

JS

JavaScript Best Practices



JS



Saad Irfan
@DevWithSaad



1. Avoid global variables

Global variables and function names are an incredibly bad idea.

The reason is that every JavaScript file included in the page runs in the same scope.

One way to address it is by wrapping the code in an **Immediately Invoked Function Expression (IIFE)**. You can read more about IIFE in one of my posts on my profile.



Saad Irfan
@DevWithSaad



2. Triple equal

The `==` comparison operator always converts to matching types before comparison. The `===` operator forces comparison of values and type.

```
index.js

const firstNum = 1;
const secondNum = '1';

if (firstNum == secondNum) {
  console.log('true');
} else {
  console.log('false');
}

// output → true

if (firstNum === secondNum) {
  console.log('true');
}

// output → false
```



Saad Irfan

@DevWithSaad



3. Modularize

This is a general programming best practice — making sure that you create functions that fulfill **one job** at a time.

This makes it easy for other developers to debug and change your code without having to scan through all the code to work out what code block performs what function.



Saad Irfan
@DevWithSaad



4. Declare Objects w/ const

Declaring objects with const will prevent any accidental change of type.



index.js

```
let car = { type: 'Fiat', model: '500', color: 'white' };  
car = 'Fiat'; // Changes object to string  
  
const car = { type: 'Fiat', model: '500', color: 'white' };  
car = 'Fiat'; // error
```



Saad Irfan
@DevWithSaad



5. Optimize loops

Loops can become very slow if you don't do them right. One of the most common mistake is to read the length attribute of an array at every iteration.

You can avoid that by storing the length value in a different variable:

```
index.js

const arr = [1, 2, 3];

const length = arr.length;
for (let i = 0; i < length; i++) {
  console.log(arr[i]);
}
```



Saad Irfan

@DevWithSaad



6. Don't Use new Object()

- Use "" instead of **new String()**
- Use 0 instead of **new Number()**
- Use false instead of **new Boolean()**
- Use {} instead of **new Object()**
- Use [] instead of **new Array()**
- Use /(())/ instead of **new RegExp()**
- Use function (){} instead of **new Function()**

