JavaScript
Document Object Model (DOM)

Document Model Object (DOM)

- The Document Object Model (DOM) is a programming interface for HTML documents. It provides a structured representation of the document and it defines a way that the structure can be accessed from programs so that they can change the document structure, style and content.

DOM

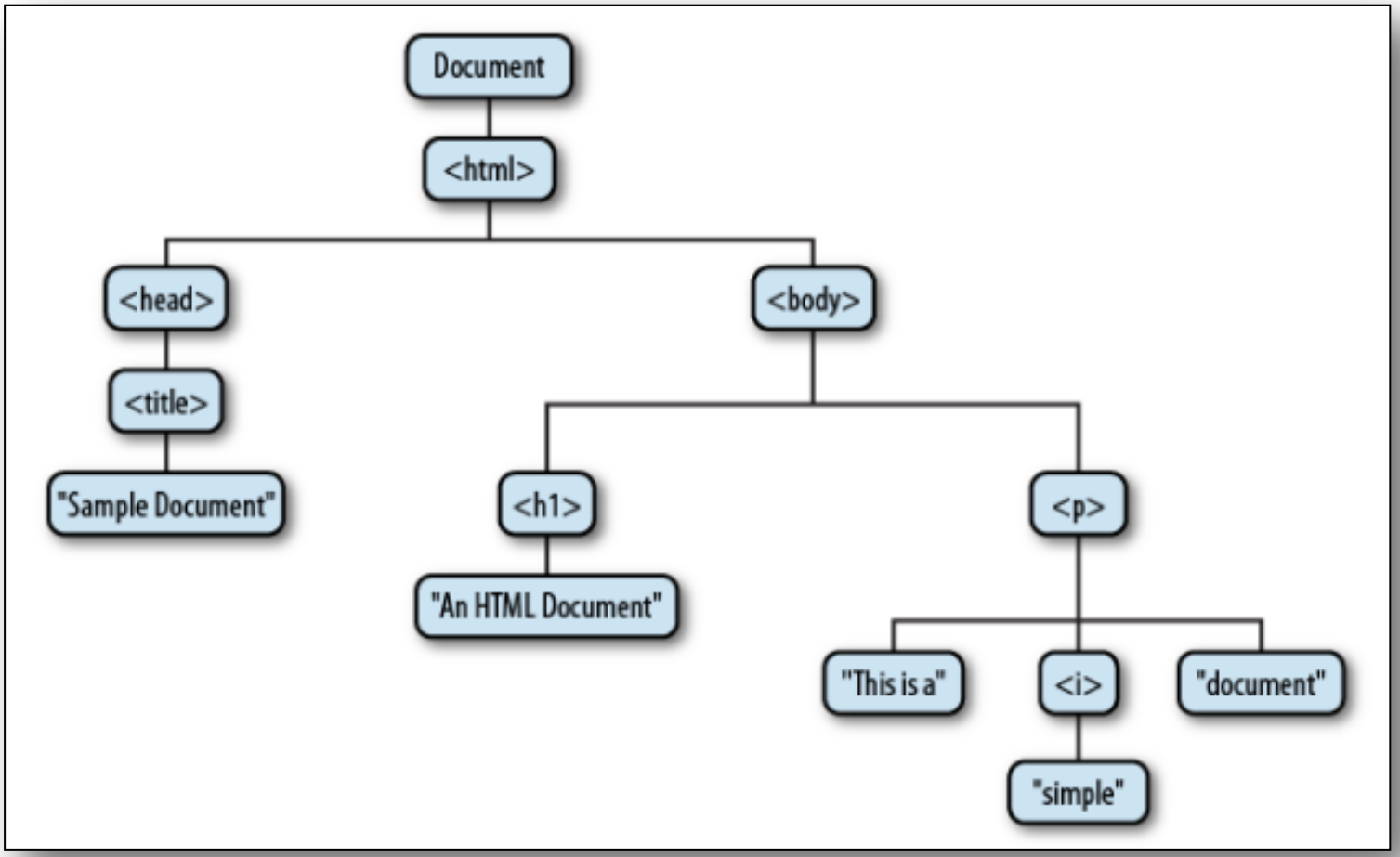
- The document Object Model (DOM) represents a web page in tree structure.
- DOM is a tree of objects each represents a single HTML element or a single literal String on the web page.
- Every node in the DOM has a relationship to its surrounding nodes.

DOM Tree

- The DOM views the HTML document as a tree-structure.
- All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.

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

Basic HTML structure to illustrate a DOM tree



The tree representation of an HTML document

DOM Tree (cont)

- The DOM is made up of items called nodes , which can have parents, children, and siblings and this determines its position in the tree.
  - Parent

    A parent node is anything that contains other nodes.
  - Child

    Children are positioned inside parent nodes.
  - Sibling

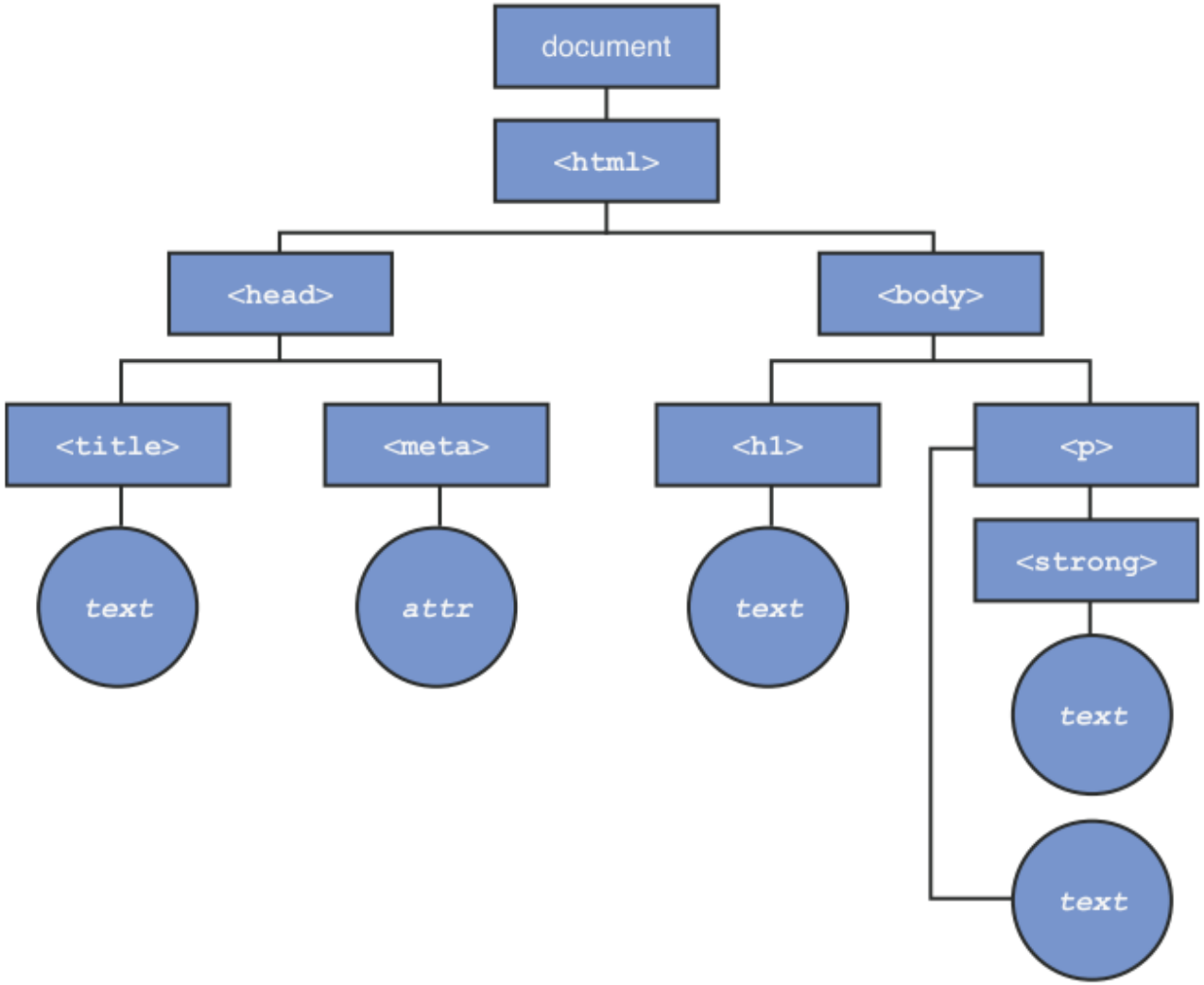
    It is the nodes that are on the same level .

