JavaScript – Tutorial

Web Designing Lab(18XW28)

MSc(SS) – Semester II

What is Script?

A script a set of instructions for calls to follow when they arrive in the system

What is Scripting Language?

A Scripting Language is a computer language with a series of commands within a file that is capable of being executed without being compiled (often interpreted)

What are the types of Scripting Language?

Client Side

Running the scripts in the client system

(e.g) JavaScript, AngularJS, JQuery, React.js etc

Server Side

Running the scripts in the server system

(e.g) PHP, Python, Node.js, Perl etc

History of JavaScript..

- Created by Brendan Eich at Netscape in December 1995
- Initially called *LiveScript*
- Officially called *ECMAScript*
- Microsoft Version is called *JScript*
- Latest version *ECMAScript 2018 (JavaScript 1.9)*

What is JavaScript?

- Client-side object-based scripting language
- Light-weight, interpreted programming language
- Embedded within the html of a document
- Allows for “preprocessing” of forms and can add “dynamic content” to a web page
- Highly case-sensitive

How to include JavaScript in html?

- JavaScript consists of statements that are placed within the *<script>…</script>* tag in .html files
- These can appear either in *<head>* or *<body>* section of a html document
- *<head>* - Functions and code that may execute multiple times (preferred)
- *<body>* - Code that needs to be executed only once

```
<script language="javascript" type="text/javascript">
          JavaScript code
</script>


<script language="javascript" type="text/javascript"
                    src="test.js" ></script>
```

```
<HTML>
<HEAD>
<TITLE>First JavaScript Example</TITLE>
</HEAD>
<BODY>
<H2>This line is straight HTML</H2>
<H3>
<SCRIPT type = "text/javascript">
    document.write("These lines are produced by<br/>");
    document.write("the JavaScript program<br/>");
    alert("Hey, JavaScript is fun!");
</SCRIPT>
</H3>
<H2>More straight HTML</H2>
<SCRIPT type = "text/javascript"
src="bogus.js"></script>
</BODY>
</HTML>
```

JavaScript Variables

- JavaScript variables have no types – *determined dynamically,* based on value stored (*typeof* operator used to check)
- Declarations are made using the *var* keyword
- Declaration outside of any function are *global*
- Declaration inside of any function are *local*

- Variables declared but not initialized have the value ***undefined***
- Variable identifiers are similar to those in other languages (ex: Java)
  - *Cannot use a keyword*
  - *Must begin with a letter, $, or _*
  - *Followed by any sequence of letters, $, _ or digits*
  - *Case sensitive*

In JavaScript variables are created using the keyword *var*

Examples:

```
var x = 10;

var y = 17;

var color = “red”;

var name = “Katie”;
```

JavaScript Data Types

- Primitive Data Types:
  - Numbers – can be integer or decimal
  - Strings - sequence of letters or numbers enclosed in single or double quotes
  - Boolean (True, False) – true or false

- Composite Data Types:
  - Arrays
  - Objects

- JavaScript is *untyped*; It does not have explicit data types
- The same variable can have different data types in different contexts
- If you have an expression which combines two numbers, it will evaluate to a number
- If you have an expression which combines a string and a number, it will evaluate to a string

```
var x = 4;

var y = 11;

var z = “cat”;

var q = “17”;
```

```
Ans = x + y;
   Ans => 15

Ans = z + x;
   Ans => cat4

Ans = x + q;
   Ans => 417
```

What is JavaScript Statements?

