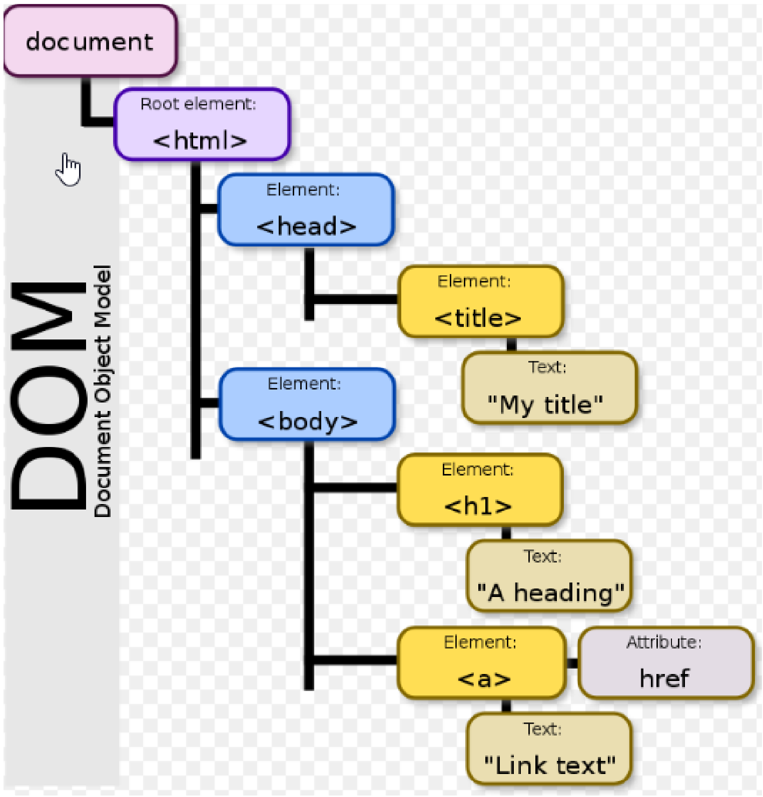
JS The HTML DOM

Lesson Time: 30 Minutes

Javascript is one of the most widely used programming languages in the world, and it’s been called “the language of the world wide web.” Javascript’s dominance of the web browser is due to DOM - the document object model.

So what is the DOM? First, we know that web pages are really just organized text--text that is organized by HTML tags to give it structure. And since we are dealing with structured text, we should be able to pull that text into Javascript as a javascript object. This is exactly what the DOM does--it loads your HTML into Javascript, converting it to a javascript object that can be worked with.

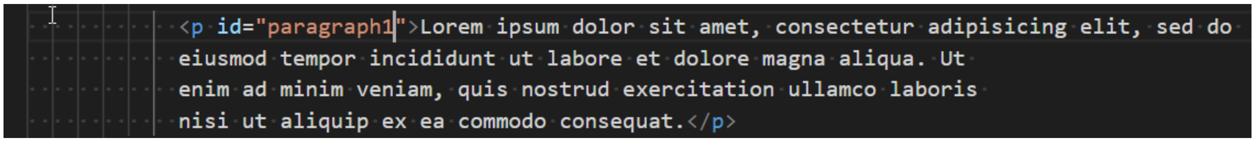


Your <html> tag becomes the javascript object “document”. The <body> tag becomes “document.body”. The entire web page is loaded into javascript as a tree of objects that can be changed by javascript. Javascript code can change, remove, add, or make any kind of edits to the page and have the changes re-displayed in the browser.

One important thing to understand is that Javascript is not making these changes on the web server--it's making the changes on your copy of the webpage downloaded to your browser. Reloading the page can reset the page to all default settings.

Let's explore a few ways javascript can change a page.

