

THE DOM

How to make our pages interactive

THE DOM

The DOM stands for document object model. In short, this is the way javascript perceives a webpage.

THE DOM

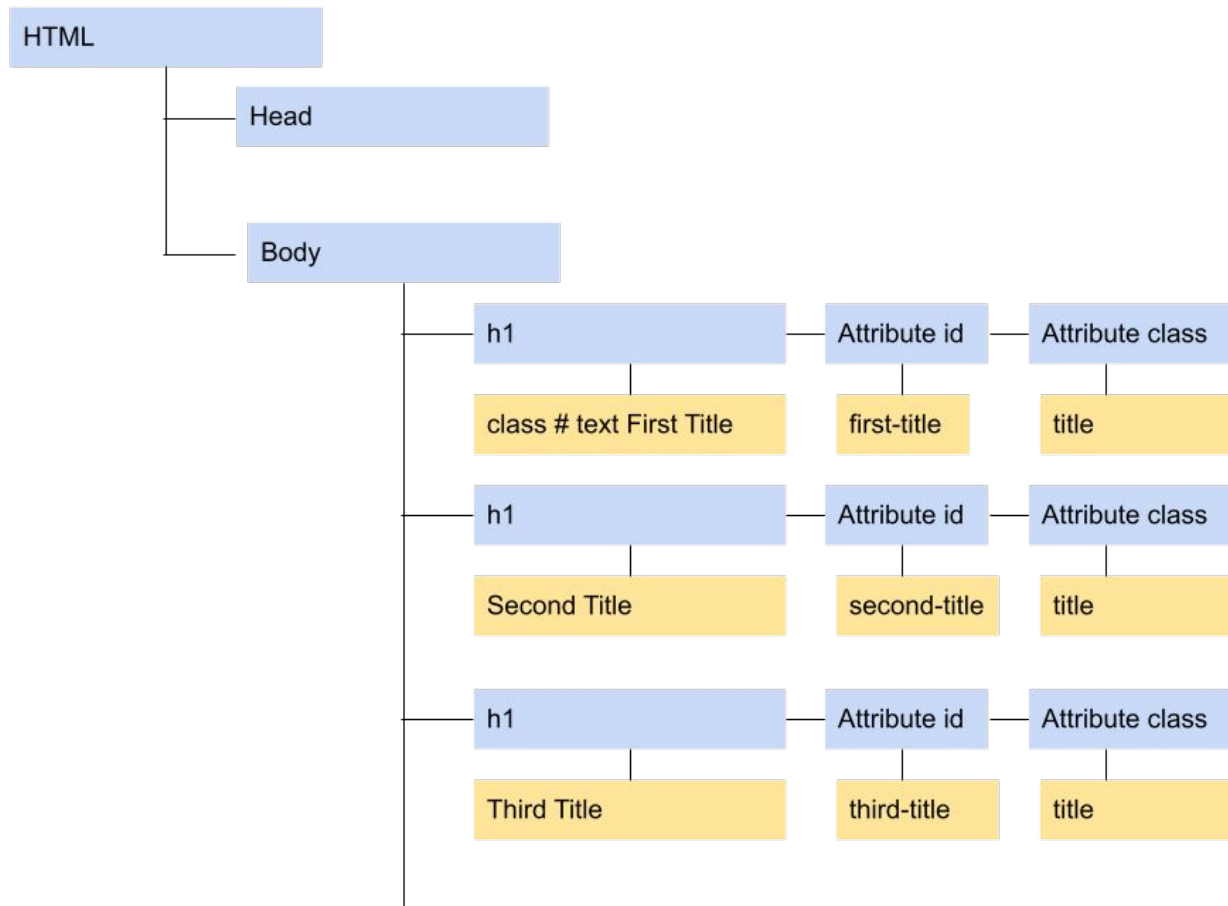
The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects.

THE DOM

DOM methods allow programmatic access to the tree. With them, you can change the document's structure, style, or content.

THE DOM

A DOM
tree
might
look
like
this:



THE DOM

Another way to say that is that every HTML document is structured as a Javascript Object. Every HTML element has a different properties that can help to manipulate it.

THE DOM

It is possible to get, create, append or remove HTML elements using JavaScript.

THE DOM

Selecting HTML element using Javascript is similar to selecting using CSS. To select an HTML element, we use tag name, id, class name or other attributes.

THE DOM

We can access already created HTML elements or elements using javascript. To access or get elements, we use different built-in methods on the document object.

THE DOM

getElementsByTagName(): This method takes a tag name as a string parameter. This method returns an `HTMLCollection` object.

THE DOM

An `HTMLCollection` is an array like object of HTML elements. The `length` property provides the size of the collection.

THE DOM

Whenever we use this method we access the individual elements using index or after loop through each individual items.

THE DOM

An `HTMLCollection` does not support all array methods therefore we should use regular for loop instead of the `forEach` method.