- A statement is a section of JavaScript that can be evaluated by a Web browser
- A script is simply a collection of statements

Examples:

```
Last_name = “Dunn”;
x = 10 ;
y = x*x ;
```

JavaScript Operators

- Arithmetic: +, -, *, /, %, ++, --
- Comparison: ==, !=, >, <, >=, <=
- Logical: &&, ||, !
- Assignment: =, +=, -=, *=, /=, %=
- Bitwise: &, |, ^, ~, <<, >>, >>>
- Conditional: ?:

JavaScript – Control Structures

- There are three basic types of control structures in JavaScript: *selection*, *loops* and *jump*
- Each control structure manipulates a block of JavaScript expressions beginning with { and ending with }

- Selection
  - *if, if..else, if..else if..*
  - *switch*
- Loops
  - *while*
  - *do..while*
  - *for*
  - *for..in*
- Jump
  - *break;*
  - *continue;*

```
if ( x  = =  10)
{          y  =  x*x;
}
else
{          x  =  0;
}
```

```
count = 0;
while (count <= 10) {
    document.write(count);
    count++;
}
```

JavaScript Fuctions

- Functions are a collection of JavaScript statement that performs a specified task
- Functions are used whenever it is necessary to repeat an operation
- Functions are declared by a name and invoked by the same name
- It has four parts
  - function keyword
  - function name
  - comma separated list of arguments
  - statements enclosed within curly braces

- Syntax

```
function functionname(parameters-list)
{
   statements;
}
```

- The function should be invoked for execution by using the syntax

```
functionname(parameters);
```

```
function square(x)
{
  return x*x;
}

z = 3;
sqr_z = square(z);
```

Name of Function:
square

Input/Argument: x

Output: x*x

JavaScript Objects

- Objects
- Array
- String
- Date
- Math
- RegExp

JavaScript - Array

- An array is a compound data type that stores numbered pieces of data
- Each numbered datum is called an *element* of the array and the number assigned to it is called an *index*.
- The elements of an array may be of any type, single array can even store elements of different type.

Creating an Array

- There are several different ways to create an array in JavaScript
- Using the Array() constructor:

```
var a = new Array(1, 2, 3, 4, 5);
var b = new Array(10);
```

- Using array literals:

```
var c = [1, 2, 3, 4, 5];
var c = ["we", "can", 50, "mix", 3.5, "types"];
```

Accessing Array elements

- Array elements are accessed using the [ ] operator
- Example:

```
var colors = [“red”, “green”, “blue”];
colors[0] => red
colors[1] => green
```

Adding Elements into an Array

- To add a new element to an array, simply assign a value to it
- Example:

```
var a = new Array(10);
a[50] = 17;
```

- JavaScript also has 2-Dimensional arrays

Array Methods

- concat two arrays into one
- join array items into a single string (commas between)
- push & pop appends and removes the element at the end -"right stack"
- shift, unshift appends and removes the element at the beginning - "left stack"
- sort sorts the value in the array
- reverse reverses the items in an array
- slice returns the subset of the array
- splice adds/removes the items and return

What are the popup boxes supported in JavaScript?

alert box

- Allows to alert the user about some action or result on the web page
- A small window that has “OK” button and displays a short textual message

```
alert(“message”);
```

confirm box

- To verify the decision of a user to perform a given action (or) task
- Display a message with “OK” and “Cancel” button

```
confirm(“Are you want to proceed?”);
```

Example - confirm box

```
<script>
function checkPassword( )
{
        if(myForm.txtPassword.value==“”)
                alert(“Pls. Enter password!!”);
        else
                confirm(“Are you want to proceed?”);
}
</script>
```

prompt box

- Allows to prompt the user of a web page to enter a string/textual information
- Displays a message with “OK” and “Cancel” button

```
prompt(“message”,”value”);
```

Example - prompt box

```
<script>
function checkPassword( )
{
        var identity=prompt(“Enter your name”,””);
        alert(identity);
}
</script>
```

What are events?

An ***event*** is something that happens, especially when it is unusual or important. You can use ***events*** to describe all the things that are happening in a particular situation.

What is events in JavaScript?

JavaScript's interaction with HTML is handled through ***events*** that occur when the user or the browser manipulates a page. When the page loads, it is called an ***event***. When the user clicks a button, that click too is an ***event***.

