Javascript Global Functions, Events, DOM, Validation

Javascript Global Functions

- eval():   Evaluates a string and executes it as if it was script code

```
var x=eval("56") * 8;
alert(x);
```

- isFinite(): The isFinite function evaluates an argument to determine whether it is a finite number.

```
if(isFinite("56"))        {
        alert("finite number")
}
else {
        alert("not a finite number")
}
```

Javascript Global Functions

- isNaN(): The isNaN function evaluates an argument to determine if it is “Not a Number”

```
if(isNaN("x56")) {
        alert("not a number")
}
else  {
        alert("it is a number")
}
```

Javascript Global Functions

- Number(): Convert different object values to their numbers.

```
var test1 = new Boolean(true);
var test2 = new Boolean(false);
var test3 = new Date();
var test4 = new String("999");
var test5 = new String("999 888");
var n = Number(test1) + "<br>" + Number(test2) + "<br>" +
Number(test3) + "<br>" + Number(test4) + "<br>" + Number(test5);
document.write(n);
```

Javascript Global Functions

- parseFloat(): Parses a string and returns a floating point number.

```
var a = parseFloat("10") + "<br>";
var b = parseFloat("10.00") + "<br>";
var c = parseFloat("10.33") + "<br>";
var d = parseFloat("34 45 66") + "<br>";
var e = parseFloat(" 60 ") + "<br>";
var f = parseFloat("40 years") + "<br>";
var g = parseFloat("He was 40") + "<br>";
var n = a + b + c + d + e + f + g;
document.write(n)
```

Javascript Global Functions

- parseInt(): Parses a string and returns an integer.

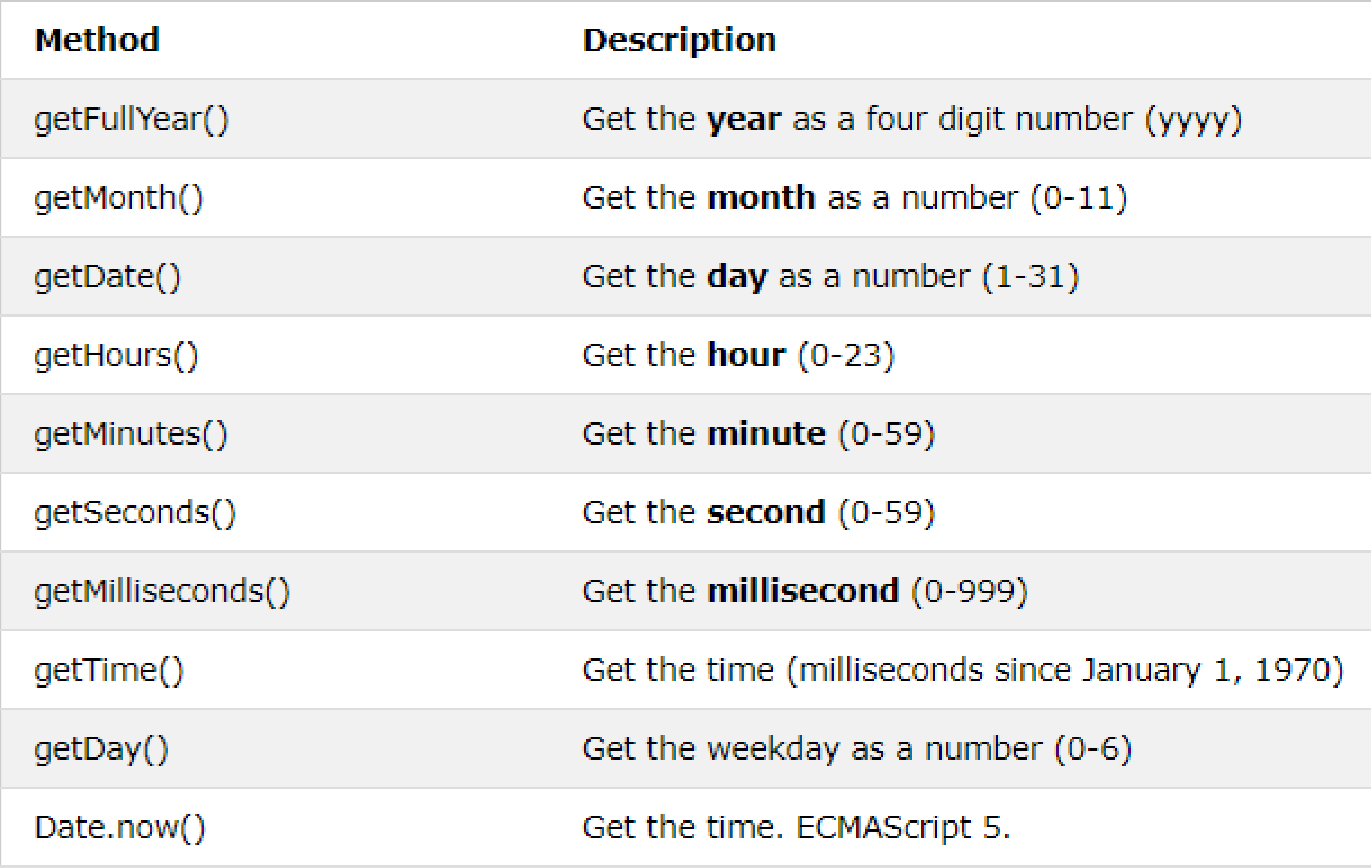
```
var a = parseInt("10") + "<br>";
var b = parseInt("10.00") + "<br>";
var c = parseInt("10.33") + "<br>";
var d = parseInt("34 45 66") + "<br>";
var e = parseInt(" 60 ") + "<br>";
var f = parseInt("40 years") + "<br>";
var g = parseInt("He was 40") + "<br>";
var n = a + b + c + d + e + f + g;
document.write(n)
```

