

GOOD JAVASCRIPT DEVELOPERS DO THIS



Declare Local Variables

- Minimize the use of global variables. Global variables and functions can be overwritten by other scripts.
- All variables used in a function should be declared as local variables.
- Local variables must be declared with the `var` or `let` or `const` keyword, otherwise they will become global variables.

Declarations on Top

- It's a good coding practice to put all declarations at the top of each script or function.
- This gives the code a cleaner look and makes it easier to avoid unwanted global variables.

```
// Declare at the beginning  
let firstName, lastName, age;  
  
// Use later  
firstName = "John";  
lastName = "Doe";  
age = 30;
```

Initialize Variables

- It's a good coding practice to initialize variables when you declare them.
- This gives the code a cleaner look and provides a single place to initialize variables.

```
// Declare and initiate at the beginning  
let firstName = "",  
let lastName = "",  
let price = 0,  
let discount = 0,  
let fullPrice = 0,  
const myArray = [],  
const myObject = {};
```


Declare Objects & Arrays with const

- ▶ Declaring objects and Arrays with const prevents any accidental change of type.

```
let person = {name:"John", age:"30"};  
person = "James"; // Changes object to string
```

With const:

```
const person = {name:"John", age:"30"};  
person = "James"; // Not possible
```

Beware of Automatic Type Conversions

- ▶ Javascript is a loosely typed language which means you don't specify the type of variable while declaring like int, float, or char. You just give it data of any type and that could be overwritten.

```
let x = "Hello"; // typeof x is a string  
x = 5;           // changes typeof x to a number
```

A good solution to it is to declare all the variables which you don't want to reassign later with the `const` keyword.

Use Parameter Defaults

- If a function is called with a missing argument, the value of the missing argument is set to `undefined` and that could break your code.

Avoid it using default Parameters:

```
function (a=1, b=1){  
    /*function code*/  
}
```

Now if arguments will be missing then automatically `a` and `b` will set to value 1