



JAVASCRIPT VARIABLES

CIT 260:03 – MOBILE APPLICATION DEVELOPMENT

JAVASCRIPT DATATYPES

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 DATATYPES

JavaScript allows you to work with three primitive data types:

- ✓ **Numbers:** 123, 120.50 etc.
- ✓ **Strings** of text: "This text string" etc.
- ✓ **Boolean:** true or false.

JAVASCRIPT DATATYPES

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 DATATYPES

JavaScript does not make a distinction between integer values and floating-point values. All numbers in JavaScript are represented as floating-point values.

JAVASCRIPT VARIABLES

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

JAVASCRIPT VARIABLES

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the **var** keyword as follows.

JAVASCRIPT VARIABLES

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


JAVASCRIPT VARIABLES

You can also declare multiple variables with the same **var** keyword as follows.

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

JAVASCRIPT VARIABLES

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

JAVASCRIPT VARIABLES

For instance, you might create a variable named **money** and assign the value 2000.50 to it later. For another variable, you can assign a value at the time of initialization as follows.

JAVASCRIPT VARIABLES

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

JAVASCRIPT VARIABLES

Use the **var** keyword only for declaration or initialization, once for the life of any variable name in a document. You should not re-declare same variable twice.

JAVASCRIPT VARIABLES

JavaScript is **untyped** language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold.

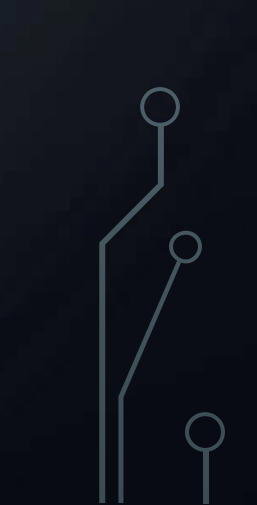

JAVASCRIPT VARIABLES

The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.



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.



JAVASCRIPT VARIABLE SCOPE

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

JAVASCRIPT VARIABLE SCOPE

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

JAVASCRIPT VARIABLE SCOPE

Within the body of a function, a local variable takes precedence over a global variable with the same name. If you declare a local variable or function parameter with the same name as a global variable, you effectively hide the global variable.

JAVASCRIPT VARIABLE SCOPE

Take a look into the following example.

```
<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>
```



JAVASCRIPT VARIABLE NAMES

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



JAVASCRIPT VARIABLE NAMES

You should not use any of the JavaScript reserved keywords as a variable name. These keywords are mentioned in the next section.



JAVASCRIPT VARIABLE NAMES

For example:

break or **boolean** variable names are not valid.



JAVASCRIPT VARIABLE NAMES

JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or an underscore character.

JAVASCRIPT VARIABLE NAMES

For example:

123test is an invalid variable name
but **_123test** is a valid one.



JAVASCRIPT VARIABLE NAMES

JavaScript variable names are case-sensitive.



JAVASCRIPT VARIABLE NAMES

For example:

Name and **name** are two different variables.

JAVASCRIPT VARIABLE WORDS

A list of all the reserved words in JavaScript are given in the following table. They cannot be used as JavaScript variables, functions, methods, loop labels, or any object names.

JAVASCRIPT VARIABLE WORDS

✓ abstract

✓ boolean

✓ break

✓ byte

✓ case

JAVASCRIPT VARIABLE WORDS

✓ catch

✓ char

✓ class

✓ const

✓ continue

JAVASCRIPT VARIABLE WORDS

✓ debugger

✓ default

✓ delete

✓ do

✓ double

JAVASCRIPT VARIABLE WORDS

- ✓ else
- ✓ enum
- ✓ export
- ✓ extends
- ✓ false

JAVASCRIPT VARIABLE WORDS

- ✓ final
- ✓ finally
- ✓ float
- ✓ for
- ✓ function

JAVASCRIPT VARIABLE WORDS

✓ goto

✓ if

✓ implements

✓ import

✓ in

JAVASCRIPT VARIABLE WORDS

✓ instanceof

✓ int

✓ interface

✓ long

✓ native

JAVASCRIPT VARIABLE WORDS

✓ new

✓ null

✓ package

✓ private

✓ protected

JAVASCRIPT VARIABLE WORDS

✓ public

✓ return

✓ short

✓ static

✓ super

JAVASCRIPT VARIABLE WORDS

✓ switch

✓ synchronized

✓ this

✓ throw

✓ throws

JAVASCRIPT VARIABLE WORDS

✓ transient

✓ true

✓ try

✓ typeof

✓ var

JAVASCRIPT VARIABLE WORDS

- ✓ void
- ✓ volatile
- ✓ while
- ✓ with



JAVASCRIPT VARIABLES

SOURCE:

[HTTPS://WWW.TUTORIALSPOINT.COM/JAVASCRIPT/JAVASCRIPT_VARIABLES.HTM](https://www.tutorialspoint.com/javascript/javascript_variables.htm)