

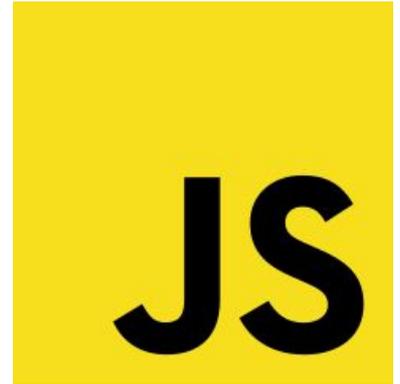
# Lecture 07: JS Events and Traversal

---

ITP 303 Full-Stack Web Development

# 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**

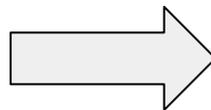
# Document Object Model (DOM)

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

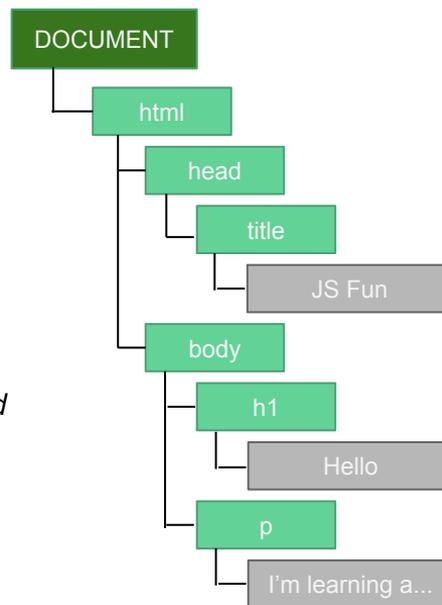
# Document Object Model (DOM)

- 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>
```



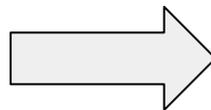
*Your HTML file  
gets read into  
the browser and  
loads into DOM*



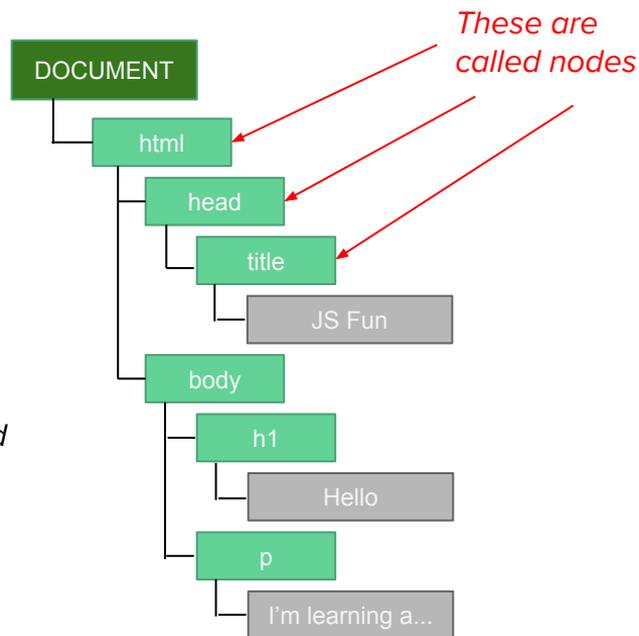
# Document Object Model (DOM)

- 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>
```



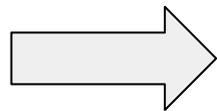
*Your HTML file gets read into the browser and loads into DOM*



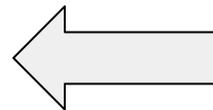
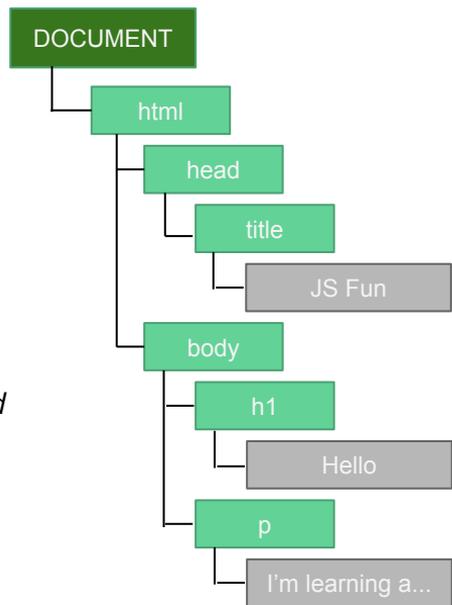
# Ok... so what does JS do?