Javascript Date

- JavaScript provides Date object to work with date & time including days, months, years, hours, minutes, seconds and milliseconds.
- Below example shows how to display current date and time using Date object in JavaScript.

```
Date(); //current date
//or
var currentDate = new Date();  //current date
```

Javascript Date – Methods



Method	**Description**
getFullYear()	Get the **year** as a four digit number (yyyy)
getMonth()	Get the **month** as a number (0-11)
getDate()	Get the **day** as a number (1-31)
getHours()	Get the **hour** (0-23)
getMinutes()	Get the **minute** (0-59)
getSeconds()	Get the **second** (0-59)
getMilliseconds()	Get the **millisecond** (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

Javascript Date

```
<html>
<body>
        <h1>Demo: Current Date</h1>
        <p id="p1"></p>
        <p id="p2"></p>
        <script>
                document.getElementById("p1").innerHTML = Date();
                var currentDate = new Date();
                document.getElementById("p2").innerHTML = currentDate;
        </script>
</body>
</html>
```

Javascript Live Clock

```
<html><head><title>JavaScript Live Clock</title></head>
<body onload="clockStart()">
        <div id="clock">
        <span id="hour">hh</span>:<span id="min">mm</span>:<span id="sec">ss</span>
        </div>
<script>
        var timerId; // current timer if started
        function clockStart() {
                if (timerId)
                        return;
                timerId = setInterval(update, 1000);
                update();  // (*)
        }
```

Javascript Live Clock

```
function update() {
  var date = new Date();
  var hours = date.getHours();
  if (hours < 10)
            hours = '0'+hours;
  document.getElementById('hour').innerHTML = hours;
  var minutes = date.getMinutes();
  if (minutes < 10)
            minutes = '0'+minutes;
  document.getElementById('min').innerHTML = minutes;
  var seconds = date.getSeconds();
  if (seconds < 10)
            seconds = '0'+seconds;
  document.getElementById('sec').innerHTML = seconds;
}
</script></body></html>
```

Javascript Events

- interaction with HTML page and HTML elements is handled through events
- Events can be page loads, button click, pressing a key, select data in form controls, focus on control, mouse over and mouse out on any element, etc
- Events are a part of the Document Object Model (DOM) and every HTML element contains a set of events which can trigger JavaScript Code.