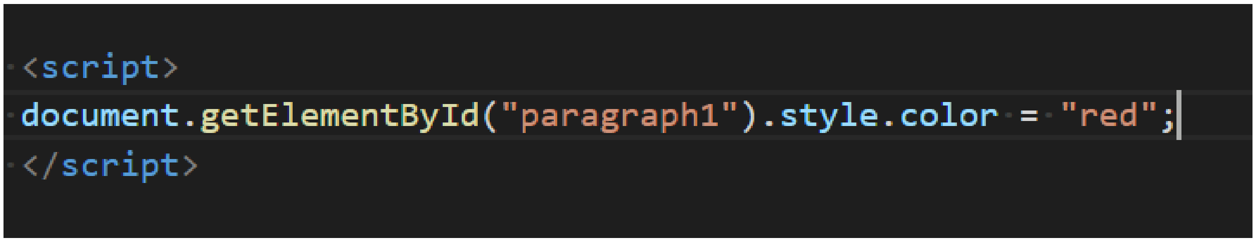
First thing to recall from our previous HTML lessons is that every element has an **id** attribute.   You get to choose the name of the ID when creating the element.



```
                <p id="paragraph1">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
                eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
                enim ad minim veniam, quis nostrud exercitation ullamco laboris
                nisi ut aliquip ex ea commodo consequat.</p>
```

Here we have a paragraph element we've given the ID *paragraph1*

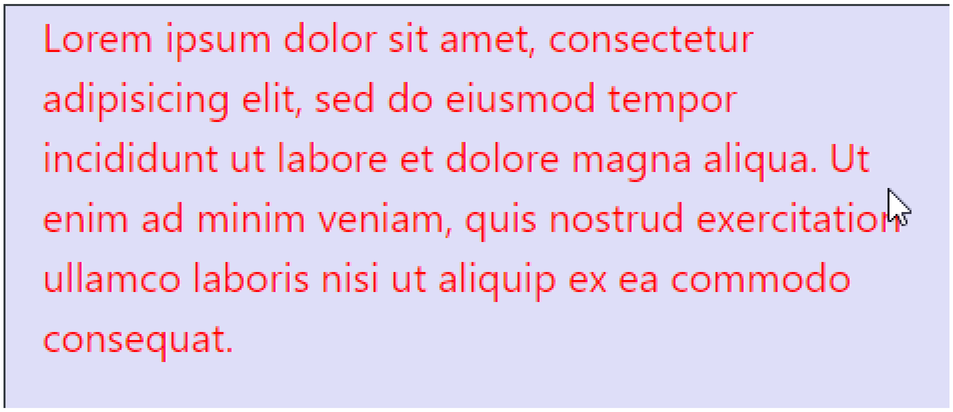
In the <script> tag of our document, we can access the element with JS by using the **getElementByID** method.



```
<script>
document.getElementById("paragraph1").style.color = "red";
</script>
```

In this line of code, we change the text color to red for the elements that belong to paragraph1

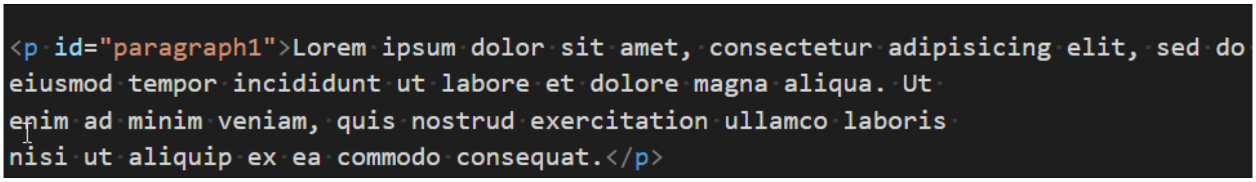
And the results appears in browser.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

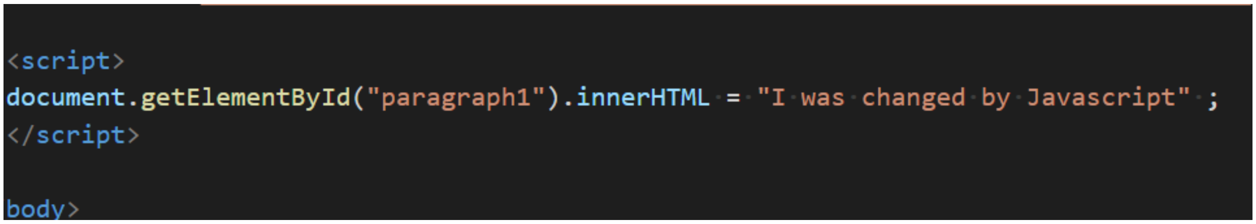
In another example, we can change the text itself, again by getting the element by its ID, and then working with a property of the HTML element

