IS 242 Web Application Development 1

Lecture 10: Introduction to JavaScript (Part 3)
HTML DOM

Outlines of today’s lecture

- What’s DOM?
- How to find HTML elements in the document.
- How to change the content, attributes and style of HTML elements.

What is DOM?

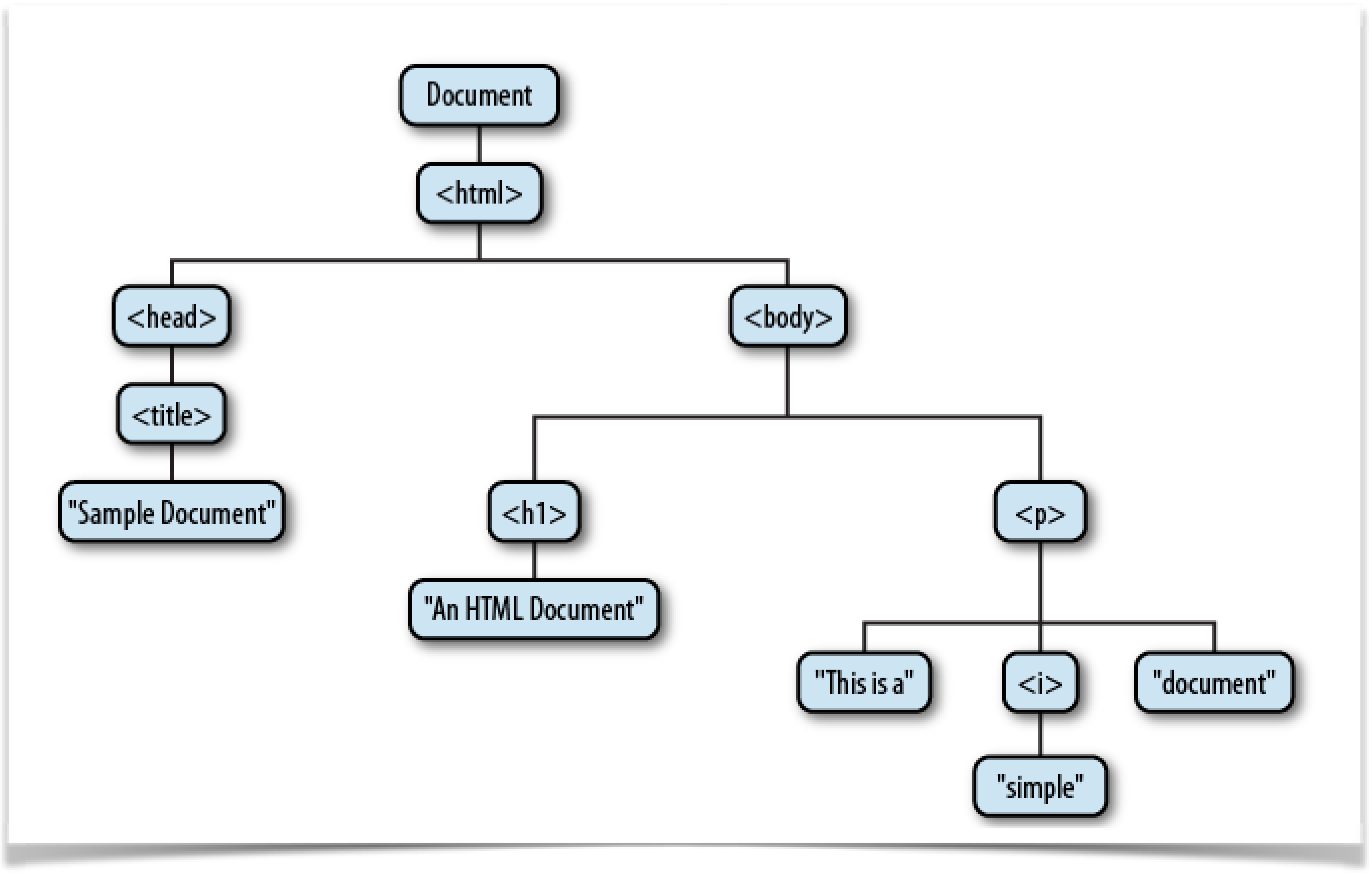
The **Document Object Model**, or **DOM**, is the fundamental API for representing and manipulating the content of HTML documents. It defines:

- 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.

