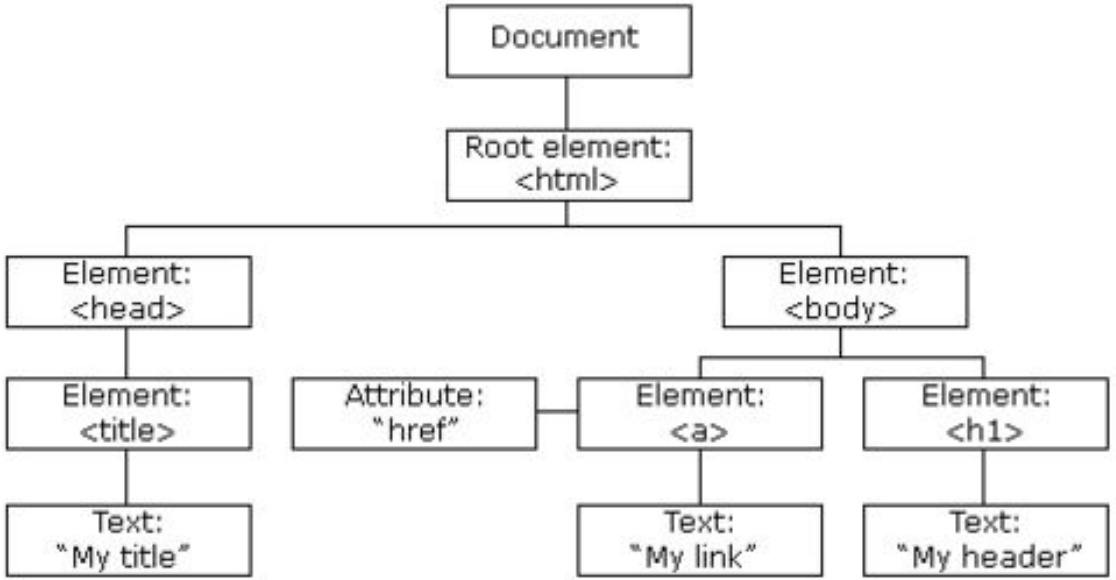
JS - DOM Interaction

You’re ready...

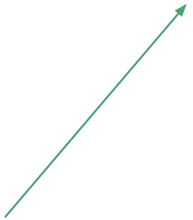
JS lives inside of your browser and has access to the DOM via *document*

document is a global variable, and it looks like this:

The HTML DOM Tree of Objects

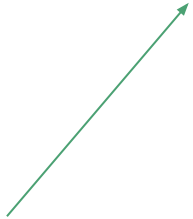


```
var element = document.getElementById("name");
```



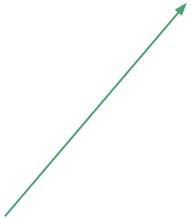
Returns an *object* that represents the HTML element in your website with the given Id

```
var elements =
document.getElementsByClassName("class");
```



Returns an *array* that represents the HTML elements in your website with the given class

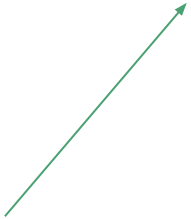
```
var element = document.createElement("h1");
```



Creates a brand new HTML element with the given tagname, but it won't be in the DOM (not until you tell it to)

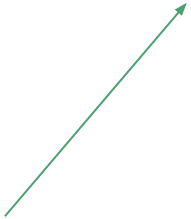
```
var element1 = document.createElement("h1");
var element2 = document.getElementById("el2");

element2.appendChild(element1);
```



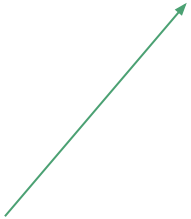
Adds element1 inside element2 as its last child (Now this new element is in the DOM!)

element1.innerHTML = “Hello”;



Replaces all of the inner contents of element1 with “Hello”

element1.style.color = “#4433AA”;

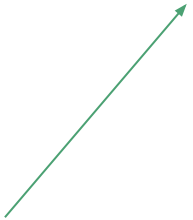


Changes the inline-style of element1

JS is nice for DOM manipulation, but JQuery does it better!

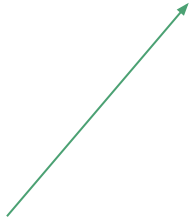
You versus The one she told you not to worry about

```
var element = $("#elementId");
```



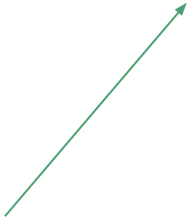
Returns an *object* that represents the HTML element in your website with the given Id

```
var elements = $(“.className”);
```



Returns an *object* that represents the HTML **elements** in your website with the given **class**

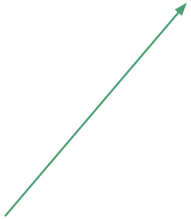
```
var element = $("<div></div>")
```



Creates a brand new HTML element with the given tagname, but it won't be in the DOM (not until you tell it to)

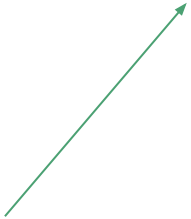
```
var element1 = $("<div></div>")
var element2 = $("#elementId");

element2.append(element1);
```



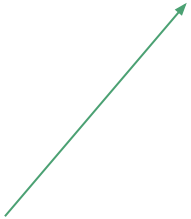
Adds element1 inside element2 as its last child (Now this new element is in the DOM!)

element1.html(“Hello”);



Replaces all of the inner contents of element1 with “Hello”

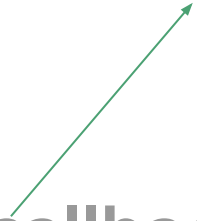
element1.css(“color: #4433AA;”);



Changes the inline-style of element1

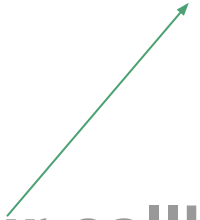
The real beauty is that JQuery enables user interaction with callbacks!

```
element1.click(function() {
    alert("hey, you clicked me!");
});
```



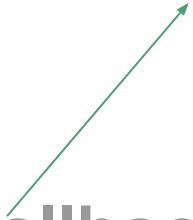
Invokes your callback function whenever you click element1

```
element1.dblclick(function() {
    alert("hey, you double clicked me!");
});
```



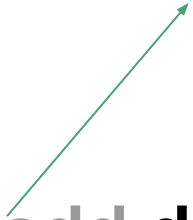
Invokes your callback function whenever you double-click element1

```
element1.hover(function() {
    alert("hey, you hovered over me!");
});
```



Invokes your callback function whenever you hover over element1

```
element1.hide();
element1.show();
```



hide() will add display: “none”; to element1

show() will remove display: “none”; from element1

JQuery does way more than DOM interaction, but we'll get to that later