

Chapter 6

CLIENT SIDE SCRIPTING USING JAVASCRIPT

Java Script - is a client side scripting language used to validate data. It is embedded in HTML document. JavaScript was developed by Brendan Eich. It is supported by almost all web browsers.

<SCRIPT> Tag - used to include(embed) script in an HTML page. The Language attribute of <SCRIPT> tag specifies the type of scripting language used.

Example:

```
<SCRIPT Language="JavaScript">
</SCRIPT>
```

<SCRIPT> Tag ഉപയോഗിക്കുന്നു. ഇതിലെ Language attribute ഏതുതരം ഭാഷയാണുപയോഗിക്കുന്നതെന്ന് വ്യക്തമാക്കുന്നു..

Example:

```
<SCRIPT Language="JavaScript">
</SCRIPT>
```

Data types in JavaScript

Number:-They include integers and floating point numbers. എല്ലാവിധത്തിലുമുള്ള സംഖ്യകളെ സൂചിപ്പിക്കുന്നു

Strings:-A string is a combination of characters, numbers or symbols enclosed within double quotes. double quotes നുള്ളിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള എല്ലാ ക്യാരക്റ്ററുകളും

Boolean:-A boolean data can be either True or False.

Variables in JavaScript

A variable is a data type whose value changes during program execution. Variables in JavaScript are declared using the var keyword. Javascript determines the type of variable only when assigning a value.

പ്രോഗ്രാമുകളിൽ മൂല്യത്തിന് മാറ്റം വരുന്നതാണ് വേരിയബിൾ. var കീവേർഡ് ഉപയോഗിച്ചാണ് വേരിയബിളുകളെ ഡിക്ലെയർ ചെയ്യുന്നത്. വേരിയബിളുകൾക്ക് മൂല്യം നൽകുമ്പോൾ മാത്രമാണ് അവയുടെ ഡേറ്റാ ടൈപ്പ് തീരുമാനിക്കപ്പെടുന്നത്.

Example:

```
var a; Declares a variable a.
var msg="Hello";
```

Operators in JavaScript

Operators in C++ are used in JavaScript with the same use. C++ൽ ഉപയോഗിക്കുന്ന ഓപ്പറേറ്ററുകൾ അതുപോലെതന്നെ ജാവാസ്ക്രിപ്റ്റിലും ഉപയോഗിക്കുന്നു.

Arithmetic Operator is used to perform arithmetic operations. (+, -, *, /, %)

Relational Operator is used for comparison operations. (<, <=, >, >=, ==, !=)

Assignment Operator (=)

Arithmetic Assignment Operator (+=, -=, *=, /=, %=) are used to simplify the use of assignment operator.

Logical Operator - (Logical and &&, Logical or ||, logical not !)

Operator Precedence

() Expression grouping

++ -- Increment and decrement

* / % Multiplication, division, and modulo division

+ - Addition and subtraction

typeof Operator is used to find the type of a javascript variable. It returns a string indicating the data type of its operand. ഡേറ്റാ ടൈപ്പ് എന്താണെന്ന് മനസ്സിലാക്കുവാൻ typeof ഓപ്പറേറ്റർ ഉപയോഗിക്കാം

Example:-

```
typeof"Hello" //returns String
```

```
typeof 3.14 //returns Number
```

```
typeof false //returns Boolean
```

```
var p;
```

```
typeof p //returns Undefined (മൂല്യം നൽകിയിട്ടില്ല).
```

String addition (+) operator can be used to add(join) two strings. രണ്ട് string കളെ യോജിപ്പിക്കുവാൻ String addition + ഉപയോഗിക്കുന്നു.

Example:-

```
var a,b,c
a="Hello";
b="How are you";
c=a+b;
document.write(c);
```

Output:-

Hello How are you

Number function in JavaScript

The Number() function is used to convert a string data into a number.

Example:

Number("42");//returns 42

Number("eggs");//returns NaN String that can't be converted to number returns NaN.

Number(true);//returns 1

Number(false);//returns 0

Note:-8

NaN stands for Not a Number.