1. Paragraph1 element in HTML, before the javascript code executes.



```
<p id="paragraph1">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.</p>
```

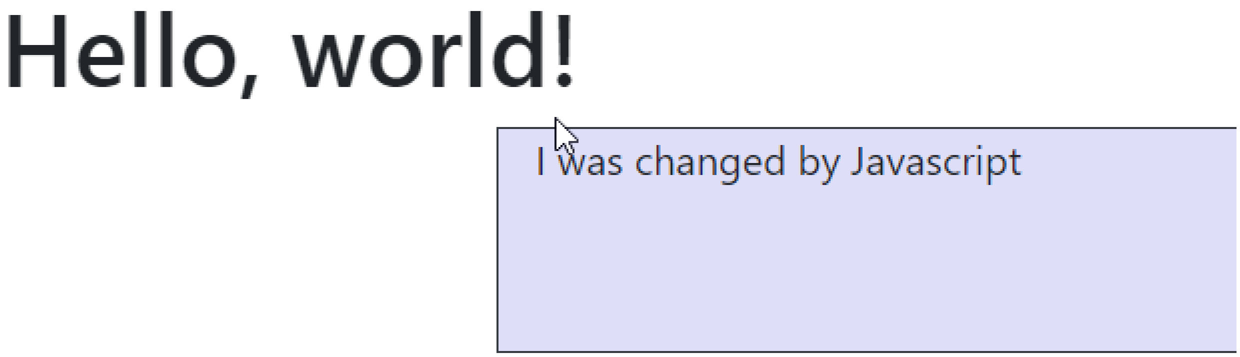
2. The javascript in the <script> tag



```
<script>
document.getElementById("paragraph1").innerHTML = "I was changed by Javascript" ;
</script>

body>
```

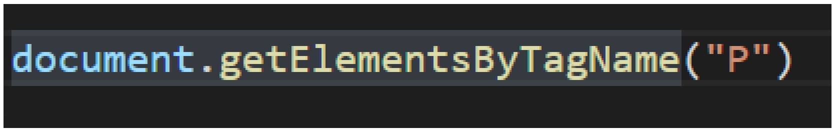
3. Results when the page loads.



Hello, world!

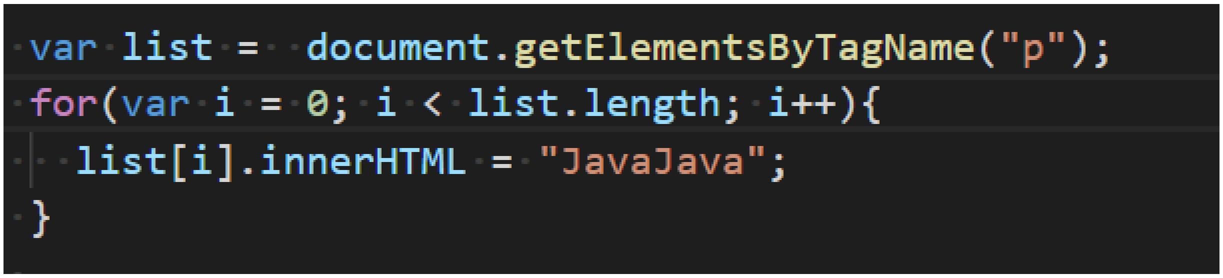
I was changed by Javascript

The most common way to manipulate an HTML element is by getElementByID. However, you can also use getElementsByTagName or getElementsByClassName. Be aware of the plural ***getElements*** . These javascript methods will return ***all*** elements that have the same html tag or class as the one you pass to Javascript.



```
document.getElementsByTagName("P")
```

