Manipulating the DOM with JavaScript

Or what would we do without JQuery

JQuery allows us to select a node using CSS selector syntax:

```
var n = $("article#first p.summary");
```

This code lets us find all paragraphs with the class summary in the article with id “first”

IRON HACK

The native equivalent in JavaScript is:

```
var n = document.querySelectorAll("article#first p.summary");
```

This code lets us find all paragraphs with the class summary in the article with id "first"

IRON HACK

If you remember, to select this

```
<p>Hi.</p>
```

with JQuery, you use

```
$('p')
```

IRON HACK

With JavaScript, you use

```
elements = document.getElementsByTagName('p');
```

Also, you could select elements from the DOM searching them by other attributes...

IRON HACK

document.getElementById();

```
<!--HTML-->
<div id="features">
    <p>He has big ears</p>
</div>

<script>
    var attr = document.getElementById("features");
</script>
```

document.getElementsByName();

```
<!--HTML-->
<div name="song">
    <p>A boy named Sue </p>
</div>

<script>
    var johnny_cash = document.getElementsByName("song");
</script>
```

IRON HACK

document.getElementsByClassName();

```
<!--HTML-->
<button class="btn-green">Start</button>

<script>
   var johnny_cash = document.getElementsByClassName("btn-green");
</script>
```

IRON HACK

document.querySelector();


```
<!--HTML→
<h1>Shopping list</h1>
<ul>
     <li>Honey</li>
     <li>Sugar</li>
</ul>
<ul>
     <li>Salt</li>
     <li>Soy</li>
</ul>

<script>
   var list = document.querySelector("ul");
</script>
```

IRON HACK

document.querySelectorAll();

```
<!--HTML-->
<h1>Shopping list</h1>
<ul>
     <li>Honey</li>
     <li>Sugar</li>
</ul>
<ul>
     <li>Salt</li>
     <li>Soy</li>
</ul>

<script>
   var lists = document.querySelectorAll("ul");
</script>
```

IRON HACK

Adding elements

Create a HTML file with one <p> element. It could be something as simple as:

```
<p id = 'hi'>Good morning, world!</p>
```

IRON HACK

Adding elements

To add a new element to the HTML DOM, you must *first* create the element (element node) first,

```
var node = document.createElement('p');
var ih = document.createTextNode('This is Ironhack!!!');
node.appendChild(ih);
```

This creates a new empty element with the tag <p>

In this lines the node is created and appended to the element node

IRON HACK

Adding elements

Now append the created element to the existing paragraph

```
document.getElementsByTagName('p')[0].appendChild(node);
```

Why are we appending the node to the first object of an array?

IRON HACK

In addition to append, you can also use the prepend, before and after functions to add elements. Try adding the following elements to the container using these functions.

How are they different?

prepend()

```
var node = document.createElement('p');
var ih = document.createTextNode('This is Ironhack!!!');
document.body.insertBefore(node, document.body.childNodes[0]);
```

before()

```
var my_sentence = document.getElementById('p');
span.innerHTML = '***';
my_sentence.parentNode.insertBefore(span, my_sentence);
```

IRON HACK

after()

```
var my_sentence = document.getElementById('p');
span.innerHTML = '***';
my_sentence.parentNode.insertBefore(span,
my_sentence.nextSibling);
```

IRON HACK

Removing elements

In JQuery, you can remove the selected elements from the document by calling the `.remove()` method.

```
$('p').remove();
$('button').remove();
```

IRON HACK

In JavaScript it is a little bit more complex...

```
<!--HTML-->
<h1>Shopping list</h1>
<ul>
     <li>Honey</li>
     <li>Sugar</li>
</ul>
<ul>
     <li>Salt</li>
     <li>Soy</li>
</ul>

<script>
   var title = document.body.getElementByTagName("h1")[0];
   title.parentNode.removeChild("h1");
</script>
```