JavaScript Variable

1. JavaScript variable
2. JavaScript Local variable
3. JavaScript Global variable

A **JavaScript variable** is simply a name of storage location. The actual value of a variable can be changed at any time. Variables are helpful in various situations such as:

- **Variables as Placeholders for unknown values:** They can be used in places where the values they represent are unknown when the code is written.
- **Variables as Code Clarifies:** They can make the purpose of your code cleaner.

Rules of naming variables in JavaScript:

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

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

Naming Variables: Before starting to name variables, you should need to aware of JavaScript's naming rules. The factors you need to consider when selecting variable names are case sensitivity, invalid characters and the reserved words used in JavaScript.

1. **Using Case in Variables:** In JavaScript variables are case-sensitive. **For Example:** "total" and "TOTAL" have different meanings in JavaScript.
   - If you are using a variable name that consists of only a word, it is make sure that easier way to use lowercase letters.
   - If you are using a variable name with two words such as "total count". It's better to capitalize the first letter of the word. **For example**. "Total_Count", etc.

2. Allowed special characters: An important rule to remember is that variable name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign. **For example:** "_totalpay", Total_Count etc.

- After the first letter, we can use digits (0 to 9), **for example:** "value1".

3. Avoiding Reserved Words: When naming variables in JavaScript avoid the use of the reserved word. **For example:** "if", "case" etc.

Declaring Variables:

To declare text as a variable, you can use the "var" or "let" keyword. The Following syntax is used for declaring a variable in JavaScript:

Syntax:

1. var variable_name;

In the above syntax, "var" is a keyword and "variable_name" is a name given to a variable.

For Example:

1. var total_amount;

In the above example, "var" is a keyword and "total_amount" is a variable name.

Assign Value to Variables:

For assigning a value to a variable, you can use the JavaScript assignment operator (=).

Syntax:

1. var variable_name = value;

In the above syntax, "var" is a keyword, "variable_name" is a name given to a variable and value is used to assign a value to a variable.

For Example:

1. var total_amount = 500;

In the above example, "var" is a keyword, "total_amount" is a variable name and 500 is a value assigned to a variable.

Example of JavaScript variable

Let's see a simple example of JavaScript variable.

```
<script>
var x = 10;
var y = 20;
var z=x+y;
document.write(z);
```

```
</script>
```

Types of Variables:

There are two types of variables in JavaScript:

- Local variable
- Global variable

Let us explain one by one.

JavaScript 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>
```

Or,

```
<script>
If(10<13){
var y=20;//JavaScript local variable
}
</script>
```

JavaScript 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>
```

JavaScript Global Variable

A **JavaScript global variable** is declared outside the function or declared with window object. It can be accessed from any function.

Let's see the simple example of global variable in JavaScript.

```
<script>
var value=50;//global variable
function a(){
alert(value);
}
function b(){
alert(value);
}
</script>
```

Declaring JavaScript global variable within function

To declare JavaScript global variables inside function, you need to use **window object**. For example:

1. window.value=90;

Now it can be declared inside any function and can be accessed from any function. For example:

```
function m(){
window.value=100;//declaring global variable by window object
}
```

```
function n(){
alert(window.value);//accessing global variable from other function
}
```

Internals of global variable in JavaScript

When you declare a variable outside the function, it is added in the window object internally. You can access it through window object also. For example:

```
var value=50;
function a(){
alert(window.value);//accessing global variable
    }
```

Javascript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. 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:

1. var a=40;//holding number
2. var b="Rahul";//holding string

JavaScript primitive data types

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

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

The non-primitive data types are as follows:

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