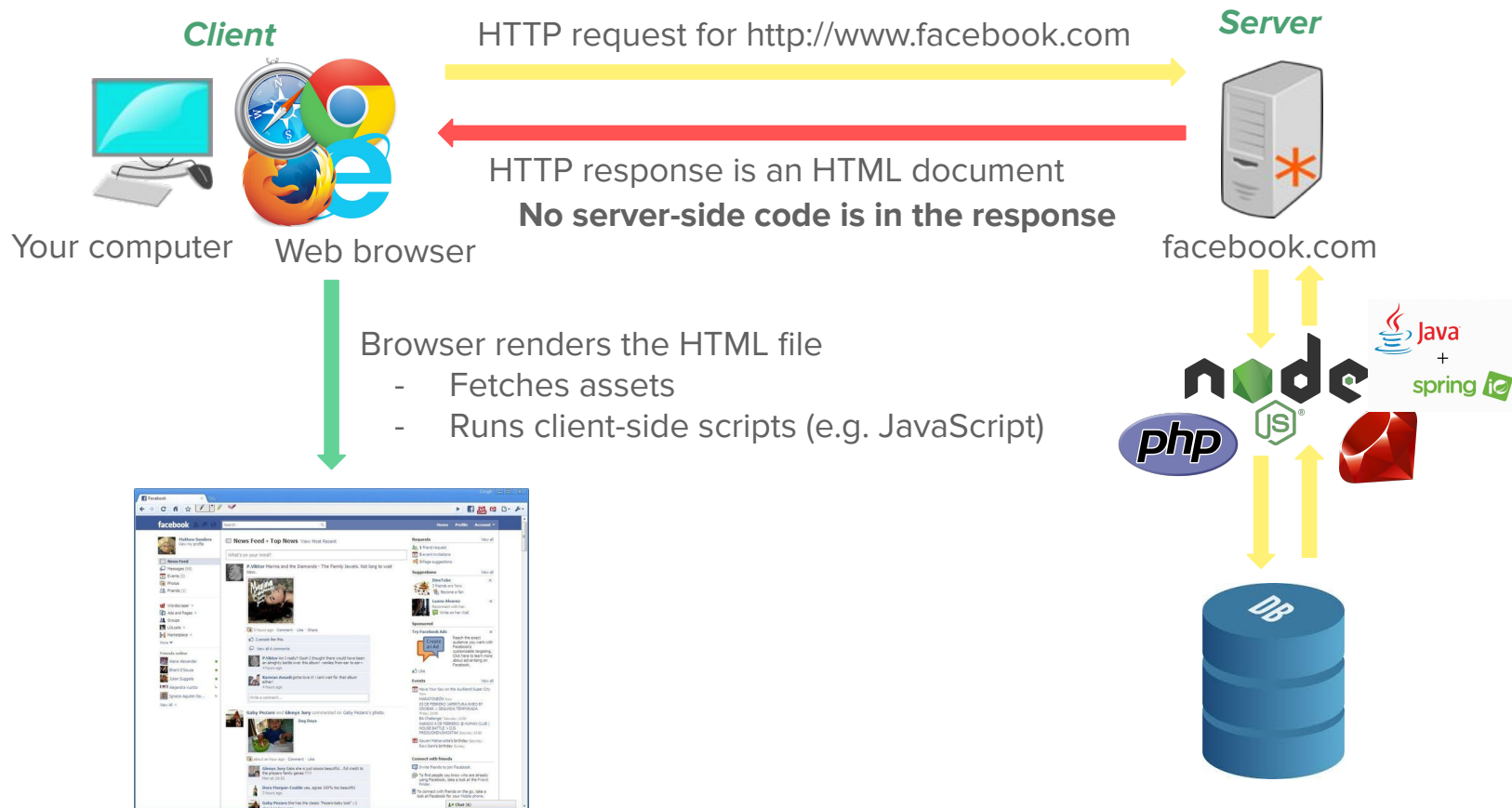
```
/* Medium devices (tablets, 768px and up) */  
@media (min-width: 768px) { ... }
```

```
/* Large devices (desktops, 992px and up) */  
@media (min-width: 992px) { ... }
```

```
/* Extra large devices (large desktops, 1200px and up) */  
@media (min-width: 1200px) { ... }
```

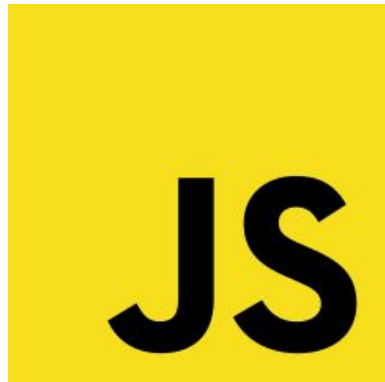
Intro to JavaScript

The journey so far...