Overview of the DOM

```
<html>
  <head>
    <title>Sample
Document</title>
  </head>
  <body>
    <h1>An HTML
Document</h1>
    <p>This is a
<i>simple</i> document.
</html>
```

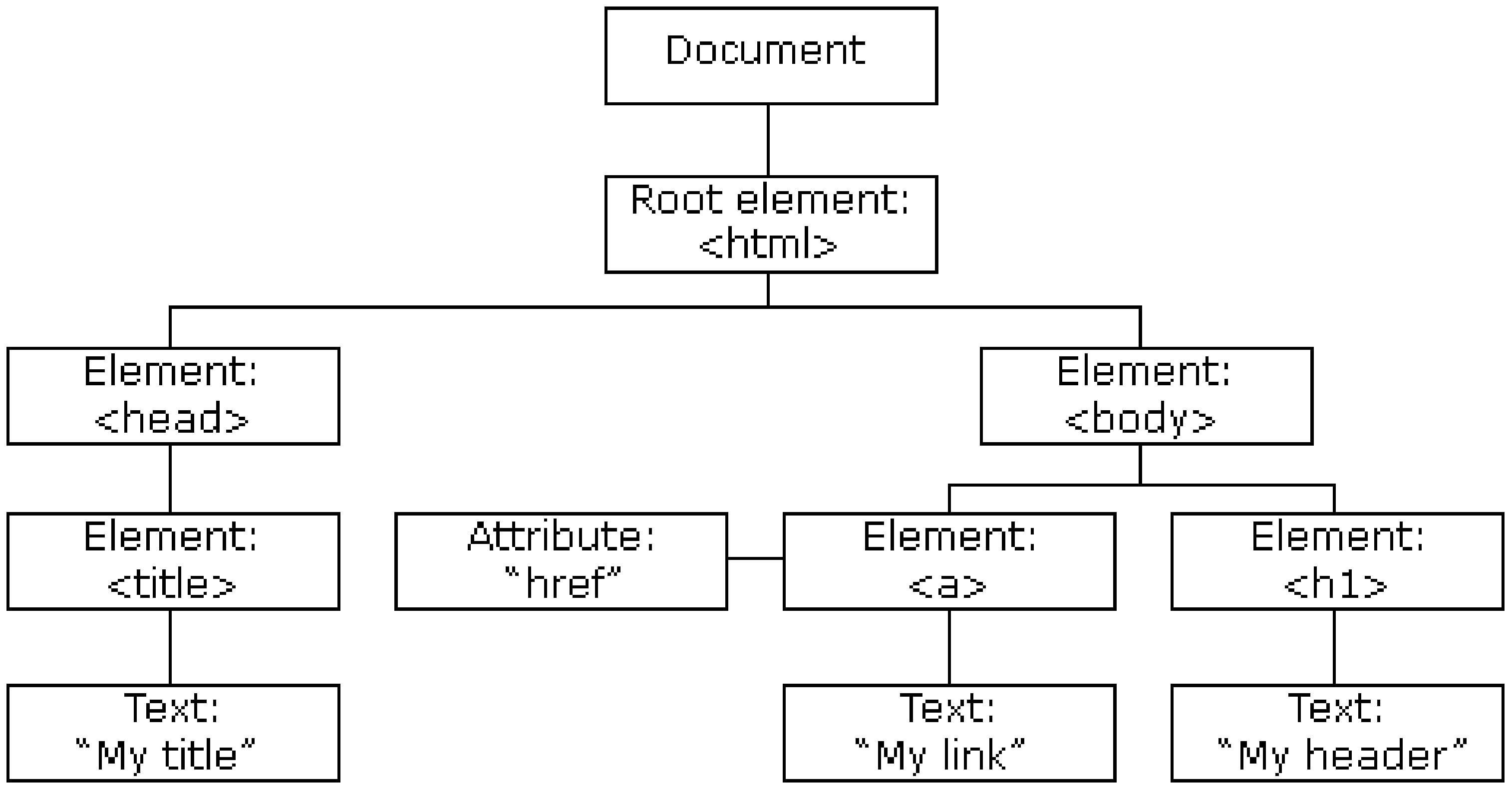


Overview of the DOM (Cont.)

- The node directly above a node is the **parent** of that node.
- The nodes one level directly below another node are the **children** of that node.
- Nodes at the same level, and with the same parent, are **siblings**.
- The set of nodes any number of levels below another node are the **descendants** of that node.
- And the parent, grandparent, and all other nodes above a node are the **ancestors** of that node.

How Is It Created?