DOM nodes

- The type of nodes in the DOM can be classified to :
  - Nodes that represent HTML elements are called **element nodes**
  - The text inside HTML elements are called **text nodes**
  - Every HTML attribute is **attribute node.**

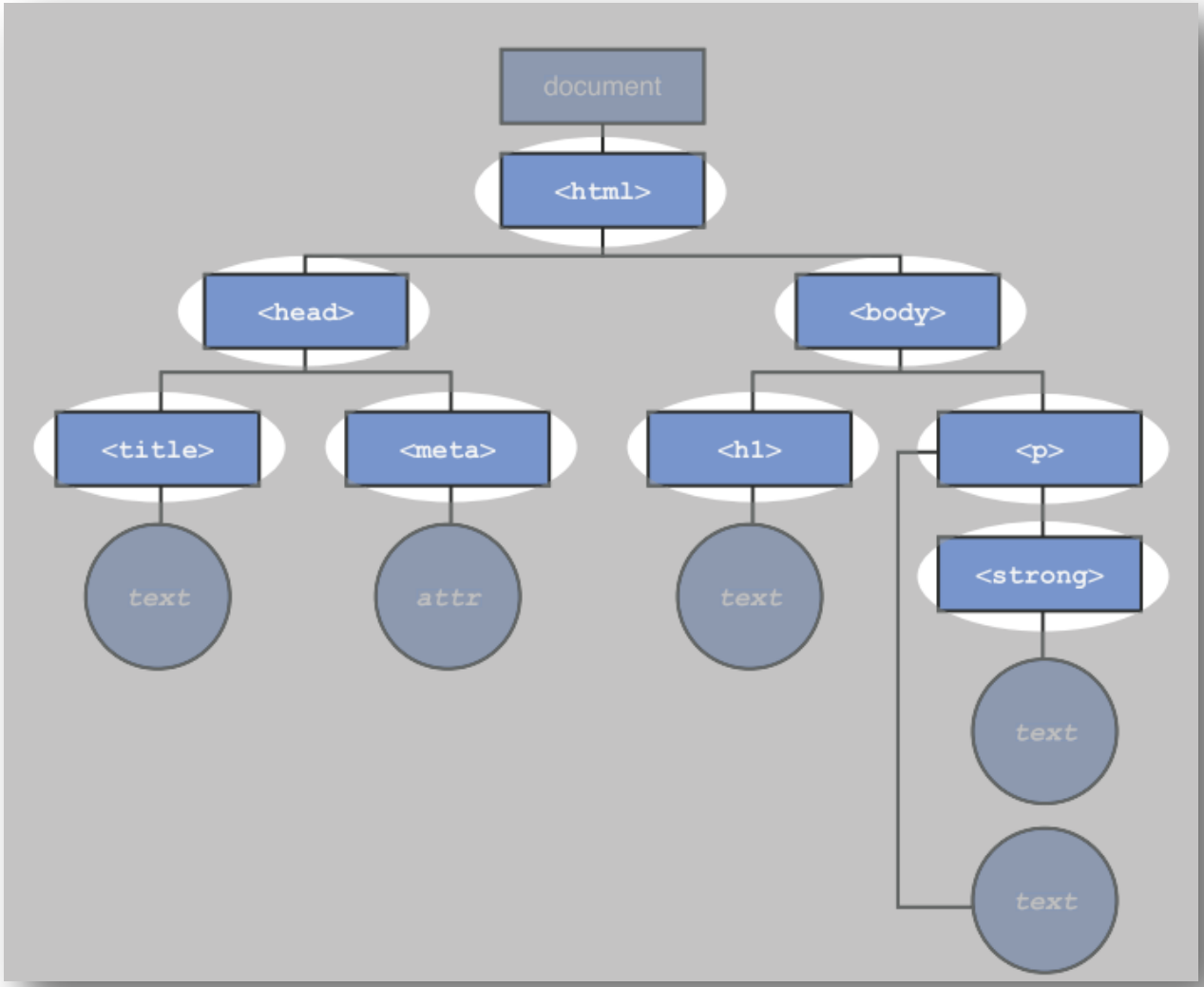
Node Types Example

```
<!--
node-types.html
-->
<html>
    <head>
        <title>  DOM Nodes  example</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <h1>Hello World!</h1>
        <p>This is a <strong> basic HTML document</strong></p>
    </body>
</html>
```

