
JAVASCRIPT

WEEK 6 – 09/30/2019



THE JAVA SCRIPT PROGRAMMING LANGUAGE

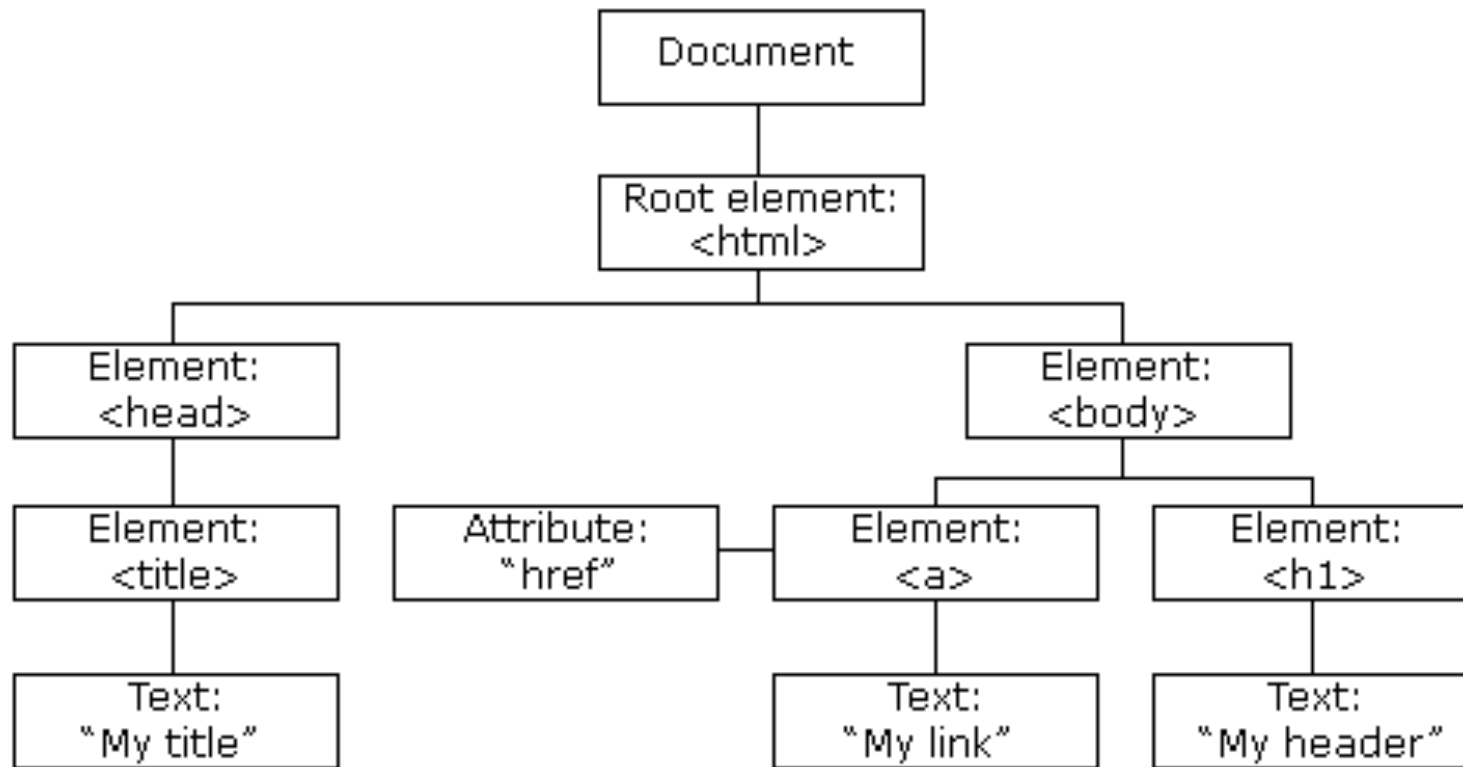
- Really has nothing to do with the Java programming language
- “JavaScript” is just a name (a.k.a. JS conforms to ECMAScript)
 - Java is to Javascript what Ban is to Banana
- It was originally called mocha, then LiveScript
- Then Netscape and Sun reached a licensing deal and renamed it
- JavaScript is the most widely-used programming language **

ONE LANGUAGE; TWO PLATFORMS

- Client side – embedded within HTML, executed by browser
- Server side – embedded within server-side scripts, executed by the web server
- We begin with Client Side scripting ...

THE HTML DOM (DOCUMENT OBJECT MODEL)

- For an HTML page, the browser creates a DOM when the page is loaded
- The DOM provides a “glossary” of objects on the page
- Each `<tag>` on the page is defined in the DOM
- The DOM contains a hierarchy of objects on the page
- JavaScript can manipulate objects “real time” as the page is being rendered by the browser.



