

JS Best Practices

```
//1. Variable Naming = Name deya e jano buja jai ki kora hosse
//2. Function Namin = Function mening full hote hobe
//Example
//bad practice
function email(user) {
    //implementation
}
//Good practice
function sendEmailToUser(email) {
    //implementation
}

//3. function Argument kom thaka e valo
function fetProduct({friend, from, to}){/////object pathalam number of urgument ekta ai khane
}

//4. default argument use korbo, conditional na
//Bad
function create(type){
    const shae = type || 'circle';
}
//Good
function create(type = 'circle'){
    //....
}

//5. ekta function er modda multiple kaj na kora e valo, function bashi likle problem nai

//6. === use kora e valo

//7. default object value
const shapeConfig = {
    type: 'Circle',
    with: 123,
    // height: null          //null dekhe aita ai khane rakbo e na, Exclude the 'height key
}
function createShape(config){
    config = Object.assign(//aita function jeta notun object return kore, ai function 2 ta paramitter nai
        source, target
        { //ai target amr aita output dorkar
            type:'Circle',
            with: 300,
            height: 300,
        },
        config//source objcet k target er modda niya nibe & finally target k return kore
    );
}
create(shapeConfig);
//confucion thakle object.assign er documentation theke dakhe nite parbo

//8. global prototype pollute na kora e valo.
//Bad
Array.prototype.myFunction = function myFunction() { //ai vabe prototype er modda jodi function/propety du
    kiya dei pore ja koto array tai user korte parbo
    //...
};
//Good
class myArray extends Array {
    myFunc() {
        //..
    }
}
```

//9. Conditional shorthand ব্যবহার

//Bad

```
if(isValid === true) {
```

```
}
```

```
if(isValid === false) {
```

```
}
```

//Good

```
if(isValid) {
```

```
}
```

```
if(!isValid) {
```

```
}
```

//10. method chaning ব্যবহার

//Bad

```
class Product {
```

```
  constructor(name) {
```

```
    this.name = name;
```

```
  }
```

```
  setUnits(units) {
```

```
    this.units = units;
```

```
  }
```

```
}
```

```
const product = new Product("Bag");
```

```
product.setUnits(12);
```

//Good

```
class Product {
```

```
  constructor(name) {
```

```
    this.name = name;
```

```
  }
```

```
  setUnits(units) {
```

```
    this.units = units;
```

```
    return this; //this k return kore dilam
```

```
  }
```

```
}
```

```
const product = new Product("Shirt").setUnits(2); //method chaning
```

//11. avoid using eval ekdom e use korbo na

```
eval("alert('Hi;')") //aita k onek protect kore use korte hoi
```

//12. Curly Braces use korbo

//Bad

```
if(something)
```

```
x = false;
```

//13. var use korbo na const let use korbo

//14. Semicolon use korbo

```
//15. IIFE use korbo as much as possible = imidiate invoke function
(function a() {

})();

//16. Avoid using global
//use module patern
//Bad
function chaning() {

}

//Good
const current = (function a() {

})();
```