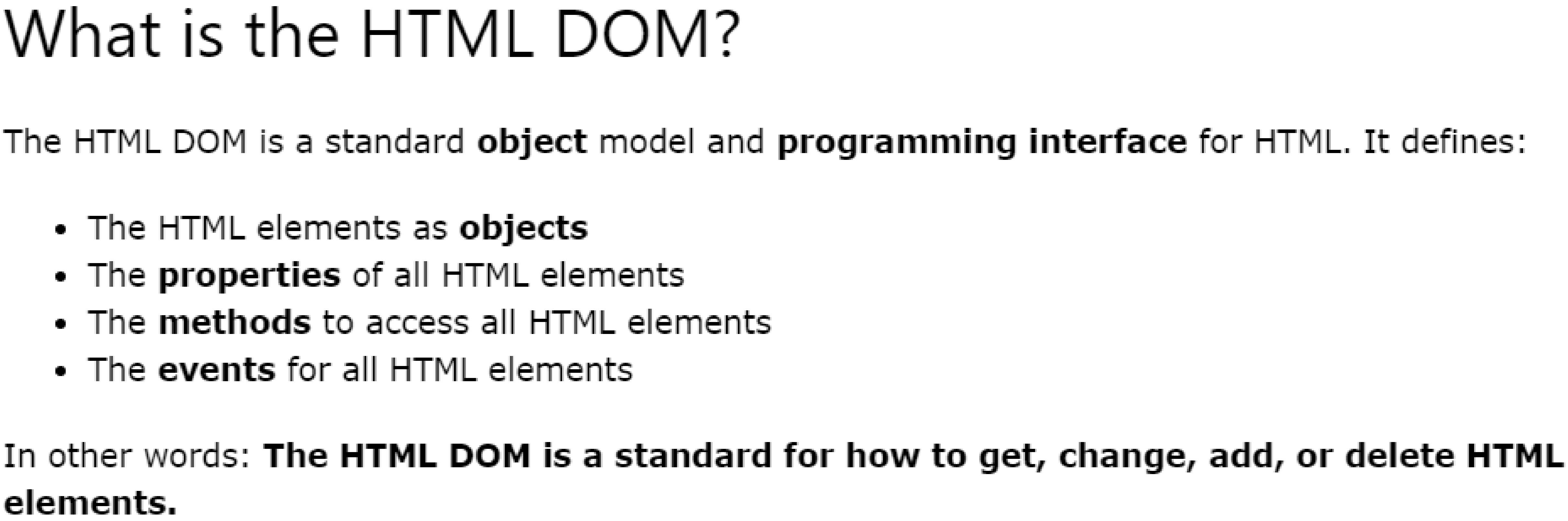
HTML DOM

https://www.w3schools.com/js/js_htmldom.asp

2025 Fall

w3schools



What is the HTML DOM?

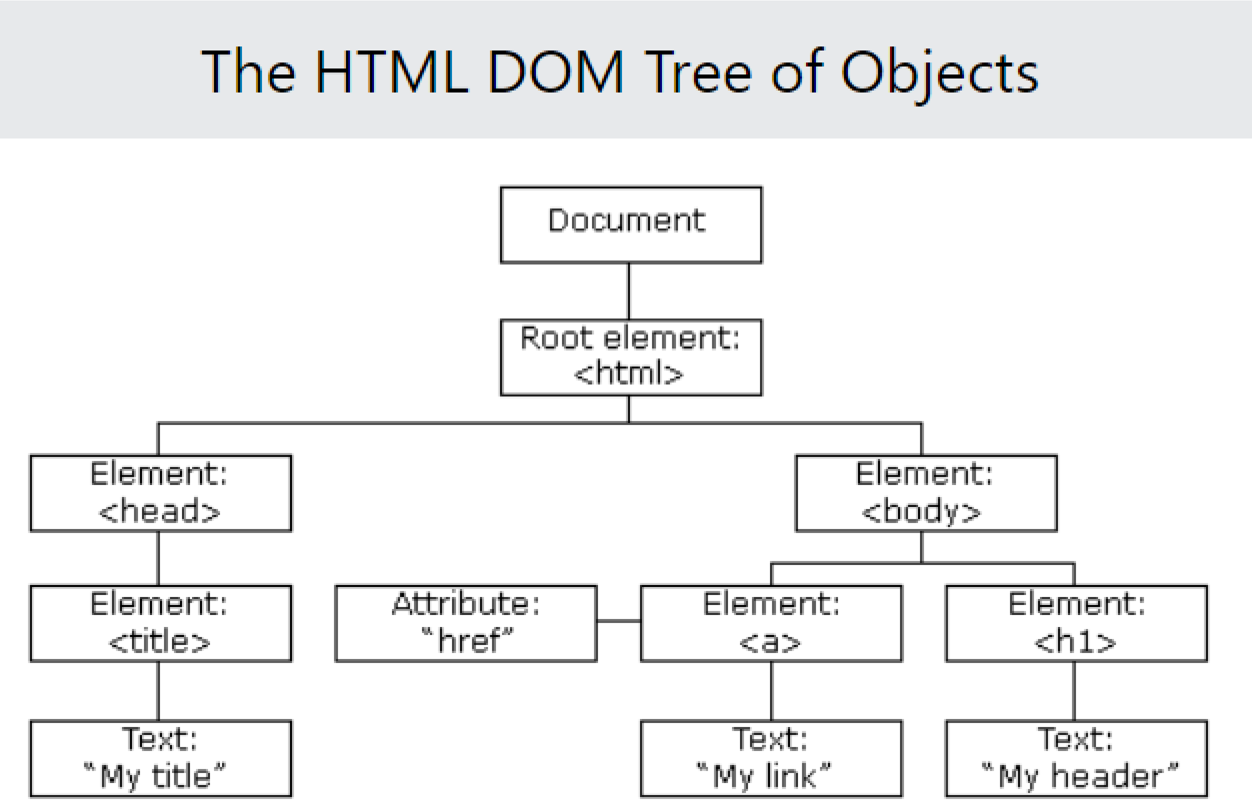
The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

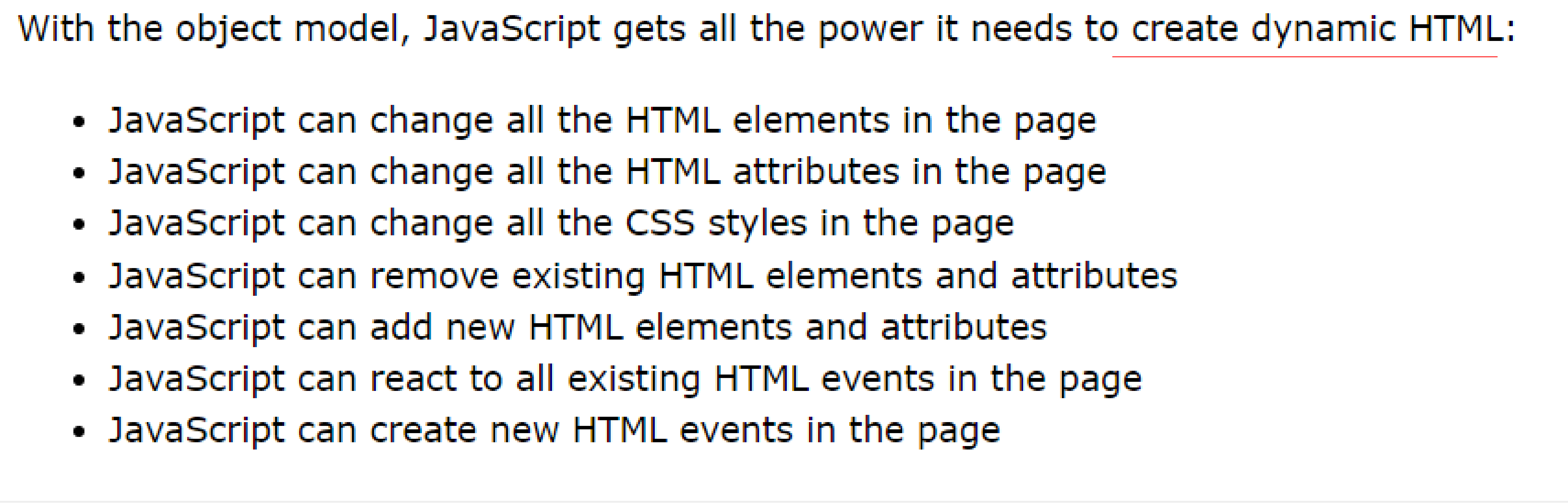
In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page. The **HTML DOM** model is constructed as a tree of **Objects**:



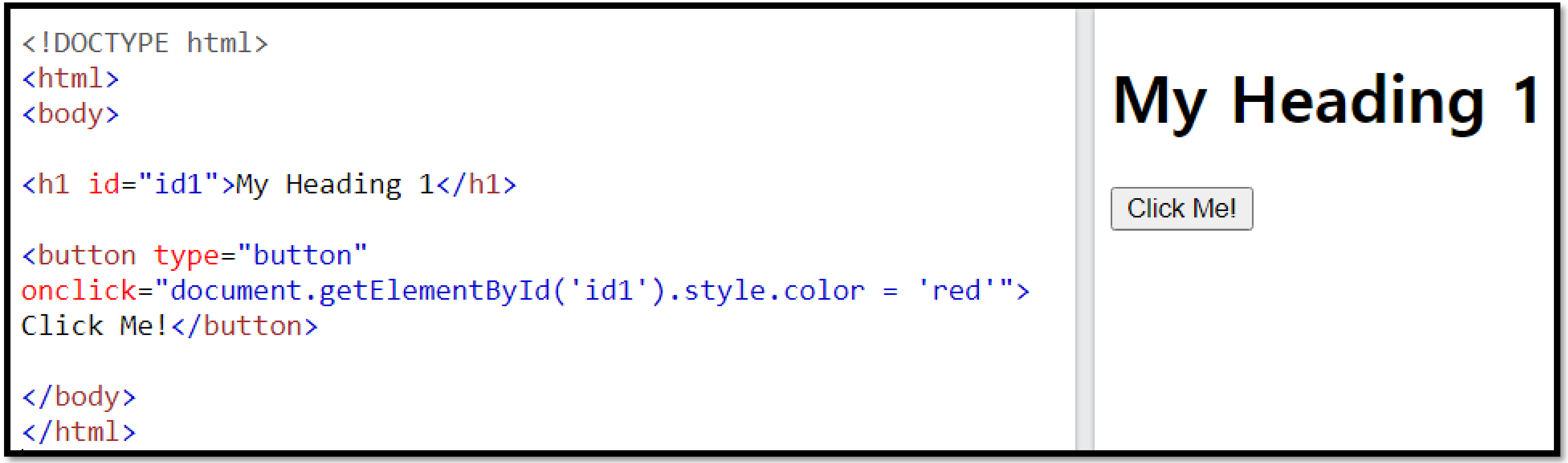
The HTML DOM Tree of Objects



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

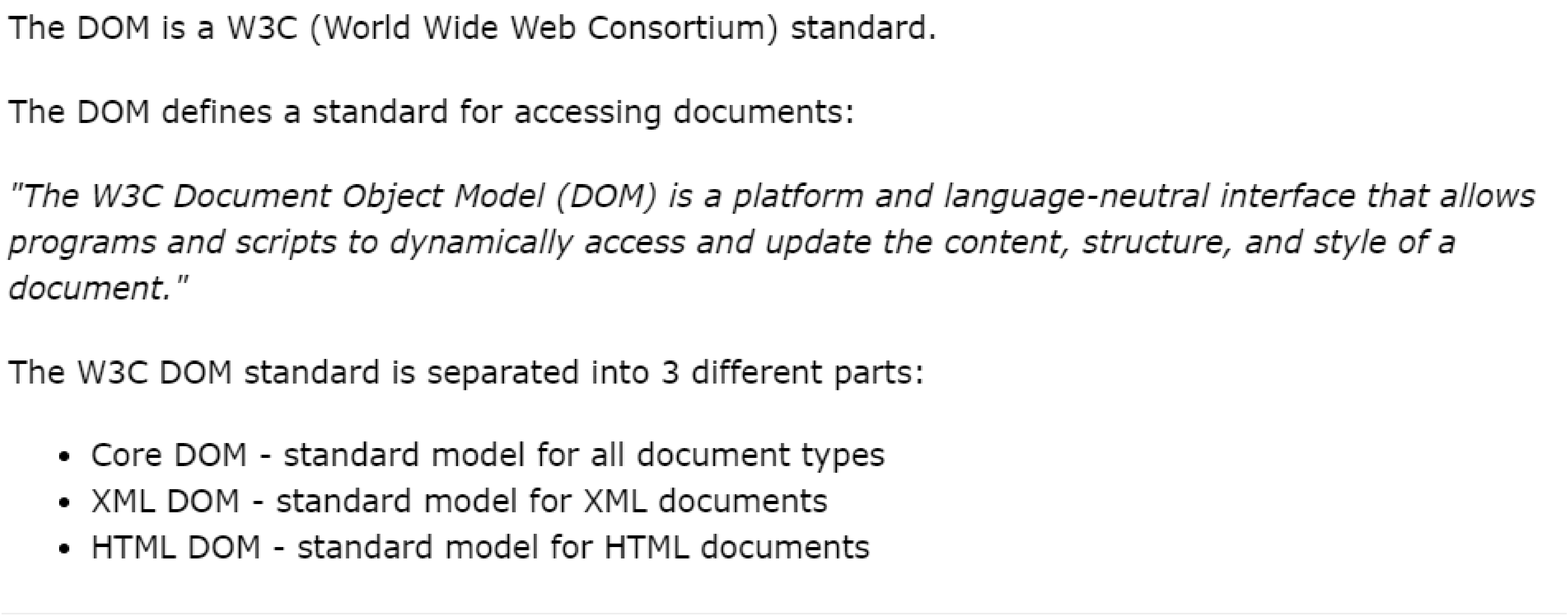
https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_color2



```html
<!DOCTYPE html>
<html>
<body>

<h1 id="id1">My Heading 1</h1>

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>

</body>
</html>
```

Using Events

The HTML DOM allows you to execute code when an event occurs.



The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The W3C DOM standard is separated into 3 different parts:

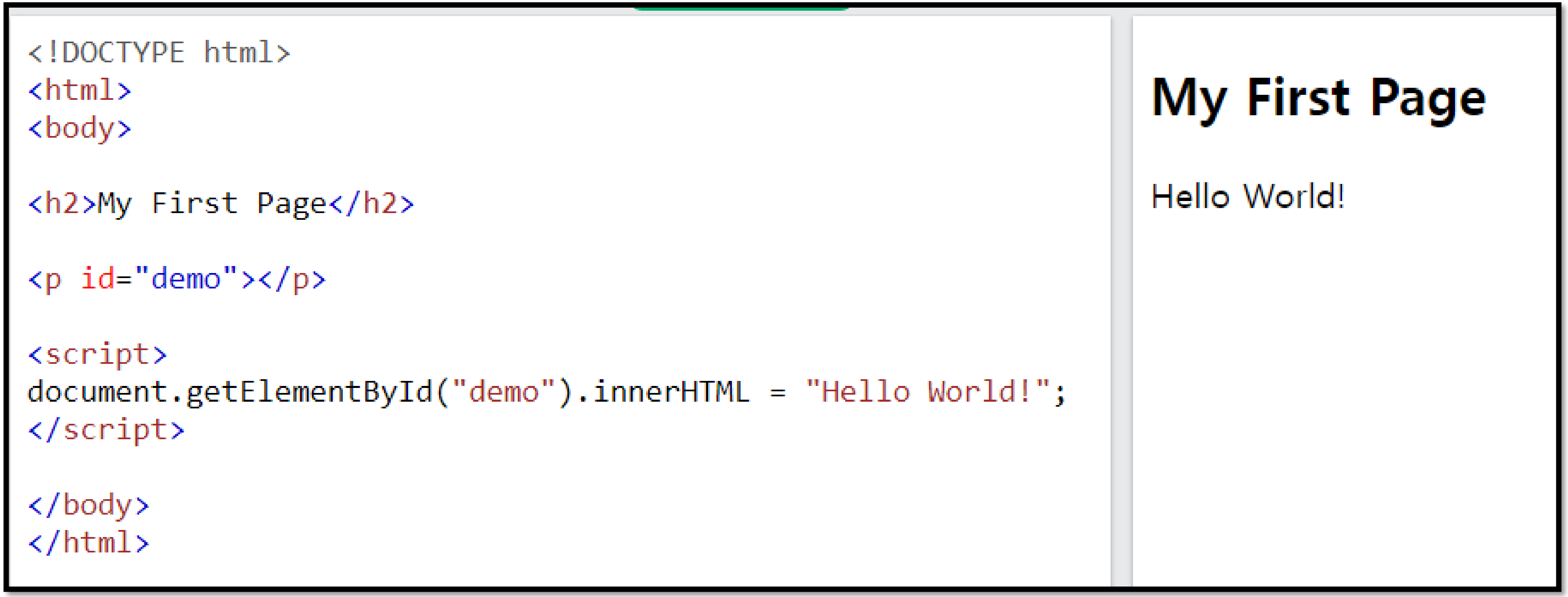
- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

HTML DOM methods are **actions** you can perform (on HTML Elements).

HTML DOM properties are **values** (of HTML Elements) that you can set or change.

The following example changes the content (the `innerHTML`) of the `<p>` element with `id="demo"`:

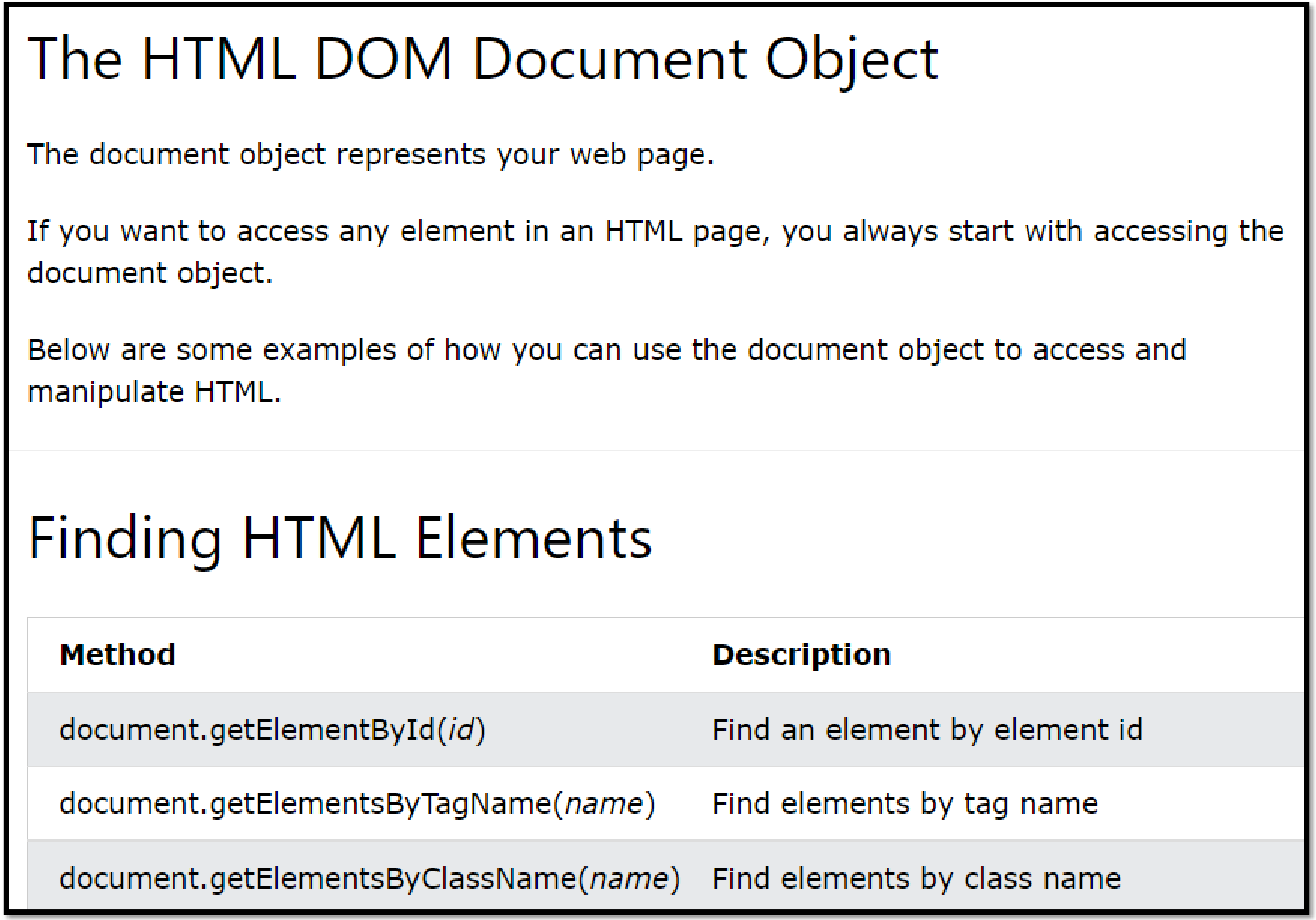
https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method



```
<!DOCTYPE html>
<html>
<body>

<h2>My First Page</h2>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

My First Page

Hello World!



The HTML DOM Document Object

The document object represents your web page.

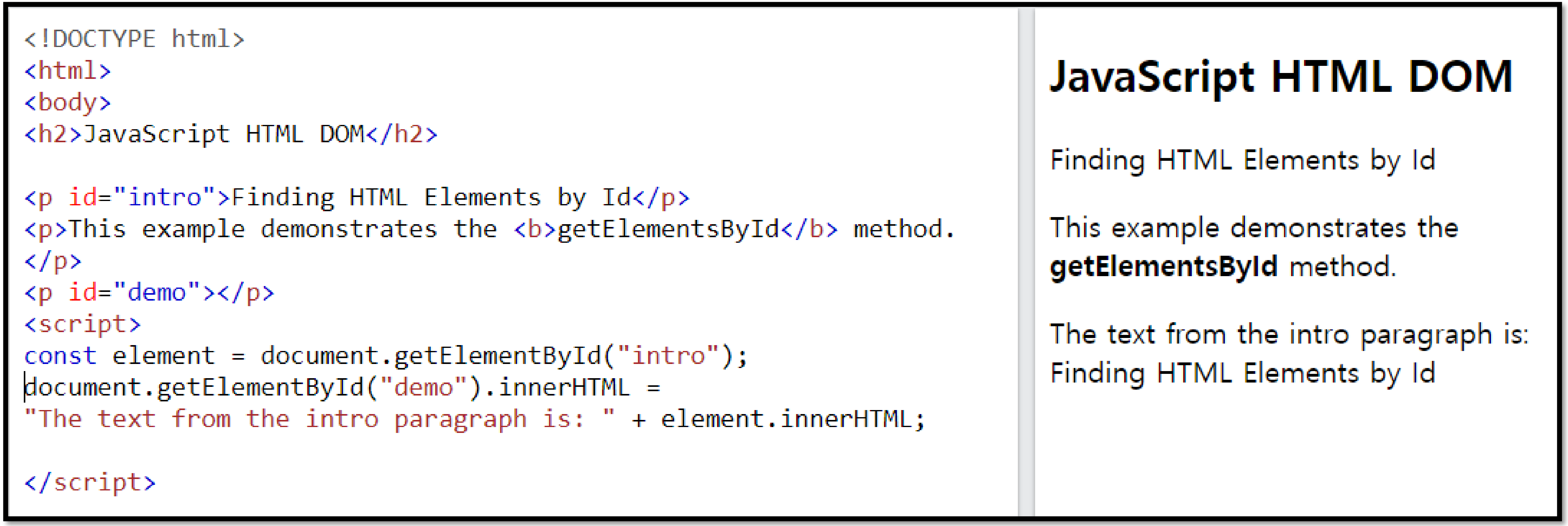
If you want to access any element in an HTML page, you always start with accessing the document object.

Below are some examples of how you can use the document object to access and manipulate HTML.

Finding HTML Elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementbyid



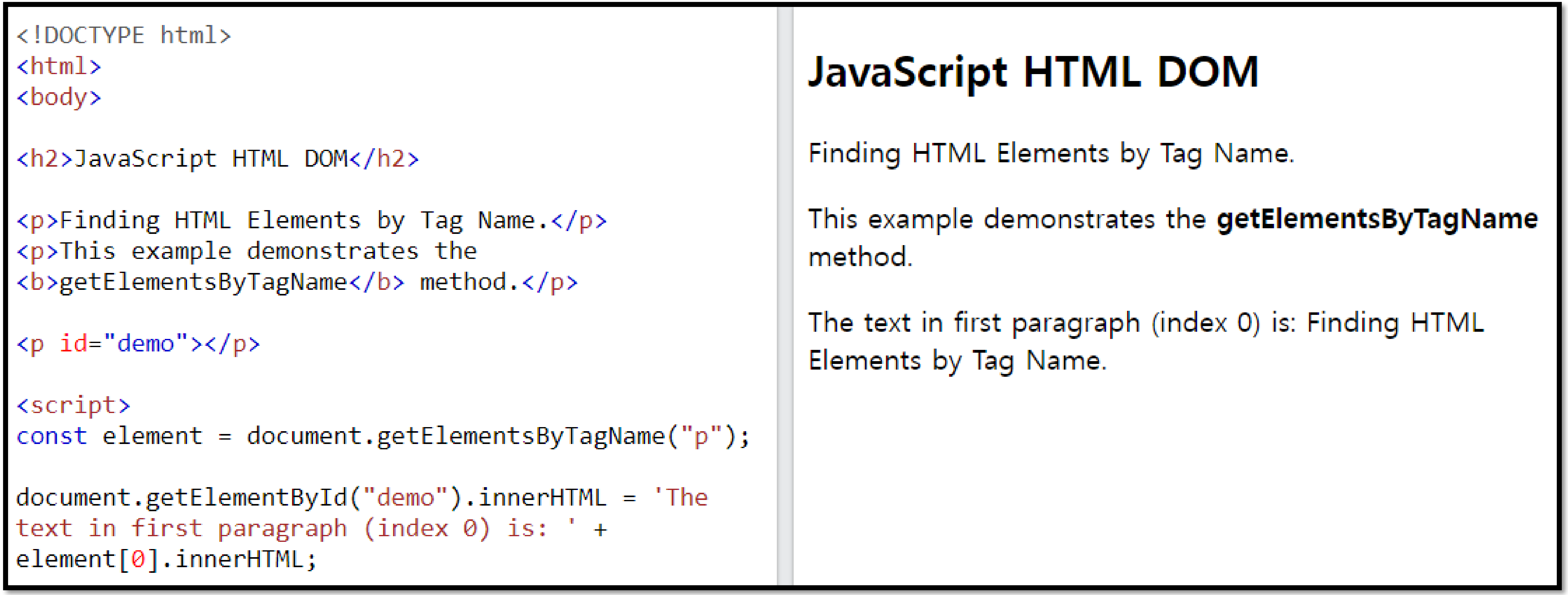
```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>