Common JavaScript Events

- onclick
- onchange
- onfocus
- onabort
- onblur
- onload
- onunload
- onkeydown
- onkeypress
- onkeyup
- onmouseover
- onmouseup
- onselect
- onsubmit

Example – JavaScript event

```
<!doctype html>
<html>
  <head>
    <script>
      function hiThere() {
        alert('Hi there!!');
      }
    </script>
  </head>
  <body>
    <button type="button" onclick="hiThere()">Click
   me !!!</button>
  </body>
</html>
```

What is an error?

An ***error*** is an action which is inaccurate or incorrect

What are the types of error?

Syntax Error: Also called *parsing errors*, occur at interpret time

```
<script type = "text/javascript">
        window.printme(;
</script>
```

Logical Error: most difficult error to be traced as it is the error on the logical part of the coding in a program generate unexpected output

Runtime Error: an error that occurs during the running of the program, also known as the *exceptions*

```
<script type = "text/javascript">
        window.printme();
</script>
```

What do you mean by exception?

An ***exception*** is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions

What do you mean by exception handling?

A ***process*** of responding to exceptions when a computer program runs. It attempts to gracefully handle these situations so that a program (or worse, an entire system) does not crash.

How you handle exceptions in JavaScript?

```
try {
  // attempt to execute this code
} catch (error-object) {
  // this code handles exceptions
} finally {
  // this code always gets executed
}
```

try - test a block of code for errors

catch – handle the error

throw – create custom error

finally – execute code, after try and catch, regardless of the result

Example 1 – JavaScript Exceptions

```
<html>
  <head>
     <script type = "text/javascript">
         function myFunc() {
           var a = 100;
           try {
              alert("Value of variable a is : " + a );
           }
           catch ( e ) {
              alert("Error: " + e.message );
           }
         }
     </script>
   </head>
```

```
<body>
   <p>Click the following to see the result:</p>
   <form>
      <input type = "button" value = "Click Me"
onclick = "myFunc();" />
   </form>
  </body>
</html>
```

Example 2 – JavaScript Exceptions

```
<html>
  <head>
     <script type = "text/javascript">
         function myFunc() {
           var a = 100;
           try {
              alert("Value of variable a is : " + a );
           }
           catch ( e ) {
              alert("Error: " + e.message );
           }
            finally {
              alert("Finally block!!");
           }
         }
     </script>
   </head>
```

```
<body>
   <p>Click the following to see the result:</p>
   <form>
      <input type = "button" value = "Click Me"
onclick = "myFunc();" />
   </form>
 </body>
</html>
```

Example 3 – JavaScript Exceptions

```
<html>
  <head>
     <script type = "text/javascript">
         function myFunc() {
           var a = 100;
           var b = 0;
           try {
              if ( b == 0 ) {
                 throw( "Divide by zero error." );
              } else {
                 var c = a / b;
              }
           }
           catch ( e ) {
              alert("Error: " + e );
           }
         }
     </script>
   </head>
```

```
<body>
   <p>Click the following to see the result:</p>
   <form>
      <input type = "button" value = "Click Me"
onclick = "myFunc();" />
   </form>
 </body>
</html>
```

What are the type of Error in JavaScript?

- ***SyntaxError:*** It represents a syntax error.
- ***RangeError:*** It represents an error in the range.
- ***ReferenceError:*** It represents an illegal reference.
- ***TypeError:*** It represents a type error.
- ***EvalError:*** It represents an error in the eval() function.
- ***URIError:*** It represents an error in the encodeURI().

Example – SyntaxError

```
<!DOCTYPE html>
<html>
<body>
    <h3>
      JavaScript Error Name Property
    </h3>
    <p id="gfg"> </p>
      <script>
          try {
               eval("alert('Geeks for Geeks’)");
          }
          catch (err) {
               document.getElementById("gfg").innerHTML = err.name;
          }
      </script>
</body>
</html>
```

References

- https://www.geeksforgeeks.org/javascript-tutorial/
- http://www.htmldog.com/guides/javascript/
- https://www.tutorialspoint.com/javascript/index.htm
- https://www.javatpoint.com/javascript-tutorial