Javascript Events

- Document Level Events
  - onload, onunload
- Form Level Events
  - Onsubmit, Onreset,
  - Onchange, onselect, onblur, onfocus
- Keyboard Events
  - Onkeydown, onkeypress, onkeyup
- Mouse Events
  - Onclick, ondblclick
  - onmouseover, onmouseout

Call Function on Event – Ex 1

```
<html>
  <head>
    <script>
     function callme() {
        alert("Hello");
        document.write("Hello");
     }
    </script>
  </head>
  <body>
    <form>
      <input type = "button" onclick = "callme()" value = "Click Me">
    </form>
 </body>
</html>
```

Call Function on Event – Ex 2

```
<html>
<head>
<title>Title of the Page</title>
<script>
function goodbye(){
        alert("Goodbye!")
}
function hello(){
        alert("Hello World")
}
</script>
</head>
<body onLoad="hello()“ onUnload="goodbye()">
        Now you are leaving this page <a href="page2.htm">for another</a>.
</body>
</html>
```

Event – Ex 3

```
<html>
<head>
<script type="text/javascript">
<!–
function over()
{
        alert("Mouse Over");
}
function out()
{
        alert("Mouse Out");
} //-->
</script>
</head>
<body>
<div onmouseover="over()" onmouseout="out()">
<h2> This is inside the division </h2> </div> </body> </html>
```

Javascript Event

- **onFocus** means “when the cursor is placed on this element.” This event handler is placed in one- and multiple-line entry fields as well as in drop-down lists, for example, in order to check zip codes that the user has entered for typos or extra digits.
- **onBlur** is the opposite of **onFocus**. This event handler is used with exactly the same elements, namely one- and multiple-line text fields and drop-down lists. It becomes active if the focus is removed from the element, for example, as soon as the mouse is clicked on another element.

Event – Ex 4

```
<html>
<head>
<title>Title of the Page</title>
<script
language="JavaScript">
<!--
function message(field){
window.status="You have
placed the cursor in the " +
field + "
field!"
}
//-->
</script>
</head>
```

```
<body>
<form>
   <input type="text"
          onFocus="message('one-line text')">
   <textarea rows=2 cols=20 wrap=virtual
          onFocus="message('multipleline text’)”>
   </textarea>
<select onFocus="message('Field of choice')">
   <option>Point 1
   <option>Point 2
   <option>Point 3
</select>
</form>
</body>
</html>
```

