Array Destructuring in ES6

- Destructuring arrays is a new feature in ES6.
- It makes possible to extract values from arrays and assign them to variables.
- Example:

```
let arr = [1, 2, 3];
let [a, b, c] = arr;
console.log(a, b, c); // 1 2 3
```

- We can also destructure arrays returning from a function.

```
function getArr() {
  return [1, 2, 3];
}
let [a, , c] = getArr();
```

- Notice that we left the second value empty, so it will be undefined.

- The destructuring syntax makes it easier to also swap values.

```
let a = 1;
let b = 2;
[a, b] = [b, a];
console.log(a, b); // 2 1
[a, b = 3] = [2];
console.log(a, b); // 2 3
```

Quick Quiz

- What is the output of the following code?

```
let names = ["John", "Jane", "Mary"];
let [Ann, Bob, Charlie] = names;
console.log(Ann, Bob, Charlie);
```

> John Jane Mary

Object Destructuring

- Destructuring objects is a new feature in ES6.
- It makes possible to extract values from objects and assign them to variables.

- Example:

```
let obj = {
  a: 1,
  b: 2,
  c: 3,
};
let { a, b, c } = obj;
console.log(a, b, c); // 1 2 3
```

- We can assign without declaration, but theres syntax to remember.

```
let a, b;
({ a, b } = { a: 1, b: 2 });
console.log(a, b); // 1 2
```

- The () with semi-colon `;` is a way to assign without declaration.