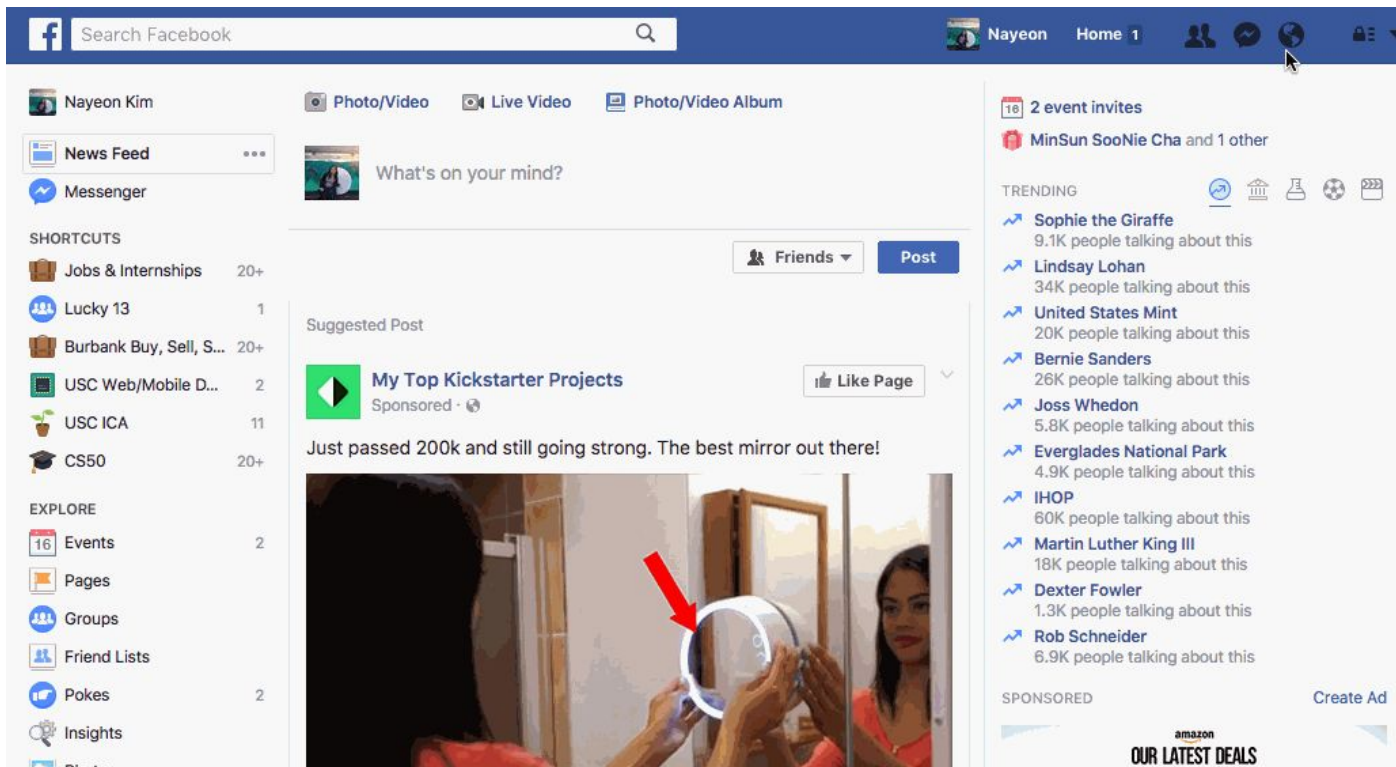


JavaScript (JS)

- A programming language that can be read and executed by the browser.
- While many ideas are borrowed from the language Java, **Java and JavaScript are two entirely different languages.**
- Primarily used on client-side, but with NodeJS on the backend, it's possible to build an entire web application with just JavaScript
- Some key things JS is good at:
 - DOM manipulation
 - Listening to user events (click, hover, keypress, etc)
 - Client-server communication without reloading pages
 - And more...



