ES6

FOR ... OF ITERATORS

FOR ... OF

ARRAY

```
let iterable = [10, 20, 30];

for (const value of iterable) {
  console.log(value);
}
// 10
// 20
// 30
```

STRING

```
let iterable = "boo";

for (let value of iterable) {
  console.log(value);
}
// "b"
// "o"
// "o"
```

MAP

```
let iterable = new Map([["a", 1], ["b", 2], ["c", 3]]);

for (let entry of iterable) {
  console.log(entry);
}
// [a, 1]
// [b, 2]
// [c, 3]
```

SET

```
let iterable = new Set([1, 1, 2, 2, 3, 3]);

for (let value of iterable) {
  console.log(value);
}
// 1
// 2
// 3
```

DOM ELEMENTS

```
let articleParagraphs = document.querySelectorAll("article > p");

for (let paragraph of articleParagraphs) {
  paragraph.classList.add("read");
}
```

ITERABLE PROTOCOL

ALLOWS JAVASCRIPT OBJECT TO DEFINE OR CUSTOMIZE THEIR ITERATION BEHAIVOR.

```
var iterable = {
  [Symbol.iterator]() {
    return {
      i: 0,
      next() {
        if (this.i < 3) {
          return { value: this.i++, done: false };
        }
        return { value: undefined, done: true };
      }
    };
  }
};

for (var value of iterable) {
  console.log(value);
}
// 0
// 1
// 2
```

```
let fibonacci = {
    [Symbol.iterator]() {
        let pre = 0, cur = 1;
        return {
            next () {
                [ pre, cur ] = [ cur, pre + cur ];
                return { done: false, value: cur };
            }
        };
    }
}

for (let n of fibonacci) {
    if (n > 1000)
        break;
    console.log(n);
}
```

QUESTIONS?