- JavaScript gives us a language to interact with the DOM

```
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*



*Use JavaScript to manipulate the DOM to your liking*

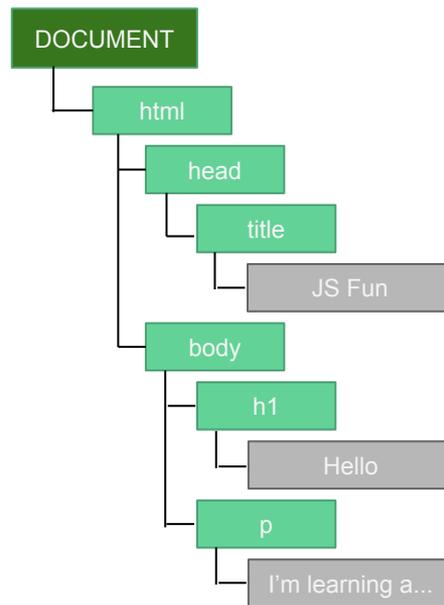
# How JS “Finds elements” aka accesses the DOM

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

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

h1 { font-size: 24px }

p { color: red }



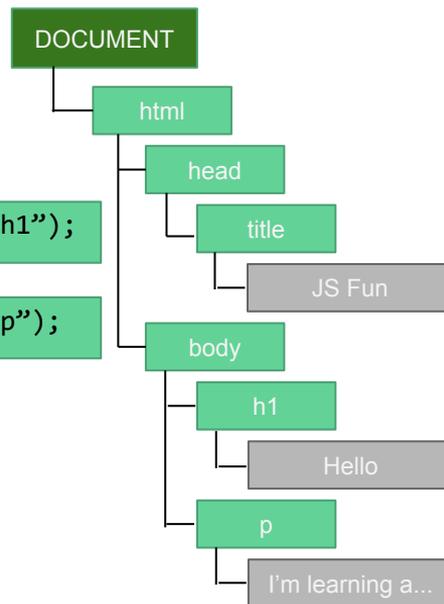
# How JS “Finds elements” aka accesses the DOM

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

```
<!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");`

`document.querySelector("p");`



# Accessing DOM Nodes

There are several 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

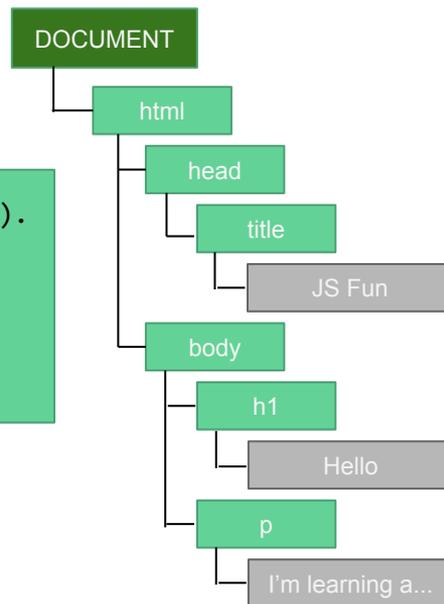
The image is a screenshot of a Facebook profile page for Nayeon Kim. The top navigation bar includes the Facebook logo, a search bar, and the user's name 'Nayeon' with 'Home 1' and various utility icons. The left sidebar contains navigation options: News Feed, Messenger, and a 'SHORTCUTS' section with links to Jobs & Internships, Lucky 13, Burbank Buy, Sell, S..., USC Web/Mobile D..., USC ICA, and CS50. An 'EXPLORE' section follows with links to Events, Pages, Groups, Friend Lists, Pokes, and Insights. The main content area starts with a 'What's on your mind?' prompt and a 'Post' button. Below this is a 'Suggested Post' for 'My Top Kickstarter Projects', which is a sponsored video. The video shows a person holding a white circular object, with a red arrow pointing to it. To the right of the video is a 'Like Page' button. Further right is a 'TRENDING' section listing various entities like Sophie the Giraffe, Lindsay Lohan, United States Mint, Bernie Sanders, Joss Whedon, Everglades National Park, IHOP, Martin Luther King III, Dexter Fowler, and Rob Schneider. At the bottom right, there is a 'SPONSORED' section for 'amazon OUR LATEST DEALS'.

# 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.