JavaScript

- JavaScript is *the* scripting language of the Web.
- JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more.

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license

Example

```
<html>
<body>

<script type="text/javascript">
      document.write("This is my first JavaScript!");
</script>

</body>
</html>
```

This is my first JavaScript!

How to Handle Simple Browsers

- Browsers that do not support JavaScript, will display JavaScript as page content.
- The HTML comment tag should be used to "hide" the JavaScript.
- Just add an HTML comment tag <!-- before the first JavaScript statement, and a --> (end of comment) after the last JavaScript statement.

Example

```
<html>
<body>
<script type="text/javascript">
<!--
document.write("Hello World!");
//-->
</script>
</body>
</html>
```

The two forward slashes at the end of comment line (//) is the JavaScript comment symbol. This prevents JavaScript from executing the --> tag.

What can a JavaScript do?

- JavaScript gives HTML designers a programming tool
- JavaScript can put dynamic text into an HTML page JavaScript can react to events
- JavaScript can read and write HTML elements
- JavaScript can be used to validate data
- JavaScript can be used to detect the visitor's browser
- JavaScript can be used to create cookies

How To Use?

- <script> tag is used to insert a JavaScript into an HTML page.
- Between <body> tag
- Between <head> tag

How To Use (Cont.)?

- JavaScripts in the **body** section will be executed WHILE the page loads.
- JavaScripts in the **head** section will be executed when CALLED.

JavaScript Statements

- JavaScript is case sensitive.
- Use of semicolon(;) in the end of statement is ***optional***.

```
document.write("Hello");

document.write("Hello")
```


JavaScript Comments

- Single line comments start with //.

```
<script type="text/javascript">
// Write a heading
document.write("<hl>This is a heading</hl>");
// Write two paragraphs:
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

- Multi line comments start with /* and end with */.

```
<script type="text/javascript">
/*
The code below will write
one heading and two paragraphs
*/
document.write("<hl>This is a heading</hl>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

JavaScript Variables

- JavaScript variables are used to hold values or expressions.

x=5, y=6, z=x+y

Rules for JavaScript variable names:

- Variable names are case sensitive (y and Y are two different variables)
- Variable names must begin with a letter or the underscore character

JavaScript Variables

- Declaration

  var x;
  var carname;

- Assign Values

  x=5;
  carname="Toyota";

JavaScript Arithmetic Operators

- Y=5

Operator	Description	Example	Result
+	Addition	x=y+2	x=7
-	Subtraction	x=y-2	x=3
*	Multiplication	x=y*2	x=10
/	Division	x=y/2	x=2.5
%	Modulus (division remainder)	x=y%2	x=1
++	Increment	x=++y	x=6
--	Decrement	x=--y	x=4

JavaScript Assignment Operators

- x = 5 and y=10

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

+ Operator and Strings

- To add two or more string variables together, use the + operator.

```
<script type="text/javascript">
txt1="What a very";
txt2="nice day";
txt3=txt1+" "+txt2;
document.write(txt3);
</script>
```

What a very nice day

Comparison Operators

Given that **x=5**, the table below explains the comparison operators:

Operator	Description	Example
==	is equal to	x==8 is false
===	is exactly equal to (value and type)	x===5 is true x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

```
if (age<18) document.write("Too young");
```

Logical Operators

Given that **x=6 and y=3**, the table below explains the logical operators:

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

If...Else Statements

- used to perform different actions based on different conditions.
- **if statement** - use this statement to execute some code only if a specified condition is true
- **if...else statement** - use this statement to execute some code if the condition is true and another code if the condition is false
- **if...else if....else statement** - use this statement to select one of many blocks of code to be executed
- **switch statement** - use this statement to select one of many blocks of code to be executed

Example (If)

```
if (condition)
  {
  code to be executed if condition is true
  }
```

Note that if is written in lowercase letters. Using uppercase letters (IF) will generate a JavaScript error!

Example

```
<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10

var d=new Date();
var time=d.getHours();

if (time<10)
  {
  document.write("<b>Good morning</b>");
  }
</script>
```


Example (If...else)

```
if (condition)
  {
  code to be executed if condition is true
  }
else
  {
  code to be executed if condition is not true
  }
```

Example

```
<script type="text/javascript">
//If the time is less than 10, you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.

var d = new Date();
var time = d.getHours();

if (time < 10)
  {
  document.write("Good morning!");
  }
else
  {
  document.write("Good day!");
  }
</script>
```

Example(If...else if...else)

```
if (condition1)
  {
  code to be executed if condition1 is true
  }
else if (condition2)
  {
  code to be executed if condition2 is true
  }
else
  {
  code to be executed if condition1 and condition2 are not true
  }
```

Example

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
  {
  document.write("<b>Good morning</b>");
  }
else if (time>10 && time<16)
  {
  document.write("<b>Good day</b>");
  }
else
  {
  document.write("<b>Hello World!</b>");
  }
</script>
```


Switch Statement

- used to perform different actions based on different conditions.

