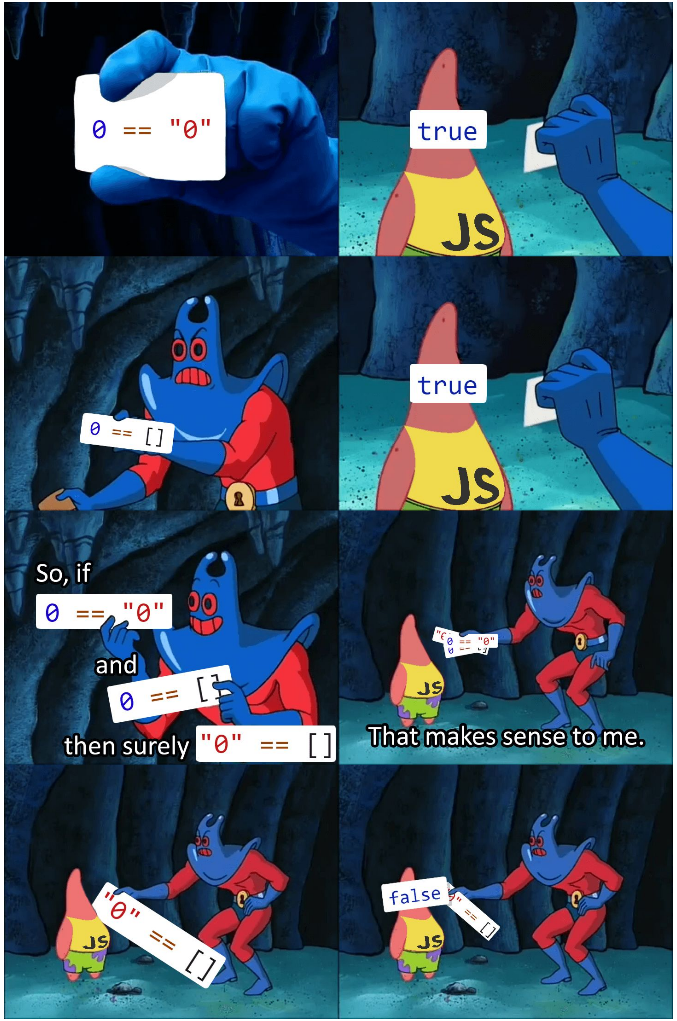


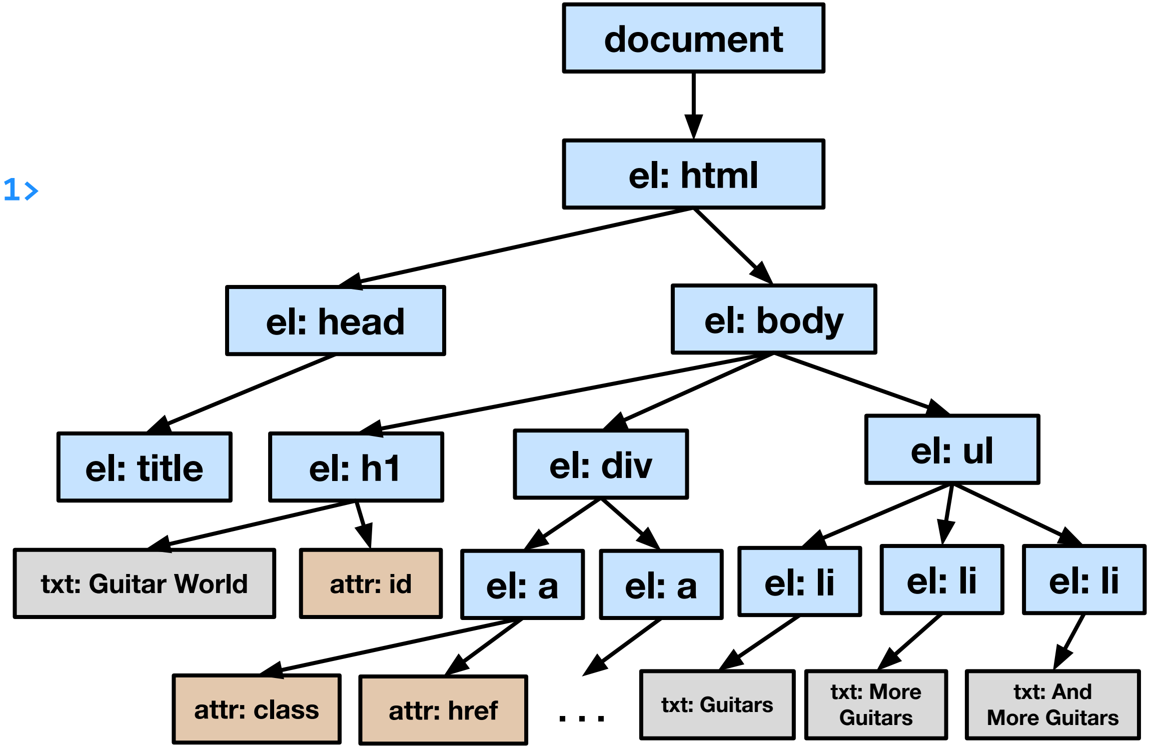
CSc 337
DOM

Benjamin Dicken

Document Object Model

- The DOM is the tree-structured, “behind the scenes” representation of the elements, attributes, etc of a webpage
- Can explore the dom via the inspector

```html
<html>
<head>
  <title>About</title>
</head>
<body>
  <h1 id="mainTitle">Guitar World</h1>
  <div>
    <a class="mainLink"
       href="index.html">Home</a>
    <a class="mainLink"
       href="store.html">Store</a>
  </div>
  <ul>
    <li>Guitars</li>
    <li>More Guitars</li>
    <li>And More Guitars!</li>
  </ul>
</body>
</html>
```



Activity

```html
<html>
<head>
  <title>Microsoft</title>
</head>
<body>
  <h1 id="aTitle">Microsoft</h1>
  <img src="./img/ms.jpg"/>
  <div id="mainContent">
    <span>
      Click
      <a href="buy.html">here</a>
      To buy a surface pro!
    </span>
  </div>
</body>
</html>
```

How many DOM elements?

Activity

```html
<html>
<head>
  <title>Microsoft</title>
</head>
<body>
  <h1 id="aTitle">Microsoft</h1>
  <img src="./img/ms.jpg"/>
  <div id="mainContent">
    <span>
      Click
      <a href="buy.html">here</a>
      To buy a surface pro!
    </span>
  </div>
</body>
</html>
```

Activity

```html
<html>
<head>
  <title>Microsoft</title>
</head>
<body>
  <h1 id="aTitle">Microsoft</h1>
  <img src="./img/ms.jpg"/>
  <div id="mainContent">
    <span>
      Click
      <a href="buy.html">here</a>
      To buy a surface pro!
    </span>
  </div>
</body>
</html>
```

What is the deepest DOM element level?

Use Js to access DOM

- The DOM is the tree-structured, “behind the scenes” representation of the elements, attributes, etc of a webpage
    - Js Objects
- Can explore the dom via the developer console
- Access via the **`document`** variable

Use Js to access DOM

```
function printDOM(element) {
  console.log(element.tagName);
  if (element.children == undefined) { return; }
  else {
    for (c in element.children) {
