



Published in JavaScript in Plain English



fatfish

Follow

Mar 6 · 2 min read · ✨ · Listen



Save



7 Cool Useful JavaScript Coding Techniques That You Should Use

Do you know all of these tricks?



1. Use object instead of “switch”

We often use `switch` to handle different things, but have you ever thought of using an object to greatly simplify your code?(It works in some simple scenarios)

```
1  // ❌
2  const n = 1
3  let result
4  switch (n) {
5    case 1:
6      result = 'res-1'
7      break
8    case 2:
9      result = 'res-2'
10     break
11    case 3:
12      result = 'res-3'
13      break
14    // ...There are a lot more
15  }
```

switch.js hosted with ❤ by GitHub

[view raw](#)

You only need to use an object to achieve your goal.

```
1  // ✅
2  const n = 1
3  const nMap = {
4    1: 'res-1',
5    2: 'res-2',
6    3: 'res-3'
7  }
8  const result = nMap[ n ]
```

switch2.js hosted with ❤ by GitHub

[view raw](#)

2. Use the “includes” method instead of multiple “if”

Do you often write code like this? Multiple conditions can trigger a piece of logic. As your business grows, you may need to write more “||”, that’s terrible.

```

1 // ❌
2 const n = 1
3 if (n === 1 || n === 2 || n === 3 || n === 4 || n === 5) {
4   // ...
5 }

```

includes.js hosted with ❤️ by GitHub

[view raw](#)

Use `includes` to make your code extra maintainable.

```

1 // ✅
2 const n = 1
3 const conditions = [ 1, 2, 3, 4, 5 ] // You just need to add new numbers here
4 if (conditions.includes(n)) {
5   // ...
6 }

```

includes2.js hosted with ❤️ by GitHub

[view raw](#)

3. Use the default parameters of the ES6 function

Why don't you use default parameters?

```

1 // ❌
2 const func = (name) => {
3   name = name || 'fatfish'
4   console.log(name)
5 }
6 // ✅
7 const func = (name = 'fatfish') => {
8   console.log(name)
9 }

```

es6-parameters.js hosted with ❤️ by GitHub

[view raw](#)

4. Use “`? ... : ...`” instead of “`if... else...`”

Many times simple conditional judgments do not need to use “`if`”.

```
1 // ❌
2 const n = 18
3 let result
4 if (n % 2 === 0) {
5   result = 'even number'
6 } else {
7   result = 'odd number'
8 }
```

if.js hosted with ❤️ by GitHub

[view raw](#)

Simplify your code with just a ternary expression.

```
1 // ✅
2 const n = 18
3 let result = n % 2 === 0 ? 'even number' : 'odd number'
```

if2.js hosted with ❤️ by GitHub

[view raw](#)

5. Use “+” to convert string to number

You might be using `Number()` and `parseInt()` to convert string to number.

```
1 // ❌
2 let str = '123'
3 let num = Number(str) // 123
4 let num2 = parseInt(str) // 123
```

+.js hosted with ❤️ by GitHub

[view raw](#)

It’s actually easier to use “+”.

```
1 // ✅
2 let str = '123'
3 let num = +str // 123
```

+2.js hosted with ❤️ by GitHub

[view raw](#)

6. Use “JSON.stringify” to output prettier information

Here’s a deeply nested object, you might use `console.log` to print it.

```
1 // ✖
2 const bigObj = {
3   name: 'fatfish',
4   obj: {
5     name: 'fatfish',
6     obj: {
7       name: 'fatfish',
8       obj: {
9         name: 'fatfish',
10        obj: {
11          name: 'fatfish',
12          // ...
13        }
14      }
15    }
16  }
17 }
18 console.log(bigObj)
```

JSON.stringify.js hosted with ❤ by GitHub

[view raw](#)

```
> const bigObj = {
  name: 'fatfish',
  obj: {
    name: 'fatfish',
    obj: {
      name: 'fatfish',
      obj: {
        name: 'fatfish',
        obj: {
          name: 'fatfish',
          // ...
        }
      }
    }
  }
}
```

```
console.log(bigObj)
```

```
▼ {name: 'fatfish', obj: {...}} ⓘ
  name: "fatfish"
  ▼ obj:
    name: "fatfish"
    ▼ obj:
      name: "fatfish"
      ▼ obj:
        name: "fatfish"
        ► [[Prototype]]: Object
        ► [[Prototype]]: Object
        ► [[Prototype]]: Object
        ► [[Prototype]]: Object
        ► [[Prototype]]: Object
```

```
⏏ undefined
```

But that is not convenient for viewing specific properties, we need to manually expand each level to see the data.

```
1 // ✅
2 const bigObj = {
3   name: 'fatfish',
4   obj: {
5     name: 'fatfish',
6     obj: {
7       name: 'fatfish',
8       obj: {
9         name: 'fatfish',
10        obj: {
11          name: 'fatfish',
12          // ...
13        }
14      }
15    }
16  }
17 }
18 console.log(JSON.stringify(bigObj, null, 2))
```


JSON.stringify2.js hosted with ❤️ by GitHub

[view raw](#)

It's really convenient and intuitive.

[Open in app](#) ↗



 Search Medium



```

> const bigObj = {
  name: 'fatfish',
  obj: {
    name: 'fatfish',
    obj: {
      name: 'fatfish',
      obj: {
        name: 'fatfish',
        obj: {
          name: 'fatfish',
          // ...
        }
      }
    }
  }
}
console.log(JSON.stringify(bigObj, null, 2))

```

```

{
  "name": "fatfish",
  "obj": {
    "name": "fatfish",
    "obj": {
      "name": "fatfish",
      "obj": {
        "name": "fatfish",
        "obj": {
          "name": "fatfish"
        }
      }
    }
  }
}

```

4. undefined



57



4



7. Initialize the array with “fill”

Initialize an array of a fixed length and each item is “fatfish”.

```

1 // ❌
2 let array = []
3 const len = 100
4 for(let i = 0; i < len; i++){
5   array[i] = 'fatfish'
6 }

```

fill.js hosted with ❤ by GitHub

[view raw](#)

Actually, using `fill`, it will be very easy.


```
1 // ✅  
2 let array = Array(100).fill('fatfish')
```

fill2.js hosted with ❤️ by GitHub

[view raw](#)

Finally

Thanks for reading. Looking forward to your following and reading more high-quality articles.

Interviewer: Can “x !== x” Return True in JavaScript?

Five magical JavaScript knowledge points you may not know!

javascript.plainenglish.io

What does (123[‘toString’].length + 123) Print Out in JavaScript?

A question that 95% of front-end developers answer incorrectly.

javascript.plainenglish.io

Interviewer: You Have Been Working for 3 Years, and You Can’t Answer This Algorithm Question?

A girl’s interview experience

javascript.plainenglish.io

My Boss: You Know ES6, But Why Don’t You Use It? 🙄

10 complaints from my boss that helped me learn a lot.

javascript.plainenglish.io

More content at [PlainEnglish.io](https://plainenglish.io).

Sign up for our [free weekly newsletter](#). Follow us on [Twitter](#), [LinkedIn](#), [YouTube](#), and [Discord](#).

Interested in scaling your software startup? Check out [Circuit](#).

Front End Development

JavaScript

Programming

React

Vuejs