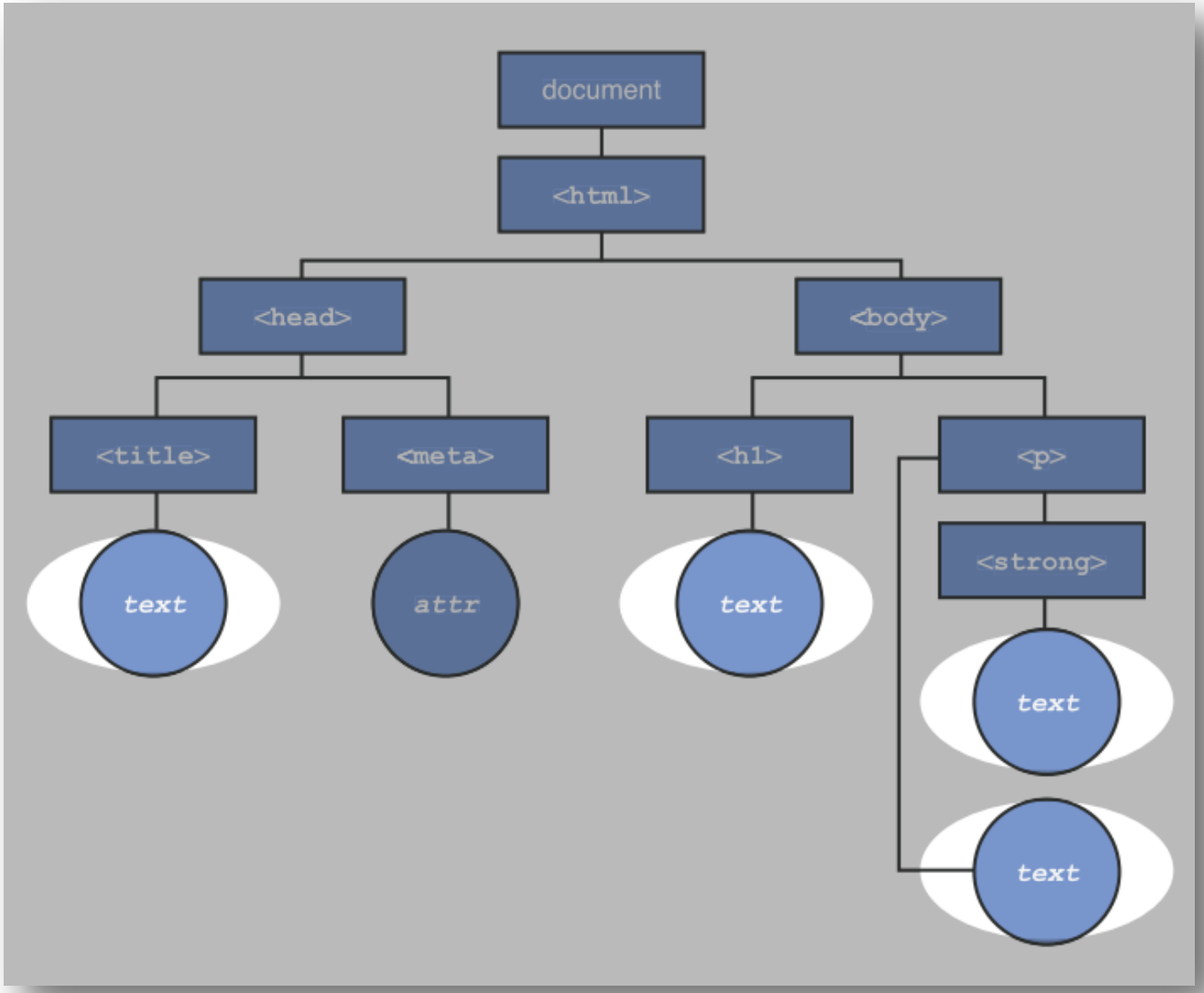


A graphical representation of the Document Object Model

Element Nodes

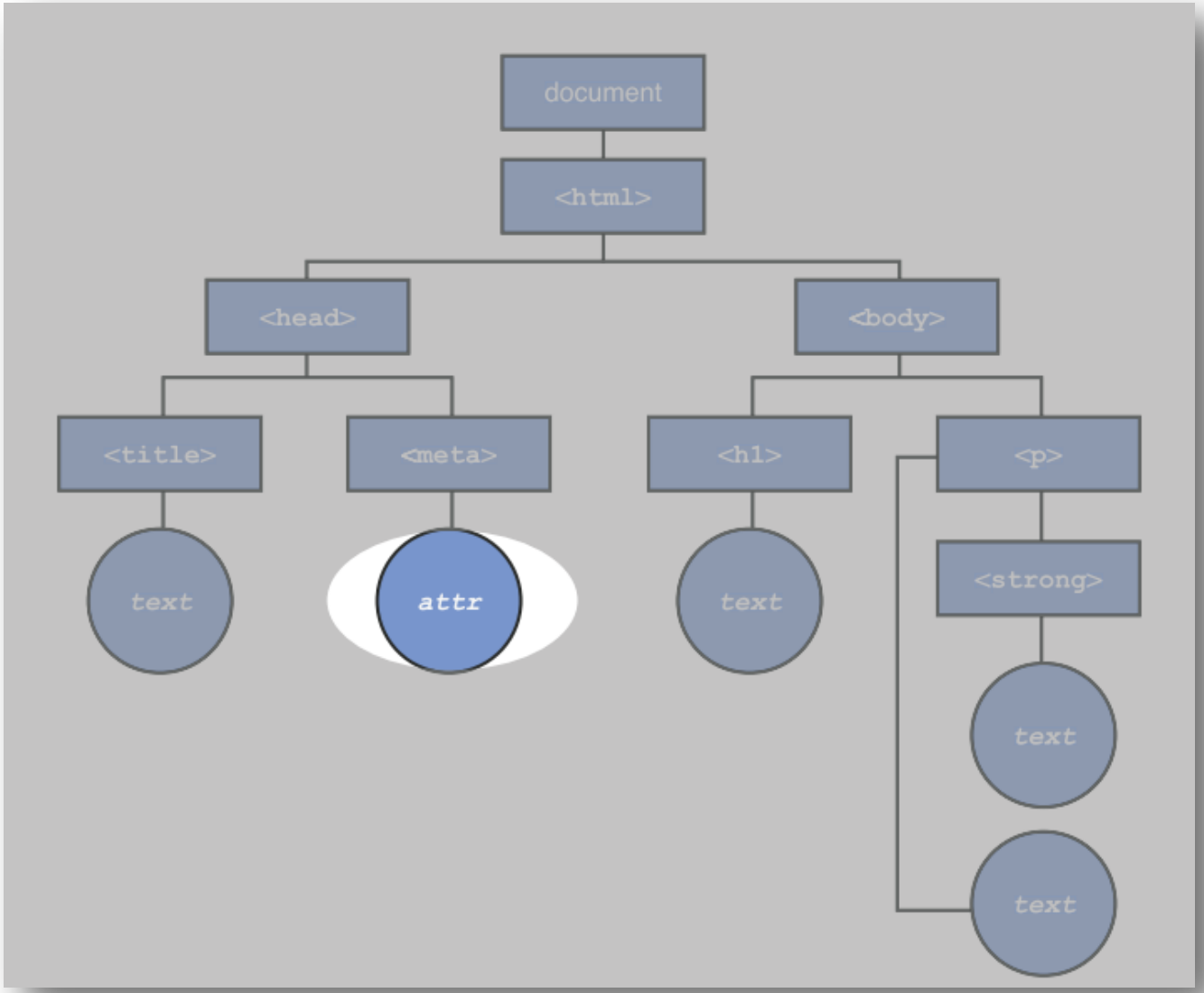


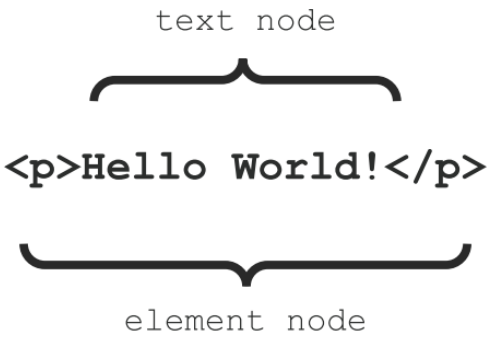
Text Nodes


document
<html>
<head>
<body>
<title>
<meta>
<h1>
<p>

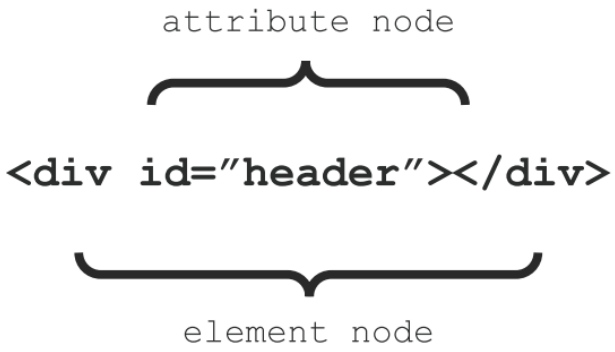
text
attr
text
text
text

Attribute Nodes





The difference between an element node and a text node



The position of an attribute node in relation to an element node

Selecting Element Node

- Every element node in the DOM tree can be selected using different methods.

Example-2

```
<html>
 <head>
  <title>Working with element node</title>
 </head>
 <body>
    <div>Working with element node</div>
    <p > First paragraph</p>
    <p id ="p-id"> second  paragraph</p>
    <p class ="p-class"> Third paragraph</p>
    <p class ="p-class"> Fourth  paragraph</p>
 </body>
</html>
```

Selecting Element Node

- Every element node in the DOM tree can be reached using different methods.
- **Selecting by ID**

```
document. getElementById ("p-id")
```

```
<p id ="p-id"> second  paragraph</p>
```

- **Selecting by Class**

```
document. getElementsByClassName ("p-class")
```

```
<p class ="p-class"> Third paragraph</p>
<p class ="p-class"> Fourth  paragraph</p>
```

- **Selecting by Name**

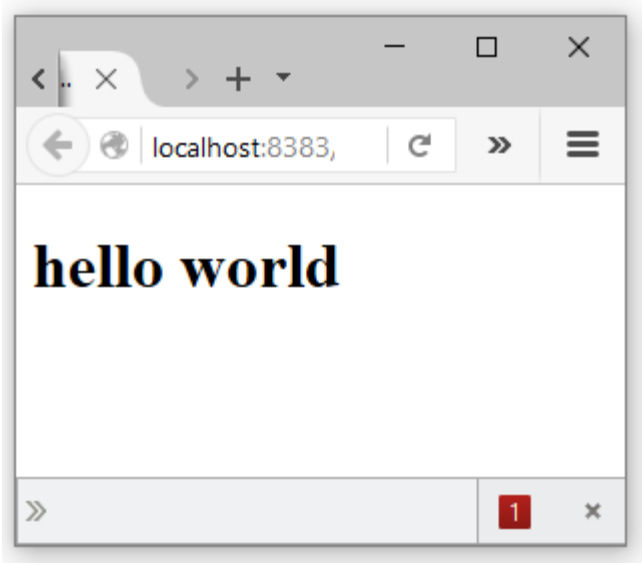
```
document.getElementsByTagName ("p")
```

```
<p > First paragraph</p>
<p id ="p-id"> second  paragraph</p>
<p class ="p-class"> Third paragraph</p>
<p class ="p-class"> Fourth  paragraph</p>
```

Working with Text node

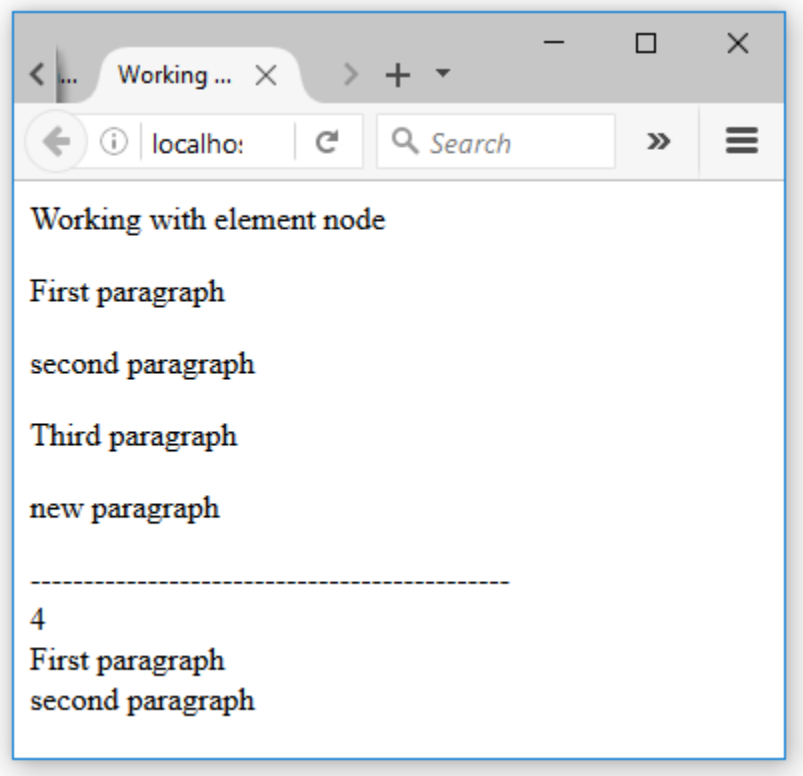
- The method innerHTML is used to get and change the content of HTML node.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Working with Text node Example</title>
  </head>
  <body>
    <div id ="div1">To do write content</div>
    <script>
      document.getElementById("div1").innerHTML = "<h1>hello world</h1>";
    </script>
  </body>
</html>
```



- **Example-4**

```
<!DOCTYPE html>
<!-- element-nodes.html -->
<html>
  <head>
    <title>Working with element node</title>
  </head>
  <body>
      <div>Working with element node</div>
      <p > First paragraph</p>
      <p id ="p-id"> second  paragraph</p>
      <p class ="p-class"> Third paragraph</p>
      <p class ="p-class"> Fourth  paragraph</p>
      ------------------------------------------
      <br/>
       <script>
      document.write( document.getElementsByTagName ("p").length+"<br>");
      document.write(document. getElementsByTagName ("p").item(0).innerHTML+"<br>");
      document.write( document. getElementById ("p-id").innerHTML+"<br>");
     document. getElementsByClassName ("p-class").item(1).innerHTML="new paragraph";
      </script>
  </body>
</html>
```


Working ...
localho:
Search
Working with element node
First paragraph
second paragraph
Third paragraph
new paragraph

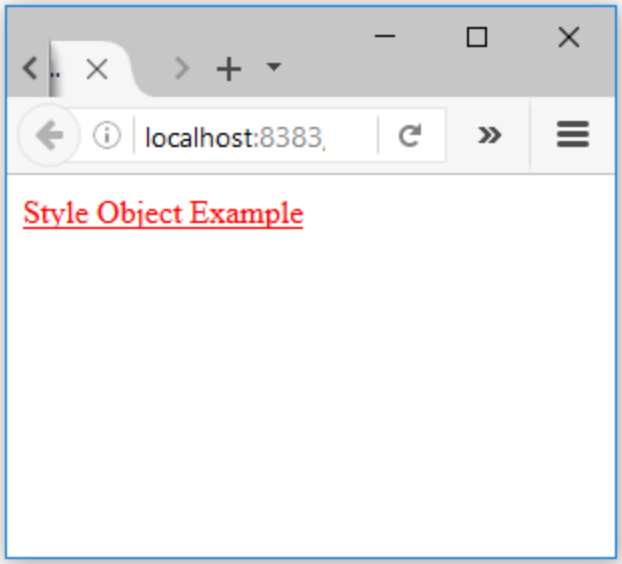
4
First paragraph
second paragraph

Changing Text node appearance

- Changing the CSS properties with the style property of the text node.
- Changing the class attribute value of the text node.

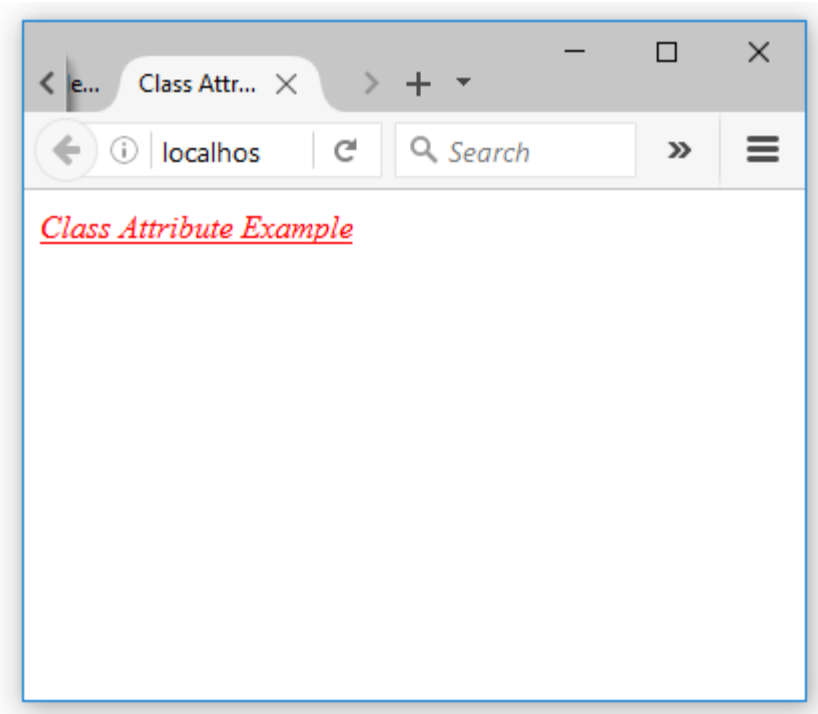
Using Style Property

