fatfish Follow

Jan 30 · 2 min read · ✦ · ▶ Listen

🔖 Save   🐦   f   in   🔗   •••

7 ES6 Spread Operator Tricks And Tips Every Developer Should Know About

7 ES6 Tricks Make You Write Less And Do More.

Photo by Gaelle Marcel on Unsplash

Preface

ES6 provides a new operator called the spread operator, which is composed of three points (…).

The spread operator allows you to expand the elements of an iterable object, such as map, array, set, and object. That even allows us to write less and do more.

Example:

```js
1   const arr = [ 1, 2, 3, 4, 5 ]
2
3   console.log(arr) // [ 1, 2, 3, 4, 5 ]
4   console.log(...arr) // 1 2 3 4 5
5
6   const { name, ...other } = { name: 'fatfish', age: 100, luckyNumber: 6 }
7
8   console.log(name) // fatfish
9   console.log(other) // { age: 100, luckyNumber: 6 }
```

In this article, I want to teach you 7 spread operator tricks.

1. A better way to use the array's push() method

We all know that the push() method supports passing multiple indeterminate parameters.

Usually, we use the apply() method to fill the array, but that is not convenient after all.

```js
1   const arr = [ 'fatfish', 'medium' ]
2
3   arr.push.apply(arr, [ 'JavaScript', 'NodeJs' ])
4   console.log(arr) // ['fatfish', 'medium', 'JavaScript', 'NodeJs']
```

In fact, it doesn't need to be so troublesome. The spread operator can help us do this easily.

```
1   const arr = [ 'fatfish', 'medium' ]
2
3   arr.push(...[ 'JavaScript', 'NodeJs' ])
4
5   console.log(arr) // ['fatfish', 'medium', 'JavaScript', 'NodeJs']
```

2. Copy a new array

Copying an array is one of the most convenient functions of the spread operator.

```
1   const arr = [ 'fatfish', 'medium', 'JavaScript', 'NodeJs' ]
2   const copyArr1 = arr.slice(0) // [ 'fatfish', 'medium', 'JavaScript', 'NodeJs' ]
3   // A more convenient way
4   const copyArr2 = [ ...arr ] // [ 'fatfish', 'medium', 'JavaScript', 'NodeJs' ]
```

> **Tip** (Sourced from <u>here</u>): *The spread operator only copies the array itself to the new one, not the elements. This means that the copy is shallow, not deep.*

Example:

```
1   const arr = [ 'fatfish', 'medium', [ 'JavaScript', 'NodeJs' ] ]
2   const copyArr = [ ...arr ]
3   // If we change the third element, the "arr" will also be change.
4   copyArr[ 2 ][ 0 ] = 'FrontEnd'
5   console.log(copyArr) // [ 'fatfish', 'medium', [ 'FrontEnd', 'NodeJs' ] ]
6   // The arr also changed
7   console.log(arr) // [ 'fatfish', 'medium', [ 'FrontEnd', 'NodeJs' ] ]
```

3. Remove duplicate values from the array

Arrays of duplicate values can be removed through the set data structure and the spread operator.

```
1    const arr = [ 'fatfish', 'fatfish', 'medium', 'medium' ]
2    const uniqueArray = [ ...new Set(arr) ]
3
4    console.log(uniqueArray) // ['fatfish', 'medium']
```

4. Connect multiple arrays

Yes, we can use the spread operator to chain multiple arrays to get brand-new data.

```
1    const arr1 = [ 'fatfish', 'medium' ]
2    const arr2 = [ 'JavaScript', 'NodeJs' ]
3    const arr = [ ...arr1, ...arr2 ]
4
5    console.log(arr) // ['fatfish', 'medium', 'JavaScript', 'NodeJs']
```

5. Convert NodeList to a real array

Manipulating the DOM is what front-end developers do every day, such as using **document.querySelectorAll** to get a NodeList. We can convert it to an array with the spread operator. Then you can use forEach, map, etc.