<p id="intro">Finding HTML Elements by Id</p>
<p>This example demonstrates the <b>getElementsById</b> method.
</p>
<p id="demo"></p>
<script>
const element = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"The text from the intro paragraph is: " + element.innerHTML;

</script>
```

JavaScript HTML DOM

Finding HTML Elements by Id

This example demonstrates the **getElementsById** method.

The text from the intro paragraph is: Finding HTML Elements by Id https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbytagname2



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<p>Finding HTML Elements by Tag Name.</p>
<p>This example demonstrates the
<b>getElementsByTagName</b> method.</p>

<p id="demo"></p>

<script>
const element = document.getElementsByTagName("p");

document.getElementById("demo").innerHTML = 'The
text in first paragraph (index 0) is: ' +
element[0].innerHTML;
```

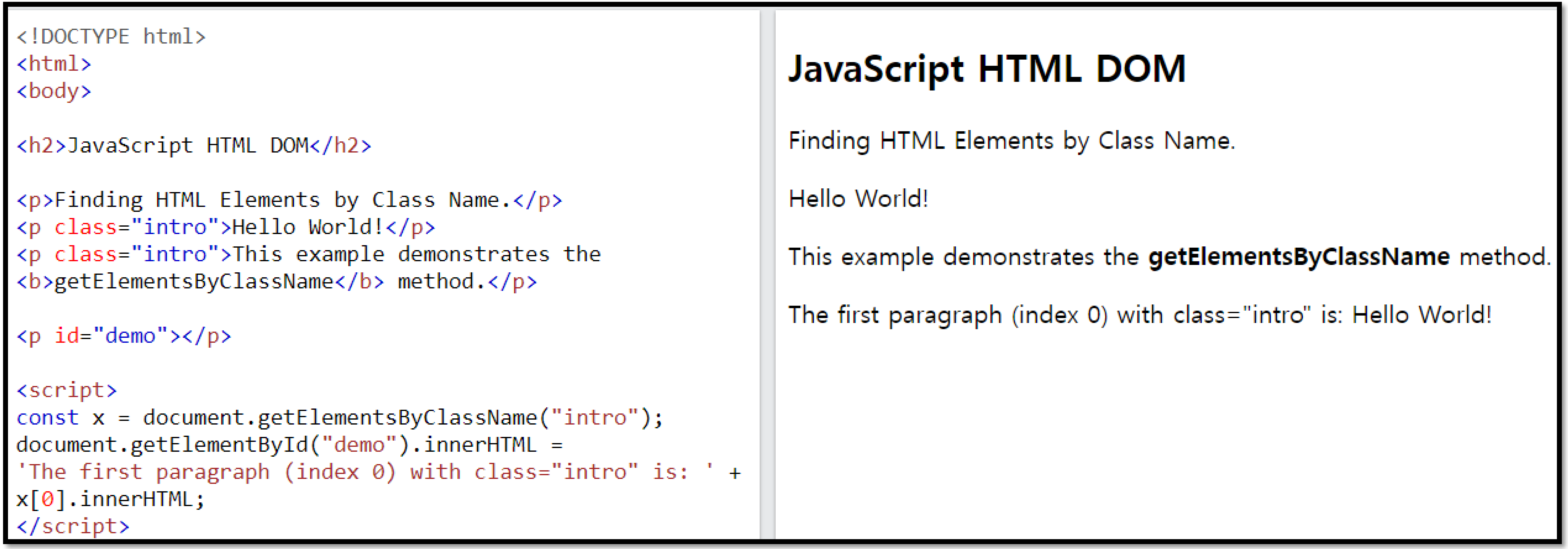
JavaScript HTML DOM

Finding HTML Elements by Tag Name.

This example demonstrates the **getElementsByTagName** method.

The text in first paragraph (index 0) is: Finding HTML Elements by Tag Name.

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbyclassname



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<p>Finding HTML Elements by Class Name.</p>
<p class="intro">Hello World!</p>
<p class="intro">This example demonstrates the
<b>getElementsByClassName</b> method.</p>

<p id="demo"></p>

<script>
const x = document.getElementsByClassName("intro");
document.getElementById("demo").innerHTML =
'The first paragraph (index 0) with class="intro" is: ' +
x[0].innerHTML;
</script>
```

JavaScript HTML DOM

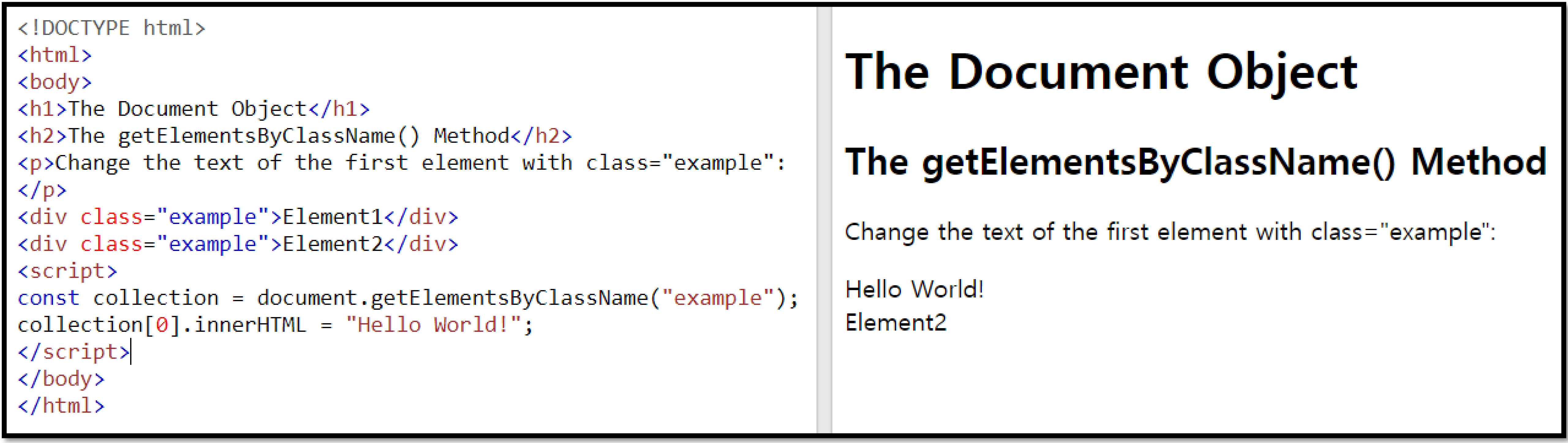
Finding HTML Elements by Class Name.

Hello World!

This example demonstrates the **getElementsByClassName** method.

The first paragraph (index 0) with class="intro" is: Hello World!

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_getelementsbyclassname



```
<!DOCTYPE html>
<html>
<body>
<h1>The Document Object</h1>
<h2>The getElementsByClassName() Method</h2>
<p>Change the text of the first element with class="example":
</p>
<div class="example">Element1</div>
<div class="example">Element2</div>
<script>
const collection = document.getElementsByClassName("example");
collection[0].innerHTML = "Hello World!";
</script>
</body>
</html>
```

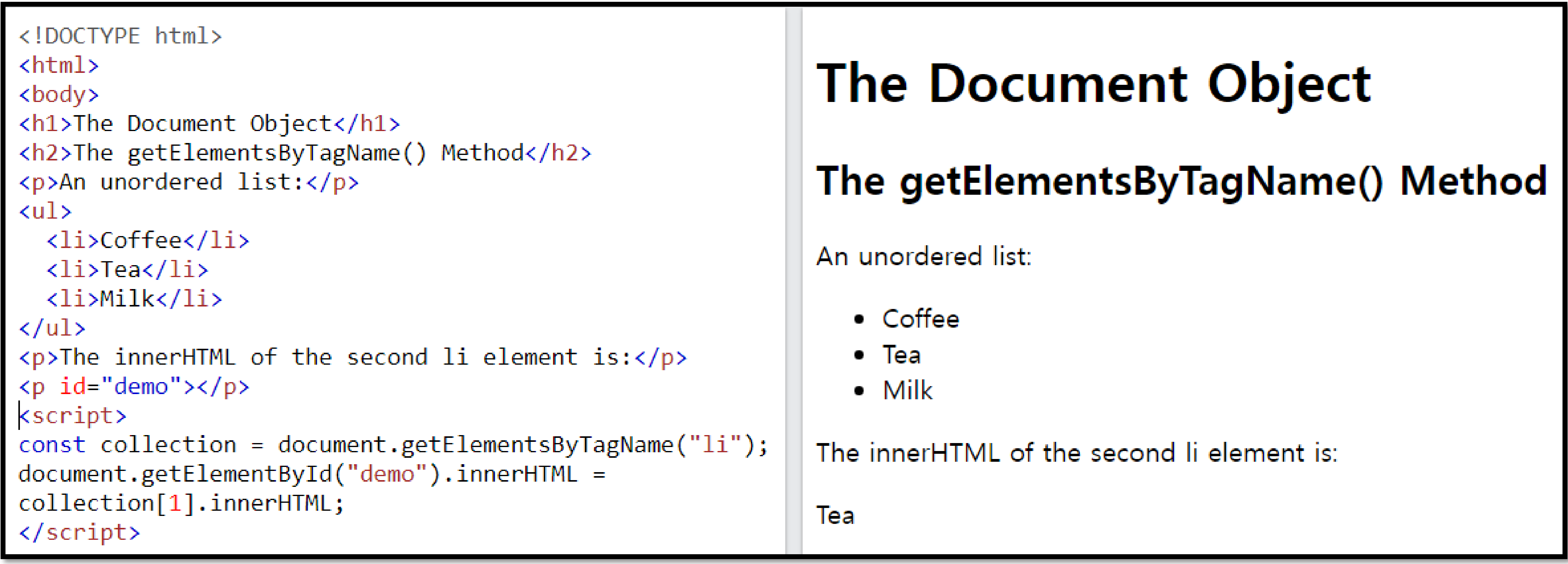
The Document Object

The getElementsByClassName() Method

Change the text of the first element with class="example":

Hello World!
Element2

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_document_getelementsbytagname



