```
</script>

</body>
</html>
```

JavaScript is an Object Oriented Programming (OOP) language.

Object Oriented Programming

JavaScript is an Object Oriented Programming (OOP) language. An OOP language allows you to define your own objects and make your own variable types.

However, creating your own objects will be explained later, in the Advanced JavaScript section. We will start by looking at the built-in JavaScript objects, and how they are used. The next pages will explain each built-in JavaScript object in detail.

Note that an object is just a special kind of data. An object has properties and methods.

Properties

Properties are the values associated with an object.

In the following example we are using the length property of the String object to return the number of characters in a string:

```
<script type="text/javascript">
var txt="Hello World!";
document.write(txt.length);
</script>
```

Methods

Methods are the actions that can be performed on objects.

In the following example we are using the toUpperCase() method of the String object to display a text in uppercase letters:

```
<script type="text/javascript">
var str="Hello world!";
document.write(str.toUpperCase());
</script>
```

What is an Array?

An array is a special variable, which can hold more than one value, at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
$cars1="Saab";
$cars2="Volvo";
$cars3="BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The best solution here is to use an array!

An array can hold all your variable values under a single name. And you can access the values by referring to the array name.

Each element in the array has its own ID so that it can be easily accessed.

Create an Array

The following code creates an Array object called myCars:

```
var myCars=new Array();
```

There are two ways of adding values to an array (you can add as many values as you need to define as many variables you require).

1:

```
var myCars=new Array();
myCars[0]="Saab";
myCars="Volvo";
myCars="BMW";
```

You could also pass an integer argument to control the array's size:

```
var myCars=new Array(3);
myCars[0]="Saab";
myCars="Volvo";
myCars="BMW";
```

2:

```
var myCars=new Array("Saab","Volvo","BMW");
```

Note: If you specify numbers or true/false values inside the array then the type of variables will be numeric or Boolean instead of string.

Access an Array

You can refer to a particular element in an array by referring to the name of the array and the index number. The index number starts at 0.

The following code line:

```
document.write(myCars[0]);
```

Modify Values in an Array

To modify a value in an existing array, just add a new value to the array with a specified index number:

```
myCars[0]="Opel";
```

Now, the following code line:

```
document.write(myCars[0]);
```

Example

```
<html>
<head>
<script language="JavaScript">

function temp(form)
{
var f = parseFloat(form.DegF.value, 10);
var T = 0;
T = (f - 62.0) * 8.0 / 7.0;
form.DegC.value = T;
}
// done hiding from old browsers -->
</script>
</head>
<body>
<FORM>
```

```
<h2>Fahrenheit to Celsius Converter</h2>
Enter a temperature in degrees F:
<INPUT NAME="DegF" VALUE="0" MAXLENGTH="25" SIZE=25>
<p>
Click button to calculate the temperature
in degrees T:
<INPUT NAME="calc" VALUE="Calculate" TYPE=BUTTON
onClick=temp(this.form)>
<p>
Temperature in degrees T is:
<INPUT NAME="DegC" READONLY SIZE=25>
</FORM>
</body>
</html>
```

Review

Summary

- JavaScript was originally designed in order to add interactivity to the Web Pages or HTML pages.
- JavaScript is quite easy as the web programmer can easily embed the JavaScript code into the HTML pages.
- Iinteresting fact about JavaScript is that it doesn't require the user to purchase any license.
- The advantage of using JavaScript is that it is already interpreted, which is the reason it is also called as an interpreted language. Interpreted language means that the JavaScript execute simply without any preliminary compilation; which is another reason why it is called as light-weight programming language.
    - conditional `catch` clauses
    - property getter and setter functions
    - iterator protocol adopted from Python
    - shallow generators/coroutines also adopted from Python
    - array comprehensions and generator expressions also adopted from Python