**JavaScript uses keywords; var, let and const to declare a new variable**.

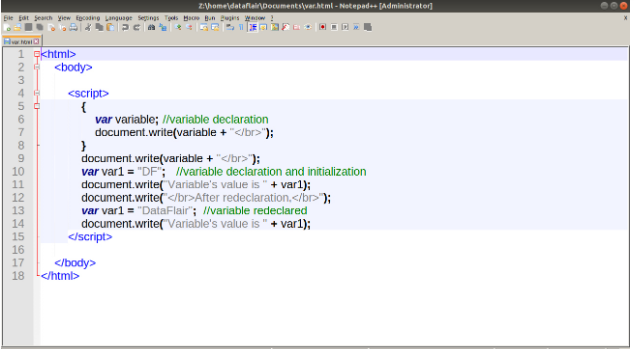
**Variable:**

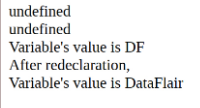
**Var is function scoped and redefine the variables.**

**Let and const are block scoped**

**Var**

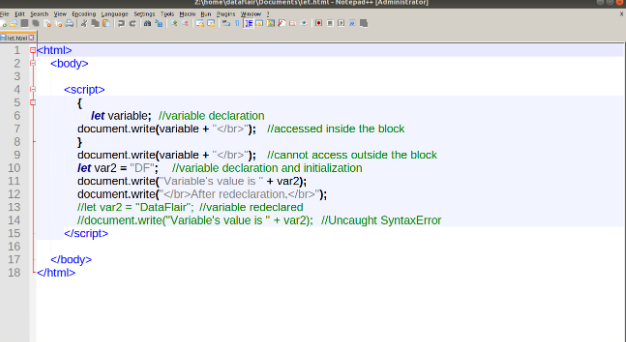
 We declare a variable, optionally initializing it to a value. We can reassign the values to the variables or **redefine the variables.** If you re-declare a variable, it will not lose its value.





**2.Let**

**You cannot redeclare** the let declared variables. You will face a **SyntaxError** if you try to do so. The scope of these variables is the block they are defined in, as well as any sub-blocks. The browser will throw a **ReferenceError** if you try to access these variables outside of their block.



**3.const**

The value of a constant is read-only; you cannot change it through reassignment, and you can’t redeclare it. An initializer for a constant is required; i.e.,

1.

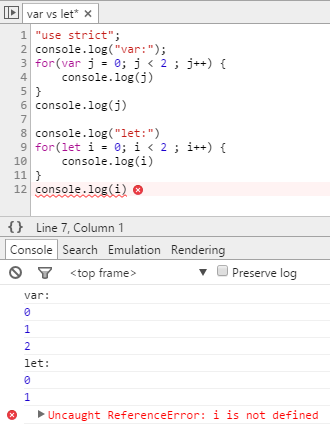
document.write(x);//undefined

var x=20;

document.write(y);//uncaught reference:error

let y=10;

**When global and local variables have the same name, the local variable gets precedence.** When we access the variable, JavaScript interpreter will use the value of the local variable to perform its tasks.

Here is an example for the difference between the two Let and Var(support just started for chrome):  


As you can see the var j variable is still having a value outside of the for loop scope (Block Scope), but the let i variable is undefined outside of the for loop scope.