```
<!DOCTYPE html>
<html>
<body>
<h1>The Document Object</h1>
<h2>The getElementsByTagName() Method</h2>
<p>An unordered list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
<p>The innerHTML of the second li element is:</p>
<p id="demo"></p>
<script>
const collection = document.getElementsByTagName("li");
document.getElementById("demo").innerHTML =
collection[1].innerHTML;
</script>
```

The Document Object

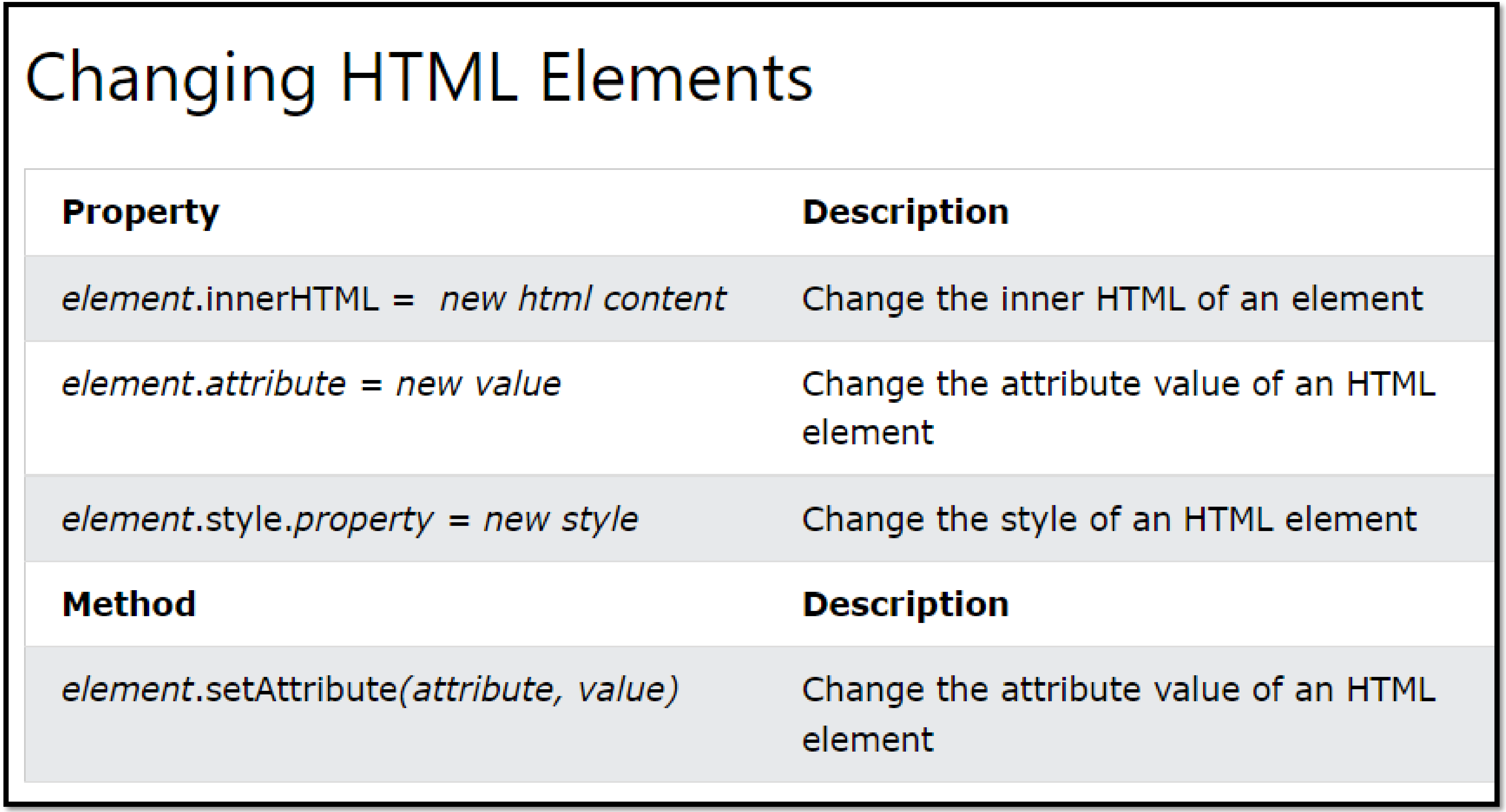
The getElementsByTagName() Method

An unordered list:

- Coffee
- Tea
- Milk

The innerHTML of the second li element is:

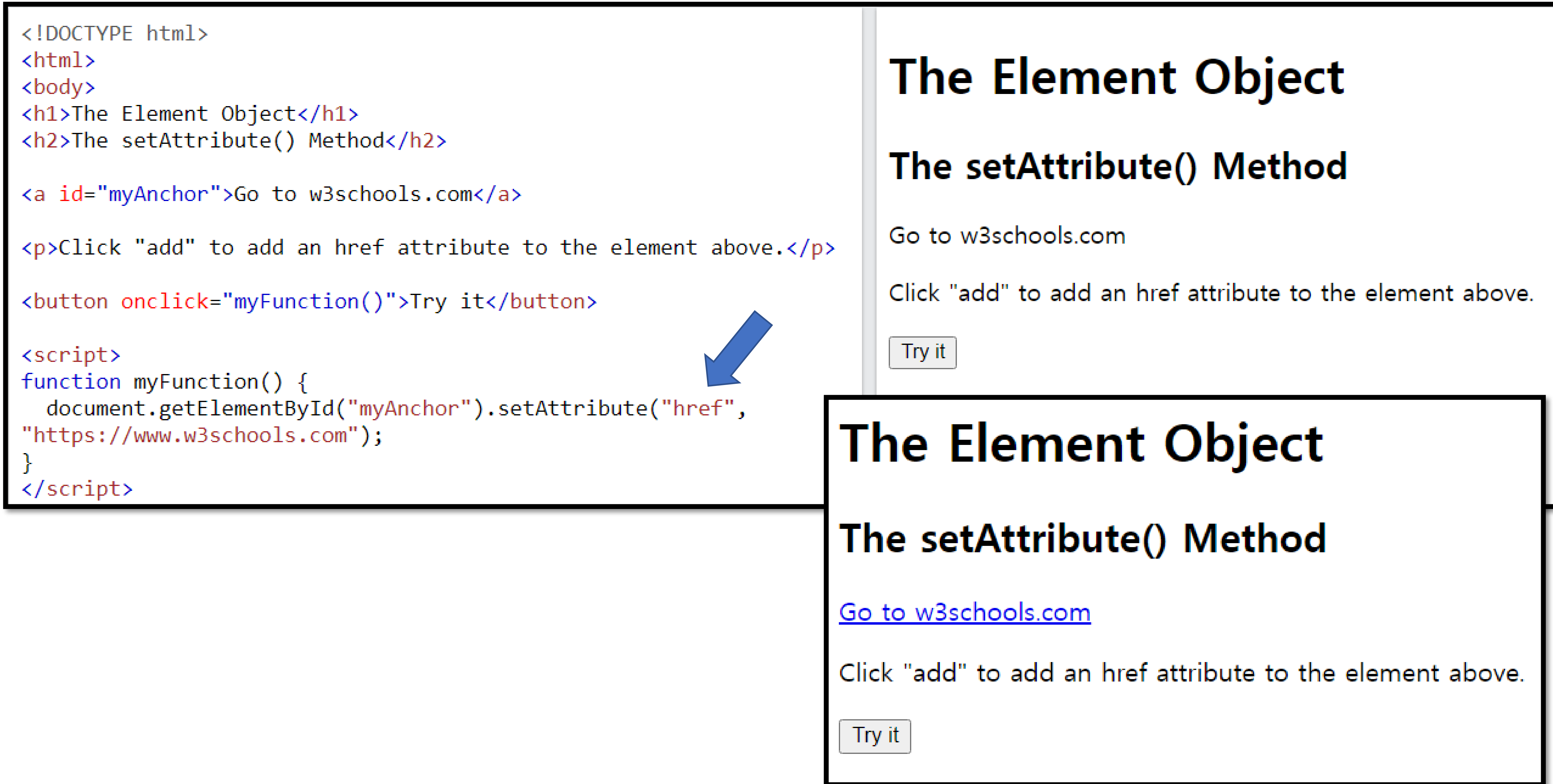
Tea



Changing HTML Elements

Property	**Description**
element.innerHTML = *new html content*	Change the inner HTML of an element
element.*attribute* = *new value*	Change the attribute value of an HTML element
element.style.*property* = *new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_setattribute2



```
<!DOCTYPE html>
<html>
<body>
<h1>The Element Object</h1>
<h2>The setAttribute() Method</h2>