      printDOM(element.children[c]);
    }
  }
}

printDOM(document);
```

Accessing specific elements

Methods in the document object:

```
getElementById(string)

getElementsByClassName(string)

getElementsbyTagName(string)
```

Returns object(s) representing element(s)
in the DOM

Text vs HTML

One of the most basic things to do with a DOM element is set/get the text and HTML inside of it

```
element.innerHTML
element.innerText
```

What is the difference?

Activity

What does this print?

```
<html>
<head>
  <title>Microsoft</title>
</head>
<body>
  <h1 id="aTitle">Microsoft</h1>
  <img src="./img/ms.jpg"/>
  <div id="mainContent">
    <span>
      Click
      <a href="buy.html">here</a>
      To buy a surface pro!
    </span>
  </div>
</body>
</html>
```

```
var c = document.getElementById('mainContent')
console.log(c.innerHTML)
console.log(c.innerText)
```

Activity

```html
<html>
<head>
  <title>Microsoft</title>
</head>
<body>
  <h1 id="aTitle">Microsoft</h1>
  <img src="./img/ms.jpg"/>
  <div id="mainContent">
    <span>
      Click
      <a href="buy.html">here</a>
      To buy a surface pro!
    </span>
  </div>
</body>
</html>
```

What does this print?

```javascript
var c = document.getElementById('mainContent')
c.innerHTML = 'hi'
console.log(c.innerText)
```

Changing Styles

Can change style (CSS) properties by updating the **style** information in the DOM

```
element.style.PROPERTY = VALUE
```

DOM Reference

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

Activity

```
function getColor() {
    let randomValue = Math.random();
    if (randomValue < 0.3) { return 'red'; }
    else if (randomValue < 0.7) { return 'blue'; }
    return 'white';
}


function patriotify() {
    var elements = document.getElementsByTagName("*");
    for(var i = 0; i < elements.length; i++) {
        elements[i].style.backgroundColor = getColor();
    }
}

window.setInterval( patriotify, 1000 );
```

How would this affect a webpage?