```
<!DOCTYPE html>
<!--
style-object.html
-->
<html>
    <head>
        <title>Style Object Example</title>
    </head>
    <body>
        <div id="div1">Style Object Example</div>
        <script>
            var div1 = document.getElementById("div1");
            div1.style.color="red";
            div1.style.textDecoration="underline";
        </script>
    </body>
</html>
```


localhost:8383
Style Object Example

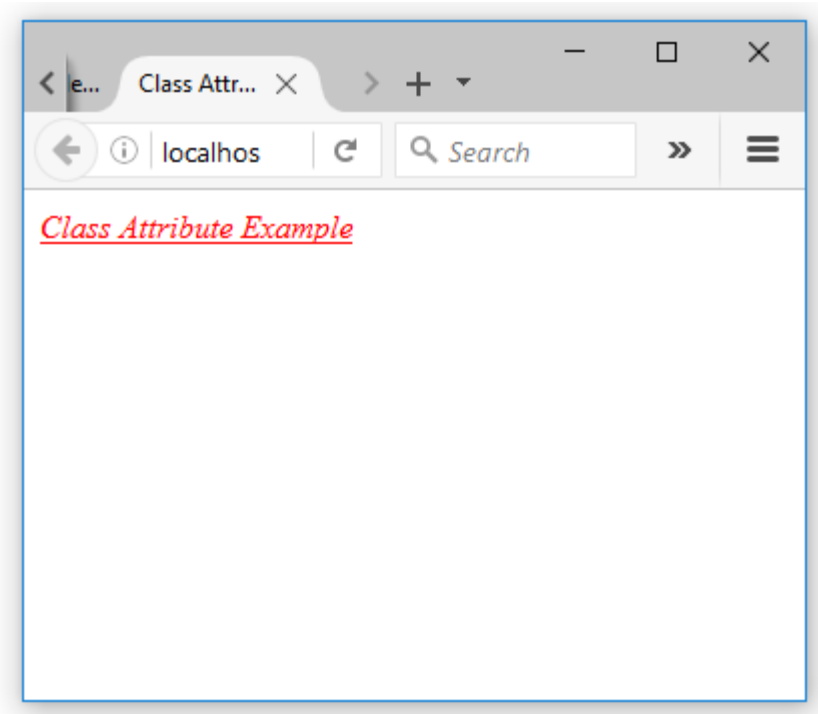
Changing the class attribute

```
<!DOCTYPE html>
<!--
class-attribute.html
-->
<html>
    <head>
        <title>Class Attribute Example</title>
        <style>
            .show {
                color: red;
                font-style: italic;
                text-decoration: underline;
            }
        </style>
    </head>
    <body>
        <div id="id1">Class Attribute Example</div>
        <script>
            var div1 = document.getElementById("id1");
            div1.setAttribute("class", "show");
        </script>
    </body>
</html>
```


e...
Class Attr...
localhos
Search
Class Attribute Example

Changing the class attribute

```
<!DOCTYPE html>
<!--
class-attribute1.html
-->
<html>
   <head>
        <title>Class Attribute Example</title>
        <style>
            .show {
                color: red;
                font-style: italic;
                text-decoration: underline;
            }
        </style>
    </head>
    <body>
        <div id="id1">Class Attribute Example</div>
        <script>
            var div1 = document.getElementById("id1");
            div1.className = "show";
        </script>
    </body>
</html>
```

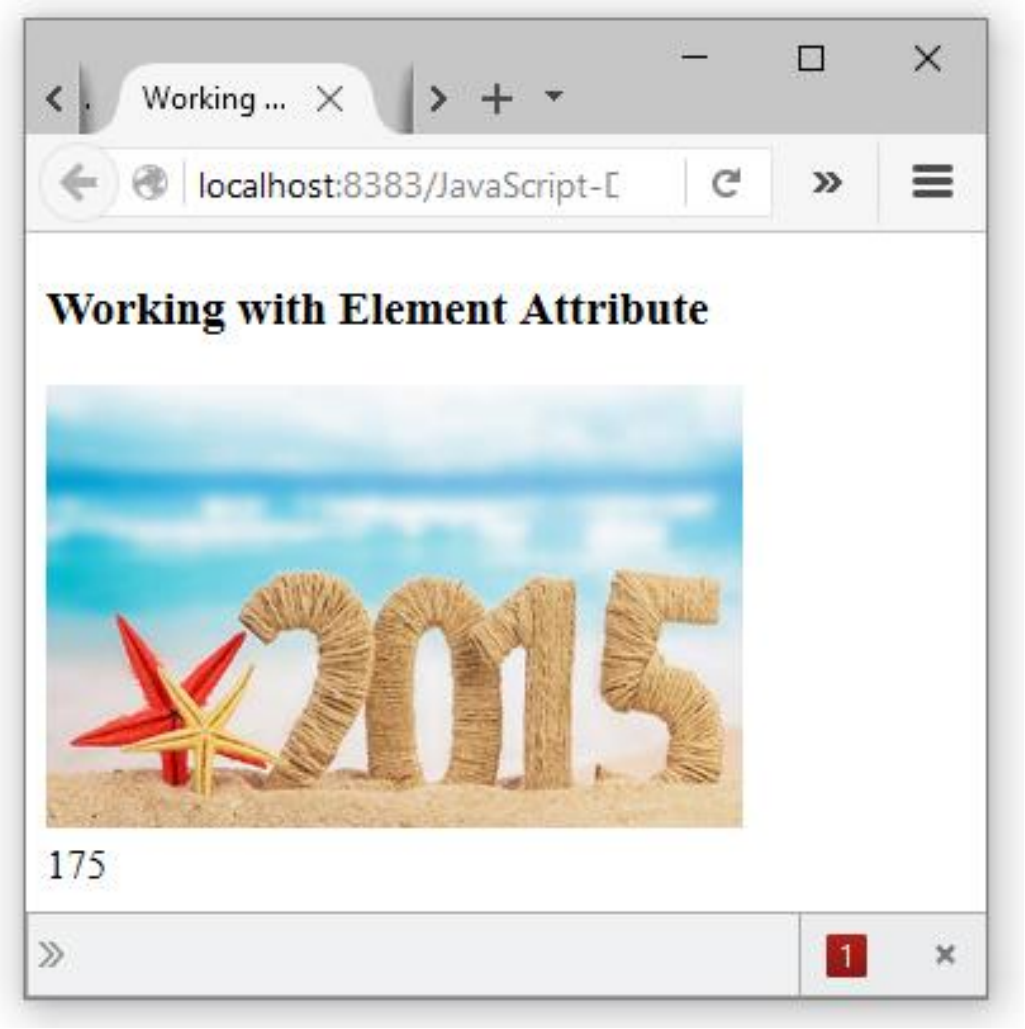

e...
Class Attr...
localhos
Search
Class Attribute Example

Working with Attributes

- The DOM defines set of methods for getting and setting the values of HTML attributes. Those methods are :
  - getAttribute()
  - setAttribute()
  - removeAttribute()
  - hasAttribute()

- **Example-8**