<a id="myAnchor">Go to w3schools.com</a>

<p>Click "add" to add an href attribute to the element above.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
  document.getElementById("myAnchor").setAttribute("href",
"https://www.w3schools.com");
}
</script>
```

The Element Object

The setAttribute() Method

Go to w3schools.com

Click "add" to add an href attribute to the element above.

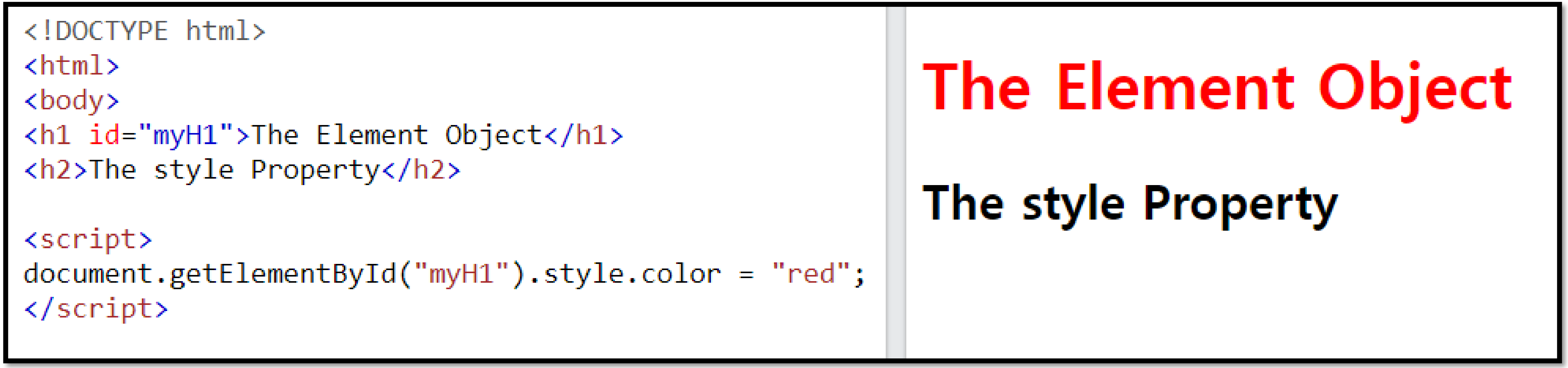
The Element Object

The setAttribute() Method

Go to w3schools.com

Click "add" to add an href attribute to the element above.

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_element_style

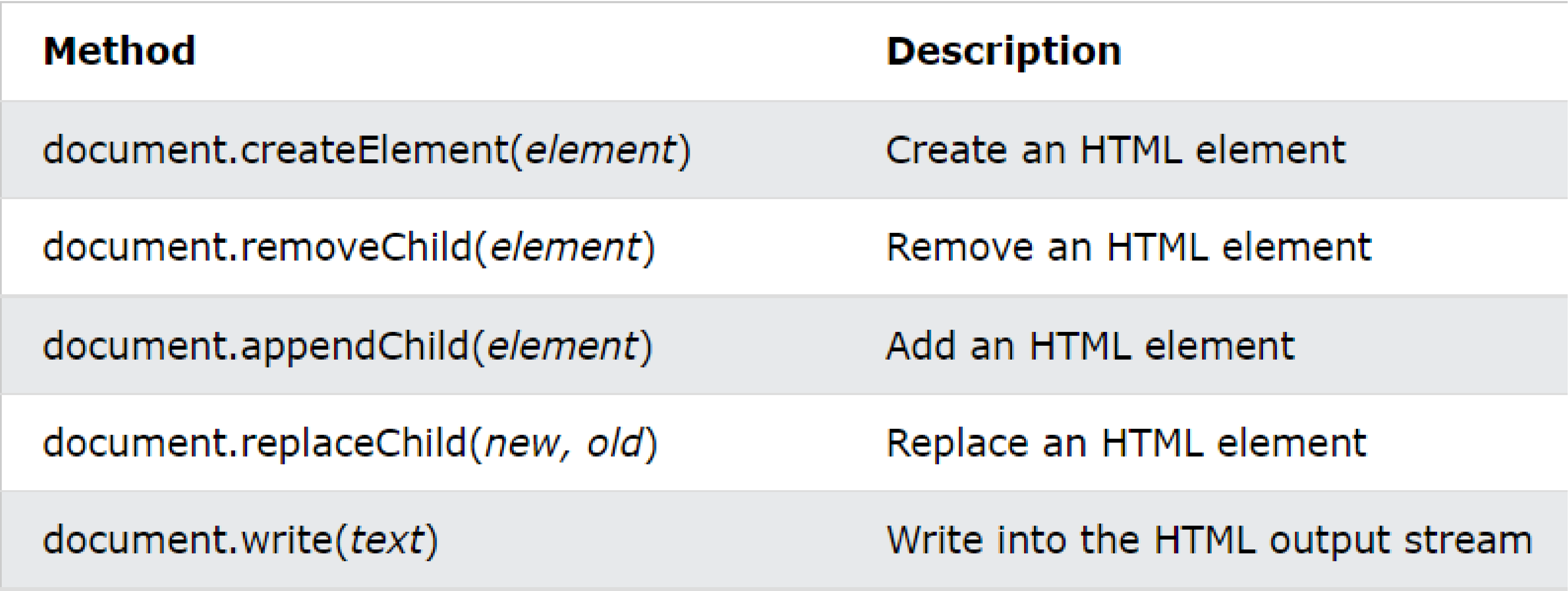


```
<!DOCTYPE html>
<html>
<body>
<h1 id="myH1">The Element Object</h1>
<h2>The style Property</h2>

<script>
document.getElementById("myH1").style.color = "red";
</script>
```

The Element Object

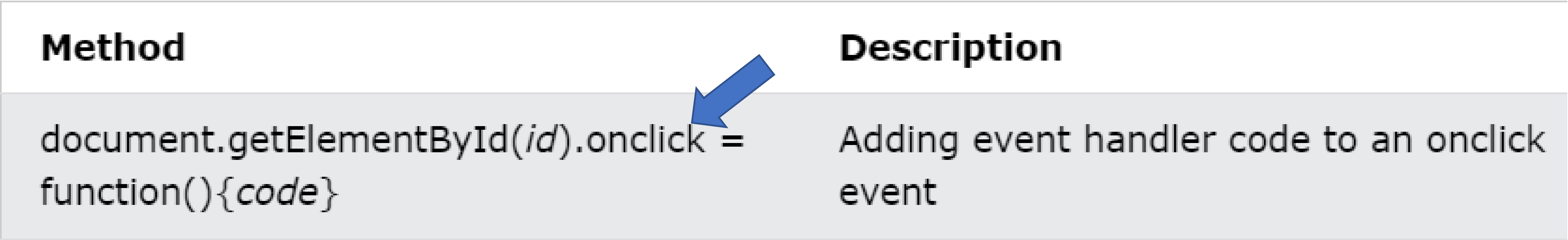
The style Property

Adding and Deleting Elements

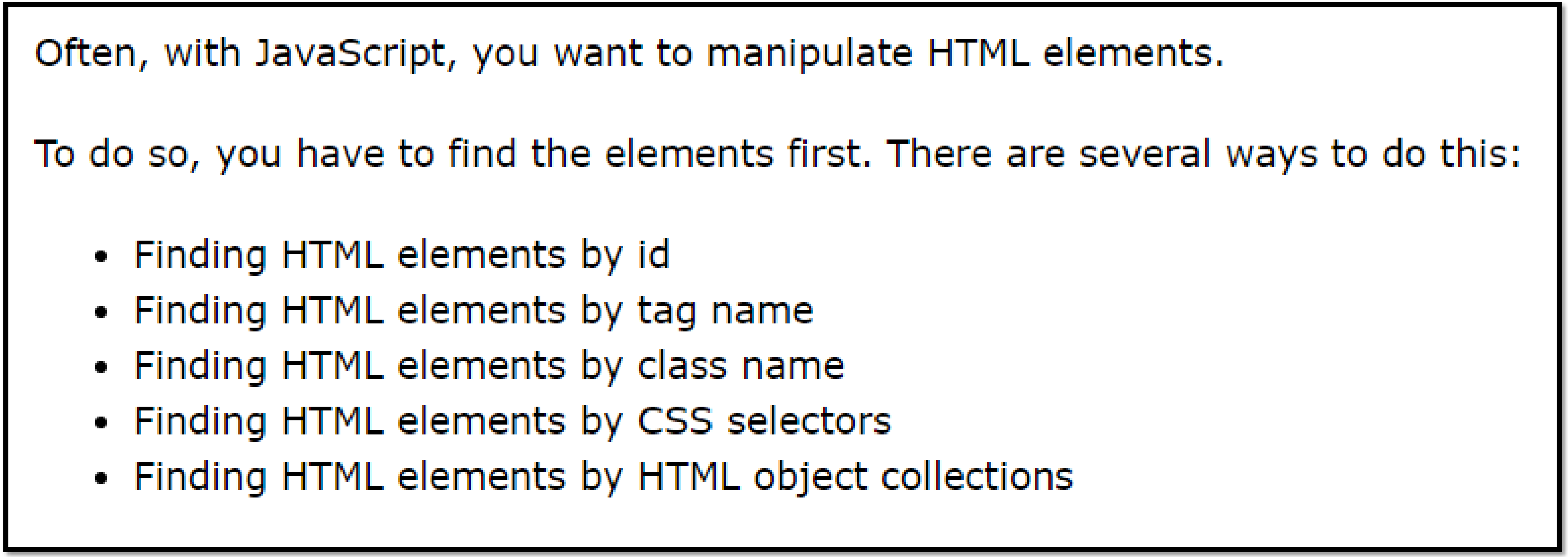


Method	**Description**
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new, old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

Adding Events Handlers



Method	Description
document.getElementById(*id*).onclick = function(){*code*}	Adding event handler code to an onclick event

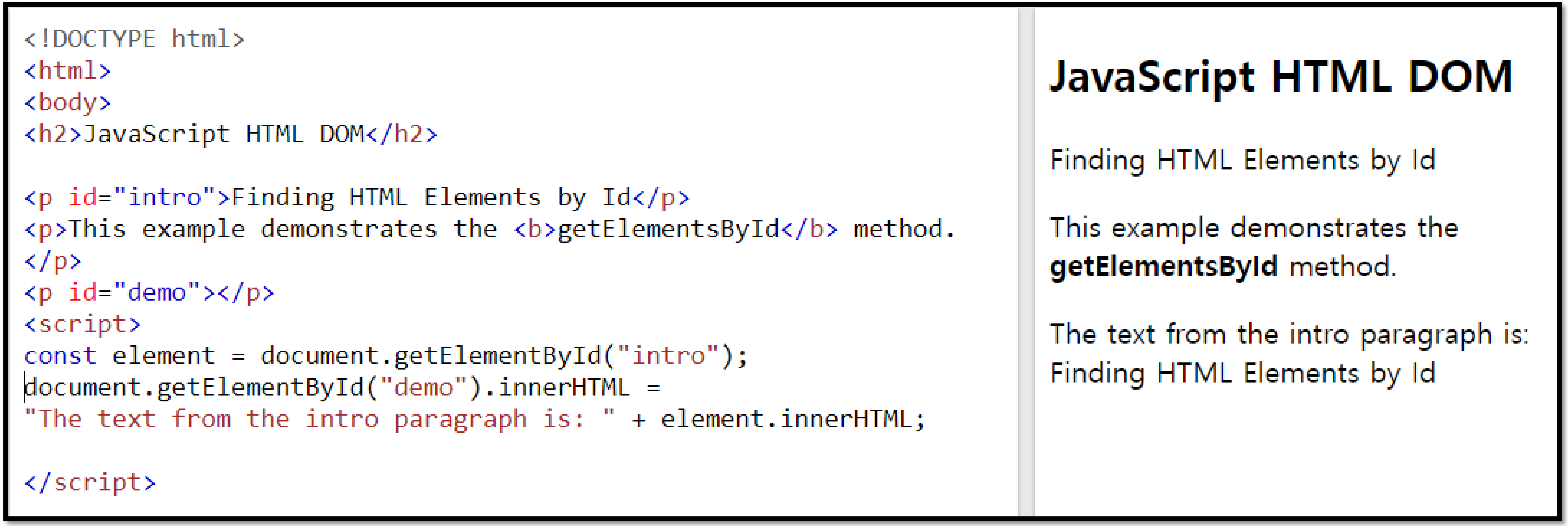


Often, with JavaScript, you want to manipulate HTML elements.

To do so, you have to find the elements first. There are several ways to do this:

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementbyid



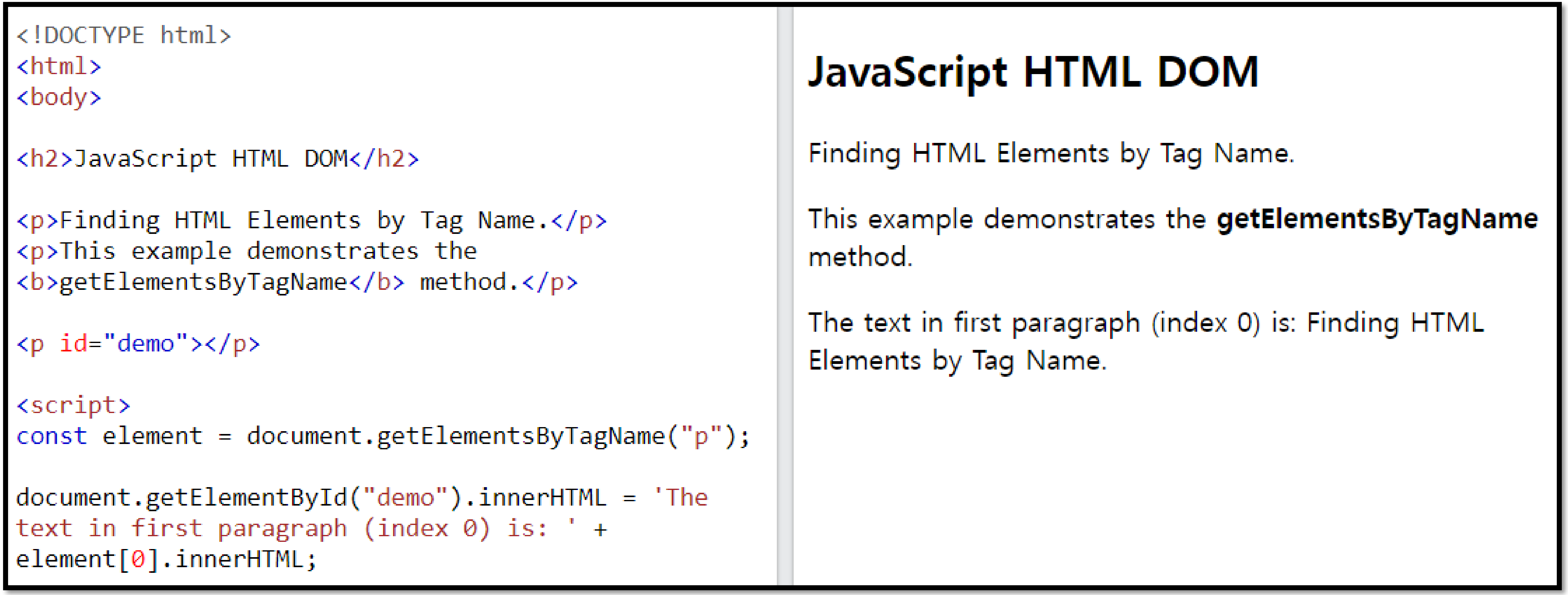
```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>