JavaScript in live action



JavaScript in live action

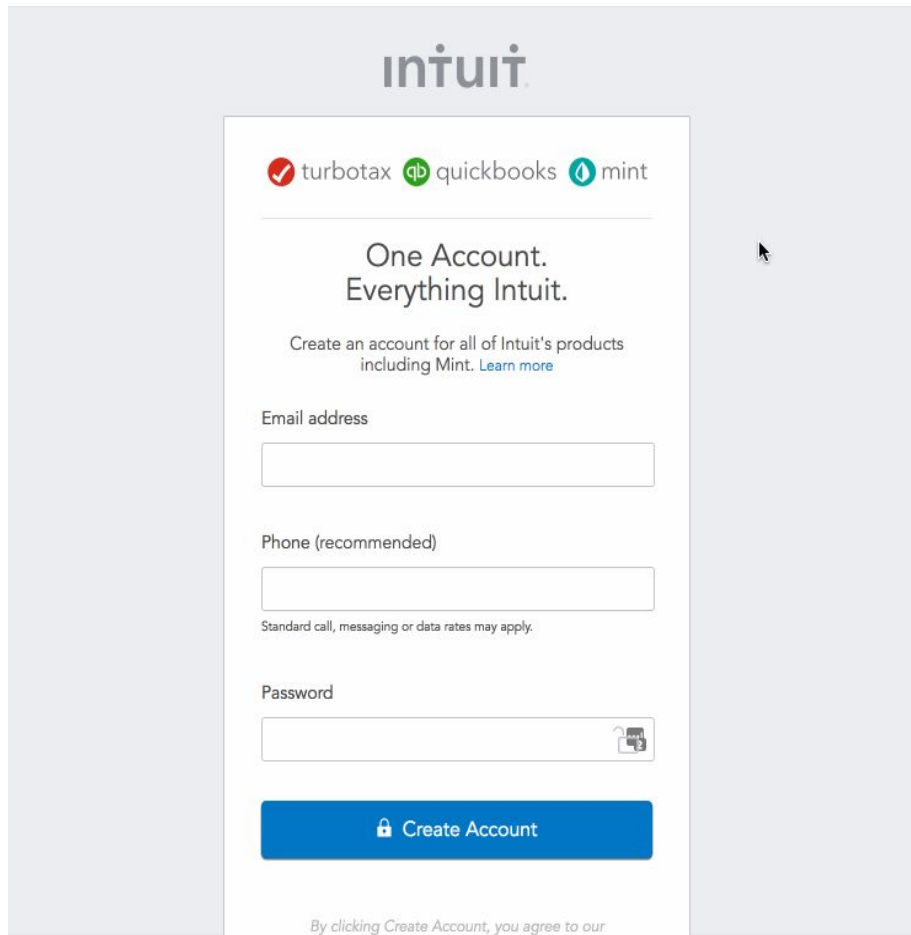
The image is a screenshot of a Facebook profile page for Nayeon Kim. The top navigation bar is dark blue with the Facebook logo, a search bar, and user information (Nayeon, Home 1). The left sidebar contains navigation links: News Feed, Messenger, SHORTCUTS (Jobs & Internships, Lucky 13, Burbank Buy, Sell, S..., USC Web/Mobile D..., USC ICA, CS50), and EXPLORE (Events, Pages, Groups, Friend Lists, Pokes, Insights). The main content area shows a post by Nayeon Kim with the text "What's on your mind?". Below this is a "Suggested Post" for "My Top Kickstarter Projects" by a user named "Like Page". The post features a video of a person holding a circular object, with a red arrow pointing to it. The video caption reads: "Just passed 200k and still going strong. The best mirror out there!". The right sidebar shows "2 event invites" (MinSun SooNie Cha and 1 other), a "TRENDING" list (Sophie the Giraffe, Lindsay Lohan, United States Mint, Bernie Sanders, Joss Whedon, Everglades National Park, IHOP, Martin Luther King III, Dexter Fowler, Rob Schneider), and a "SPONSORED" section with a "Create Ad" button.