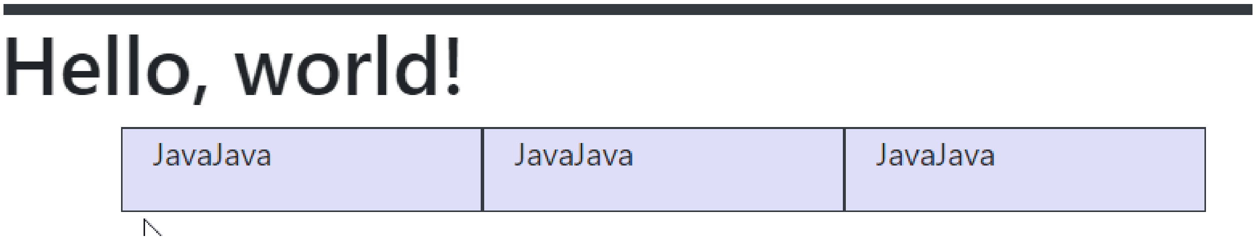
This returns all elements that are <p> tags. There is more than one.



```
var list =  document.getElementsByTagName("p");
for(var i = 0; i < list.length; i++){
   list[i].innerHTML = "JavaJava";
}
```

This code says...
Get all elements with the tag of p and assign them to the variable “list”. *For each item in “list”, change it’s innerHTML to “JavaJava*

Results:



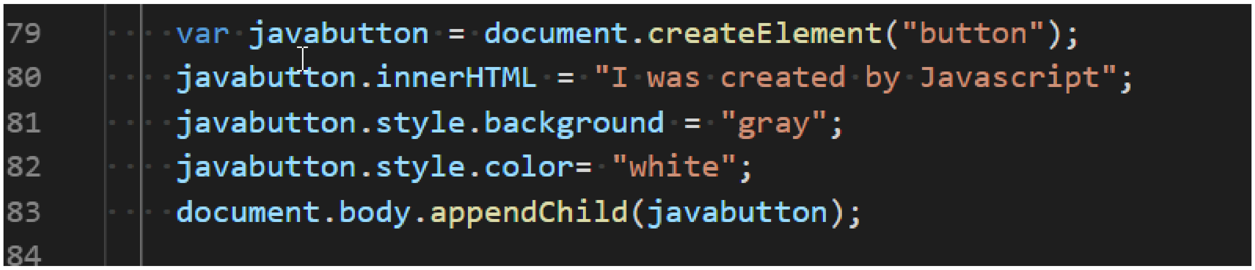
Hello, world!

JavaJava

JavaJava

JavaJava

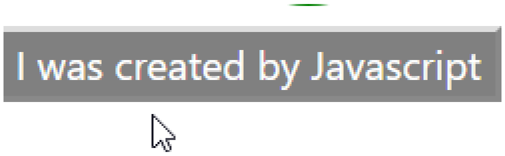
In addition to finding and changing existing elements in our documents, we can create or delete brand new elements. Example.



```
    var javabutton = document.createElement("button");
    javabutton.innerHTML = "I was created by Javascript";
    javabutton.style.background = "gray";
    javabutton.style.color= "white";
    document.body.appendChild(javabutton);
```

This code says...
1. Create a variable called javabutton and assign it the results of the createElement method as a new button. 2. Set the new button’s innerHTML 3. Set the new button’s background color. 4. Set the new button’s text color. 5. Add the new button to the HTML document.

Results:



Key Terms	getElementByID(), getElementByTagName(), GetElementByClassName()
Lesson Files	
Additional Resources	https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

Further Learning	How did javascript get so popular? https://news.codecademy.com/javascript-history-popularity/

Important DOM Objects and Properties

Lesson Time: 30 Minutes

The three most important DOM objects are

DOM Object	Meaning
Document	The HTML document
Element	An element in the HTML
Window	The Browser/Browser Window

The most important & common properties are

document.getElementByID	Find an element by HTML ID
document.addEventListener element.addEventListener	Create a new HTML event on an element
document.cookie	Work with cookies
document.getElementsByClassName	Get all elements by their HTML Class
document.querySelector	Get first element by it’s CSS Selector
document.querySelectorAll	Get all elements that match CSS Selector

element.InnerHTML	Work with the HTML content of an element

The general workflow of DOM manipulation goes like this.

1. Lookup an element by using getElementById, querySelector, ect.
2. Change something about that element, such as change HTML content, add an event, move it around the screen, change it’s color, ect.

There is a complete list of all of the DOM object properties on W3 Schools, and there is more than we could cover here. The goal is that you understand how to use JS to change your HTML document and can learn and practice over time. No one is an expert overnight!