Java? Script?:
An Introduction to Javascript

1. OBJECTIVES

At the end of the session, the student should be able to:

a. answer the question “What is Javascript?”; and
b. create a simple Javascript program observing proper syntax;

2. DISCUSSION

Before starting to learn what Javascript is, make sure you already understood HTML / XHTML.

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)
- A JavaScript consists of lines of executable computer code
- A 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

Are Java and JavaScript the Same?

NO! Java and JavaScript are two completely different languages in both concept and design!
Java (developed by Sun Microsystems) is a powerful and much more complex programming language - in the same category as C and C++.

What can a JavaScript Do?

- JavaScript gives HTML designers a programming tool - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- JavaScript can put dynamic text into an HTML page - A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page
- JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element
- JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing
- JavaScript can be used to detect the visitor's browser - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- JavaScript can be used to create cookies - A JavaScript can be used to store and retrieve information on the visitor's computer

How do I put a JavaScript into an HTML Page?

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

The code above will produce this output on an HTML page:

```
Hello World!
```

Example Explained

To insert a JavaScript into an HTML page, we use the <script> tag (also use the type attribute to define the scripting language). So, the <script type="text/javascript"> and </script> tells where the JavaScript starts and ends.

The word **document.write** is a standard JavaScript command for writing output to a page.
By entering the **document.write** command between the <script type="text/javascript"> and </script> tags, the browser will recognize it as a JavaScript command and execute the code line. In this case the browser will write Hello World! to the page:

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

<u>Where do I put the JavaScript?</u>

<u>Scripts in the head section</u>: Scripts to be executed when they are called, or when an event is triggered, go in the head section. When you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

<u>Scripts in the body section</u>: Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page.

<u>Scripts in both the body and the head section</u>: You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

<u>Using an External JavaScript</u>: Sometimes you might want to run the same JavaScript on several pages, without having to write the same script on every page. To simplify this, you can write a JavaScript in an external file. Save the external JavaScript file with a .js file extension.

Note: The external script cannot contain the <script> tag!
To use the external script, point to the .js file in the "src" attribute of the <script> tag:

```
<html>
<head>
<script src="xxx.js"></script>
</head>
<body>
</body>
</html>
```

Note: Remember to place the script exactly where you normally would write the script!

<u>What is the syntax of Javascript?</u>

<u>Variables</u>
A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.

Rules for variable names:

- Variable names are case sensitive
- They must begin with a letter or the underscore character

IMPORTANT! JavaScript is case-sensitive! A variable named strname is not the same as a variable named STRNAME!

Declare a Variable

```
var strname = some value
or
strname = some value
```

Assign a Value to a Variable

```
var strname = "Hege"
or
strname = "Hege"
```

Note: Be careful of making you variables global or local.

Conditional Statements

If Statement	***If...else Statement***	***If...else if...else Statement***

`<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>")`
`}`
 	
`if (time==11)`
`{`
`document.write("<b>Lunch-time!</b>")`
`}`
`</script>`
	`<script type="text/javascript">`
	`//If the time is less than 10,`
	`//you will get a "Good morning"`
	`//Otherwise you will get “Good day"`
	
	`var d = new Date()`
	`var time = d.getHours()`
	
	`if (time < 10)`
	`{`
	`document.write("Good morning!")`
	`}`
	`else`
	`{`
	`document.write("Good day!")`
	`}`
	`</script>`
		`<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

```
<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>
```

Operators

Arithmetic Operators

Operator	***Description***	***Example***	***Result***

+	Addition	x=2
		y=2
		x+y
			4

-	Subtraction	x=5
		y=2
		x-y
			3

*	Multiplication	x=5
		y=4
		x*y
			20

/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operators

Operator	***Example***	***Is The Same As***
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Comparison Operators

Operator	***Description***	***Example***
==	is equal to	5==8 returns false

===	is equal to (checks for both value and type)	x=5
		y="5"
		x==y returns true
		x===y returns false

!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operators

Operator	***Description***	***Example***

&&	and	x=6
		y=3
		(x < 10 && y > 1) returns true

||	or	x=6
		y=3
		(x==5 || y==5) returns false

!	not	x=6
		y=3
		!(x==y) returns true

String Operator

`txt1="What a very"`
`txt2="nice day!"`
`txt3=txt1+" "+txt2`
	`txt1="What a very "`
	`txt2="nice day!"`
	`txt3=txt1+txt2`

Conditional Operator

`Syntax:`	`variablename=(condition)?value1:value2`
`Example:`	`greeting=(visitor=="PRES")?"Dear President ":"Dear "`

Loops

For Loops	***While Loops***	***Do...While Loops***

`<html>`
`<body>`
`<script type="text/javascript">`
`var i=0`
`for (i=0;i<=10;i++)`
`{`
`document.write("The number is " + i)`
`document.write("<br />")`
`}`
`</script>`
`</body>`
`</html>`
	`<html>`
	`<body>`
	`<script type="text/javascript">`
	`var i=0`
	`while (i<=10)`
	`{`
	`document.write("The number is " + i)`
	`document.write("<br />")`
	`i=i+1`
	`}`
	`</script>`
	`</body>`
	`</html>`
		`<html>`
		`<body>`
		`<script type="text/javascript">`
		`var i=0`
		`do`
		`{`
		`document.write("The number is " + i)`
		`document.write("<br />")`
		`i=i+1`
		`}`
		`while (i<0)`
		`</script>`
		`</body>`
		`</html>`

Break	***Continue***
The break command will stop the loop and continue on with the code after the loop (if there are)	The continue command will stop the current loop and continue with the next value

Functions

- A *function* is a group of codes that can be executed when an event occurs or when the function is called.
- Functions are defined at the beginning of the page, at the <head> section

```
<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>
```

```
<html>
<head>
<script type="text/javascript">
function product(a,b){
return a*b
}
</script>
</head>
<body>
<script type="text/javascript">
document.write(product(4,3))
</script>
</body>
</html>
```

If you can see on the first example, we used used something called "***onclick***". These are what we call Javascript events.

<u>What are events?</u>

- Events are actions that can be detected by Javascript
- Every element on a web page can trigger an event.
- We define events inside the HTML tags. eg. The **onclick** event in the button.
- Events are used in combination with functions.
- Go to http://www.w3schools.com/jsref/jsref_events.asp for a list of Javascript events.

Examples of Events

Event	*Event Description*
onload, onUnload	The onload and onUnload events are triggered when the user enters or leaves the page.
onFocus, onBlur, onChange	The onFocus, onBlur and onChange events are often used in combination with validation of form fields.
onSubmit	The onSubmit event is used to validate ALL form fields before submitting it.
OnMouseOver, on MouseOut	onMouseOver and onMouseOut are often used to create "animated" buttons.

<u>What are some ways that I can use get input from the user?</u>

We can use these tree types of dialog box to get started with user interaction.

Alert Box	*Confirm Box*	*Prompt Box*
alert("sometext")	confirm("sometext") *confirm() function returns either a true or a false	prompt("sometext","defaultvalue") *prompt() function returns the String

Source: Introduction to Javascript, http://www.w3schools.com/js/js_intro.asp