```
<!DOCTYPE html>
<!-- element-attribute.html -->
<html>
   <head>
      <title>Working with Element Attribute</title>
   </head>
   <body>
      <div>To do write content</div>
      <img src="image1.jpg" height="175" width="175" id="pic">
      <script>
         if (document.getElementById("pic").hasAttribute("width")) {
            document.write( document.getElementById("pic"). getAttribute ("width"));
            document.getElementById("pic").setAttribute("width","275");
         }
      </script>
   </body>
</html>
```

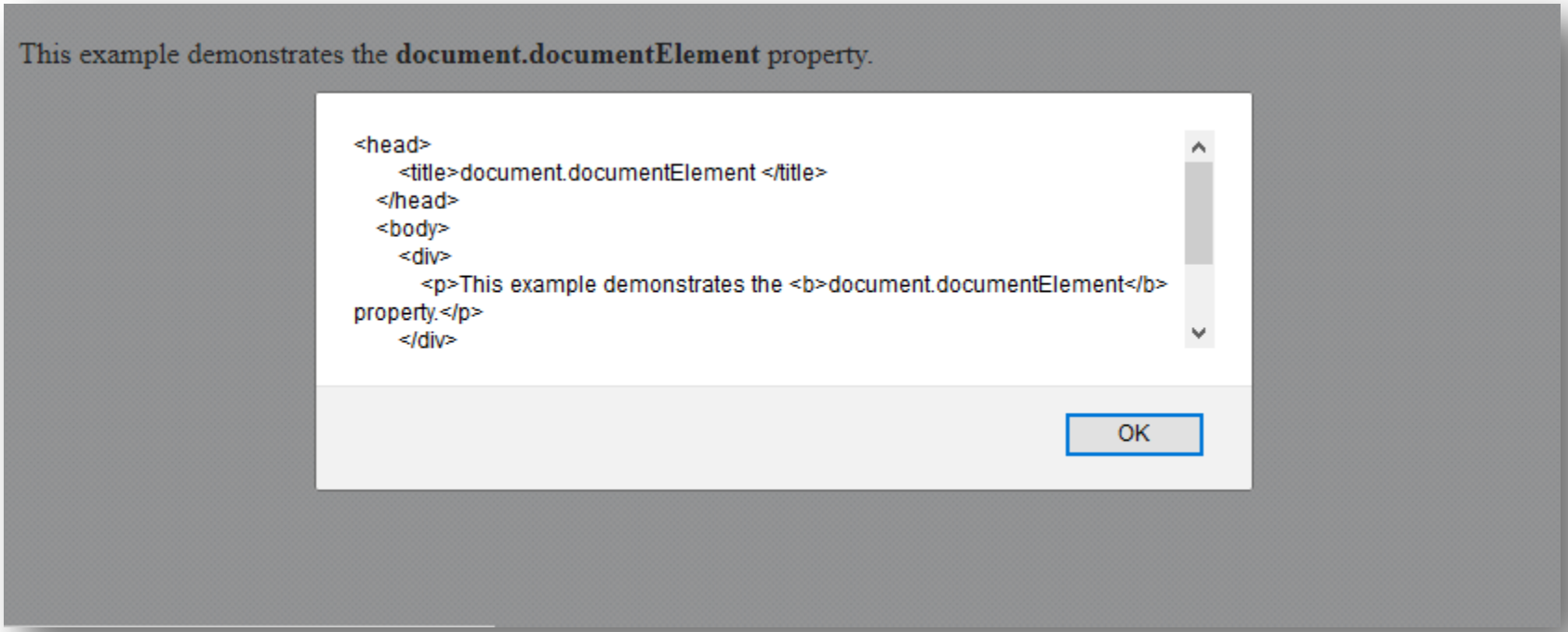

Working ...
localhost:8383/JavaScript-[
Working with Element Attribute
175

HTML Navigation (Tree Node Traverse)

- The node tree can be navigated using the node relationship.
- With DOM, all nodes in the node tree can be accessed by JavaScript.
- New nodes can be created, and all nodes can be modified or deleted.
- There are two special properties that allow access to the full document:
  - document.body - The body of the document
  - document.documentElement - The full document

- **Example-9**

```
1   <!DOCTYPE html>
2   <!-- document-documentElement.html -->
3   <html>
4     <head>
5       <title>document.documentElement </title>
6     </head>
7     <body>
8       <div>
9         <p>This example demonstrates the <b>document.documentElement</b> property.</p>
10      </div>
11      <script>
12        alert(document.documentElement.innerHTML);
13      </script>
14    </body>
15  </html>
```

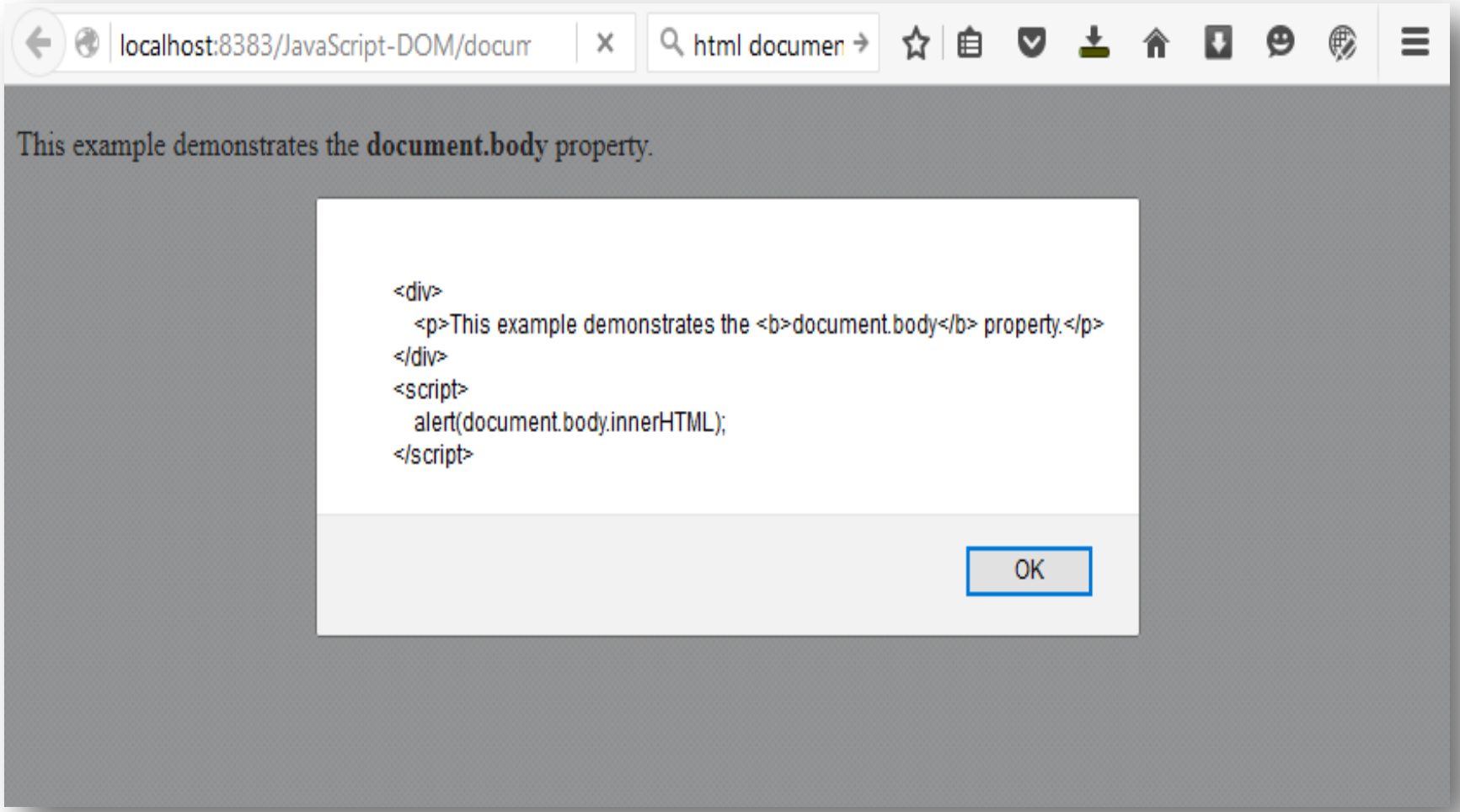

This example demonstrates the document.documentElement property.
<head>
<title>document.documentElement </title>
</head>
<body>

This example demonstrates the document.documentElement
property.

OK

- **Example-10**