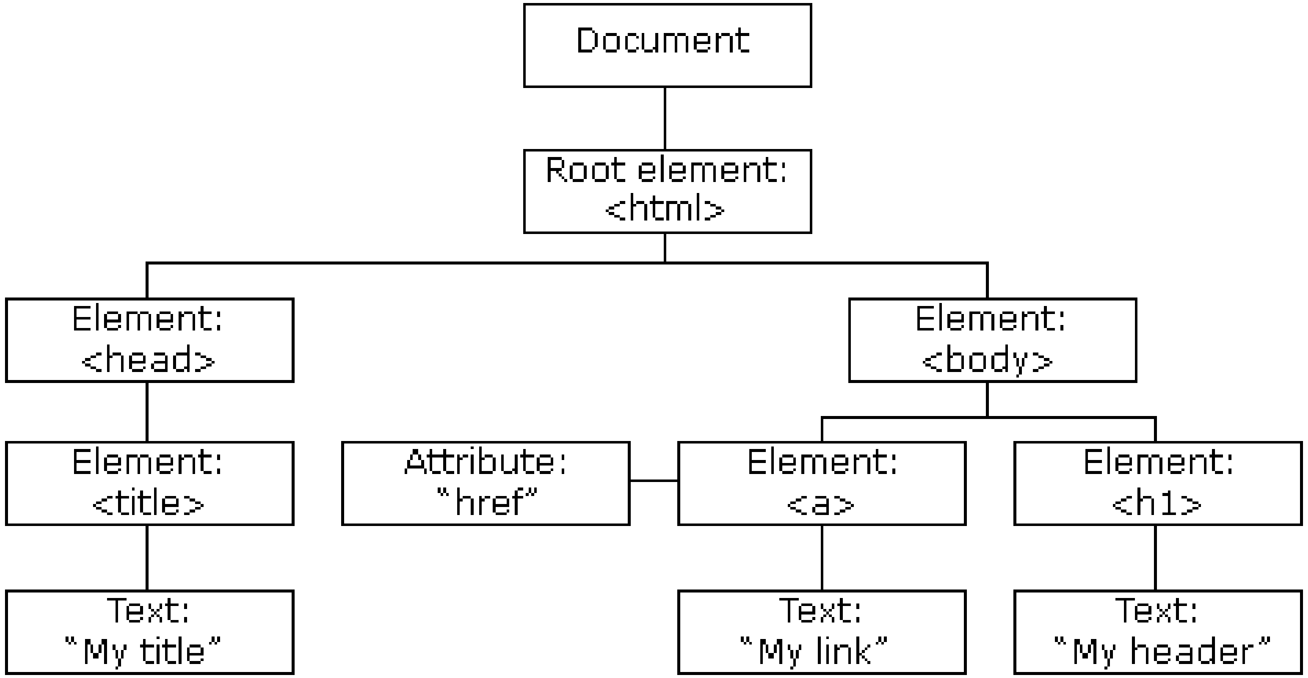
Event – Ex 5

```
<html>
<head>
<title>Title of the Page</title>
<script
language="JavaScript">
<!--
function message(field){
//window.status="You have
placed the cursor in the " +
field + " field!"
alert("You have now left the "
+ field + " field!")
}
//-->
</script>
</head>
```

```
<body>
<form>
<input type="text" onBlur="message('one-line
text')">
<textarea rows=2 cols=20 wrap=virtual
onBlur="message('multipleline text ')"></textarea>
<select onBlur="message('Field of choice')">
   <option>Point 1
   <option>Point 2
   <option>Point 3
</select>
</form>
</body>
</html>
```

HTML DOM

- When a web page is loaded, browser creates a Document Object Model of the page.
- The HTML DOM model is constructed as a tree of Objects:



HTML DOM

With the object model, JavaScript can do following:

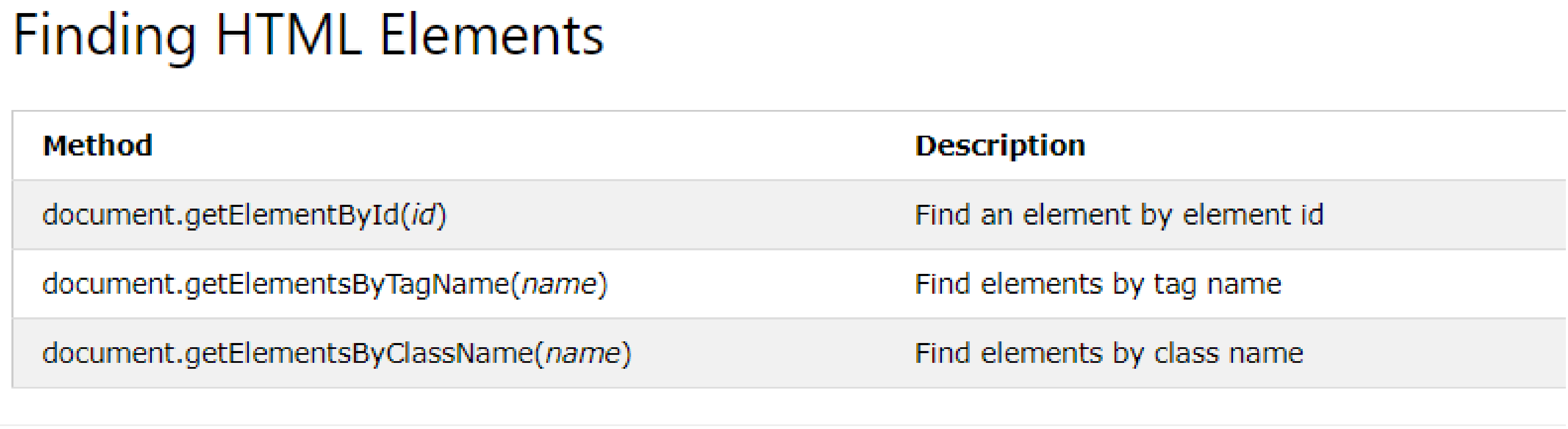
- modify all the HTML elements and attributes in the
- change all the CSS styles in the page
- Add remove existing HTML elements and attributes
- add new HTML elements and attributes
- react to all existing HTML events in the page
- create new HTML events in the page