```
switch(n)
{
case 1:
  execute code block 1
  break;
case 2:
  execute code block 2
  break;
default:
  code to be executed if n is different from case 1 and 2
}
```

Example (Switch)

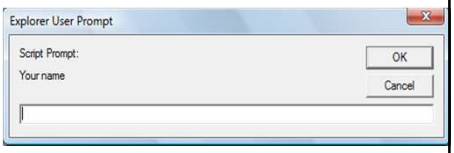
```
<script type="text/javascript">
//You will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.

var d=new Date();
theDay=d.getDay();
switch (theDay)
{
case 5:
  document.write("Finally Friday");
  break;
case 6:
  document.write("Super Saturday");
  break;
case 0:
  document.write("Sleepy Sunday");
  break;
default:
  document.write("I'm looking forward to this weekend!");
}
</script>
```


Popup Boxes

- **Alert Box**
  alert(“Hello Every Body");

- **Confirm Box**
  confirm(“Press a button");

- **Prompt Box**
  prompt("Your name","")



Functions

- A function will be executed by an event or by a call to the function.

```
function functionname(var1,var2,...,varX)
{
some code
}

function displaymessage()
{
alert("Hello World!");
}
```


Example (Function)

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!");
}
</script>
</head>

<body>
<form>
<input type="button" value="Click me!" onclick="displaymessage()" />
</form>
</body>
</html>
```

JavaScript Loops

You want the same block of code to run over and over again in a row. Instead of adding several almost equal lines in a script we can use loops to perform a task like this.

In JavaScript, there are two different kind of loops:

- **for** - loops through a block of code a specified number of times
- **while** - loops through a block of code while a specified condition is true

The for Loop

```
for (var=startvalue;var<=endvalue;var=var+increment)
{
code to be executed
}
```

```
<script type="text/javascript">
var i=0;
for (i=0;i<=5;i++)
{
document.write("The number is " + i);
document.write("<br />");
}
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5

While Loop

```
while (var<=endvalue)
  {
  code to be executed
  }
```

```
<script type="text/javascript">
var i=0;
while (i<=5)
  {
  document.write("The number is " + i);
  document.write("<br />");
  i++;
  }
```


The break Statement

- The break statement will break the loop and continue executing the code that follows after the loop (if any).

```
<script type="text/javascript">
var i=0;
for (i=0;i<=10;i++)
  {
  if (i==3)
    {
    break;
    }
  document.write("The number is " + i);
  document.write("<br />");
  }
</script>
```

The number is 0
The number is 1
The number is 2

The continue Statement

- The continue statement will break the current loop and continue with the next value.

```
<script type="text/javascript">
var i=0
for (i=0;i<=10;i++)
  {
  if (i==3)
    {
    continue;
    }
  document.write("The number is " + i);
  document.write("<br />");
  }
</script>
```

The number is 0
The number is 1
The number is 2
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10

JavaScript Events

- Events are actions that can be detected by JavaScript.
- Examples of events:
  - A mouse click
  - A web page or an image loading
  - Mousing over a hot spot on the web page
  - Selecting an input field in an HTML form
  - Submitting an HTML form
  - A keystroke

Events

- **onLoad and onUnload**
- **onFocus, onBlur and onChange**
- **onSubmit**
- **onMouseOver and onMouseOut**

JavaScript Objects

- JavaScript is an Object Oriented Programming (OOP) language.
- An OOP language allows you to define your own objects and make your own variable types.
- We will start by looking at the built-in JavaScript objects.

JavaScript Objects and Properties

- Properties are the values associated with an object.

```
<script type="text/javascript">
var txt="Hello World!";
document.write(txt.length);
</script>
```

12

JavaScript String Object

- The String object is used to manipulate a stored piece of text.
- Some String Methods

search()

toLowerCase()

toUpperCase()

JavaScript Date Object

- The Date object is used to work with dates and times.

```
new Date() // current date and time
new Date(milliseconds) //milliseconds since 1970/01/01
new Date(dateString)
new Date(year, month, day, hours, minutes, seconds, millisecond
```

```
today = new Date()
d1 = new Date("October 13, 1975 11:13:00")
d2 = new Date(79,5,24)
d3 = new Date(79,5,24,11,33,0)
```


Data Comparison

?

JavaScript Array Object

- The Array object is used to store multiple values in a single variable.
- An array can be defined in three ways.

1:

```
var myCars=new Array(); // regular array (add an optional integer
myCars[0]="Saab";       // argument to control array's size)
myCars[1]="Volvo";
myCars[2]="BMW";
```

2:

```
var myCars=new Array("Saab","Volvo","BMW"); // condensed array
```

3:

```
var myCars=["Saab","Volvo","BMW"]; // literal array
```


JavaScript Math Object

- The Math object allows you to perform mathematical tasks.

Math.PI

Math.E

Math.sqrt(number)

Math.round(4.7)

JavaScript Navigator Object

- The Navigator object allows you to check/validate your Browser.

- Navigator.appName,;
- Navigator.appVersion