

ES6 Tips

Write Minimal ES6 Code

← Swipe



Joe Harrison
@frontendjoe

A large yellow square containing the letters 'ES' in a bold, black, sans-serif font.

ES

Boolean Casting

Today's recommended method
according to Airbnb's style guide 🤔

OLD

```
const age = Boolean(input.value)
```

NEW

```
const age = !!input.value
```

* I'm a bit undecided on this syntax,
I guess it's nice if you're familiar with it
and want to keep things minimal



Joe Harrison
@frontendjoe



Nullish Coalescing

Returns its right-hand side when its left-hand side operand is null or undefined

OLD

```
const addId = (user, id) => {  
  user.id =  
    id !== null && id !== undefined  
      ? id  
      : "Unknown"  
  return user  
}
```

NEW

```
const addId = (user, id) => {  
  user.id = id ?? "Unknown"  
  return user  
}
```



Joe Harrison
@frontendjoe



Destructuring Objects

Write less code by unpacking properties from objects into distinct variables

OLD

```
const save = params => {  
  saveData(  
    params.name,  
    params.email,  
    params.dob  
  )  
}
```

NEW

```
const save = ({name, email, dob}) => {  
  saveData(name, email, dob)  
}
```



Joe Harrison
@frontendjoe



Destructuring Arrays

Write less code by unpacking values from arrays into distinct variables

OLD

```
const data = [  
  "axios", "recharts",  
  "flocked", "flick"  
];  
  
const plugins = data[0], apps = data[1]
```

NEW

```
const data = [  
  "axios", "recharts",  
  "flocked", "flick"  
];  
  
const [plugins, apps] = data
```



Joe Harrison
@frontendjoe



Optional Chaining

Allows you to read the value of a deeply nested property without checking if it's a valid chain

OLD

```
const hasValidPostcode = u =>  
  u &&  
  u.address &&  
  u.address.postcode &&  
  u.address.postcode.valid
```

NEW

```
const hasValidPostcode = u =>  
  u?.address?.postcode?.valid
```



Joe Harrison
@frontendjoe



Spread Operator

Merge two objects into one using this cool syntax, also very clean when cloning objects

OLD

```
const details = { name: "Man Utd" }  
const stats = { games: 7, points: 21 }  
  
const team = Object.assign(  
  {},  
  details,  
  stats  
)
```

NEW

```
const details = { name: "Man Utd" }  
const stats = { games: 7, points: 21 }  
  
const team = {  
  ...details,  
  ...stats  
}
```



Joe Harrison
@frontendjoe



Default Parameters

Function parameters default to undefined, so it's useful to set a value for this eventuality

OLD

```
const createUser = (name, email) => {  
  const user = {  
    email,  
    name: name ?? "Unknown",  
  }  
  // create user  
}
```

NEW

```
const createUser = (  
  name = "Unknown",  
  email  
) => {  
  const user = { email, name }  
  // create user  
}
```



Joe Harrison
@frontendjoe



For (of)

Arguably the same amount of code required but for (of) is known to be 24% faster than forEach

OLD

```
const array = []
const fillArray = items => {
  items.forEach(i =>
    array.push(i)
  )
}
```

NEW

```
const array = []
const fillArray = items => {
  for (let i of items) {
    array.push(i)
  }
}
```



Joe Harrison
@frontendjoe