HTML DOM TREE OF OBJECTS

THE HTML DOM

- 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

DEMO

```
<html>
```

```
<body>
```

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

```
<script>
```

Method

Property

```
document.getElementById("demo").innerHTML="My paragraph of text.";
```

```
</script>
```

```
</body>
```

```
</html>
```

DEMO 1

```
<html>
  <body>
    <h2>Javascript Statements</h2>
    <p>A javascript program can manipulate the window alert object.</p>
    <script>
      alert("Hello World");
    </script>
  </body>
</html>
```


DEMO 2

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>What can Javascript do?</h2>
```

```
<p id="demo">
```

```
  Javascript can change HTML content.
```

```
</p>
```

```
<button type="button" onclick="document.getElementById('demo').innerHTML='Hello Javascript!'">Click  
Here</button>
```

```
</body>
```

```
</html>
```

DEMO 3

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>What can Javascript Do?</h2>
```

```
<p id="demo">Javascript can change the style attribute.</p>
```

```
<button type="button"  
onclick="document.getElementById('demo').style.fontSize='25px'">Click Here</button>
```

```
</body>
```

```
</html>
```

JAVASCRIPT SYNTAX

- Javascript code is embedded between a `<script>..</script>` tag set within your HTML file.
- Can be within `<head>` section or the `<body>` section, or both.
- `{ }` encloses a block of code
- `;` marks the end of each java script statement
- Functions are identified by `()` following the function name
 - Pre-defined
 - User-defined

JAVASCRIPT SYNTAX

You can use single or double quotes

- “This isn’t fair”
- ‘He said “This isn’t fair ” ’

Variables – Strings or Numeric or Boolean

- Defined by var
- Variable name begins with a letter, \$, or _
- Case sensitive
 - “Value” does not equal “value”
- JS programmers usually use camelCaps

SETTING VARIABLES

- `var firstName = 'Sreesha';`
- `var lastName = 'Nath';`
- `var fullName = firstName + ' ' + lastName`
- `var isInstructor = true;`
- `var score = 0;`
- `score = 50;`
- `score = score + 10;`
- `score += 10;` (adds value on right to variable on left)
- `var name = 'Sreesha';`
- `var message = 'Hey';`
- `message = message + ' ' + name;`
- `message += ' ' + name;`

CREATING ARRAYS

```
var days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri'];  
alert (days[0]);
```

```
document.write('<p>');  
document.write(days[2]);  
document.write('</p>');
```

```
var i = 0;  
while (i < 5) {  
  document.write('<p>');  
  document.write(days[i]);  
  document.write('</p>');  
  i += 1;  
}
```

USING FUNCTIONS

- Step One: Create the Function (often done in <head> the section)

```
function printToday() {  
    var today = new Date();  
    document.write(today.toString());  
}
```

USING FUNCTIONS

- Step Two: Call the Function. (in the <body> section)

```
<h1>Using Functions</h1>
```

```
<p>Today is
```

```
<script>
```

```
  printToday();
```

```
</script>
```

```
</p>
```


USING FUNCTIONS

- Step One: add TIME to the Function

```
function printToday() {  
    var today = new Date();  
    var hours = today.getHours();  
    var minutes = today.getMinutes();  
    var seconds = today.getSeconds();  
    document.write(today.toString());  
    document.write(" ");  
    document.write(hours + ":" + minutes + ":" + seconds);  
}
```

USING FUNCTIONS

- Step Two: Call the function.

```
<h1>Using Functions</h1>
```

```
<p>Today is
```

```
<script>
```

```
printToday();
```

```
</script>
```

```
</p>
```

JAVASCRIPT CLIENT SIDE EVENTS

Common HTML Events

Here is a list of some common HTML events:

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

EVENTS

- Onclick

```
<html>
```

```
<body>
```

```
<h1 onclick="alert('You clicked this heading');">Click Here!</h1>
```

```
</body>
```

```
</html>
```

EVENTS

```
<html>
  <body>
    <div onmouseover="mOver(this);" onmouseout="mOut(this);"
style="background-color:yellow; width:120px; height:20px;
padding:40px;"> Mouse Over the Box </div>
    <script>
      function mOver(obj){
        obj.innerHTML = "Moused Over";
      }

      function mOut(obj){
        obj.innerHTML = "Mouse Over This Box";
      }
    </script>
  </body>
</html>
```

GO PLAY!

- <http://www.w3schools.com/js/default.asp>