4. JavaScript Functions

We will discuss how to create our own user-defined functions and use system predefined functions. Reusable program modules are normally called functions in JavaScript. JavaScript provides a rich collection of functions for performing the following tasks:

- Mathematical computations
- String manipulations
- String and number conversions: parseInt(), parseFloat()
- Date and Time manipulations
- Array data manipulation

Some commonly used predefined functions such as Math.random(), Math.pow() are also called methods of Math object.

User Defined Functions

If you use only a small number of functions, you should consider to place your functions inside the <script> tags or elements and within the <head>.

- Creating a function must consider
    - Function constructor
    - Input Arguments
    - Return value
- Function calling can be
    - Call by value
    - Call by reference
    - Recursive function - a function calls itself
- The format of a function definition is:
  function function_name(parameter_list)
  {

// Variable declaration;

// Statements;

return | return expression; // | means OR

}

- Scope Rules
  - Variables defined inside a function have a function or local scope
  - Variables defined outside a function have global scope
- Examples of function definition:

```
function multiplyf(x, y)  { return x*y;  }
function square(x)  { return x*x;  }
// Using Function Object
var multiply = new Function("x", "y", "return x* y");
function add(x, y)  { return x + y;  }
function sub(x, y)  { return x - y;  }
function mul(x, y)  { return x * y;  }
function div(x, y)   { return x / y;  }
```

Recursive Function Example

```
function factorial (num)
        {
        // num! = num * (num-1) !
        // 5! = 5 * 4 * 3 * 2 * 1
        if (num <= 1) return 1;
        else
        return num * factorial(num -1);
       }
```

How to Invoke JavaScript Functions

After you define JavaScript Functions, you can call or invoke functions through the following approaches:

- Call a function within a function
- Call a function using event handlers such as onClick(), and onLoad()

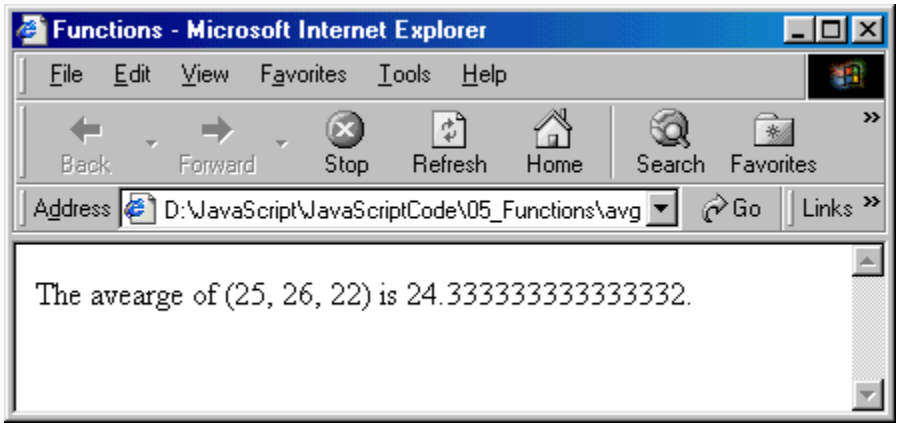
Anchor and Link

- <a> anchor or link can be used with event handler to invoke JavaScript functions

```
< a href ="javascript:void(0)" onClick="someF1()">HERE</a> An example of onClick event handler.<br>

< a href ="javascript:void(0)" onDblClick="fsomeF2()">HERE</a> An example of DblClick event handler.<br>

< a href ="javascript:void(0)"
        onMouseOver="return someF3( )"
        onMouseDown="return someF4( )"
        onMouseUp="return someF5( )"
        onMouseOut="return someF6( )"
>
<img name="ImgName" src="somePic.gif" height=16 width=16 border=0></a>
```