THE DOM

The basic syntax looks like this:

```
document.getElementsByTagName("tagname");
```

THE DOM

With an HTML page like this..

```
<body>
  <h1 class='title' id='first-title'>First Title</h1>
  <h1 class='title' id='second-title'>Second Title</h1>
  <h1 class='title' id='third-title'>Third Title</h1>
  <h1></h1>
  <h1></h1>
  <ul></ul>
  <script src="./script.js"></script>
</body>
```

THE DOM

If we wanted to see all of our
h1s in our console, we could do
something like this:

THE DOM

```
const allTitles = document.getElementsByTagName("h1");  
  
console.log(allTitles);  
console.log(allTitles.length);
```

THE DOM

In our console, we'll see this:

```
HTMLCollection(5) [h1#first-title.title,  
h1#second-title.title,  
h1#third-title.title, h1, h1, first-title:  
▶ h1#first-title.title, second-title:  
h1#second-title.title, third-title:  
h1#third-title.title]
```

THE DOM

If we wanted to see each individual h1 in our console...

```
for (let i = 0; i < allTitles.length; i++) {  
  console.log(allTitles[i]);  
}
```

THE DOM

In our console, we'll see this:

▶ h1#first-title.title	<u>script.js:27</u>
------------------------	---------------------

▶ h1#second-title.title	<u>script.js:27</u>
-------------------------	---------------------

▶ h1#third-title.title	<u>script.js:27</u>
------------------------	---------------------

▶ h1	<u>script.js:27</u>
------	---------------------

▶ h1	<u>script.js:27</u>
------	---------------------

THE DOM

Similarly to what we just did,
we can also select elements by
class name.

THE DOM

getElementsByClassName(): This method returns an HTMLCollection object.

THE DOM

An `HTMLCollection` is an array like list of HTML elements. The `length` property provides the size of the collection. It is possible to loop through all the `HTMLCollection` elements.

THE DOM

And, just like we did before, we can use `getElementsByClassName` to log all our titles to the console.

THE DOM

```
const allClassTitles = document.getElementsByClassName("title");  
  
console.log(allClassTitles);  
console.log(allClassTitles.length);
```

THE DOM

In our console, we'll see this...

```
HTMLCollection(3) [h1#first-title.title,  
h1#second-title.title,  
h1#third-title.title, first-title:  
▶ h1#first-title.title, second-title:  
h1#second-title.title, third-title:  
h1#third-title.title]
```

THE DOM

Just like we did before, we can loop through the array-like object that is returned to us.

```
for (let i = 0; i < allClassTitles.length; i++) {  
  console.log(allClassTitles[i]);  
}
```

THE DOM

In our console, we'll see this...

script.js:42

```
<h1 class="title" id="first-title">First  
Title</h1>
```

script.js:42

```
<h1 class="title" id="second-title">  
Second Title</h1>
```

script.js:42

```
<h1 class="title" id="third-title">Third  
Title</h1>
```

THE DOM

Similar to what we saw in CSS,
our most specific way of
selecting a DOM node is with id.

THE DOM

The basic syntax looks like this. You'll notice this one isn't plural.

```
document.getElementById("id");
```

THE DOM

We can use it like this:

```
let firstTitle = document.getElementById("first-title");  
console.log(firstTitle);
```

THE DOM

In our console, we'll see this...

```
script.js:52  
<h1 class="title" id="first-title">First  
Title</h1>
```

THE DOM

Another way to do the same stuff is to use the

document.querySelector() method.

It can be used to select an HTML element by its tag name, id or class.

THE DOM

If the tag name or class is used
it selects only the first
element.

THE DOM

```
// select the first available h1 element
let firstTitleElement = document.querySelector("h1");
// select id with first-title
let firstTitleId = document.querySelector("#first-title");
// select the first available element with class title
let firstTitleClass = document.querySelector(".title");
```

THE DOM

querySelectorAll(): can be used to select html elements by their tag name or class. It returns a nodeList which is an array like object.

THE DOM

This supports array methods. We can use, for example, a for loop or forEach method to loop through each nodeList elements.

THE DOM

Here's an example...

```
const allTitlesQuery = document.querySelectorAll("h1");

console.log(allTitlesQuery.length);

for (let i = 0; i < allTitles.length; i++) {
  console.log(allTitles[i]);
}
```

THE DOM

In our console, we'll see this...

```
5 script.js:67
```

```
script.js:70
```

```
<h1 class="title" id="first-title">First  
Title</h1>
```

```
script.js:70
```

```
<h1 class="title" id="second-title">  
Second Title</h1>
```

```
script.js:70
```

```
<h1 class="title" id="third-title">Third  
Title</h1>
```

```
script.js:70
```

```
<h1></h1>
```

```
script.js:70
```

```
<h1></h1>
```