WEBISOFTECH

Web-www.webisoftech.com
E-mail- enquiry@webisoftech.com
Phone- +91 9766962674

JavaScript Variables

JavaScript Data types

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types −

Numbers, eg. 123, 120.50 etc.

Strings of text e.g. "This text string" etc.

Boolean e.g. true or false.

JavaScript also defines two trivial data types, null and undefined, each of which defines only a single value. In addition to these primitive data types, JavaScript supports a composite data type known as object.

JavaScript Variables

```
<script type = "text/javascript">
  <!--
    var money;
    var name;
  //-->
</script>
```

```
<script type = "text/javascript">
  <!--
    var money, name;
  //-->
</script>
```

```
<script type = "text/javascript">
  <!--
    var name = "Ali";
    var money;
    money = 2000.50;
  //-->
</script>
```

JavaScript Variable Scope

The scope of a variable is the region of your program in which it is defined. JavaScript variables have only two scopes.

Global Variables − A global variable has global scope which means it can be defined anywhere in your JavaScript code.

Local Variables − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

```
<html>
   <body onload = checkscope();>
      <script type = "text/javascript">
         <!--
            var myVar = "global";      // Declare a global variable
            function checkscope( ) {
               var myVar = "local";    // Declare a local variable
               document.write(myVar);
            }
         //-->
      </script>
   </body>
</html>
```

This produces the following result −

```
local
```

JavaScript Variable Names

While naming your variables in JavaScript, keep the following rules in mind.

You should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section. For example, break or Boolean variable names are not valid.

JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character. For example, 123test is an invalid variable name but _123test is a valid one.

JavaScript variable names are case-sensitive. For example, Name and name are two different variables.