Example:

```
1    // $divs is a NodeList
2    const $divs = document.querySelectorAll('div')
3    // $arrayDivs is An Array
4    const $arrayDivs = [ ...$divs ]
5    console.log(Array.isArray($divs), Array.isArray($arrayDivs)) // false true
```

Open in app ↗

Search Medium

```
 , _]
> const $divs = document.querySelectorAll('div')
<· undefined
> $divs
<·  NodeList(1295) [div#root, div.a.b.c, div.d.e.f.g.h.i.j.k, div, div.y.c, div.iu.iv.y, div.am.iw.ix.iy.iz.ja.jb
    , div.jc.jd.je.jf.jg.jh.ji.jj.jk, div.ha.jl.jm.jn.d, div.jo.jp.jq.am.bq.jb.ag.ah.aj.ai.jr.c, div.am.bq,
    div.js.y.jt, div, div.kd.ke.hb, div.pl.y, div.jq.y, div.cx.h.k.j.i.hz, div.hs.am.cy.jb.hv.ah.kp.c,
    div.kl.km.kn.ko, div.y, div.kr.y, div.am.cu, div.ho.la.am.ks.kt.ku.kv, div.y.kw.kx, div, div.cp, div.ke.ky.kx
    , div.ke.ky.bv.b.ec.eb, div.kr.y, div.am.cu, div.ho.la.am.ks.kt.ku.kv, div.y.kw.kx, div, div.cp, div.ke.ky.kx
    , div.ke.ky.bv.b.ec.eb, div.bv.b.oq.hg.ca.mt.uf.ug.gm.uh.ui.uj.jt.cp, div.kr.y, div.am.cu,
    div.ho.la.am.ks.kt.ku.kv, div.y.kw.kx, div, div.cp, div.ke.ky.kx, div.ke.ky.bv.b.ec.eb, div.kr.y, div.am.cu,
 ▶  div.ho.la.am.ks.kt.ku.kv, div.y.kw.kx, div, div.cp, div.ke.ky.kx, div.ke.ky.bv.b.ec.eb, div.lb.lc.y, div.kr.y
    , div.am.cu, div.ho.la.am.ks.kt.ku.kv, div.y.kw.kx, div, div.cp, div.ke.ky.kx, div.ke.ky.bv.b.ec.eb, div.mc.y
    , div, div.y, div.am.bq, div.po.pp, div.y.cu, div.ps.pi.y.pt.pu.mt.ah, div.ke.pq.pr.ky, div.ha.jl.jm.jn.d,
    div.y.ag.aj.ig.ai.jr.c, div.lk.jq.cu.ll, div.cx.am.jb, div.y.cu, div.ps.pi.y.qc.lc.mt.ah, div.nt.am.cy.c,
    div.y, div.am.an, div.au.ae.ao.ap.lu.lv.lw.lx.ly.lz.ma, div.ng.vu.vv.vw.vx.vy.y,
    div.vz.wa.wb.wc.wd.we.wf.wg.wh.wi.wj.wk.wl.wm.wn.wo.wp.wq.wr.ws.am, div.wt.wu.wv.ww.wx.am.bq.wy.lt,
    div.ek.j.i.d, div.y.cu, div.ps.pi.y.rq.rr.mt.ah, div.y, div.y, div.j.i.d, div.xp.j.i, div.am, div.xq.y,
    div.xp.h.k.d, div.am, div.xq.y, div.am.bq.ni.nj.nk.nl.nm, div.mc.x.np.nq.y, div.y, div.mc.hx.np.nq.y, div.y
    , _]
```

6. Destructuring

The spread operator is often used to destructure arrays and objects, check it out!

Destructure array:

Destructure object:

7. Convert string to array

Isn't it amazing that a string can be turned into an array like this?

Finally

Thanks for reading. I am looking forward to your following and reading more high-quality articles.

"I Lost a Job Opportunity Just Because of Promise.all"

An interview experience that made me so sad.

javascript.plainenglish.io

Interviewer: What Happened to "npm run xxx"?

A secret that most people don't know.

javascript.plainenglish.io

Interviewer: Can "x !== x" Return True in JavaScript?

Five magical JavaScript knowledge points you may not know!

javascript.plainenglish.io

It's 2022, Don't Abuse the Arrow Function Anymore

4 cases where you shouldn't use arrow functions.

javascript.plainenglish.io

More content at _PlainEnglish.io_.

Sign up for our _free weekly newsletter_. Follow us on _Twitter_, _LinkedIn_, _YouTube_, and _Discord_.

Build awareness and adoption for your tech startup with _Circuit_.

Front End Development   JavaScript   Programming   React   Vuejs