

# 5. JavaScript Functions

---

## Function

JavaScript functions are defined with the **function** keyword. You can use a function declaration or a function expression.

### Function Declarations

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).

#### Example

```
function myFunction(a, b) {  
  return a * b;  
}
```

Semicolons are used to separate executable JavaScript statements.

Since a function declaration is not an executable statement, it is not common to end it with a semicolon.

### Function Expressions

A JavaScript function can also be defined using an expression.

A function expression can be stored in a variable:

#### Example

```
const x = function (a, b) {return a * b};
```

After a function expression has been stored in a variable, the variable can be used as a function:

#### Example

```
const x = function (a, b) {return a * b};  
  
let z = x(4, 3);
```

## Function Parameters and Arguments

Function parameters are the names listed in the function definition. Function arguments are the real values passed to (and received by) the function.

```
function functionName (parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

### Invoking a JavaScript Function

The code inside a function is not executed when the function is defined.

The code inside a function is executed when the function is invoked.

It is common to use the term "call a function" instead of "invoke a function".

It is also common to say "call upon a function", "start a function", or "execute a function".

#### Example

```
function myFunction(a, b) {  
    return a * b;  
}  
myFunction(10, 2);    // Will return 20
```

#### Example

```
function myFunction(a, b) {  
    return a * b;  
}  
window.myFunction(10, 2);
```

## JavaScript Dialog Boxes

JavaScript has three kind of popup boxes:

- Alert box
- Confirm box
- Prompt box.

### 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

```
alert("I am an alert box!");
```

### 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

```
if (confirm("Press a button!")) {  
  txt = "You pressed OK!";  
} else {  
  txt = "You pressed Cancel!";  
}
```

### 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

```
let person = prompt("Please enter your name", "Harry Potter");
```

```
let text;
```

```
if (person == null || person == "") {
```

```
text = "User cancelled the prompt.";
```

```
} else {
```

```
text = "Hello " + person + "! How are you today?";
```

```
}
```

## Form Validation

```
<html>
<head>
<title> Form Validation </title>
<script type="text/javascript">
<!--
// Form validation code will come here.
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;
}
return(true);
}

function validateEmail() {
Var emailID=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(validate());">
<table cellpadding="2" cellspacing="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>

```