<p id="intro">Finding HTML Elements by Id</p>
<p>This example demonstrates the <b>getElementsById</b> method.
</p>
<p id="demo"></p>
<script>
const element = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"The text from the intro paragraph is: " + element.innerHTML;

</script>
```

JavaScript HTML DOM

Finding HTML Elements by Id

This example demonstrates the **getElementsById** method.

The text from the intro paragraph is: Finding HTML Elements by Id https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbytagname2



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<p>Finding HTML Elements by Tag Name.</p>
<p>This example demonstrates the
<b>getElementsByTagName</b> method.</p>

<p id="demo"></p>

<script>
const element = document.getElementsByTagName("p");

document.getElementById("demo").innerHTML = 'The
text in first paragraph (index 0) is: ' +
element[0].innerHTML;
```

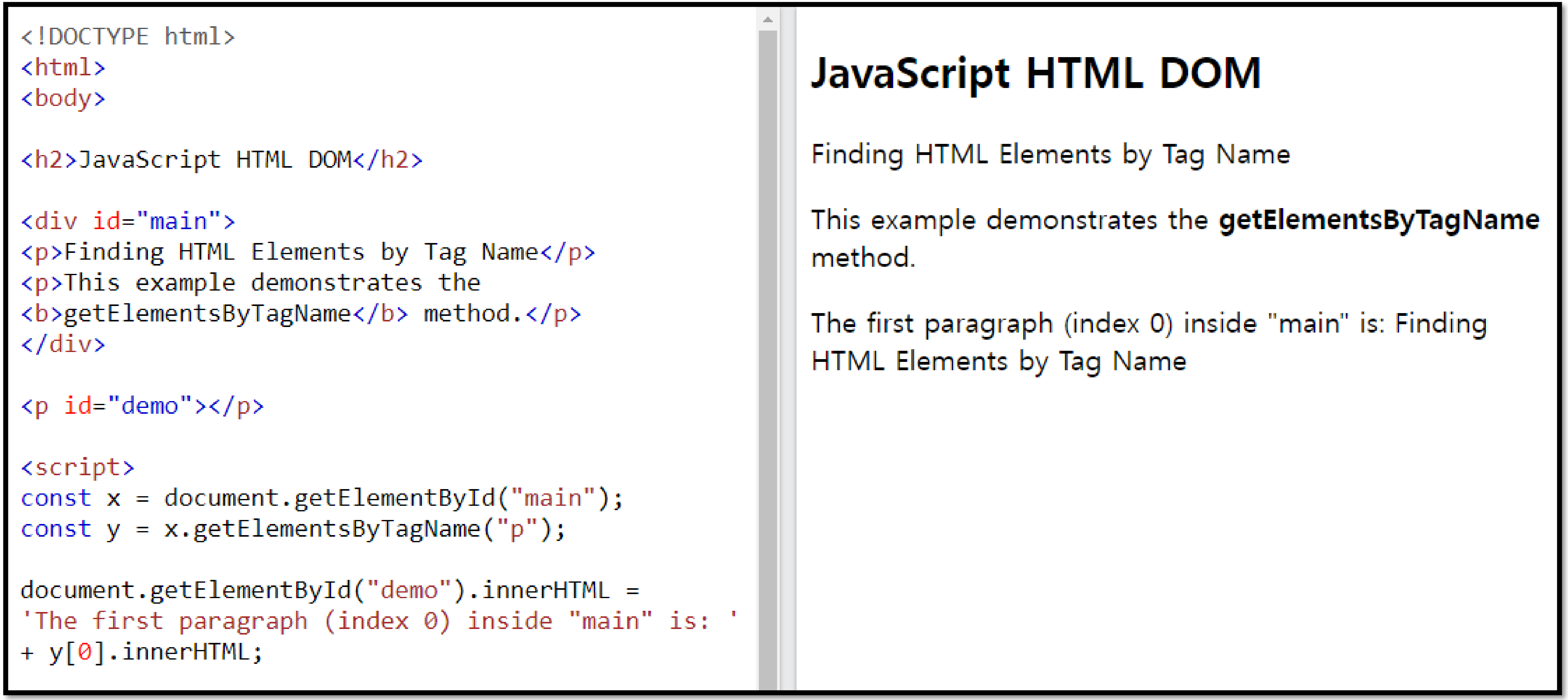
JavaScript HTML DOM

Finding HTML Elements by Tag Name.

This example demonstrates the **getElementsByTagName** method.

The text in first paragraph (index 0) is: Finding HTML Elements by Tag Name.

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbytagname



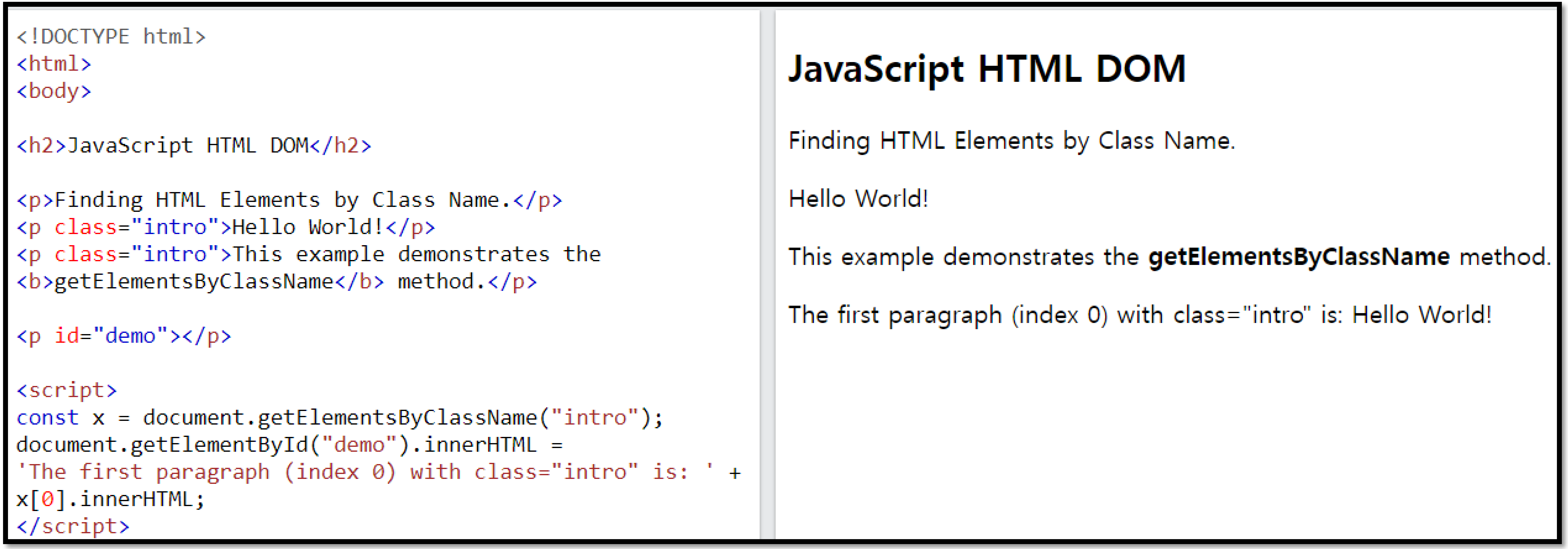
```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<div id="main">
<p>Finding HTML Elements by Tag Name</p>
<p>This example demonstrates the
<b>getElementsByTagName</b> method.</p>
</div>

<p id="demo"></p>

<script>
const x = document.getElementById("main");
const y = x.getElementsByTagName("p");

document.getElementById("demo").innerHTML =
'The first paragraph (index 0) inside "main" is: '
+ y[0].innerHTML;
```

JavaScript HTML DOM

Finding HTML Elements by Tag Name

This example demonstrates the **getElementsByTagName** method.

The first paragraph (index 0) inside "main" is: Finding HTML Elements by Tag Name https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbyclassname



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<p>Finding HTML Elements by Class Name.</p>
<p class="intro">Hello World!</p>
<p class="intro">This example demonstrates the
<b>getElementsByClassName</b> method.</p>

<p id="demo"></p>

<script>
const x = document.getElementsByClassName("intro");
document.getElementById("demo").innerHTML =
'The first paragraph (index 0) with class="intro" is: ' +
x[0].innerHTML;
</script>
```

JavaScript HTML DOM

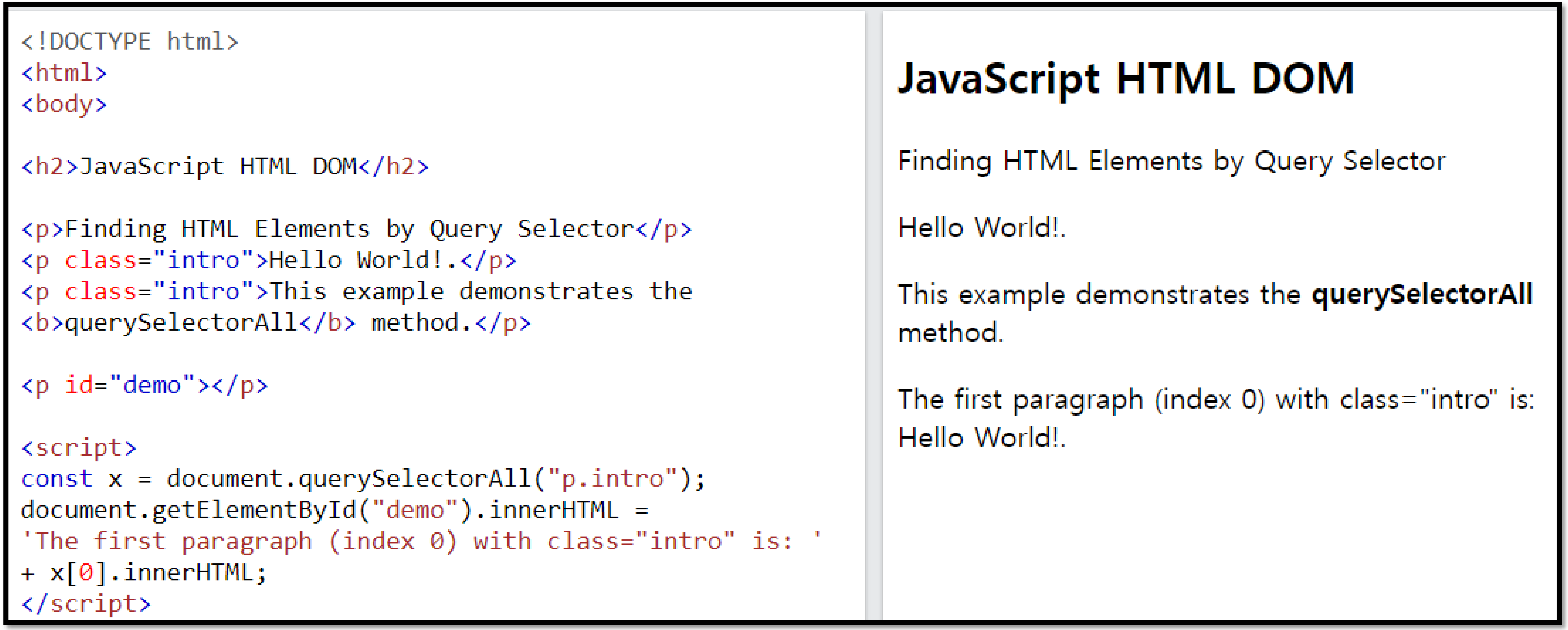
Finding HTML Elements by Class Name.

Hello World!

This example demonstrates the **getElementsByClassName** method.

The first paragraph (index 0) with class="intro" is: Hello World!

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_queryselectorall