JavaScript in live action

The image is a screenshot of a Facebook profile page for Nayeon Kim. The page layout includes a top navigation bar with the Facebook logo, a search bar, and user avatars. The left sidebar contains navigation links for News Feed, Messenger, and Shortcuts. The main content area shows a post from Nayeon Kim with the text "What's on your mind?". Below this is a "Suggested Post" for "My Top Kickstarter Projects", which is a sponsored post. The post text reads "Just passed 200k and still going strong. The best mirror out there!". The image in the post shows a person holding a circular object, with a red arrow pointing to it. The right sidebar features a "TRENDING" section with various topics and a "SPONSORED" section with an Amazon advertisement.

Facebook interface showing a profile for Nayeon Kim. The page includes a search bar, navigation links (News Feed, Messenger, Shortcuts), and a post titled "What's on your mind?". A suggested post is visible, titled "My Top Kickstarter Projects" (Sponsored), with the text "Just passed 200k and still going strong. The best mirror out there!". The image in the suggested post shows a person holding a circular object, with a red arrow pointing to it. The right sidebar displays trending topics and sponsored content.

JavaScript in live action



The image shows a screenshot of the Intuit account creation page. At the top, the Intuit logo is displayed. Below it, there are three logos: TurboTax (a red checkmark), QuickBooks (a green 'qb' icon), and Mint (a blue water drop icon). The main heading reads "One Account. Everything Intuit." followed by the text "Create an account for all of Intuit's products including Mint. [Learn more](#)". The form contains three input fields: "Email address", "Phone (recommended)", and "Password". The "Password" field has a small icon of a document with a lock. At the bottom of the form is a blue button with a white lock icon and the text "Create Account". Below the button, there is a line of text: "By clicking Create Account, you agree to our".

intuit

turbotax quickbooks mint

One Account.
Everything Intuit.

Create an account for all of Intuit's products
including Mint. [Learn more](#)

Email address

Phone (recommended)

Standard call, messaging or data rates may apply.

Password

Create Account

By clicking Create Account, you agree to our

JS syntax

```
var name = "nayeon";
```

```
var favoriteNumber = 2;
```

```
var isInstructor = true;
```

JS syntax

```
let name = "nayeon";
```

```
let favoriteNumber = 2;
```

```
let isInstructor = true;
```

Resource: More on [let](#)

JS syntax

```
let book = {  
  title: "Jane Eyre",  
  author: "Charlotte Bronte",  
  published: 1847  
}
```

JS syntax

```
if (5 > 7) {  
    // some code  
}  
  
else {  
    // some code  
}
```


JS syntax

```
for (let i = 0; i<list.length; i++) {  
    // code to iterate here  
}
```

JS syntax

```
let numberArray = [1,2,3,4,5];
```

JS syntax

```
let numberArray = [1,2,3,4,5];
```

```
numberArray.push(6);
```

```
// now numberArray is has 1,2,3,4,5,6
```

JS syntax

```
function sayHello() {  
    console.log("Hello!");  
}
```

JS syntax

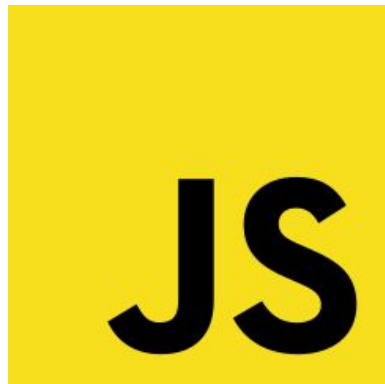
```
let sayHello = function() {  
    console.log("Hello!");  
}
```

JS syntax

```
let sayHello = function() {  
    console.log("Hello!");  
}  
  
// To call the function  
  
sayHello();
```


Two key features of JS

1. Searching and changing elements in the DOM
2. Listening to user events



Searching elements??? What does that mean?

