The Basics of JavaScript

How to Add JavaScript in XHTML

<html> <head>
<script type = ”text/javascript” > JavaScript Code </script> => Using this, JavaScript is directyly Embedded in XHTML
<script type = ”text/javascript” src=”source.js”> </script> => Using this, JavaScript is indirectyly Embedded in XHTML
</head></html>

Comment Lines

//single-line Comment,   /* Multi-line Comment */,   <!-- JavaScript/XHTML Style Comment -->

Primitive Data Types

Number, String, Boolean, Undefined, Null

General Syntax of Variables/Declaring Variables

General Syntax for Variable Declaration: var var_name;
Examples: var count = 100; var str = ”hello”; var pi = 3.142; var flag = true;
Variables in JS are basically objects that supports various built-in properties & methods.

Input(Keyboard/Document) & Output(Screen/Document) Methods

I/P Methods: window.prompt(), window.confirm(), document.getElementById()
O/P Methods: window.alert(), document.write(), document.writeln()

Operators with Precedence & Associativity

Operators	Associativity	Operators	Associativity	Operators	Associativity
++, --, unary -	Right	>, <, >=, <=	Left	&&	Left
*, /, %	Left	==, !=	Left	||	Left
+, -	Left	===, !==	Left	=, +=, -=, *=, /=, %=, &&=, ||=	Right
Note: Highest precendence operators are listed first. Special operator: typeof()					

Data Type Conversion Methods

Implicit: “August ” + 2012 => “August 2012”, 7 * “3” => 21
Explicit: String(value), var_name.toString(), var_name.toString(2), var_name.toString(8), var_name.toString(16), parseInt(), parseFloat()

<table>
<tr><td colspan="2">Control Statements</td></tr>
<tr><td>Selection Statements:
if(expression)
{
single or compound statement(s);
}
else
{
single or compound statement(s);
}</td><td>Switch Statement:
switch(expression)
{
case value_1: statement(s); break;
case value_2: statement(s); break;
case value_3: statement(s); break;
case value_n: statement(s); break;
default: statement(s);
}</td></tr>
<tr><td>Loop statements:
while(control expression)
{
statement or compound statement(s);
}

do
{
statement or compound statement(s);
} while(control expression);</td><td>Loop Statements:
for(initial expression; control expression; increment expression)
{
statement or compound statement(s);
}

for(identifier in object)
{
statement or compound statement(s);
}</td></tr>
<tr><td colspan="2">Built-in Objects, Their properties & Methods</td></tr>
<tr><td>Object Name: Math
Methods:
floor(), round(), sqrt(), pow(), max(), min(), sin(),</td><td>Object Name: Number
Properties:
MAX_VALUE, MIN_VALUE, NaN, POSITIVE_INFINITY, NEGATIVE_INFINITY, PI</td></tr>
<tr><td>Object Name: String
property: length
Methods: charAt(), indexOf(), substring(), toLowerCase(), toUpperCase()</td><td>Object Name: Date
Methods:
toLocaleString(), getDate(), getMonth(), getDay(), getFullYear(), getTime(), getHours(), getMinutes(), getSeconds(), getMilliseconds()</td></tr>
<tr><td>Object Name: Array
Property: length
Methods: join(), sort(), concat(), slice(), push(), pop(), shift(), unshift()</td><td>Object Name: window
Methods: confirm(), prompt(), alert()
Object Name: document
Methods: getElementById(), write(), writeln()</td></tr>
</table>

User Object Creation & Modification

General Syntax to Create Object: var obj_name = new Object();
Example to create object & Assign properties to Object:
var my_car = new Object(); my_car.make = “Ford”;   my_car.model = “ikon”; my_car.color = “Silver”; my_car.year = 2012;
Syntax to delete Object Properties: delete obj_name.property_name; Example: delete my_car.model; delete my_car.make;

Functions

```
<html>
<head>
<title> JS Function Example </title>

<script type="text/javascript">

document.write("<h1> Function With - No Parameter </h1>");

function display()
{
document.write("<p>The function example with no parameters</p>");
}

ref_fun = display;

display(); //call to function display

ref_fun(); //call to same function display using reference name

</script>
</head> <body> </body> </html>
```

Output of Above Function Example:

```
Function With – No Parameter
The function example with no parameters
The function example with no parameters
```

```
<html> <head> <title> JS Function Example </title>
<script type="text/javascript">

document.write("<h1> Function With - Parameters </h1>");
document.write("<h1>Method - Call By Value </h1>");

var n1 = parseInt(prompt("Enter First Number: "));
var n2 = parseInt(prompt("Enter Second Number: "));

//n1,n2 - are called actual parameters
var sum = addition(n1,n2);

document.write("<p>The Sum of "+n1+" & "+n2+" = "+sum+"</p>");

//num1,num2 - are called formal parameters
function addition(num1,num2)
{
  return(num1+num2);
}

</script>
</head> <body> </body> </html>
```

Output of Above Function Example:

```
Function With - Parameters
Method - Call By Value
The Sum of 25 & 75 = 100
```

Functions

```
<html>
<head>
<title> JS Function Example </title>
<script type="text/javascript">

document.write("<h1> Function With - Array Parameters </h1>");
document.write("<h1>Method - Call By Reference </h1>");

var digits = new Array(1, "two", 3, "four", 5);

display(digits);

function display(new_digits)
{
   document.write("<h2> Array Elements Are:</h2>");
   for(var i=0; i<new_digits.length; i++)
   {
       document.write("<h2>" + new_digits[i] + "</h2>");
   }
}

</script>
</head> <body> </body> </html>
```

Output of above Function Example:

```
Function With - Array Parameters
Method - Call By Reference
Array Elements Are:
1
two
3
four
5
```

```
<html><head><title> JS Function Example </title>
<script type="text/javascript">

document.write("<h1> Function With - Parameters </h1>");
document.write("<h1> arguments[]- Array to store parameters</h1>");

var sum = addition(11,22,33);

document.write("<p>The Sum of Argument Values = "+sum+ "</p>");

function addition()
{
   var add=0;
   var no_argu = arguments.length;
   document.write("<p>Number of Arguments = " + no_argu +"</p>");
   for(var i=0; i<arguments.length; i++)
   {
       document.write("<p>Argument["+i+"] = "+arguments[i]+"</p>");
       add = add + arguments[i];
   }
   return(add);
}

</script> </head> <body> </body> </html>
```

Output of above Function Example:

```
Function With - Parameters
arguments[]- Array to store parameters
Number of Arguments  = 3
Argument[0] = 11
Argument[1] = 22
Argument[2] = 33
The Sum of Argument Values = 66
```

Constructors

```
/* JavaScript Constructors are special methods that create and initialize the properties & methods for newly created user objects */
<html> <head> <title> JS Constructor Example </title>
<script type="text/javascript">
document.write("<h1> My Car Details </h1>");

function car(new_make, new_model, new_color, new_year)
{
   this.make = new_make;
   this.model = new_model;
   this.color = new_color;
   this.year = new_year;
   this.display = display_car;
}

function display_car()
{
  document.write("<p> Make:" + this.make + "</p>");
  document.write("<p> Model:" + this.model + "</p>");
  document.write("<p> Color:" + this.color + "</p>");
  document.write("<p> Year:" + this.year + "</p>");
}

var my_car = new car("Ford","Ikon","Silver","2012");
my_car.display();

</script> </head> <body> </body> </html>
```

Output Of Above Constructor Example:

```
My Car Details
Make: Ford
Model: Ikon
Color: Silver
Year: 2012
```