

Variables

variables are the containers for storing data values. Here are some key notes about a variables:

1. **Declaration:** Variables in JavaScript are declared using the **var**, **let**, or **const** keyword.
 - **var** has been traditionally used but has some scope-related issues.
 - **let** is block-scoped and is preferable for variable declaration.
 - **const** is also block-scoped but its value cannot be reassigned once it's set.

```
//variable declared with var keyword followed by a name & assigned a value using  
assignment operator  
var x = 10;  
let y = 20;  
const PI = 3.14;
```

2. **Naming Convention:** Variable names in JavaScript can contain letters, digits, underscores, and dollar signs. They must begin with a letter, underscore, or dollar sign.

```
var myVariable = 5;  
var _myVariable = 10;  
var $myVariable = 15;
```

3. **Case Sensitivity:** JavaScript variable names are case-sensitive, meaning **myVariable** and **MyVariable** are treated as different variables.

```
var myVariable = 5;  
var MyVariable = 10;  
  
console.log(myVariable); // Outputs: 5  
console.log(MyVariable); // Outputs: 10
```

4. **Data Types:** JavaScript variables can hold various data types including numbers, strings, objects, arrays, functions, etc.

```
var num = 5;

var str = "Hello";
var arr = [1, 2, 3];
var obj = { name: "John", age: 30 };
```

5. **Dynamic Typing:** JavaScript is dynamically typed, meaning you don't have to specify the data type of a variable when declaring it. The data type of the variable is determined automatically at runtime.
6. **Scope:** Variables in JavaScript have function or block scope, depending on how they are declared.
- Variables declared with **var** are function-scoped. They are accessible anywhere within the function they are declared in.
 - Variables declared with **let** or **const** are block-scoped. They are only accessible within the block they are declared in.
7. **Hoisting:** Variable declarations are hoisted to the top of their scope during the compilation phase, but their assignments remain where they are.

```
myVar=5;
console.log(myVar); // Output: 5
var myVar; //-----This declaration is moved to top even before the code
execution-Hoisting
```

8. **Global Variables:** Variables declared outside of any function have global scope and can be accessed and modified from any part of the script.
9. **Local Variables:** Variables declared within a function have local scope and are only accessible within that function.
10. **Variable Reassignment:** Variables declared with **var** and **let** can be reassigned, while variables declared with **const** cannot be reassigned, though their properties can be modified if they are objects.

```
let x = 5;
x = 10; // This is valid
```

11. **Initialization:** It's recommended to initialize variables when declaring them to avoid unexpected behavior due to **undefined** values.

```
x = 5;
```

12. **Use of Strict Mode:** Using "**use strict**"; at the beginning of a script or function enables strict mode, which helps catch common coding errors and prevents certain actions