HTML DOM - EX

- In the DOM, all HTML elements are defined as **objects**.
- Below example changes the content (the innerHTML) of the <p> element with id="demo“
- getElementById is a method and innerHTML is a property

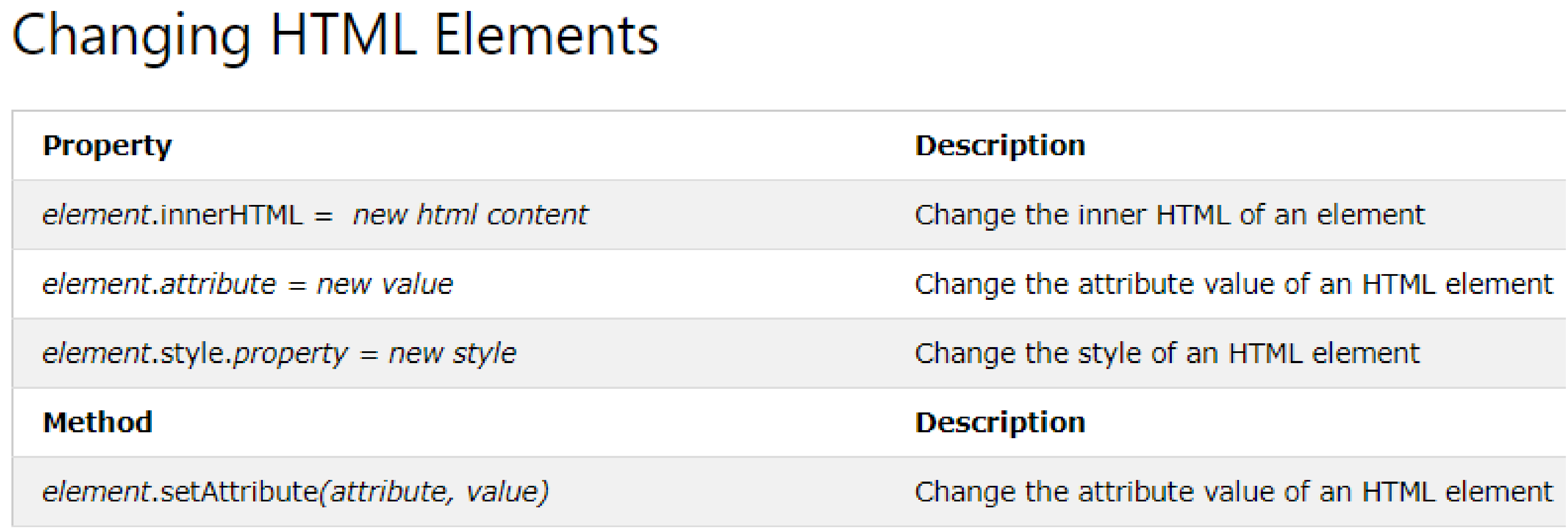
```
<body>
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML = "Hello World!";
</script>
</body>
```