```
1   <!DOCTYPE html>
2   <!-- document-body.html -->
3   <html>
4     <head>
5       <title>document.body example </title>
6     </head>
7     <body>
8       <div>
9         <p>This example demonstrates the <b>document.body</b> property.</p>
10      </div>
11      <script>
12        alert(document.body.innerHTML);
13      </script>
14    </body>
15  </html>
```


localhost:8383/JavaScript-DOM/docum
html documer
This example demonstrates the document.body property.

This example demonstrates the document.body property.

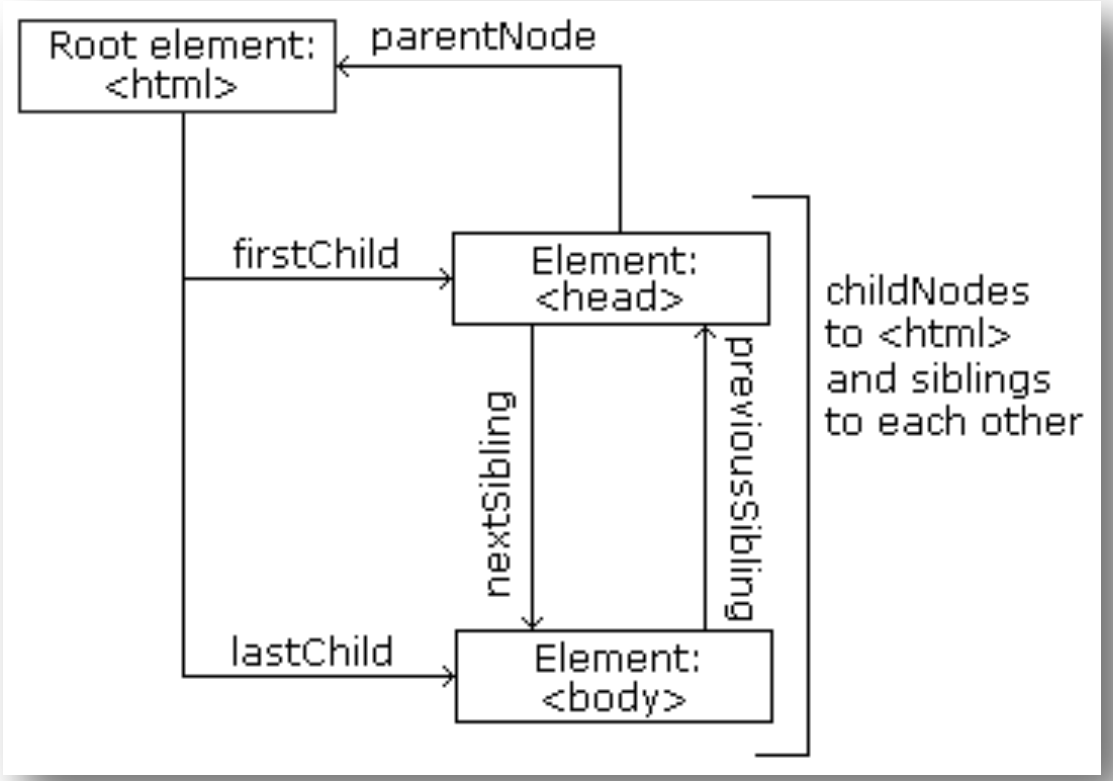
<script>
alert(document.body.innerHTML);
</script>
OK

Node Relationships

- The nodes in the node tree have a hierarchical relationship to each other.
- The terms parent, child, and sibling are used to describe the relationships.
  - The top node is called the root .
  - Every node has exactly one parent, except the root (which has no parent)
  - A node can have a number of children
  - Siblings are nodes with the same parent

Navigating Between Nodes

- The documents can be traversed as trees of Element objects using these methods:
  - parentNode
  - previousSibling
  - nextSibling
  - firstChild
  - lastChild

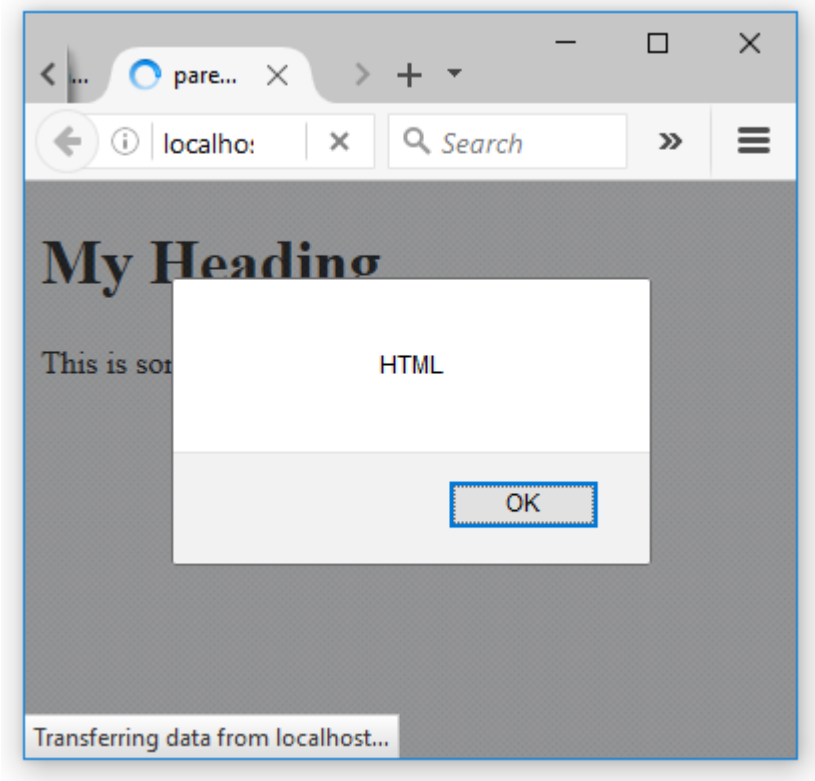


nodeName Property

- The nodeName property specifies the name of a node.
  - nodeName is read-only
  - nodeName of an element node is the same as the tag name
  - nodeName of an attribute node is the attribute name
  - nodeName of a text node is always #text
  - nodeName of the document node is always #document

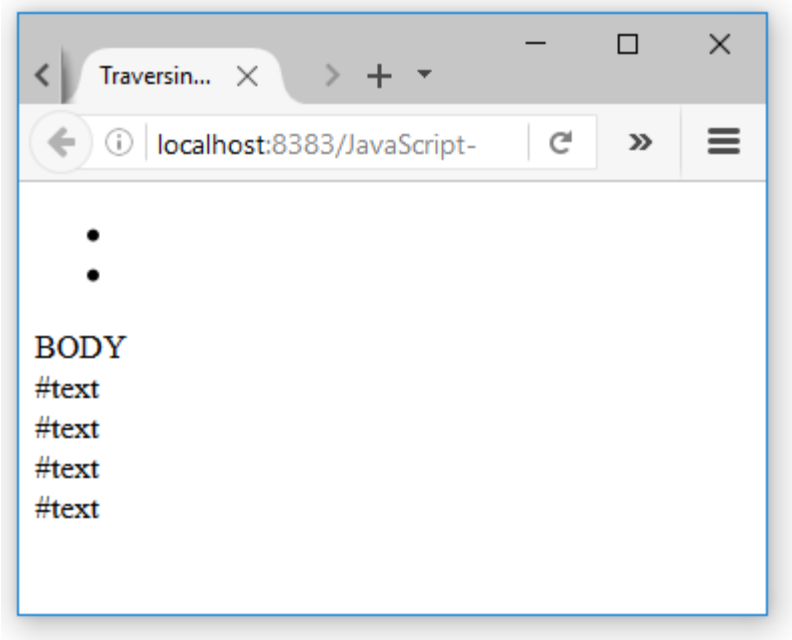
- **Example-11**

```
<!DOCTYPE html>
<!--
parent-node.html
-->
<html>
    <head>
        <title>parentNode Example</title>
    </head>
    <body id="id1">
        <h1 id="heading1">My Heading</h1>
        <p id="paragraph1">This is some text in a paragraph</p>
        <script>
            var parentElement1 = document.getElementsByTagName("body");
             alert(parentElement1[0].parentNode.nodeName);
        </script>
    </body>
</html>
```


pare...
localho:
Search
My Heading
This is so
HTML
OK
Transferring data from localhost...

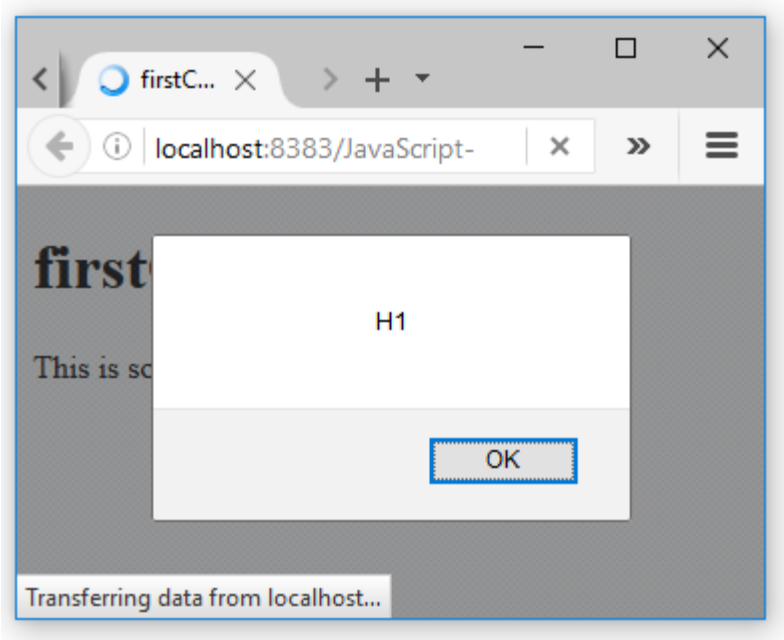
- **Example-12**

```
<!-- traverse-1.html-->
<html>
    <head>
        <title>Traversing Nodes in the DOM</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <ul id="U">
            <li id="A"></li>
            <li id="B"></li>
        </ul>
        <script>
            var ul = document.getElementById("U");
        //What is the parentNode of the ul?
            document.write(ul.parentNode.nodeName+ "</br>");
        //What is the first child of the ul?
            document.write(ul.firstChild.nodeName+ "</br>");
        //What is the last child of the ul?
            document.write(ul.lastChild.nodeName+ "</br>");
        //What is the nextSibling of the first li?
            document.write(document.getElementById("A").nextSibling.nodeName+ "</br>");
        //What is the previousSibling of the last li?
            document.write(document.getElementById("B").previousSibling.nodeName+ "</br>");
        </script>
    </body>
</html>
```


Traversin...
localhost:8383/JavaScript-
BODY
#text
#text
#text
#text

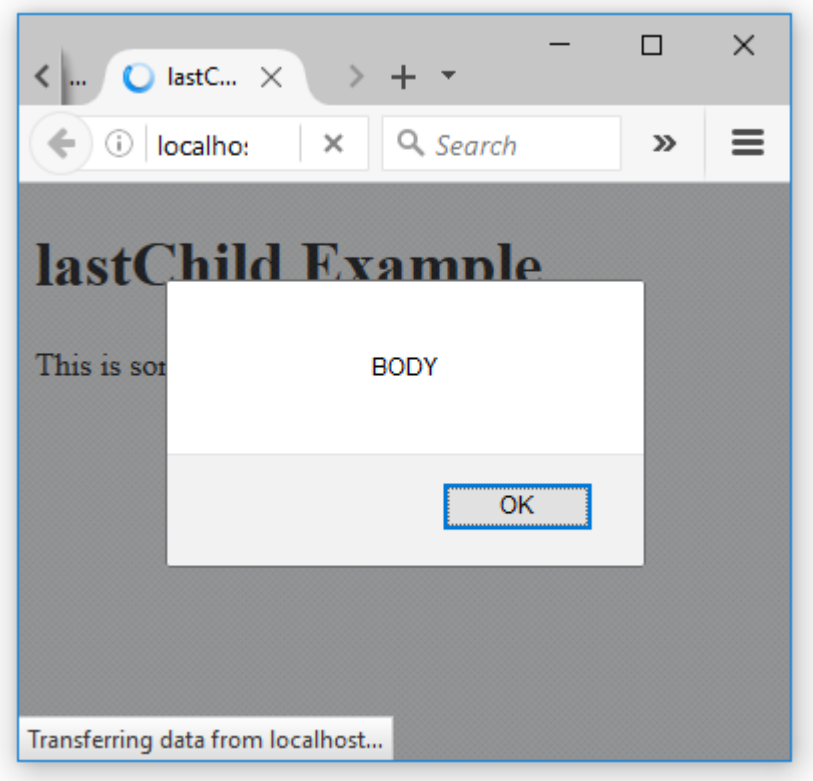
- **Example-9**