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Fun</title>
</head>
<body>
  <h1>Hello</h1>
  <p>I'm learning about JS</p>
</body>
</html>
```



JavaScript allows us to easily **find** any specific element in our HTML file.

To fully understand this, we need to first learn about the **Document Object Model**

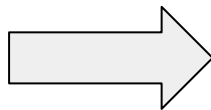
Document Object Model (DOM)

- A tree-like structure that represents a web page which can be utilized quickly access elements using JS

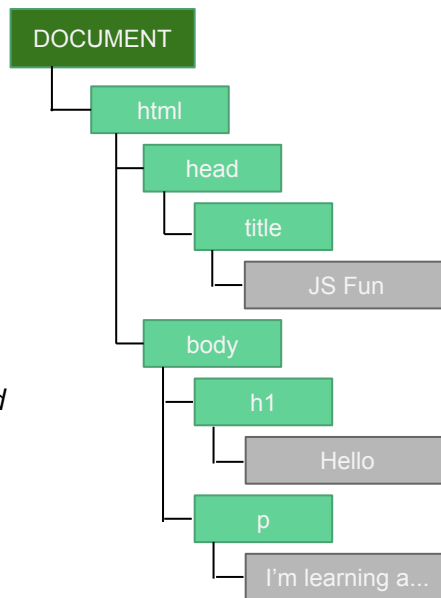
Document Object Model (DOM)

- A tree-like structure that represents a web page which can be utilized quickly access elements using JS

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Fun</title>
</head>
<body>
  <h1>Hello</h1>
  <p>I'm learning about JS!</p>
</body>
</html>
```



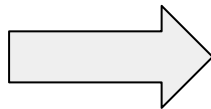
*Your HTML file
gets read into
the browser and
loads into DOM*



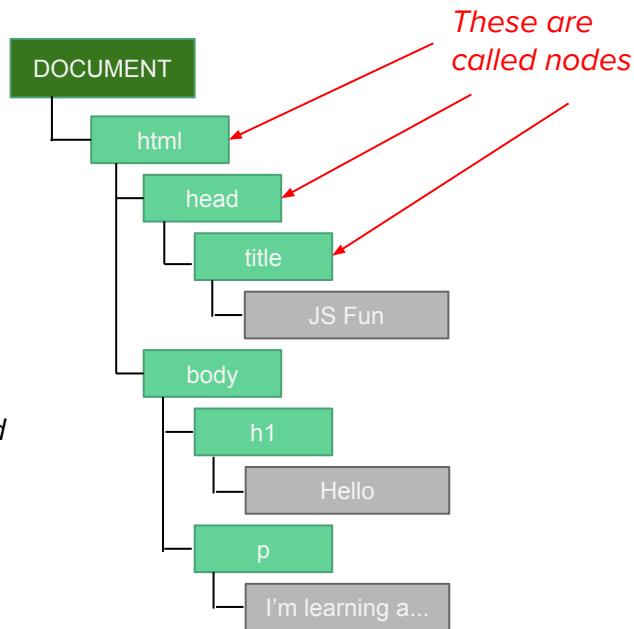
Document Object Model (DOM)

- A tree-like structure that represents a web page which can be utilized quickly access elements using JS