DOM Object – Method & Properties



Finding HTML Elements

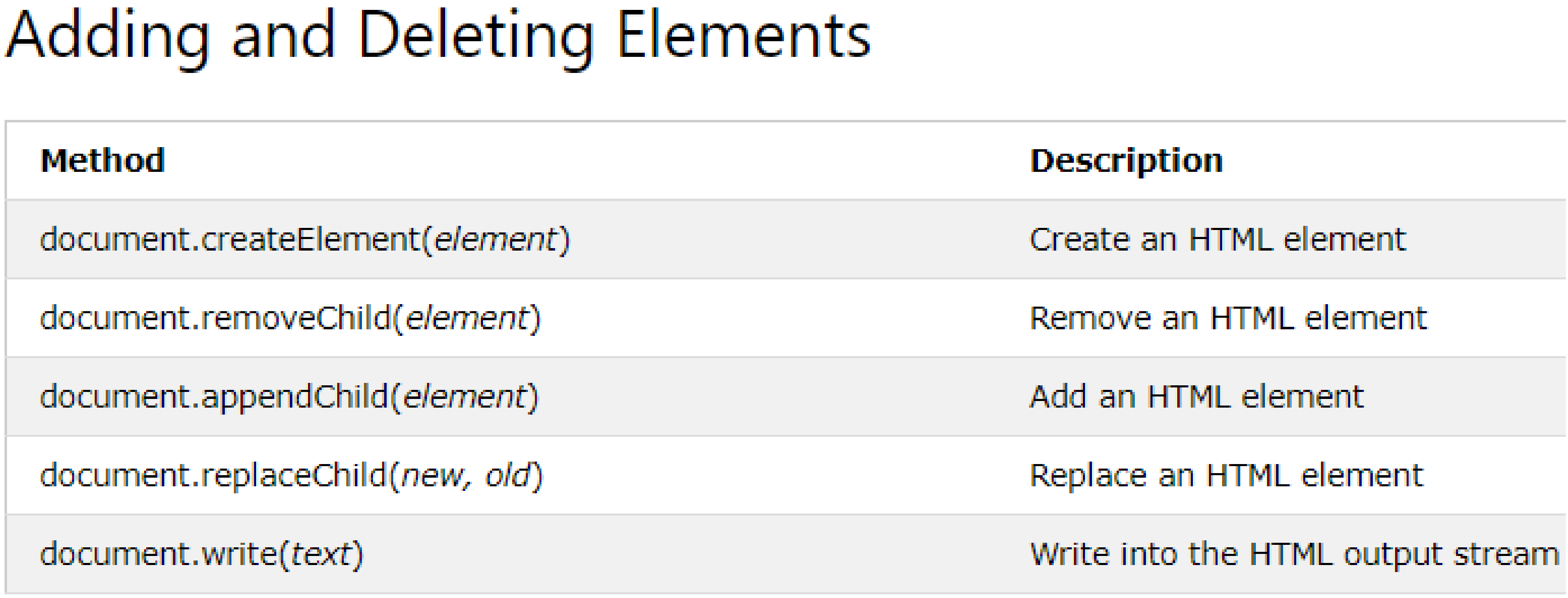
Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name