```
<!DOCTYPE html>
<!--
firstChild.html
-->
<html>
    <head>
        <title>firstChild Example</title>
    </head>
    <body><h1 id="heading1">firstChild Example</h1>
        <p id="paragraph1">This is some text in a paragraph</p>
        <script>
            var htmlElement = document.body;
            var headElement = htmlElement.firstChild.nodeName;
            alert(headElement);
        </script>
    </body>
</html>
```


firstC...
localhost:8383/JavaScript-
first
This is sc
H1
OK
Transferring data from localhost...

- **Example-10**

```
<!DOCTYPE html>
<!--
lastChild.html
-->
<html>
    <head>
        <title> lastChild Example</title>
    </head>
    <body>
        <h1 id="heading1">lastChild Example</h1>
        <p id="paragraph1">This is some text in a paragraph</p>
        <script>
            var htmlElement = document.documentElement;
            var headElement = htmlElement.lastChild.nodeName;
            alert(headElement);
        </script>
    </body>
</html>
```

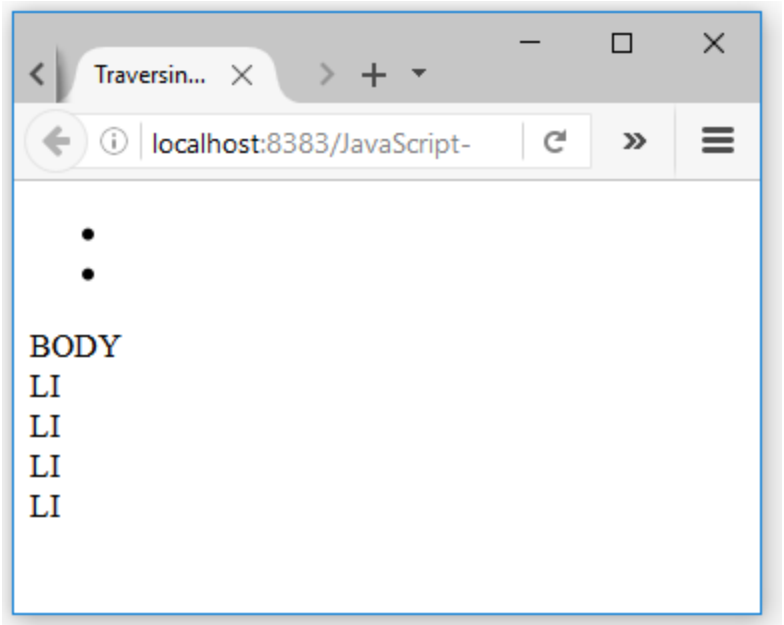

lastC...
localho:
Search
lastChild Example
This is so
BODY
OK
Transferring data from localhost...

ELEMENT TRAVERSAL

- The following set of methods are used to return the element nodes only.
  - childElementCount - Returns the number of child elements.
  - firstElementChild - Points to the fi rst child that is an element
  - lastElementChild - Points to the last child that is an element.
  - previousElementSibling - Points to the previous sibling that is an element.
  - nextElementSibling - Points to the next sibling that is an element

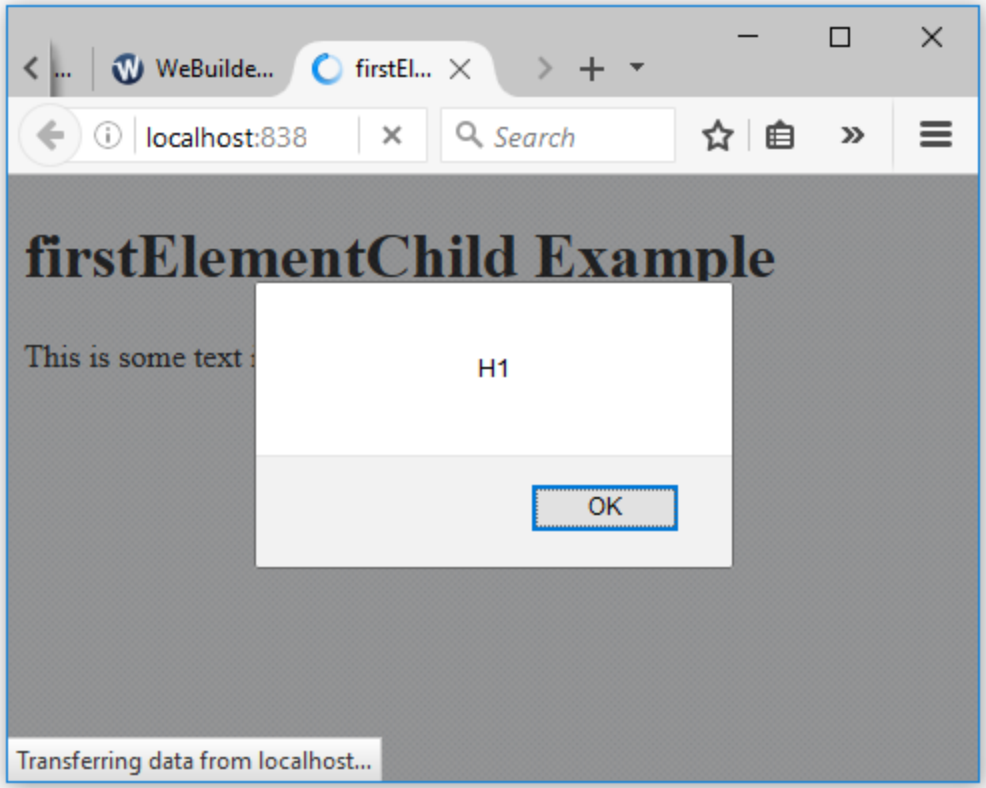
- **Example-11**

```
<!-- traverse-2.html-->
<html>
    <head>
        <title>Traversing Nodes in the DOM</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <ul id="U">
            <li id="A"></li>
            <li id="B"></li>
        </ul>
        <script>
            var ul = document.getElementById("U");
            //What is the parentNode of the ul?
            document.write(ul.parentNode.nodeName + "</br>");
            //What is the first child of the ul?
            document.write(ul.firstElementChild.nodeName+ "</br>");
            //What is the last child of the ul?
            document.write(ul.lastElementChild.nodeName+ "</br>");
            //What is the nextSibling of the first li?
            document.write(document.getElementById("A").nextElementSibling.nodeName+ "</br>");
            //What is the previousSibling of the last li?
            document.write(document.getElementById("B").previousElementSibling.nodeName+ "</br>");
        </script>
    </body>
</html>
```


Traversin...
localhost:8383/JavaScript-
BODY
LI
LI
LI
LI

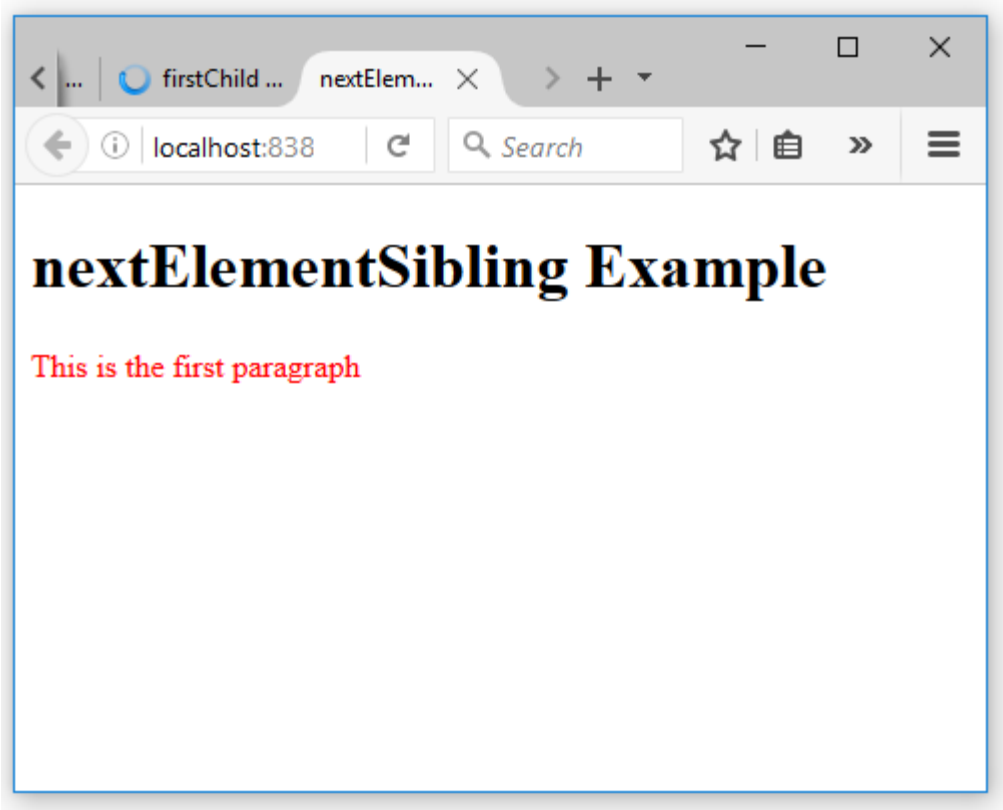
- **Example-12**

```
<!DOCTYPE html>
<!--
firstElementChild.html
-->
<html>
    <head>
        <title>firstElementChild Example</title>
    </head>
    <body>
         <h1 id="heading1">firstElementChild Example</h1>
    <p id="paragraph1">This is some text in a paragraph</p>
    <script>
       var htmlElement = document.body;
       var headElement = htmlElement.firstElementChild.nodeName;
        alert(headElement);
    </script>
    </body>
</html>
```


WeBuilde...
firstEl...
localhost:838
Search
firstElementChild Example
This is some text :
H1
OK
Transferring data from localhost...

- **Example-13**

