

305-1 WEB DESIGNING
UNIT 5 JAVASCRIPT FUNCTIONS
UNIT-5 JAVASCRIPT FUNCTIONS

5.1.1:JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability:** We can call a function several times so it save coding.
2. **Less coding:** It makes our program compact. We don't need to write many lines of code each time to perform a common task.

SYNTAX

```
function functionName([arg1, arg2, ...argN])
{
  //code to be executed
}
```

JavaScript Function Example

```
<script>
function msg()
{
  alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
```

5.1.2:JavaScript Function Arguments

```
<script>
function getcube(number)
{
  alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
```

5.1.3:Function with Return Value

```
<script>
function getInfo()
{
return "hello javatpoint! How r u?";
}
</script>
<script>
document.write(getInfo());
</script>
```

5.1.4: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:

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

```
// Simulate an HTTP redirect:
window.location.replace("http://www.w3schools.com");
```

Note: The difference between href and replace, is that replace() removes the URL of the current document from the document history, meaning that it is not possible to use the "back" button to navigate back to the original document.

5.2:JavaScript Dialogboxes

JavaScript supports three important types of dialog boxes. These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users. Here we will discuss each dialog box one by one.

Alert Dialog Box

An alert dialog box is mostly used to give a warning message to the users. For example, if one input field requires to enter some text but the user does not provide any input, then as a part of validation, you can use an alert box to give a warning message.

Nonetheless, an alert box can still be used for friendlier messages. Alert box gives only one button "OK" to select and proceed.

305-1 WEB DESIGNING

UNIT 5 JAVASCRIPT FUNCTIONS

Example

```
<html>
<head>
  <script type = "text/javascript">
    <!--
      function Warn() {
        alert ("This is a warning message!");
        document.write ("This is a warning message!");
      }
    //-->
  </script>
</head>

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

Confirmation Dialog Box

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: **OK** and **Cancel**.

If the user clicks on the OK button, the window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false. You can use a confirmation dialog box as follows.

Example

```
<html>
<head>
  <script type = "text/javascript">
    <!--
      function getConfirmation() {
        var retVal = confirm("Do you want to continue?");
        if( retVal == true ) {
          document.write ("User wants to continue!");
          return true;
        } else {
          document.write ("User does not want to continue!");
        }
      }
    </script>
  </head>
  <body>
    <p>Click the following button to see the result: </p>
    <form>
      <input type = "button" value = "Click Me" onclick = "getConfirmation();" />
    </form>
  </body>
</html>
```

305-1 WEB DESIGNING

UNIT 5 JAVASCRIPT FUNCTIONS

```
        return false;
    }
}
//-->
</script>
</head>

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

Prompt Dialog Box

The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.

This dialog box is displayed using a method called **prompt()** which takes two parameters: (i) a label which you want to display in the text box and (ii) a default string to display in the text box.

This dialog box has two buttons: **OK** and **Cancel**. If the user clicks the OK button, the window method **prompt()** will return the entered value from the text box. If the user clicks the Cancel button, the window method **prompt()** returns **null**.

Example

The following example shows how to use a prompt dialog box –

```
<html>
<head>
  <script type = "text/javascript">
    <!--
      function getValue() {
        var retVal = prompt("Enter your name : ", "your name here");
        document.write("You have entered : " + retVal);
      }
    //-->
  </script>
</head>

<body>
  <p>Click the following button to see the result: </p>
  <form>
```

305-1 WEB DESIGNING

UNIT 5 JAVASCRIPT FUNCTIONS

```
<input type = "button" value = "Click Me" onclick = "getValue();" />
</form>
</body>
</html>
```

5.3 Form Validation:

It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

JavaScript Form Validation Example

In this example, we are going to validate the name and password. The name can't be empty and password can't be less than 6 characters long.

Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

```
<script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name==""){
    alert("Name can't be blank");
    return false;
}else if(password.length<6){
    alert("Password must be at least 6 characters long.");
    return false;
}
}
</script>
<body>
<form name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form>
```

305-1 WEB DESIGNING

UNIT 5 JAVASCRIPT FUNCTIONS

JavaScript Retype Password Validation

```
<script type="text/javascript">
function matchpass(){
var firstpassword=document.f1.password.value;
var secondpassword=document.f1.password2.value;

if(firstpassword==secondpassword){
return true;
}
else{
alert("password must be same!");
return false;
}
}
</script>

<form name="f1" action="register.jsp" onsubmit="return matchpass()">
Password:<input type="password" name="password" /><br/>
Re-enter Password:<input type="password" name="password2" /><br/>
<input type="submit">
</form>
```

JavaScript Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

```
<script>
function validate(){
var num=document.myform.num.value;
if (isNaN(num)){
document.getElementById("numloc").innerHTML="Enter Numeric value only";
return false;
}else{
return true;
}
}
</script>
<form name="myform" onsubmit="return validate()" >
Number: <input type="text" name="num"><span id="numloc"></span><br/>
<input type="submit" value="submit">
</form>
```

305-1 WEB DESIGNING

UNIT 5 JAVASCRIPT FUNCTIONS

JavaScript email validation

We can validate the email by the help of JavaScript.

There are many criteria that need to be follow to validate the email id such as:

- email id must contain the @ and . character
- There must be at least one character before and after the @.
- There must be at least two characters after . (dot).

Let's see the simple example to validate the email field.

```
<script>
function validateemail()
{
var x=document.myform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
  alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
  return false;
}
}
</script>
<body>
<form name="myform" method="post" action="#" onsubmit="return validateemail();">
Email: <input type="text" name="email"><br/>

<input type="submit" value="register">
</form>
```