```html
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>

<p>Finding HTML Elements by Query Selector</p>
<p class="intro">Hello World!.</p>
<p class="intro">This example demonstrates the
<b>querySelectorAll</b> method.</p>

<p id="demo"></p>

<script>
const x = document.querySelectorAll("p.intro");
document.getElementById("demo").innerHTML =
'The first paragraph (index 0) with class="intro" is: '
+ x[0].innerHTML;
</script>
```

JavaScript HTML DOM

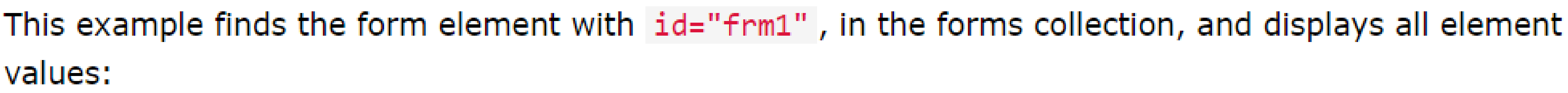
Finding HTML Elements by Query Selector

Hello World!.

This example demonstrates the **querySelectorAll** method.

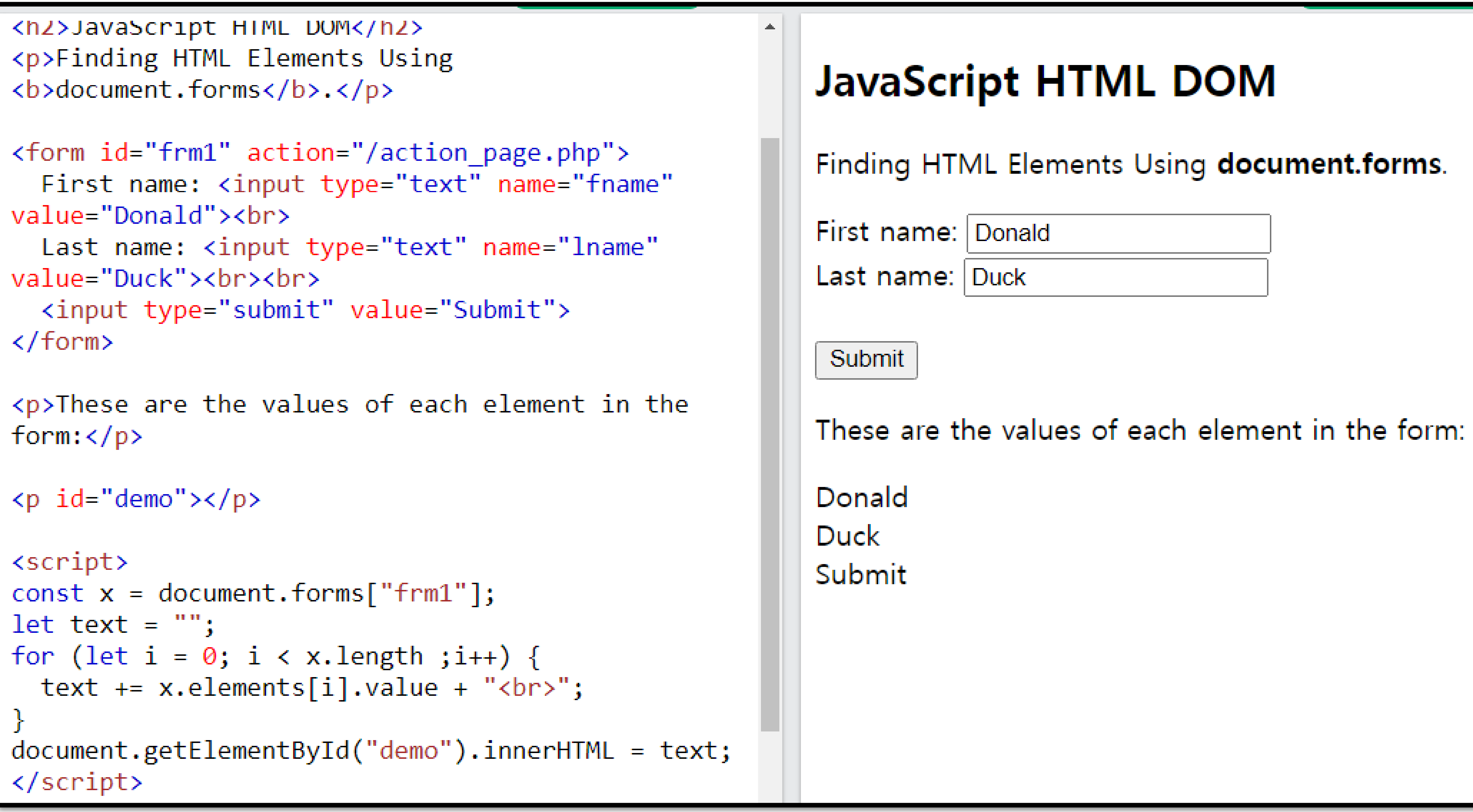
The first paragraph (index 0) with class="intro" is: Hello World!.

Finding HTML Elements by HTML Object Collections



This example finds the form element with `id="frm1"`, in the forms collection, and displays all element values:

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_form_elements



```
<h2>JavaScript HTML DOM</h2>
<p>Finding HTML Elements Using
<b>document.forms</b>.</p>

<form id="frm1" action="/action_page.php">
  First name: <input type="text" name="fname"
value="Donald"><br>
  Last name: <input type="text" name="lname"
value="Duck"><br><br>
  <input type="submit" value="Submit">
</form>

<p>These are the values of each element in the
form:</p>

<p id="demo"></p>

<script>
const x = document.forms["frm1"];
let text = "";
for (let i = 0; i < x.length ;i++) {
  text += x.elements[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
```

JavaScript HTML DOM

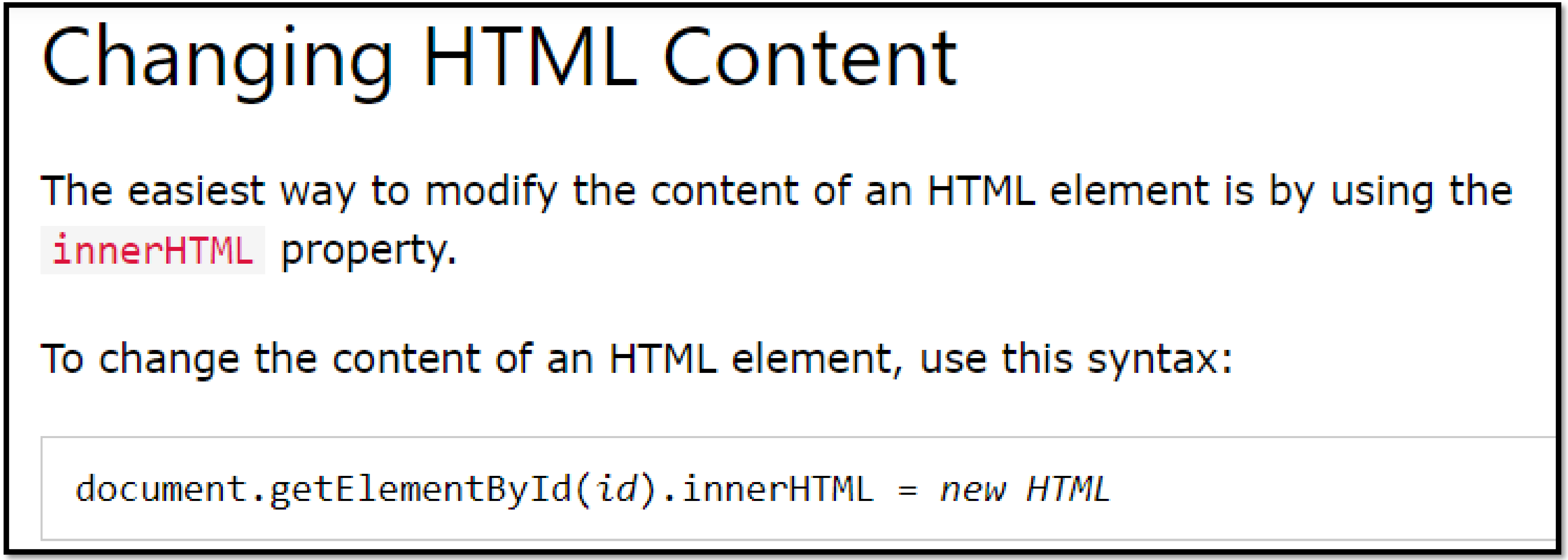
Finding HTML Elements Using **document.forms**.

First name: Donald

Last name: Duck

These are the values of each element in the form:

Donald
Duck
Submit



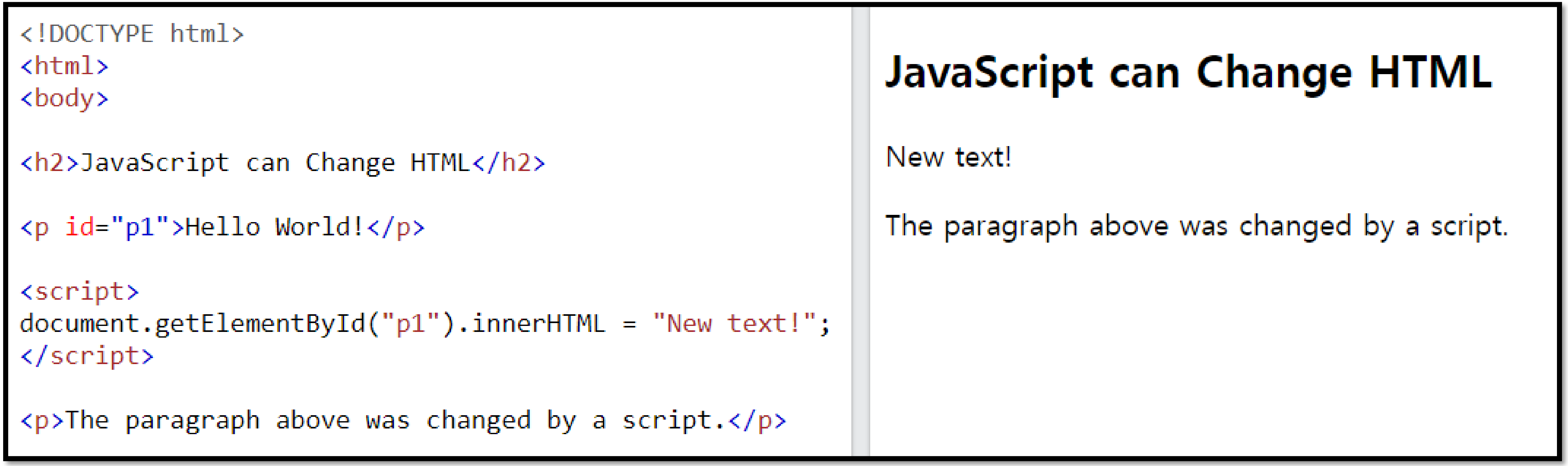
Changing HTML Content

The easiest way to modify the content of an HTML element is by using the `innerHTML` property.

To change the content of an HTML element, use this syntax:

```
document.getElementById(id).innerHTML = new HTML
```

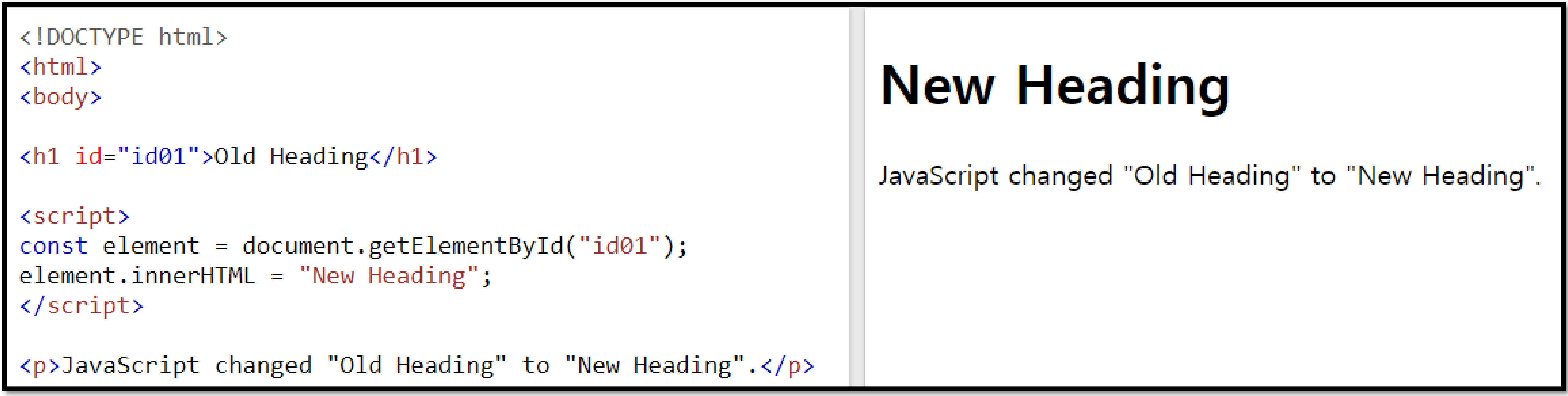
https://www.w3schools.com/js/tryit.asp?filename=tryjs_change_innerhtml



```html
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript can Change HTML</h2>

<p id="p1">Hello World!</p>

<script>
document.getElementById("p1").innerHTML = "New text!";
</script>

<p>The paragraph above was changed by a script.</p>
```

```
JavaScript can Change HTML

New text!

The paragraph above was changed by a script.
```

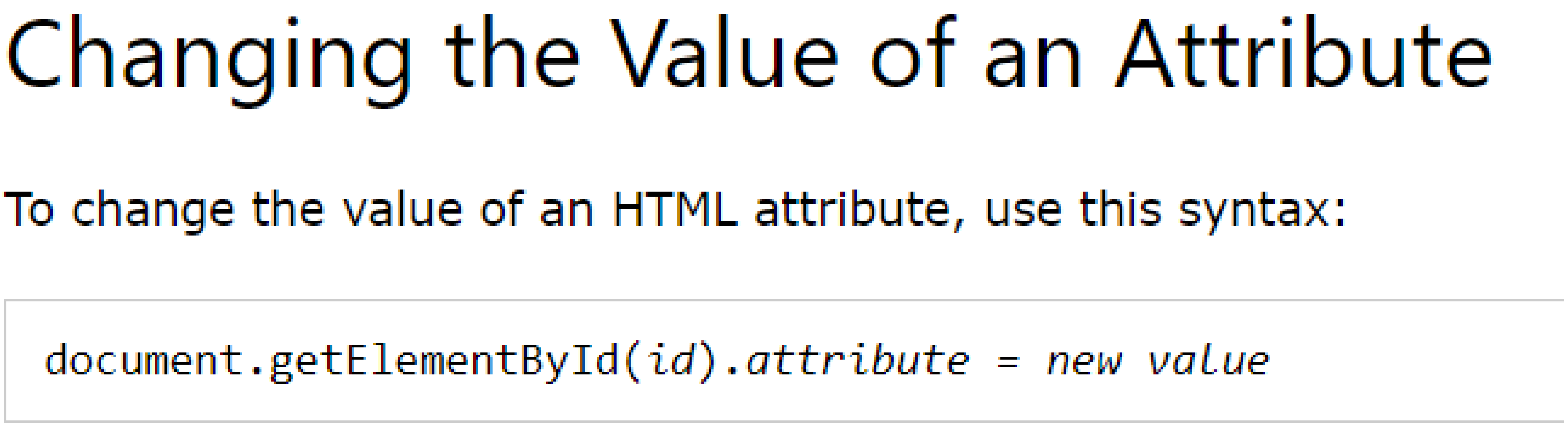
https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_innerhtml



```html
<!DOCTYPE html>
<html>
<body>

<h1 id="id01">Old Heading</h1>

<script>
const element = document.getElementById("id01");
element.innerHTML = "New Heading";
</script>

<p>JavaScript changed "Old Heading" to "New Heading".</p>
```

New Heading

JavaScript changed "Old Heading" to "New Heading".

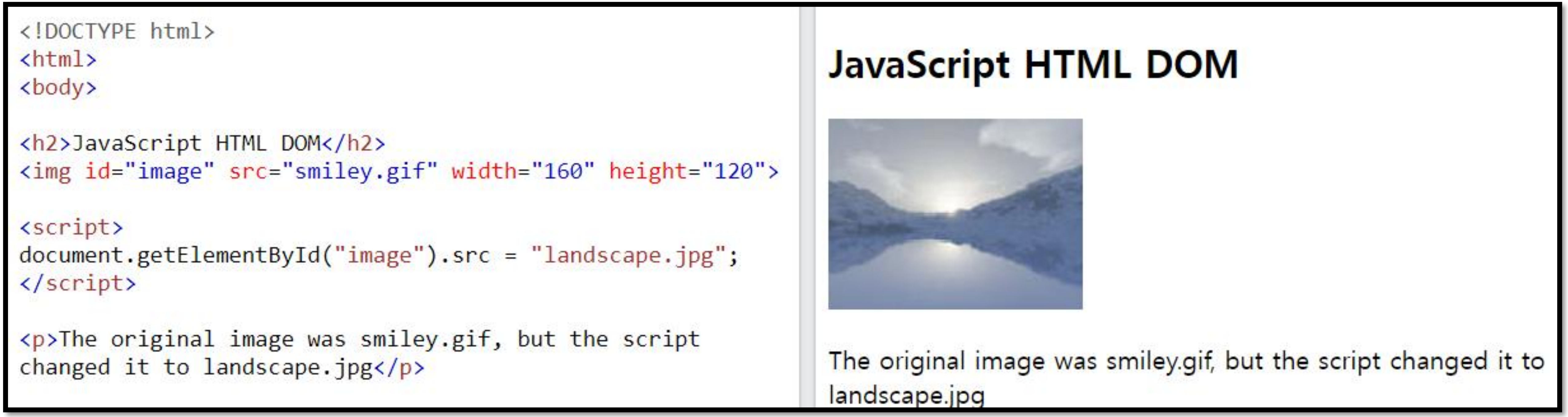


Changing the Value of an Attribute

To change the value of an HTML attribute, use this syntax:

```
document.getElementById(id).attribute = new value
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_image



```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML DOM</h2>
<img id="image" src="smiley.gif" width="160" height="120">

<script>
document.getElementById("image").src = "landscape.jpg";
</script>

<p>The original image was smiley.gif, but the script
changed it to landscape.jpg</p>
```

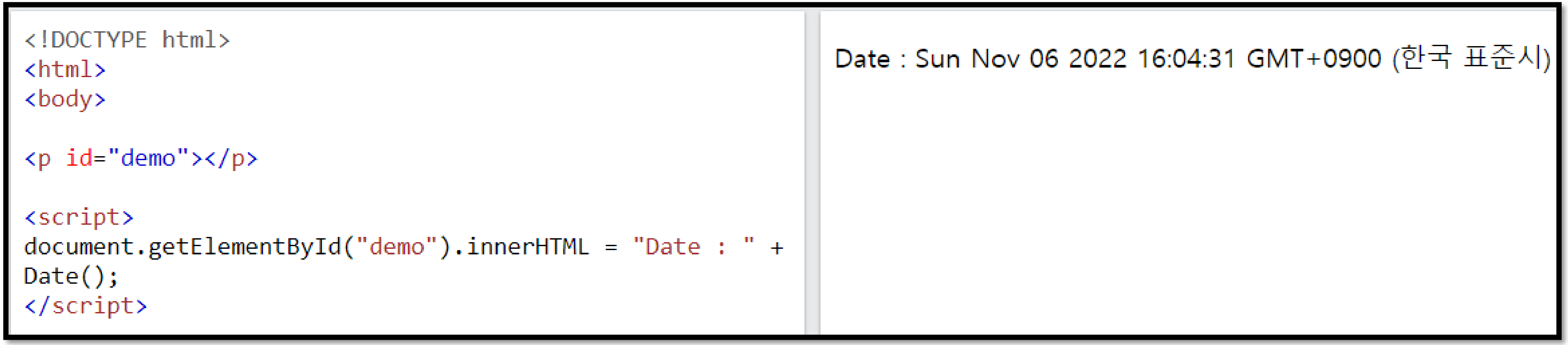
JavaScript HTML DOM

The original image was smiley.gif, but the script changed it to landscape.jpg

Dynamic HTML content

JavaScript can create dynamic HTML content:

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_date



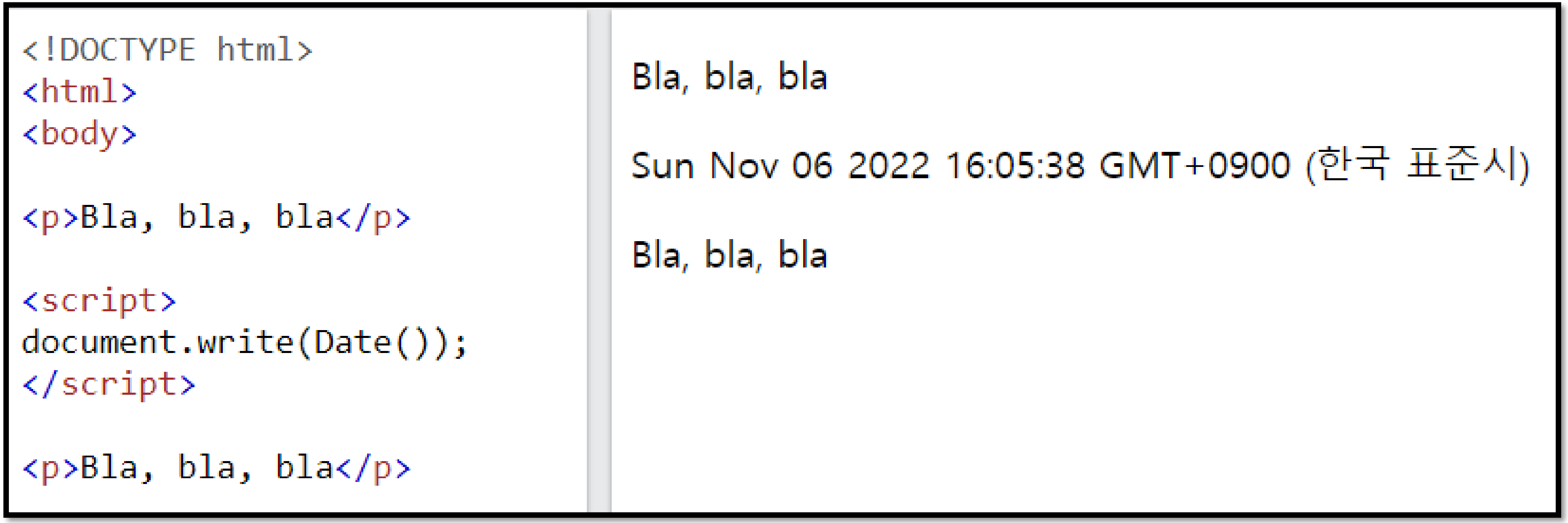
```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Date : " +
Date();
</script>
```

```
Date : Sun Nov 06 2022 16:04:31 GMT+0900 (한국 표준시)
```

document.write()

In JavaScript, `document.write()` can be used to write directly to the HTML output stream:

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_write



```html
<!DOCTYPE html>
<html>
<body>

<p>Bla, bla, bla</p>

<script>
document.write(Date());
</script>

<p>Bla, bla, bla</p>
```

```
Bla, bla, bla

Sun Nov 06 2022 16:05:38 GMT+0900 (한국 표준시)

Bla, bla, bla
```