- 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 DOM Document

- In the HTML DOM object model, the **document** object represents your web page.
- The document object is the **owner** of all other objects in your web page.
- If you want to access objects in an HTML page, you always start with accessing the **document object**.

The DOM Programming Interface

- The HTML DOM can be accessed with JavaScript (and with other programming languages).

- In the DOM, **all HTML elements are defined as objects.**

- The **programming interface** is the **properties** and **methods** of each object.

- A property is a value that you can get or set (like changing the content of an HTML element).

- A method is an action you can do (like adding or deleting an HTML element).

The DOM Programming Interface (Cont.)

- The **getElementById** Method
  - The most common way to access an HTML element is to use the id of the element.

- The **innerHTML** Property
  - The easiest way to get the content of an element is by using the innerHTML property.
  - The innerHTML property is useful for getting or replacing the content of HTML elements.
  - The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

Properties and Methods-Example

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

```
<html>
  <body>

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

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

   </body>
</html>
```

- In the example above, **getElementById** is a **method**, while **innerHTML** is a property

HTML DOM Elements

Finding HTML Elements

- Often, with JavaScript, you want to manipulate HTML elements.
- To do so, you have to find the elements first. There are a number of ways to do this, a few are listed as follows:

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

1. Finding HTML Elements by Id

- Any HTML element can have an **id** attribute.
- The value of this attribute must be unique within the document—no two elements in the same document can have the same ID.
- The easiest way to find HTML elements in the DOM, is by using the element id.
- You can select an element based on this unique ID with the **getElementById()** method of the **Document object**
- If the element is **found**, the method will return the element as an **object**.
- If the element is **not found**, the method will return **null**.

1. Finding HTML Elements by Id-Example

- This example finds the element with id=“intro":

```
<body>

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

   <p>This example demonstrates the <b>getElementById</b>
  method!</p>

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

   <script>
    myElement = document.getElementById("intro");
    document.getElementById("demo").innerHTML =
   "The text from the intro paragraph is " + myElement.innerHTML;
    </script>

</body>
```

2. Finding HTML Elements by Tag Name

- You can select all HTML elements of a **specified type (or tag name)** using the **getElementsByTagName()** method.

- getElementsByTagName() returns a **NodeList object**.

- The nodes in the node list can be accessed through their **index number** (starting from 0).

- The elements of the returned NodeList are in **document order**, so you can select the first <p> element of a document like this:

```
var firstpara = document.getElementsByTagName("p")[0];
```