Control structures in JavaScript - are used to alter the normal sequence of execution of a program. പ്രോഗ്രാമിന്റെ തുടർച്ചയായുള്ള നീക്കത്തിന് മാറ്റമുണ്ടാക്കുന്നതാണ് Control structures. C++ൽ ഉപയോഗിക്കുന്ന Control structures തന്നെയാണ് ഇവിടെയും ഉപയോഗിക്കുന്നത്.

1) **if** statement

```
if(test_expression)
```

```
{
```

```
Statements;
```

```
}
```

```
else
```

```
{
```

```
Statements;
```

```
}
```

2) **switch** statement

```
switch(expression)
```

```
{
```

```
case value1:statement1;break;
```

```
case value2:statement2;break;
```

```
-
```

```
-----
```

```
Default:statement;
```

```
}
```

3) **for** Loop

```
for(initialisation:expression:update_statement)
```

```
{
```

```
statements;
```

```
}
```

4) **while**

```
while(expression)
```

```
{
```

```
statements;
```

```
}
```

[C++ൽ ഉപയോഗിക്കുന്ന do-while ജാവാസ്ക്രിപ്റ്റിൽ ഉപയോഗിക്കുന്നില്ല]

Writing JavaScript programs

The **document.write** () function

The function document.write which writes a string into our HTML document.

HTMLdocument ൽ string output ചെയ്യാൻ document.write () ഉപയോഗിക്കുന്നു.

First JavaScript Program

```
<html>
```

We use Notepad for creating the above script and save the file with extension .HTML or

```
<body>
<script language="javascript" >
document.write("Welcome to JavaScript") ;
</script>
</body>
</html>
```

Creating functions in JavaScript

A function is a self-contained unit of a program that perform a specific task. It is an independent part of a program which is executed when it is called. Functions are usually placed in the HEAD tag of HTML.

Defining a function

Functions must be declared before they are used. A function is defined using the function keyword. Once a function is defined it can be called many times.

```
function print()
{
document.write("Welcome to JavaScript
Programming");
}
```

Note:-A function in JavaScript has no return type.

Calling a function

A function can be called using its name such as print().

Built-in Functions

1) **alert()** function - used to display a message on the screen. The syntax is

```
alert("message");
```

സ്ക്രീനിൽ ഒരു മെസേജ് ബോക്സ് കാണിക്കുവാൻ alert() ഉപയോഗിക്കുന്നു.

2) **isNaN()** function - check if a value is a number or not. The function returns True if the value is a not number. The syntax is

```
isNaN(test_value);
```

ഒരു സ്ട്രിങ് നമ്പറാണോ എന്ന് പരിശോധിക്കുന്നു.

Eg : isNaN(65) ---> False

isNaN("Big") ---> True

3) **toUpperCase()** function - This function converts a string into uppercase.

Example:-

```
var a,b;
a="abcd";
b=a.toUpperCase( );
document.write(b);
```

Output:ABCD

4) **toLowerCase()** function- This function converts a string to lowercase.

5) **charAt()** function - returns the character at the specified index in a string. The index of the first character is 0, the second character is 1, and so on.

ഒരു സ്ട്രിങ്ങിന്റെ പറഞ്ഞിരിക്കുന്ന സ്ഥാനത്തുള്ള ക്യാരക്ടർ ഏതാണെന്ന് അറിയിക്കുന്നു.

Example:-

```
var str = "HELLO WORLD";
var res = str.charAt(0);
returns H
```

.HTML. JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs. Javascript is a case-sensitive language.

നോട്ട്പാഡ് ഉപയോഗിച്ച് പ്രോഗ്രാം തയ്യാറാക്കി .HTML അല്ലെങ്കിൽ .HTM ഫോർമാറ്റിൽ സേവ് ചെയ്ത് സാധാരണ വെബ് പേജ് കാണുന്നതുപോലെ കാണാവുന്നതാണ്.