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

DOM Object – Method & Properties



Adding and Deleting Elements

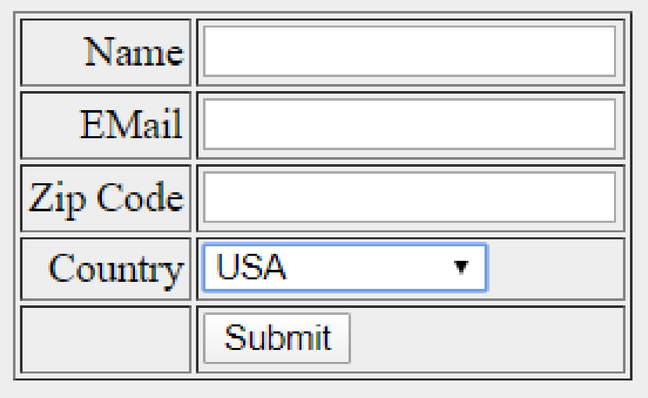
Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new, old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

```
<html>
<head>
<script>
        var btn = document.querySelector('button');
        function random(number) {
            return Math.floor(Math.random() * (number+1));
        }
        function changeBgColor() {
            var rCol = 'rgb(' + random(255) + ',' + random(255)
                            + ',' + random(255) + ') ';
            document.body.style.backgroundColor = rCol;
        }
</script>
</head>
<body>
        <button onclick= "changeBgColor()">Change
color</button>
</body>
</html>
```

Form Validation

- JavaScript provides a way to validate form's data on the client's computer before sending it to the web server.
- Form validation generally performs two functions.
  - Basic Validation – check all the mandatory fields are filled in.
  - Data Format Validation – data entered checked for correct form and value with appropriate logic to test correctness of data.

```
<html>  <head> <title>Form Validation</title>
    <script type = "text/javascript">
      <!--          // Form validation code will come here.        //-->
    </script></head>
  <body>
    <form action = "next_page" name = "myForm" onsubmit = "return(validate());">
      <table cellspacing = "2" cellpadding = "2" border = "1">
        <tr>  <td align = "right">Name</td>  <td><input type = "text" name = "Name" /></td>  </tr>
        <tr>  <td align = "right">EMail</td>  <td><input type = "text" name = "EMail" /></td>    </tr>
        <tr>  <td align = "right">Zip Code</td>   <td><input type = "text" name = "Zip" /></td>   </tr>
        <tr>   <td align = "right">Country</td>  <td>
            <select name = "Country">
           <option value = "1">USA</option>
             <option value = "2">UK</option>
             <option value = "3">Nepal</option>
            </select>
          </td>   </tr>
        <tr>  <td align = "right"></td>  <td><input type = "submit" value = "Submit" /></td>   </tr>
      </table>
    </form> </body> </html>
```



Name	
EMail	
Zip Code	
Country	USA ▾
	Submit

```
<script type = "text/javascript">
  function validate() {
     if( document.myForm.Name.value == "" ) {
       alert( "Please provide your name!" );   document.myForm.Name.focus() ;    return false;
     }
     if( document.myForm.EMail.value == "" ) {
       alert( "Please provide your Email!" );   document.myForm.EMail.focus() ;   return false;
     }
     if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
       document.myForm.Zip.value.length != 5 ) {
       alert( "Please provide a zip in the format #####." );   document.myForm.Zip.focus() ;   return false;
     }
     if( document.myForm.Country.value == "-1" ) {
       alert( "Please provide your country!" );     return false;
     }
     return( true );
   }
</script>
```

Image Slider

```
<html>
<head>
  <script type="text/javascript">
    <!--
    var image1=new Image();
    image1.src=“image1.jpg“ ;
    var image2=new Image() ;
    image2.src=“image2.jpg“ ;
    var image3=new Image() ;
    image3.src=“image3.jpg“ ;
    //-->
  </script>
</head>
```

Image Slider

```
<body>
<img src=“image1.jpg" name="slide" width="600" height="256"
style="position:absolute;left:300px" />
<script>
<!--
//variable that will increment through the images
var step=1
function slideit(){
        //if browser does not support the image object, exit.
        if (!document.images)
                return;
        document.images.slide.src=eval("image"+step+".src");
```

Image Slider

```
        document.images.slide.src=eval("image"+step+".src");
        if (step<3)
                step++;
        else
                step=1;
        //call function "slideit()" every 2.5 seconds
        setTimeout("slideit()", 2500);
}
slideit()
//-->
</script>
</body></html>
```