2. Finding HTML Elements by Tag Name-Example

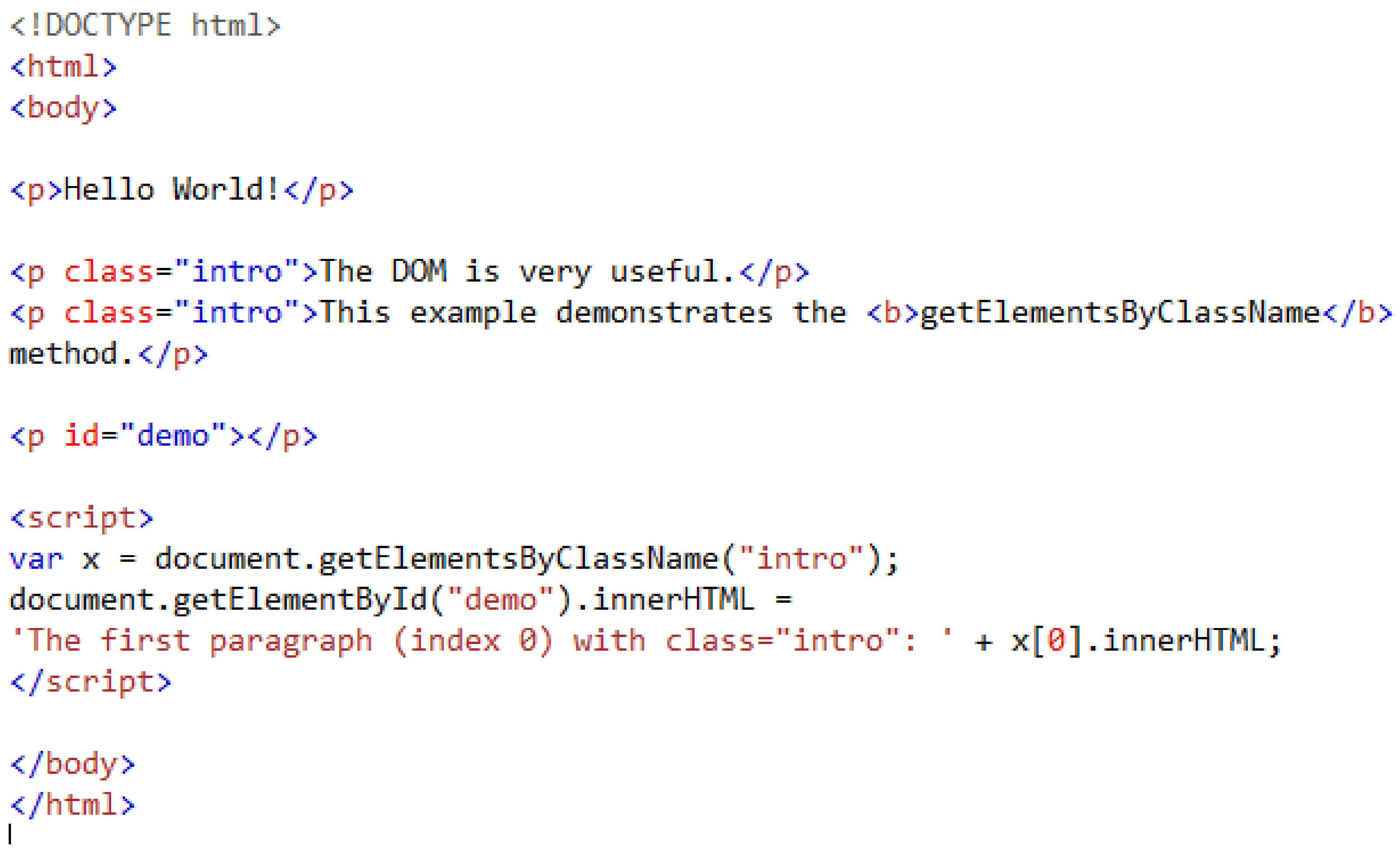
- This example finds the element with id="main", and then finds all <p> elements inside "main":

```
<div id="main">
<p>The DOM is very useful.</p>
<p>This example demonstrates the <b>getElementsByTagName</b> method</p>
</div>

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

<script>
var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
document.getElementById("demo").innerHTML =
'The first paragraph inside "main" is ' + y[0].innerHTML;
</script>
```

3. Finding HTML Elements by Class Name

- You can select all HTML elements of a **specified class** using the **getElementsByClassName()** method.
- It returns a **NodeList** containing all matching descendants of the document or element.

3. Finding HTML Elements by Class Name



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

<p>Hello World!</p>

<p class="intro">The DOM is very useful.</p>
<p class="intro">This example demonstrates the <b>getElementsByClassName</b>
method.</p>

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

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

</body>
</html>
```

4. Finding HTML Elements by HTML Object Collections

- The Document Object Model contains several **collections**, which are groups of related **objects** on a page.

- DOM collections are accessed as **properties** of DOM objects such as the document object or a DOM node. The document object has properties containing the

  - images collection
  - links collection
  - forms collection
  - anchors collection

These collections contain all the elements of the corresponding type on the page.

4. Finding HTML Elements by HTML Object Collections-Example

```
<html>

    <body>

    <a name=“html”>HTML Tutorial</a><br>

    <a name="css">CSS Tutorial</a><br>

    <a name="xml">XML Tutorial</a><br>

    <p id=“demo"></p>

    <script>

     document.getElementById("demo").innerHTML =

     "The content of the first anchor is: " + document.anchors[0].innerHTML;

   </script>

</body>

</html>
```

Changing HTML Content

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

- Write the script that will change the content of the <p> element to “Good Morning!”

```
<html>
   <body>

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

     <script>
         ?
     </script>

   </body>
</html>
```

Changing the Value of an Attribute

Write a script that changes the value of the src attribute of the <img> element to “image1.gif”:

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

          <img id="myImage" src="smiley.gif">

          <script>
           ?
          </script>

        </body>
    </html>
```

Changing HTML Style

Write a script that changes the text color of the <p> element

```
<html>
     <body>

        <p id="p2">Hello World!</p>

        <script>

        </script>

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

    </body>
</html>
```

References

- www.w3schools.com
- Flanagan, David. JavaScript: The definitive guide: Activate your web pages. O'Reilly Media, Inc., 2011.
- Deitel & Deitel (2011). *Internet and World Wide Web How to Program, 5th Edition, Harvey & Paul Deitel & Associates.*