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Fun</title>
</head>
<body>
  <h1>Hello</h1>
  <p>I'm learning about JS!</p>
</body>
</html>
```



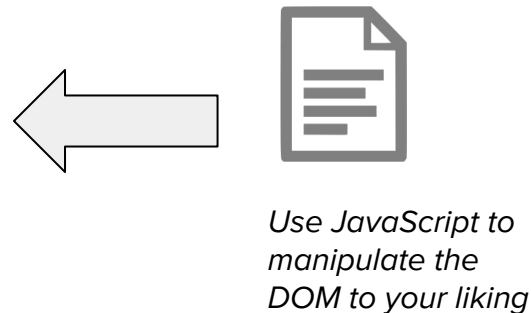
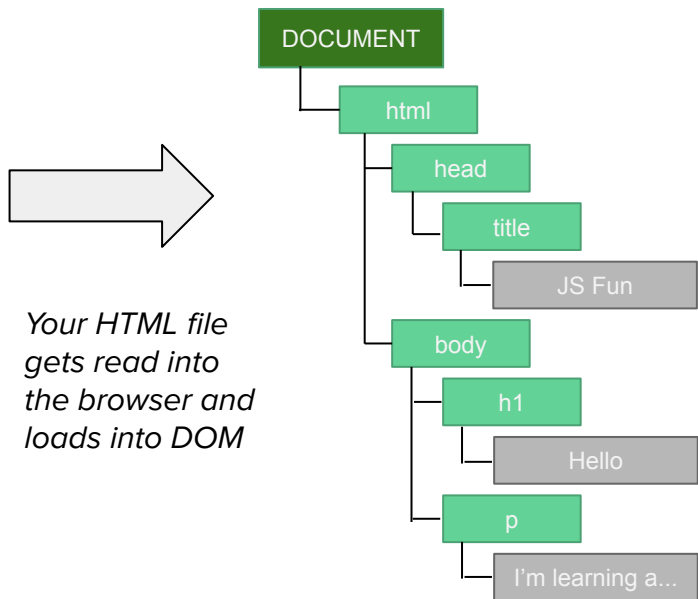
*Your HTML file
gets read into
the browser and
loads into DOM*



Ok... so what does JS do?

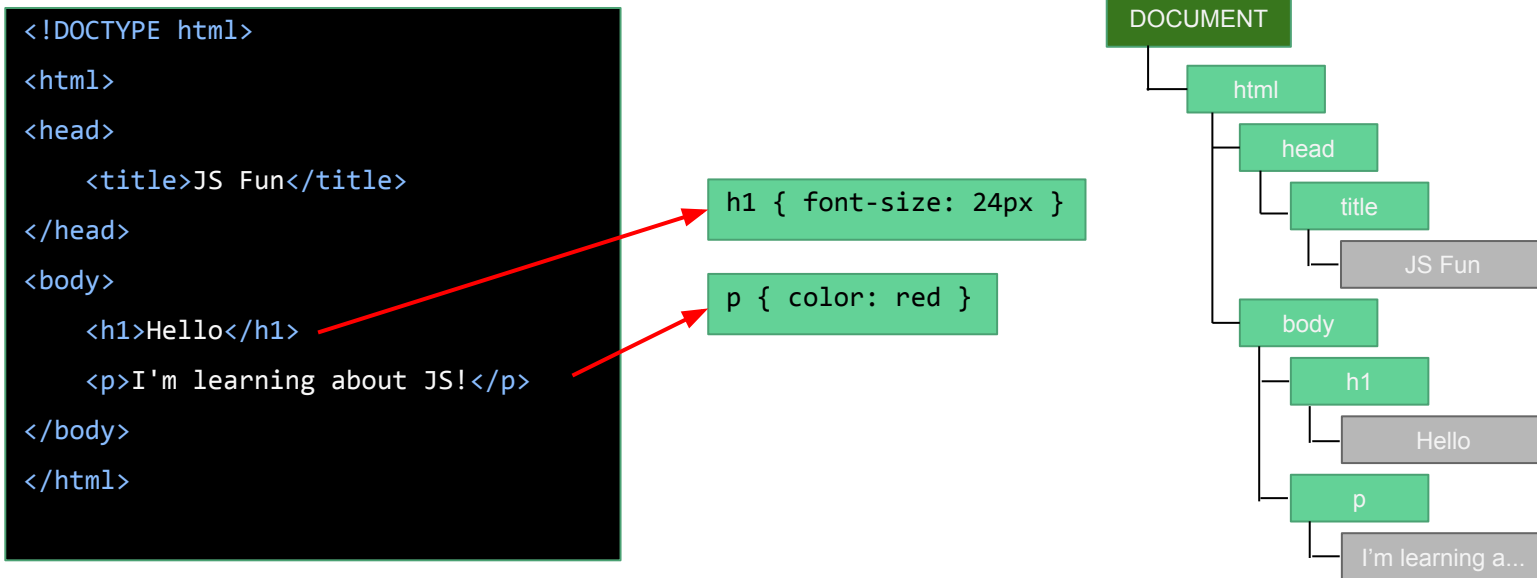
- JavaScript gives us a language to interact with the DOM

