Unit-5: JavaScript Functions:

5.1 JavaScript Functions:

- 5.1.1 Defining function (with and without parameters)
- 5.1.2 calling function
- 5.1.3 return statement
- 5.1.4 Page redirection

5.2 Dialog boxes : Alert, confirm, prompt

5.3 Form validation :

- 5.3.1 Basic validation (All form details are filled)
- 5.3.2 Data format validation

  (email, number, string, mobile number, name)

5.1 JavaScript Functions:

- 5.1.1 Defining function (with and without parameters)
- 5.1.2 calling function
- 5.1.3 return statement
- 5.1.4 Page redirection

5.1.1 Defining function (with and without parameters)

JavaScript functions are **defined** with the function keyword.

You can use a function **declaration** or a function **expression**.

with parameters

Function Declarations

Earlier in this tutorial, you learned that functions are **declared** with the following syntax:

```
function functionName(parameters) {
  // code to be executed
}
```

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are invoked (called upon).

```
<html>
<body>
<p id="demo"></p>
<script>
var x = myFunction(4, 3);
document.getElementById("demo").innerHTML = x;
functionmyFunction(a, b) {
return a * b;
}
</script>
</body>
</html>
```

without parameters

```
function<name>()
{
statements;
output statement;
}
```

function – The keyword function identify the block as a function object.
<name> – is the name of the function given by a user.
() – parenthesis does not contain any argument.
statements; – all JavaScript statements that gets executed by the function.
output statement – it is not necessary that a function that does not return anything must give output, but most of the time you will find a statement that prints the output in these type of functions.

```
//variable declarations
var side = 34;
//function declaration
functionarea_of_square()
{
var area = side * side;
console.log("Area of the Square =" + " " + area);
}
//function call
area_of_square();
```

5.1.2 calling function

The call() method is a predefined JavaScript method.

It can be used to invoke (call) a method with an owner object as an argument (parameter).

With call(), an object can use a method belonging to another object.

This example calls the **fullName** method of person, using it on **person1**:

Example

```
<html>
<body>
</p>
<p id="demo"></p>
<script>
const person = {
fullName: function() {
returnthis.firstName + " " + this.lastName;
  }
}
const person1 = {
```

```
firstName:"John",
lastName: "Doe"
}
const person2 = {
firstName:"Mary",
lastName: "Doe"
}
document.getElementById("demo").innerHTML =
person.fullName.call(person1);
</script>
</body></html>
```

call() Method with Arguments

The call() method can accept arguments:

Example

```
<html>
<body>
<p id="demo"></p>
<script>
const person = {
fullName: function(city, country) {
    return this.firstName + " " + this.lastName + "," + city + "," + country;
  }
}
const person1 = {
firstName:"John",
```

```
lastName: "Doe"
}
const person2 = {
firstName:"Mary",
lastName: "Doe"



}

document.getElementById("demo").innerHTML =
person.fullName.call(person1, "Oslo", "Norway");
</script>
</body>
</html>
```

5.1.3 Return statement

Definition and Usage

The return statement stops the execution of a function and returns a value from that function.

Syntax

return *value*;

Parameter Values

Parameter	Description

value	Optional. Specifies the value to be returned to the function caller. If omitted, it returns **undefined**

Example

```
<html>
<body>
<p id="demo"></p>
<script>
Function myFunction(name) {
return "Hello " + name;
}
document.getElementById("demo").innerHTML = myFunction("John");
</script>
</body></html>
```

Example

```
<html>
<body>
<p>This example calls a function which performs a calculation, and returns the result:</p>
<p id="demo"></p>
```

```
<script>
var x = myFunction(4, 3);
functionmyFunction(a, b) {
return a * b;
}
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

5.1.4 Page redirection

Redirect a Webpage

There are a couple of ways to redirect to another webpage with JavaScript. The most popular ones are location.href and location.replace:

Example

```
// Simulate a mouse click:
window.location.href = "http://www.w3schools.com";

// Simulate an HTTP redirect:
window.location.replace("http://www.w3schools.com");
<html>
<body>
<h2>Redirect to a Webpage</h2>
<p>The replace() method replaces the current document with a new one:</p>
<button onclick="myFunction()">Replace document</button>
```

```
<script>
functionmyFunction() {
location.replace("https://www.w3schools.com")
}
</script>
</body>
</html>
```

5.2 Dialog boxes: Alert, confirm, prompt

Alert Box

An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

window.alert("*sometext*");

The window.alert() method can be written without the window prefix.

Example

```
<html>
<body>
<h2>JavaScript Alert</h2>
<button onclick="myFunction()">Try it</button>
<script>
functionmyFunction() {
alert("I am an alert box!");
}
```

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

Confirm Box

A confirm box is often used if you want the user to verify or accept something.

When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax

window.confirm("*sometext*");

The window.confirm() method can be written without the window prefix.

Example

```
<html>
<body>
<h2>JavaScript Confirm Box</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
functionmyFunction() {
var txt;
if (confirm("Press a button!")) {
txt = "You pressed OK!";
  } else {
txt = "You pressed Cancel!";
```

```
  }
document.getElementById("demo").innerHTML = txt;
}
</script>
</body>
</html>
```

Prompt Box

A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

window.prompt("*sometext*","*defaultText*");

The window.prompt() method can be written without the window prefix.

Example