പ്രത്യേകമായി ഒരുജോലി ചെയ്യാനുള്ള നിർദ്ദേശങ്ങളെ ഫങ്ഷൻ എന്നു വിളിക്കുന്നു. സ്വതന്ത്രമായി നിൽക്കുകയും ആവശ്യമുള്ളപ്പോൾ പ്രവർത്തിക്കുകയും ചെയ്യുന്നവയാണ് ഇത്തരം പ്രോഗ്രാമുകൾ. ഇത് സാധാരണയായി HEAD ലാണ് വയ്ക്കുന്നത്.

ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിനുമുമ്പുതന്നെ തയ്യാറാക്കിവയ്ക്കണം. Function എന്ന വാക്കുപയോഗിച്ചാണ് ഇത് തയ്യാറാക്കുന്നത്. ഒരു തവണ തയ്യാറാക്കിയാൽ എത്രതവണ വേണമെങ്കിലും ഉപയോഗിക്കാം.

```
function print( )
{
document.write("Welcome to JavaScript
Programming");
}
```

ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷന് return type ഇല്ല.

6) **Number()** function - Returns value of a numeric string.

eg. : Number("65") --> 65

length Property

The length property returns the length of a string(number of characters).

Example:-

```
var a="Welcome";
var len=a.length;
document.write(len);
```

Output:-

7

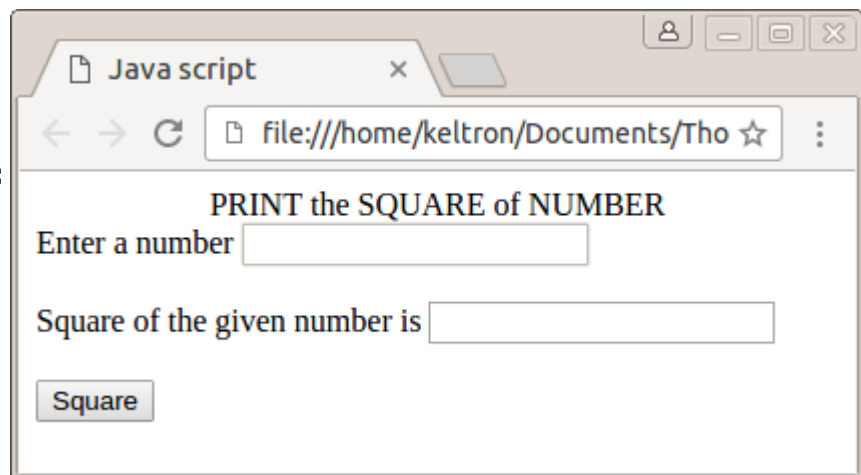
A function can be called with the help of an event. An event is an action that triggers a function call. Commonly used events are

onClick
onMouseEnter
onMouseLeave
onKeyDown
onKeyUp

ജാവാസ്ക്രിപ്റ്റ് ഫങ്ഷനുകളെ വിളിക്കുന്നത് ഇവൻ്റുകളുടെ സഹായത്തോടെയാണു്. കീബോർഡും മൗസും ഇവൻ്റുകൾ സൃഷ്ടിക്കും. പ്രധാനപ്പെട്ട ഇവൻ്റുകൾ
onClick
onMouseEnter
onMouseLeave
onKeyDown
onKeyUp

A HTML code containing script to find the square of a number using a user defined function.

```
<HTML>
<HEAD>
<TITLE> Java script </TITLE>
<SCRIPT language= "JavaScript">
function showsqr()
{
var num1, sqr;
num1 = Number(document.FRM.txtNum1.value);
sqr = num1 * num1;
document.FRM.txtSum.value =sqr;
}
</SCRIPT>
</HEAD>
<BODY>
<FORM name = "FRM">
<CENTER> PRINT the SQUARE
of NUMBER</CENTER>
Enter a number
<INPUT TYPE = "text"
name="txtNum1">
<BR> <BR>
Square of the given
number is
<INPUT TYPE = "text"
name="txtSum" >
<BR> <BR>
<INPUT TYPE = "button" value="Square" onClick="showsqr()">
</BODY>
</HTML>
```



Method to include scripts in a webpage

- inside <HEAD> Section
- inside <BODY> section
- as external file (*.js)