ES6 Tips

Useful ES6 Array Methods

← Swipe

Includes

Mainly used on Strings, but can be useful when working with Arrays of primitive types

```
const results = ["Won", "Lost", "Won"]
```

OLD

```
const hasLost =
  results.indexOf("Lost") > -1
```

NEW

```
const hasLost =
  results.includes("Lost")
```

Some

Very similar to includes, but it allows you to check if an internal property meets a condition

```
const users = [
  { name: "Ted", experience: 5 },
  { name: "Bill", experience: 10 },
]
```

OLD

```
const hasExperience =
  users
    .filter(u => u.experience > 5)
    .length > 0
```

NEW

```
const hasExperience =
  users.some(u => u.experience > 5)
```

Find

Useful when you only need to find one item in an array, no filter is required these days

```
const users = [
  { name: "Ted", experience: 5 },
  { name: "Bill", experience: 10 },
]
```

OLD

```
const ted =
  users.filter(u => u.name === "Ted")[0]
```

NEW

```
const ted =
  users.find(u => u.name === "Ted")
```

Every

Useful if you need to check if all items meet a condition, before we used to use the filter method

```
const results = ["Lost", "Lost", "Lost"]
```

OLD

```
const hasLosingStreak =
  results.filter(r => r === "Lost").length
```

NEW

```
const hasLosingStreak =
  results.every(r => r === "Lost")
```

Find Index

In the past we had to use for loops to find the index of an item, not anymore with this method

```
const users = [
  { name: "Ted", experience: 5 },
  { name: "Bill", experience: 10 },
]
```

OLD

```
let indexOfTed
for (let [index, user] of users) {
  if (user.name === "Ted") {
    indexOfTed = index
    break
  }
}
```

NEW

```
const indexOfTed =
  users.findIndex(u => u.name === "Ted")
```

Pop

Useful if you need to remove the last item in an Array, it will also return that item if you need it

```
const fruits = ["Apple", "Potato"]
```

OLD

```
fruits.splice(fruits.length -1, 1)
```

NEW

```
fruits.pop()
```

Unshift

Useful if you need to add an item to the top of an Array, before we had to use the splice method

```
const fruits = ["Banana", "Cherry"]
```

OLD

```
fruits.splice(0, "Apple")
```

NEW

```
fruits.unshift("Apple")
```

Is Array

Check if a variable is of type Array with this handy method, the old way was a bit weird lol

```
const books = ["📕", "📙", "📗"];
```

OLD

```
Object
  .prototype
  .toString
  .call(books) === "[object Array]"
```

NEW

```
Array.isArray(books);
```