```
<html>
<body>
<h2>JavaScript Prompt</h2>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
```

```
Function myFunction() {
let text;
let person = prompt("Please enter your name:", "Harry Potter");
if (person == null || person == "") {
text = "User cancelled the prompt.";
  } else {
text = "Hello " + person + "! How are you today?";
  }
document.getElementById("demo").innerHTML = text;
}
</script>
</body>
</html>
```

5.3 Form validation :

5.3.1 Basic validation (All form details are filled)

5.3.2 Data format validation (email, number, string, mobile number, name)

5.3.1 Basic validation (All form details are filled)

```
<!DOCTYPE html>
<html>
<head>
<script>
function validate() {
```

```
if(document.myForm.Name.value == "" ) {
alert( "Please provide your name!" );
document.myForm.Name.focus() ;
return false;
        }
if(document.myForm.EMail.value == "" ) {
alert( "Please provide your Email!" );
document.myForm.EMail.focus() ;
return false;
        }
if(document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
document.myForm.Zip.value.length != 5 ) {

alert( "Please provide a zip in the format #####." );
document.myForm.Zip.focus() ;
return false;
        }
if(document.myForm.Country.value == "-1" ) {
alert( "Please provide your country!" );
return false;
        }
alert("Success ");

return( true );
     }
</script>
```

```
</head>
<body>
<form  name = "myForm" onsubmit = "return(validate());">
<table cellspacing = "2" cellpadding = "2" border = "1">

<tr>
<td align = "right">Name</td>
<td><input type = "text" name = "Name" /></td>
</tr>

<tr>
<td align = "right">EMail</td>
<td><input type = "text" name = "EMail" /></td>
</tr>

<tr>
<td align = "right">Zip Code</td>
<td><input type = "text" name = "Zip" /></td>
</tr>

<tr>
<td align = "right">Country</td>
<td>
<select name = "Country">
<option value = "-1" selected>[choose yours]</option>
<option value = "1">USA</option>
```

```
<option value = "2">UK</option>
<option value = "3">INDIA</option>
</select>
</td>
</tr>

<tr>
<td align = "right"></td>
<td><input type = "submit" value = "Submit" /></td>
</tr>

</table>
</form>
</body>
</html>
```

5.3.2 Data format validation (email, number, string, mobile number, name)

Email

```
<html>
<head>
<script type = "text/javascript">
functionvalidateEmail() {
varemailID = document.myForm.EMail.value;
atpos = emailID.indexOf("@");
```

```
dotpos = emailID.lastIndexOf(".");


if (atpos< 1 || ( dotpos - atpos< 2 )) {
alert("Please enter correct email ID")
document.myForm.EMail.focus() ;
return false;
        }
return( true );
     }
</script>
</head>
<body>
<form action = "/cgi-bin/test.cgi" name = "myForm" onsubmit =
"return(validateEmail());">
<table cellspacing = "2" cellpadding = "2" border = "1">
<tr>
<td align = "right">EMail</td>
<td><input type = "text" name = "EMail" /></td>
</tr>
<tr>
<td align = "right"></td>
<td><input type = "submit" value = "Submit" /></td>
</tr>
</table>
</form>
</body>
</html>
```

Number

```
<html>
<head>
<script>
function validate(){
varnum=document.myform.num.value;
if (isNaN(num)){
document.getElementById("numloc").innerHTML="Enter Numeric value only";
return false;
}else{
return true;
  }
}
</script>
</head>

<body>
<form name="myform" action="http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return validate()" >
Number: <input type="text" name="num"><span id="numloc"></span><br/>
<input type="submit" value="submit">
</form>
</body>
</html>
```

String(Name)

```
<html>
<head>
<title>JavaScript Form Validation: Valid Zip Code </title>
<style>body{font-family:Verdana;}</style>
</head>

<body>
<script>
functioncheck_Alpha(letters){
var regex = /^[a-zA-Z]+$/;
if(regex.test(letters.yourname.value) == false){
alert("Name must be in alphabets only");
letters.yourname.focus();
return false;
    }
if(letters.yourname.value == " "){
alert("Name Field cannot be left empty");
letters.yourname.focus();
return false;
    }
return true;
  }
</script>
```

```
<form name="alphavalidate" action="form_process.php"
method="post" onSubmit="return check_Alpha(this)" >
Name :
<input type="text" size="40" name="yourname" />
<input type="submit" value="Submit" />
<input type="reset" />
</form>
</body>
</html>
```

mobile number

```
<html>
<head>
<title>Validate mobile number Javascript</title>
<script>
functionValidateNo() {
varphoneNo = document.getElementById('txtPhoneNo');

if (phoneNo.value == "" || phoneNo.value == null) {
alert("Please enter your Mobile No.");
return false;
  }
if (phoneNo.value.length< 10 || phoneNo.value.length> 10) {
alert("Mobile No. is not valid, Please Enter 10 Digit Mobile No.");
return false;
  }
```

```
alert("Success ");
return true;
}
</script>
</head>
<body>
<form name="mobileno" method="post" onSubmit="return ValidateNo()" >
<p>Mobileno</p>
<input id="txtPhoneNo" type="text"  />
<input type="button" value="Submit" onclick="ValidateNo();">
</body>
</html>
```