



# **JAVASCRIPT** **VARIABLE** **AND** **DATA TYPES**

# JAVASCRIPT VARIABLE :-

A **javascript variable** is simply a name of storage location.

**There are some rules while declaring a JavaScript variable (also known as identifiers).**

- ❖ Name must start with a letter (a to z or A to Z), underscore( \_ ), or dollar( \$ ) sign.
- ❖ After first letter we can use digits (0 to 9), for example value1.
- ❖ JavaScript variables are case sensitive, for example x and X are different variables.



# There are two types of variables in JavaScript :

- ❑ local variable
- ❑ global variable.

❑ **Local variable** -: A JavaScript local variable is declared inside block or function. It is accessible within the function or block only.

For example :

```
<script>  
function abc(){  
  var x=10;           //local variable  
}  
</script>
```

## ❑ Global variable :-

A JavaScript global variable is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable.

For example :

```
<script>
var data=200;                // gloabal variable
function a(){
document.writeln(data);
}
function b(){
document.writeln(data);
}
a();                          //calling JavaScript function
b();
</script>
```



# Declarations

JavaScript has three kinds of variable declarations.

1. **Var** : Declares a variable, optionally initializing it to a value.
2. **Let** : Declares a block-scoped, local variable, optionally initializing it to a value.
3. **Const** : Declares a block-scoped, read-only named constant.



# Var :

- ✓ Variables are containers for storing information.
- ✓ Creating a variable in JavaScript is called "declaring" a variable:  

```
var carName;           //the variable is empty (it has no value).
```
- ✓ To assign a value to the variable, use the equal sign:  

```
carName = "Volvo";
```
- ✓ You can also assign a value to the variable when you declare it:  

```
var carName = "Volvo";
```

# Let :

The keyword was introduced in ES6 (2015).

- ❖ Variable define with Let **can not be redeclared**.
- ❖ Variable define with Let **must be declared before use**.
- ❖ Variable define with Let **have block scope**.

❑ Example : **With Let can not be redeclared.**

```
let x = "John Doe";  
let x = 0;
```

❑ **Example :** 2 . These Let keywords provide **Block Scope** in JavaScript.

```
{  
    let x = "ajay";  
}  
document.write("x");  
// x can NOT be used here (outside the block)
```

❑ **Example :** 3 . Redeclaring Variables.

```
let x = 10;    // Here x is 10    //allowed  
{  
    let x = 2; // Here x is 2      //allowed  
}  
{  
    let x = 5; // Here x is 5      //allowed  
}
```



# Const :

The keyword was introduced in [ES6 \(2015\)](#).

- ❖ Variable define with Const **can not be Redeclared**.
- ❖ Variable define with Const **cannot be Reassigned**.
- ❖ Variable define with Let **have block scope**.

✓ Example : variable cannot be reassigned:

```
Const x = "John Doe";
```

```
x = 0;
```

// This will give an error



❑ **Example :** 2 . These Const keywords provide **Block Scope** in JavaScript.

```
{  
    Const x = 2;  
}  
document.write(x);  
// x can NOT be used here (outside the block)
```

❑ **Example :** 3 . Redeclaring Variables.

```
Const x = 10; // Here x is 10    // allowed  
{  
    Const x= 2;    // Here x is 2    // allowed  
}  
{  
    Const x = 5;    // Here x is 5    // allowed  
}
```

## ❑ Java script Data Type -:

JavaScript provides different data types to hold different types of values.

**There are two types of data types in JavaScript.**

- **Primitive data type -:**
- **Non-primitive (reference) data type -:**

JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use var here to specify the data type. It can hold any type of values such as numbers, strings etc.

**For example :**

```
var a=40;           //holding number  
var b="Rahul";     //holding string
```

## ➤ Primitive data type -:

There are five types of primitive data types in JavaScript. They are as follows:

**For example :**

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all



## ➤ JavaScript non-primitive data types -:

There are Three Types of primitive data types in JavaScript.

The non-primitive data types are as follows :

**For example :**

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values