What is Functions

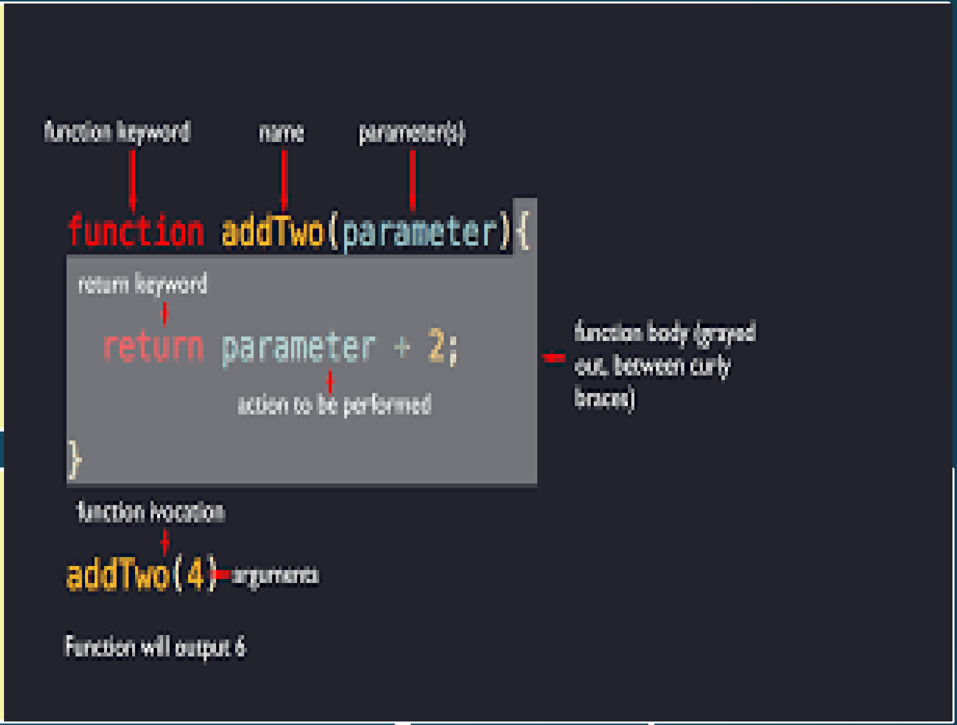
- A group of statements that is put together (or defined) once and then can be used (by reference) repeatedly on a Web page
- Also known as subprogram, procedure, subroutine
- A function is written as a code block (inside curly { } braces), preceded by the **function** keyword
- Syntax of JavaScript function

  – function *functionname*()
  {
  some code to be executed
  }

JavaScript functions

```
function function_name()
{
statement ;
statement ;
...
statement ;
}
```

```
function myFunction() {
    alert("Hello!");
    alert("How are you?");
}
```



```
function addTwo(parameter){
  return parameter + 2;
}
addTwo(4)
```

- the above could be the contents of example.js linked to our HTML page
- statements placed into functions can be evaluated in response to user events

Advantages of Functions

- Number of lines of code is reduced
- Code becomes easier to read & understand
- Code becomes easier to maintain as changes need to be made only at a single location instead multiple locations

Example

-
  ```
  <html>
  <head>
  <title>Assignment-2</title>
  </head>
  <script>
  function date()
  {
      var d=new Date();
      document.write(d);
  }
  </script>
  <body>
  <p>Press the button to show current date & time... </p>
  <input type="button" onclick="date()" value="Show date">
  </body>
      </html>
  ```

Function Defnitions

- Format of a function definition
- function *function-name*(*parameter-list*)

{

declarations and statements

}

  - Function name any valid identifier
  - Parameter list names of variables that will receive arguments
    - Must have same number as function call
    - May be empty
  - Declarations and statements
    - Function body ("block" of code)
- Return statement
  - Optional , can return either nothing, or a value

Function Parameters

- Facility to pass different parameters while calling a function
- Passed parameters can be captured inside the function and any manipulation can be done over those parameters
- A function can take multiple parameters separated by comma.

Example

-
  ```
  <html>
  <head>
  <script type="text/javascript">
  function say(name, age)
  {
     alert( "I am" + name + “and I am “+ age + " years old.");
  }
  </script>
  </head>

  <body>
  <p>Click the following button to Know about my self</p>
  <form>
  <input type="button" onclick="say(‘John’, 2)" value="MyIntro">
  </form>

  </body>
  </html>
  ```

Local and Global variable

- A variable declared (using var) within a JavaScript function becomes **LOCAL** and can only be accessed from within that function

Arrays in JavaScript

```
var name = []; // empty array
var name = [value, value, ..., value]; // pre-filled
name[index] = value; // store element
```

```
var quote= new Array(5)
quote[0]="I like JavaScript.";
quote[1]="I used to like Java.";
quote[2]="JavaScript rules.";
quote[3]="Help! JavaScript Error!";
quote[4]="Just Kidding.";
```

```
<HEAD>
<SCRIPT language="JavaScript">
function display_quote()
{
        var quote= new Array(5)
        quote[0]="I like JavaScript.";
        quote[1]="I used to like Java.";
        quote[2]="JavaScript rules.";
        quote[3]="Help! JavaScript Error!";
        quote[4]="Just Kidding.";
        var x=0;
        for (x=0; x<5; x++)
        {
                alert(quote[x]);
        }
}
</SCRIPT>
</HEAD>
<BODY>
<A HREF="javascript:display_quote()">Click Here</A>
</BODY>
```

Thank You