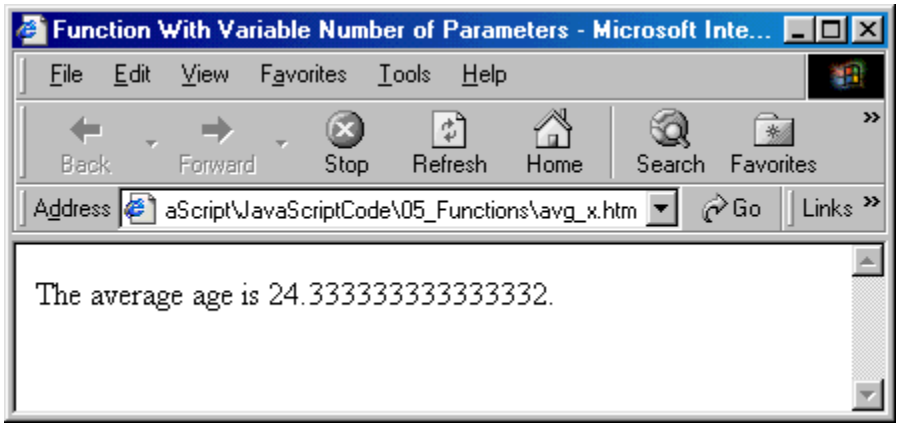
Example 4-1: An average function that accepts three numbers, calculate the average of three numbers and return the computed average value.

```
<html>
<head>
<title>Functions</title>
<script language="JavaScript">
<!--
function averageAge(age1, age2, age3)
  {
   var totalAge;

   totalAge = age1 + age2 + age3;
  return (totalAge / 3);
  }
-->
</script>
</head>
<body>
<script language="JavaScript">
<!--
  var result;      // define a variable
                  // Call averageAge() function and receive
                  // a return value for "result" variable
  result = averageAge(25, 26, 22);

  document.write("The avearge of (25, 26, 22) is " + result +
                 "." + "<br>");
-->
</script>
</body>
</html>
```



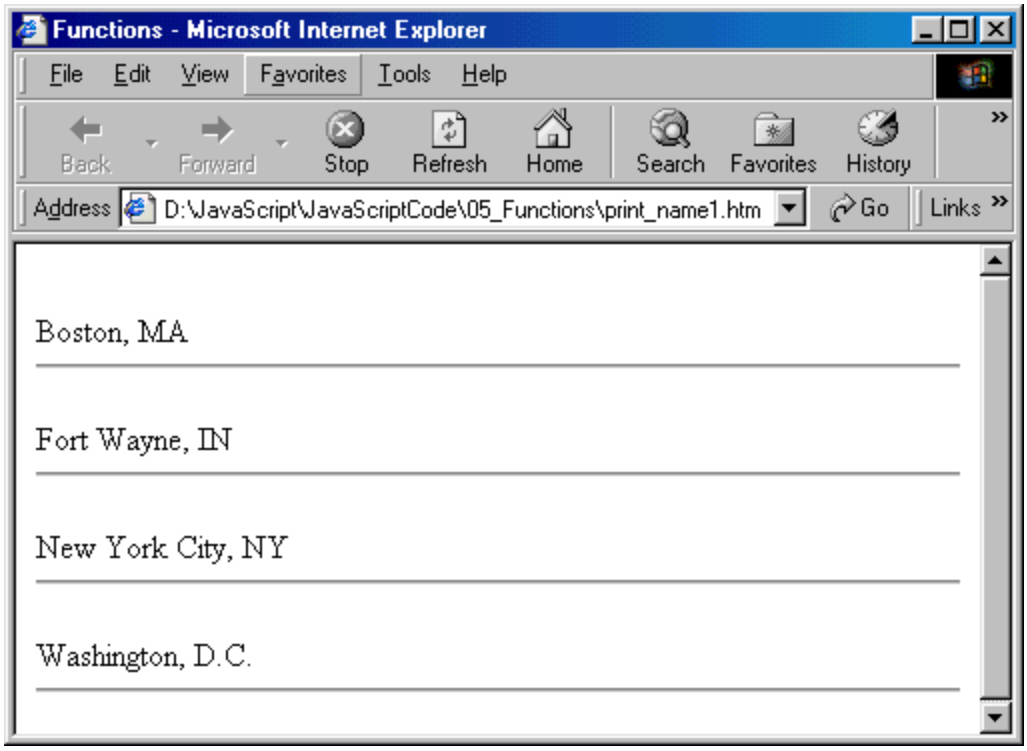
Example 4-2: An average function that can accept N different numbers, calculate the average of the N numbers, and return the computed average value.

```
<html>
<!-- avg_x.html -->
<head>
<title>Function With Variable Number of Parameters</title>
<script language="JavaScript">
<!--
function averageAge(ages)
  {
   var totalAge = 0;
   var thisArguments = averageAge.arguments;

   for ( var i = 0; i < thisArguments.length; i++)
       totalAge += averageAge.arguments[i];
   return (totalAge / thisArguments.length);
  }
-->
</script>
</head>
<body>
<script language="JavaScript">
<!--
document.write("The average age is " + averageAge(22, 25, 26) , ".");
-->
</script>
</body>
</html>
```