```
TYPE html>  
>  
>  
<title>JS Fun</title>  
</head>  
>  
<h1>Hello</h1>  
<p>I'm learning about JS!</p>  
</body>  
</html>
```



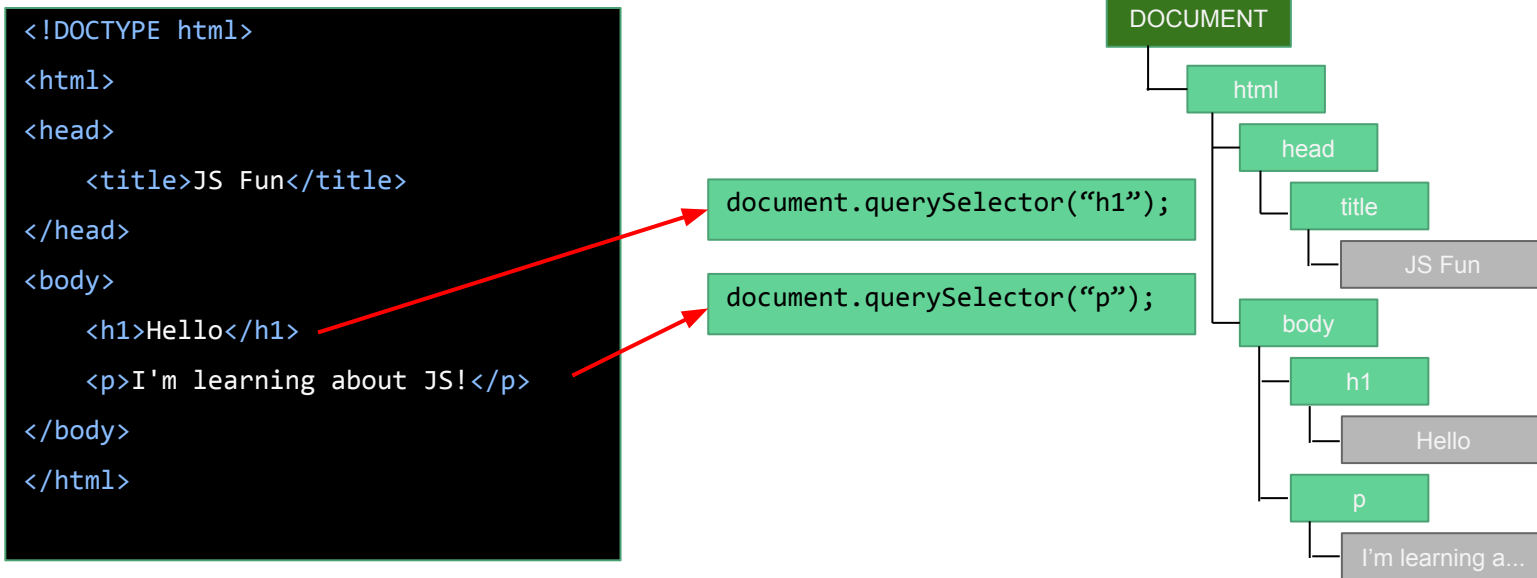
How JS “Finds elements” aka accesses the DOM

- We can select any DOM elements with JS by using the same syntax as CSS selectors



How JS “Finds elements” aka accesses the DOM

- We can select any DOM elements with JS by using the same syntax as CSS selectors



Accessing DOM Nodes

There are few ways to select DOM Nodes (HTML Elements) using JS:

- `document.querySelector(CSS Selector)`
- `document.querySelectorAll(CSS Selector)`
- `document.getElementById(ID)`
- `document.getElementsByClassName(Class)`
- `document.getElementsByTagName(Tag)`

DOM Events

Now that we can “find” elements, we can also wait and listen for an event to trigger the element.

<code>onclick</code>	Mouse right-click on an element.
<code>onmouseenter</code>	Mouse enters an element.
<code>onmouseleave</code>	Mouse leaves an element.
<code>onmouseover</code>	Mouse enters an element or its children.
<code>onmouseout</code>	Mouse leaves an element or its children.

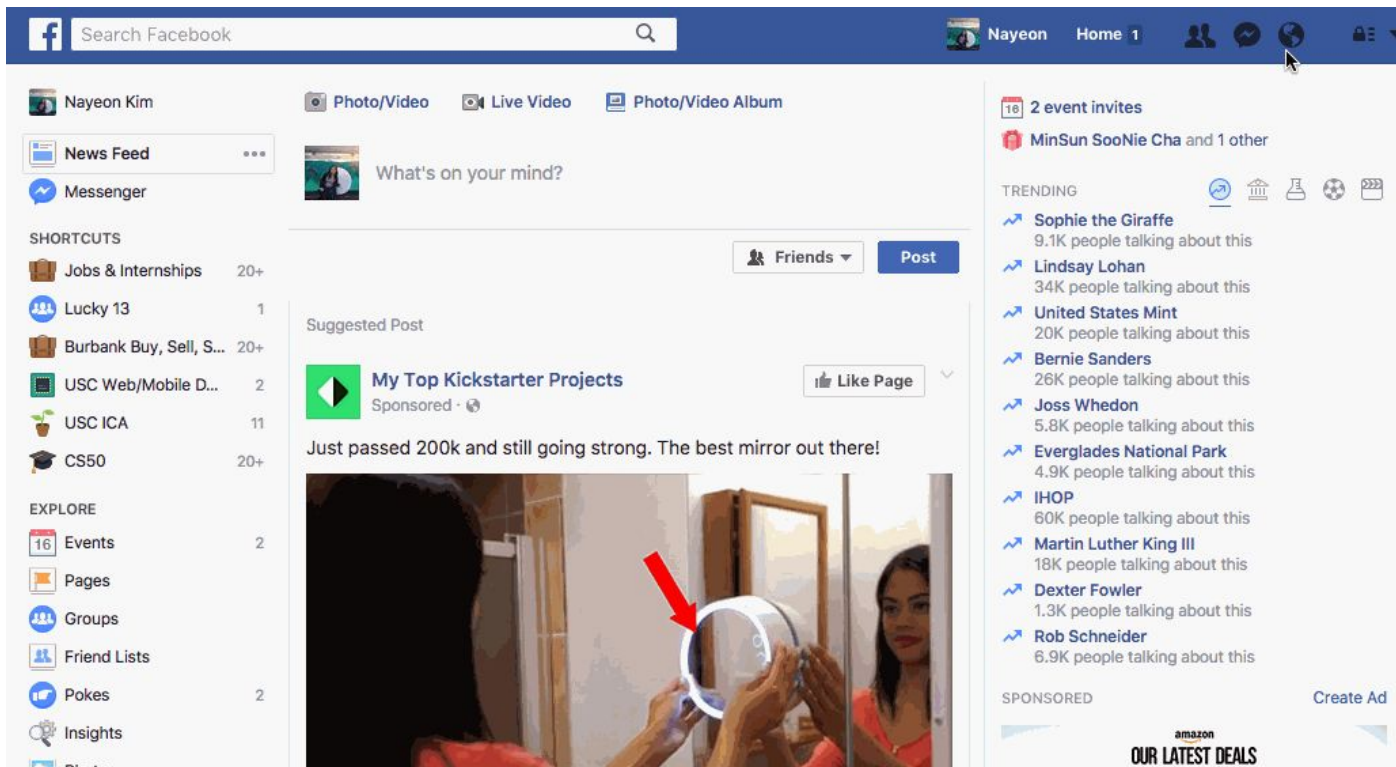
Resource: [Full list of DOM Events.](#)

