CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP

Variables: let, const & var in JavaScript | JavaScript Tutorial In Hindi #3

Overview Q&A Downloads Announcements

Variables: let, const & var in JavaScript | JavaScript Tutorial In Hindi #3

In today's tutorial, we are going to study ***variables in JavaScript*** and their different types. As we know, that variable is the name of the storage location. When we want to save some data, we store it in a variable. In any programming language, we typically do lots of calculations. The calculation results are stored in the computer's memory. Just like human memory, the memory of the computer also consists of millions of cells. The calculated values are stored in these memory cells. To make the usage and retrieval of these values easy, these memory locations are given names. The names given to these locations are called variables.
Data types in JavaScript are either Variables or Constants. ES6 has made major changes in JavaScript's syntax and has also brought new features. Initially, we used to declare variables with a keyword **"var"**. However, ES6 has brought a new variable declaration keyword, ***"let"*** and ***"const."***
Following are some rules while declaring a JavaScript variable:
1. A variable name must start with a letter (a to z or A to Z), underscore (_), or dollar($) sign.
2. A variable name cannot start with a number. After the first letter, we can use digits (0 to 9), for example, message1.
3. JavaScript variables are case sensitive. For example, a and A are different variables.
So, let us explore these variable declaration keywords in detail.

Var:-

Before the advent of ES6, var declarations were used to declare a variable. The properties of var is that it has visibility linked to the function to which it belongs. We can change its value, and it can be redeclared in the same scope. Scope means where these variables are available for use. There are two types of scope, local and global. Var declarations are globally scoped, and when they are defined inside a function, they are locally scoped.
Example of var;-

```
var age = 25; // Number
var name = "John"; // String
var developer = true;// Boolean
var location = null; // Null
var task; // undefined
```

When the variable is declared, the JavaScript engine assigns it a memory or space. Because of this, once a variable is declared, it takes a value of undefined even before the assignment. We assigned data to the variable by using the assignment operator "=". Datatypes in JavaScript are:

1. Number i.e., 11,23,45,6
2. Strings, i.e., "Hello World."
3. Boolean, i.e., true, false
4. Undefined
5. Null
6. Any of the complex data structures (Array and Objects)

let:-

The variable type let is introduced in ES6. It shares a lot of similarities with var, but unlike var, it has scope constraints. Its declaration and assignment are similar to var. The purpose of introducing let is to resolve all issues posed by variables scope, which developers face during development. The properties of let are that They have visibility linked to the block they belong with. We can change their values, but they cannot be redeclared in the same scope, unlike var.
"let" helps us by making it easier to see where variables live in our code and make our code cleaner and easier to read.

Example of let:-

```
let age = 25; // Number
let name = "John"; // String
let developer = true;// Boolean
let location = null; // Null
let task; // undefined
let age= 50;
console.log(age); // SyntaxError: identifier "age" has already been declared.
```

Const:-

Const is also introduced in ES6. It is a variable type assigned to data whose value cannot and will not be changed throughout the program. Const is more strict as compared to var and let. Const is also limited to the scope in which it operates. We declare const just like var and let. We use const when we are sure a variable will not be redeclared. The characteristic of const and their declarations are block-scoped, and they cannot be updated or redeclared.

Example:-

```
const age = 20;
const job = 'developer';
const name; // SyntaxError: missing initializer
const num = 10;
num = 20; //Compiler Error: Cannot assign to 'num' because it is a constant or read-only property
```

Single line comments in JavaScript:-

To create a single line comment in JavaScript, we have to place two slashes "//" in front of the code or text that we want the interpreter to ignore. When we place these two slashes, all text will be ignored, until the next line. Single line comments are used for writing small notes in the program

```
// This is a single line JavaScript comment
```

Multi-line comments in JavaScript:-

Although a single line comment is quite useful, but when we want to comment the long segment of code, we have to use multiline comment. Multiline comment begins with /* and ends with */.

```
/*
This is a Multiline Comment
This is a Multiline Comment
This is a Multiline Comment
 */
```

Code index.html as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Tutorial on Js</title>
</head>
<body>
    <h1>This is Js tutorial by Harry</h1>
</body>

<!-- <script src="js/tut2.js"></script> -->
<!-- <script src="js/tut3.js"></script> -->
<!-- <script src="js/tut4.js"></script> -->
```

```
<!-- <script src="js/tut5.js"></script> -->
<!-- <script src="js/tut6.js"></script> -->
<!-- <script src="js/tut7.js"></script> -->
<!-- <script src="js/tut8.js"></script> -->
<!-- <script src="js/tut9.js"></script> -->
<!-- <script src="js/tut10.js"></script> -->
<script src="js/tut11.js"></script>
</html>
```

Js code as described/written in the video

```
console.log('tut3');
// Variables in js
// var, let, const
var name = 'harry';
var channel;
var marks = 3454;
marks = 0;
channel = 'CodeWithHarry'
console.log(name, channel, marks);
// Rules for creating JavaScript Variables
/*
1. Cannot start with numbers
2. Can start with letter, numbers, _ or $
3. Are case sensitive
*/
var city = 'Delhi';
console.log(city);

const ownersName = 'Hari Ram';
// ownersName = 'Harry'; // Cannot do this (error)
console.log(ownersName);
const fruit = 'Orange';

{
 let city = 'Rampur';
 city = 'Kolkata';
 console.log(city);
}
console.log(city);


const arr1 = [12,23,12,53, 3];
// arr1.push(45);
console.log(arr1)

/* Most common programming case types:

1. camelCase
2. kebab-case
3. snake_case
4. PascalCase


*/
```

Previous Next

CodeWithHarry