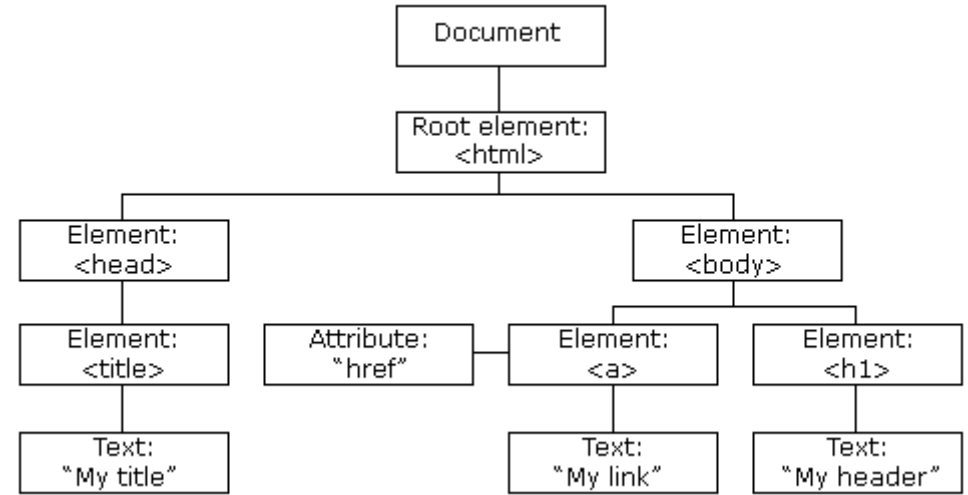# Web and Database Computing

Introduction to JavaScript: DOM Basics

DOM

the **D**ocment **O**bject **M**odel

What is the DOM?

- Object based representation of an XML document.
- Tree Structure
- Provides a way for us to naviate a document in Javascript.

From W3Schools

How can we access elements using the DOM?

Using its ID:

```
var element = document.getElementByID('SomeID');
```

Using its class name:

```
var elements = document.getElementsByClassName('SomeClass');
```

- Returns a list

Using its tag name:

```
var elements =  document.getElementsByTagName('sometag');
```

- Also returns a list

How can we access elements using the DOM?

With a CSS selector:

```
var elements = document.querySelectorAll('p.class div i');
```

- Returns a list

Using an element we already have to get it's parent:

```
var parent_element = element.parentElement;
```

Using an element we already have to get its children:

```
var child_elements = element.children;
```

And more ...

What can we do with DOM elements?

Get content:

```
var text = element.innerText;
var html = element.innerHTML;
```

Get attributes:

```
var attr = element.getAttribute('someattribute');
```

Some attributes have their own direct values

```
var title = element.title;
```

Style can be accessed directly

```
var color = element.style.color;
```

What can we do with DOM elements?

Modify content:

```
element.innerText = 'Some <text>';
element.innerHTML = 'Some <text>';
```

Modify attributes:

```
element.setAttribute('someattribute', "value");
```

Direct values don't work for all attributes:

```
element.title = "This title works";
element.class = "may not work";
```

Style can be set directly

```
element.style.color = "red";
```

An example

```javascript
function byId() {
  var item = document.getElementById("demo");
  item.style.backgroundColor = "yellow";
}

function byClass() {
  var items = document.getElementsByClassName("test");
  for (var i=0; i<items.length; i++){
    items[i].style.backgroundColor = "yellow";
  }
}

function byTag() {
  var items = document.getElementsByTagName("p");
  for (var i=0; i<items.length; i++){
    items[i].style.backgroundColor = "yellow";
  }
}

function byCSS() {
  var items = document.querySelectorAll("p#demo,span.test");
  for (var i=0; i<items.length; i++){
    items[i].style.backgroundColor = "yellow";
  }
```

https://jsfiddle.net/ian_knight_uofa/av4jw97f/

Manipulating DOM Tree

Create Element:

```
var newElement = document.createElement('P');
```

Add it to another element (with id 'parent'):

```
var parent = document.getElementById('parent');
parent.appendChild(newElement);
```

Remove that child element

```
parent.removeChild(newElement);
```

An example

JavaScript   HTML   Result   Edit in JSFiddle

```javascript
function addNew() {
  var newElement = document.createElement('P');
  newElement.innerText = "A new paragraph";
  var div = document.getElementById("demo");
  div.appendChild(newElement);
}

function removeAll() {
      var div = document.getElementById("demo");
      while(div.children.length > 0){
      div.removeChild(div.children[0]);
  }
}
```

https://jsfiddle.net/ian_knight_uofa/9ctkdzja/

A Formula!

3 steps to make writing client JS easier
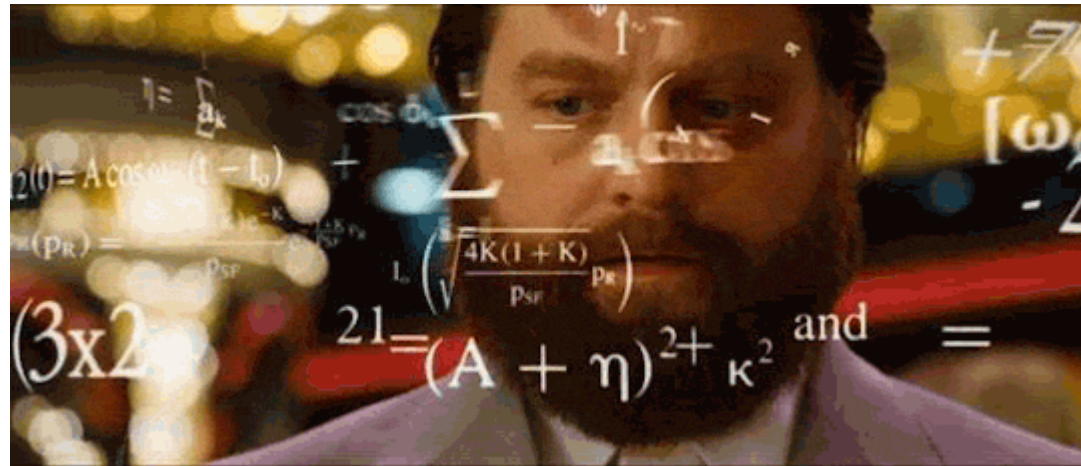
1) Identify what to Select/Change

- What elements (if any) need to be selected?
  - Are they easy to select?
  - Add class names/ids to make selection easier if needed.
- Do you need to create new elements?
  - Determine the structure of these.
- What do you need to change about the elements?
- Are there any other values you need to obtain?
  - Where do they come from?

2) Identify trigger for change

- What causes the change to occur?
  - Is it a user interaction, or does it happen on when loading the page/some element?\
  - Will you need to add an event listener/attribute?
  - https://www.w3schools.com/jsref/dom_obj_event.asp

3) Implement a function

- Write a function to perform the action
- Link it to the event.