```
document.querySelector('#content').onclick = function(){  
  this.style.backgroundColor = '#FC0';  
};
```

```
document.querySelector('div').onmouseenter = function(){  
  document.querySelector('#name').innerHTML = 'Tommy';  
};
```

```
var items = document.querySelectorAll('.item');  
for (var i=0; i < items.length; i++) {  
  items[i].onmouseleave = function(){  
    this.href = 'https://www.usc.edu/';  
  }  
}
```


JavaScript in live action

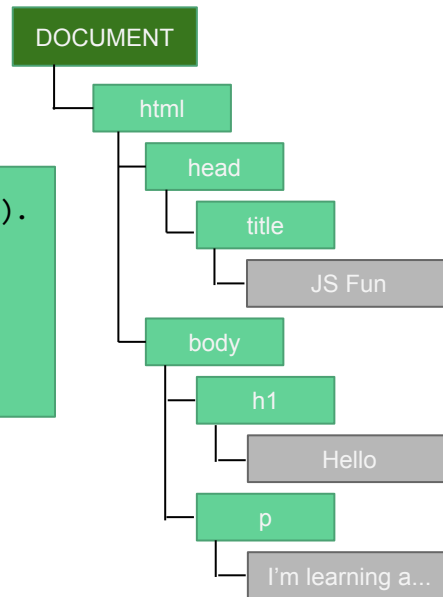


DOM Traversal is possible too

- Without specifying an element, can find neighboring elements like sibling, parent, child, etc

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Fun</title>
</head>
<body>
  <h1>Hello</h1>
  <p>I'm learning about JS!</p>
</body>
</html>
```

`document.querySelector("h1").
nextSibling;`
// returns `<p>I'm learning
about JS!</p>`



DOM Traversal properties

<code>parentNode</code>	Parent element.
<code>children</code>	Children elements.
<code>nextSibling</code>	Next sibling, including whitespace (text) nodes.
<code>nextElementSibling</code>	Next sibling, excluding whitespace (text) nodes.
<code>previousSibling</code>	Previous sibling, including whitespace (text) nodes.
<code>previousElementSibling</code>	Previous sibling, excluding whitespace (text) nodes.