```
<!DOCTYPE html>
<!--
nextElementSibling.html
-->
<html>
    <head>
        <title>nextElementSibling Example</title>
    </head>
    <body>
        <h1 id="heading1"> nextElementSibling Example</h1>
        <p id="paragraph1">This is the first paragraph</p>
        <script>
            var h1Element = document.getElementById("heading1");
            var pElement;
            pElement = h1Element.nextElementSibling;
            pElement.style.color = "red";
        </script>
    </body>
</html>
```

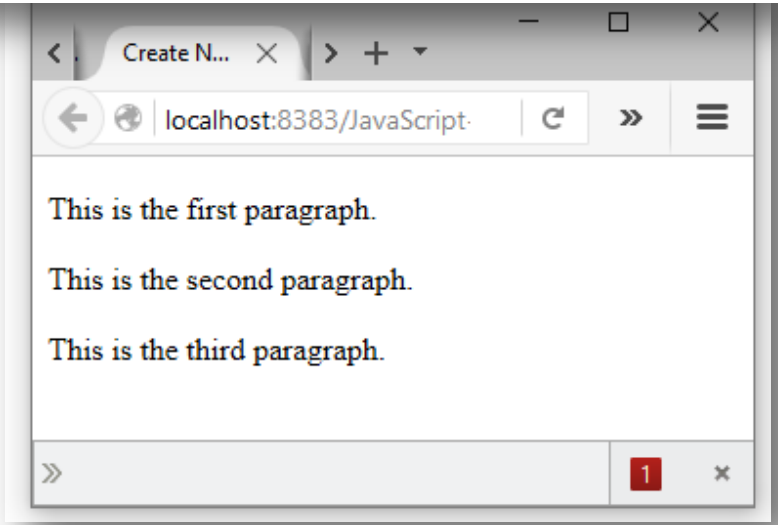

firstChild ...
nextElem...
localhost:838
Search
nextElementSibling Example
This is the first paragraph

Creating New HTML Elements (Nodes)

- To add a new element to the DOM,
- First create the element node.
- Second append it to an existing element.
  - The appendChild() method appended the new element as the last child of the parent.
  - The insertBefore() method appended the new element before a selected node.

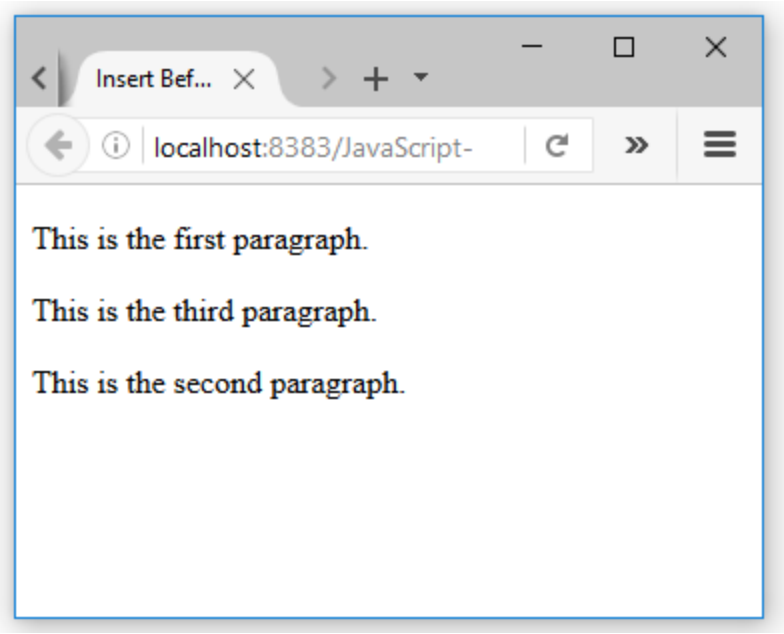
Example-14

```
1  <!DOCTYPE html>
2  <!-- create-node.html -->
3  <html>
4    <head>
5      <title>Create Node</title>
6    </head>
7    <body>
8      <div id="div1">
9        <p id="p1">This is the first  paragraph.</p>
10       <p id="p2">This is the second  paragraph.</p>
11     </div>
12     <script>
13       var para = document.createElement("p");
14       var node = document.createTextNode("This is the third paragraph.");
15       para.appendChild(node);
16
17       var element = document.getElementById("div1");
18       element.appendChild(para);
19     </script>
20   </body>
21 </html>
```



- **Example-15**

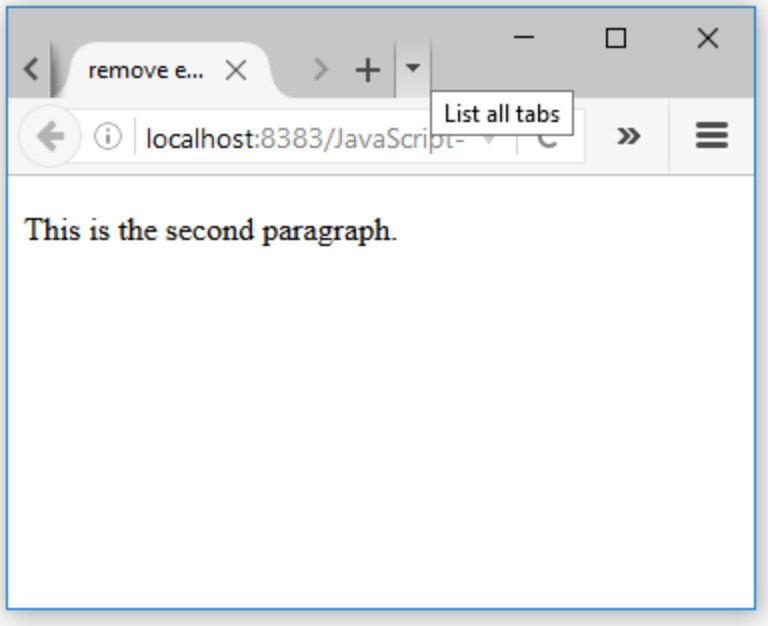
```
<!DOCTYPE html>
<!--
insert-before.html
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div id="div1">
            <p id="p1">This is the first  paragraph.</p>
            <p id="p2">This is the second  paragraph.</p>
        </div>
        <script>
            var para = document.createElement("p");
            var node = document.createTextNode("This is the third paragraph.");
            para.appendChild(node);
            var element = document.getElementById("div1");
            var p2 = document.getElementById("p2");
            element.insertBefore(para, p2);
        </script>
    </body>
</html>
```


Insert Bef...
localhost:8383/JavaScript-
This is the first paragraph.
This is the third paragraph.
This is the second paragraph.

Removing Existing HTML Elements

- To remove an HTML element, you must know the parent of the element:

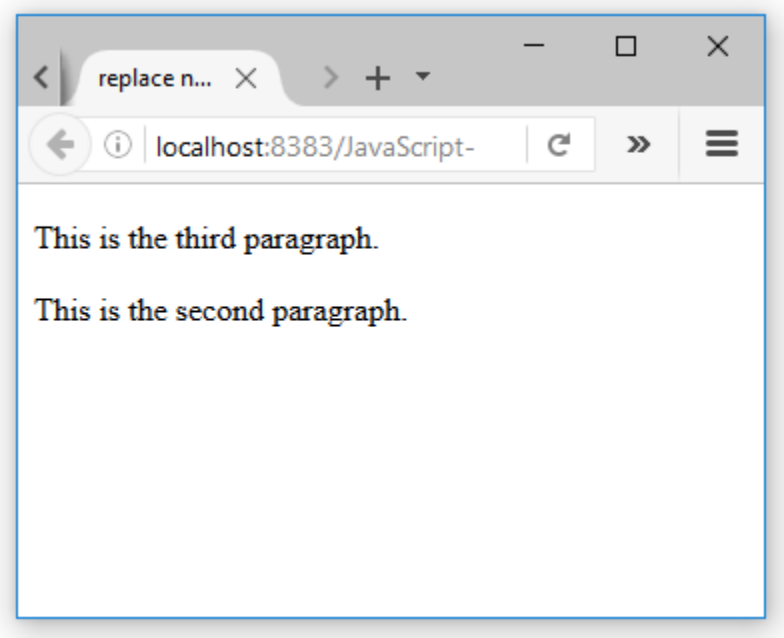
```
1   <!DOCTYPE html>
2   <!-- remove-node.html -->
3   <html>
4     <head>
5       <title>remove element </title>
6     </head>
7     <body>
8       <div id="div1">
9         <p id="p1">This is the first  paragraph.</p>
10        <p id="p2">This is the second  paragraph.</p>
11      </div>
12      <script>
13        var parent = document.getElementById("div1");
14        var child = document.getElementById("p1");
15        parent.removeChild(child);
16      </script>
17    </body>
18  </html>
```


remove e...
List all tabs
localhost:8383/JavaScript-
This is the second paragraph.

Replacing HTML Elements

- The replaceChild() method is used to replace an element to the DOM

```
<!DOCTYPE html>
<!-- replace-node.html -->
<html>
  <head>
    <title>replace node</title>
  </head>
  <body>
    <div id="div1">
      <p id="p1">This is the first paragraph.</p>
      <p id="p2">This is the second paragraph.</p>
    </div>
    <script>
      var para = document.createElement("p");
      var node = document.createTextNode("This is the third paragraph.");
      para.appendChild(node);
      var parent = document.getElementById("div1");
      var child = document.getElementById("p1");
      parent.replaceChild(para, child);
    </script>
  </body>
</html>
```


replace n...
localhost:8383/JavaScript-
This is the third paragraph.
This is the second paragraph.

Thanks!