Example 4-3: Create a screen print function for reuse.

```
<html>
<!-- print_name1.html -->
<head>
<title>Functions</title>
<script  language="JavaScript">
<!--
function printCityName(Name)
  {
    document.write("<br>", Name, "<br>");
    document.write("<hr>");
  }
-->
</script>
</head>
<body>
<script language="JavaScript">
<!--
printCityName("Boston, MA")
printCityName("Fort Wayne, IN")
printCityName("New York City, NY")
printCityName("Washington, D.C.");
-->
</script>
</body>
</html>
```



Example 4-4: Create a ChangeImage() to load a new graphics file upon the mouse click on the displayed image.

```
<html>
<!-- bulbtest.htm -->
<head>
<base href="./">
<title>Changing and Image Source Example</title>
<script language="JavaScript">
function ChangeImage(){
      MyImage.src="http://www.etcs.pfw.edu/~lin/Presentation/JavaScript
/Chapter4/bulbon.GIF"
}
</script>
</head>
<body>
<img id="MyImage"
src="http://www.etcs.pfw.edu/~lin/Presentation/JavaScript/Chapter4/bulb
off.GIF" align="center" OnClick="ChangeImage()"> <b>Click Me!</b>

</body>
</html>
```

JavaScript Math Object

Math object

Synopsis

Math.constants
Math.function()

Constants

Math.E

The mathematical constant e (Euler's constant).
e = 2.718281828459045091

Math.LN10

The mathematical constant $\log_e 10$ (natural log of 10).
$\log_e 10$ = 2.302855092994045901

Math.LN2

The mathematical constant $\log_e 2$ (natural log of 2).
$\log_e 2$ = 0.6931471805599452862

Math.LOG10E

The mathematical constant $\log_{10} e$ (log base-10 of e).
$\log_{10} e$ = 0.4342944819032518167

Math.LOG2E

The mathematical constant $\log_2 e$ (log base-2 of e).
$\log_2 e$ = 1.442695040888963387

Math.PI

The mathematical constant π.

Math.SQRT1_2

The mathematical constant$1/\sqrt{2}$.
$1/\sqrt{2}$ = 0.7071067811865475727

Math.SQRT2

The mathematical constant$\sqrt{2}$.
$\sqrt{2}$ = 1.414213562373095145

Methods (Predefined Functions)

Math.abs(x)
: Compute |x|.

Math.acos(xRadians)
: Compute an arc cosine.

Math.asin(xRadians)
: Compute an arc sine.

Math.atan(xRadians)
: Compute an arc tangent.

Math.atan2(xRadians,yRadians)
: Compute the angle from the X axis to a point.

Math.ceil(x)
: Round up a number.

Math.cos(xRadians)
: Compute a cosine.

Math.exp(x)
: Compute e^x.

Math.floor(x)
: Round down a number.

Math.log(x)
: Compute a natural logarithm.

Math.max(a,b)
: Evaluate the larger of two values.

Math.min(a,b)
: Evaluate the smaller of two values.

Math.pow(x,y)
: Compute the x^y.

Math.random()
: Generate a pseudo-random number between 0 and 1.

Math.round(x)
Round to the nearest integer.

Math.sin(xRadians)
Compute a sine.

Math.sqrt(x)
Compute a square root.

Math.tan(xRadians)
Compute a tangent.

Math Object Application Examples

Simple math:

```
Circumference = diameter * Math.PI;
```

Card Game: 52 cards:
Generate an integer between 1 and 52.

```
Math.round(Math.random()*52)
```

Generate random numbers between a different range n lowest, and m highest:

```
Math.round(Math.random() * n) + m
```

rollDice: 1, 2, 3, 4, 5, 6:

```
rollDice = Math.round(Math.radom() * 5) + 1
```