Symbols

ES6 introduced a new type of primitive, `Symbols`, let's learn what they are and how to use them

ES6 added a new type of primitive called **Symbols**. What are they? And what do they do?

The unique property of Symbols

Symbols are **always unique** and we can use them as identifiers for object properties.

Let's create a `Symbol` together:

```
const me = Symbol("Alberto");
console.log(me);
// Symbol(Alberto)
```

We said that they are always unique, let's try to create a new symbol with the same value and see what happens:

```
const me = Symbol("Alberto");
console.log(me);
// Symbol(Alberto)

const clone = Symbol("Alberto");
console.log(clone);
// Symbol(Alberto)

console.log(me == clone);
```

```
// false
console.log(me === clone);
// false
```

They both have the same value, but we will never have naming collisions with Symbols as they are always unique.

Identifiers for object properties

As we mentioned earlier, we can use them to create identifiers for object properties, so let’s see an example:

```
const office = {
  "Tom" : "CEO",
  "Mark": "CTO",
  "Mark": "CIO",
}

for (person in office){
  console.log(person);
}
// Tom
// Mark
```

Here we have our office object with three people, two of which share the same name. To avoid naming collisions we can use symbols.

```
const office = {
  [Symbol("Tom")] : "CEO",
  [Symbol("Mark")] : "CTO",
  [Symbol("Mark")] : "CIO",
}

for(person in office) {
  console.log(person);
}
// undefined
```

We got `undefined` when we tried to loop over the symbols because they are **not enumerable**, so we can't loop over them with a `for in`.

If we want to retrieve their object properties we can use `Object.getOwnPropertySymbols()`.

```
const office = {
  [Symbol("Tom")] : "CEO",
  [Symbol("Mark")] : "CTO",
  [Symbol("Mark")] : "CIO",
};

const symbols = Object.getOwnPropertySymbols(office);
console.log(symbols);
// 0: Symbol(Tom)
// 1: Symbol(Mark)
// 2: Symbol(Mark)
// length: 3
```

We retrieved the array, but to be able to access the properties we have to use `map`.

```
const symbols = Object.getOwnPropertySymbols(office);
const value = symbols.map(symbol => office[symbol]);
console.log(value);
// 0: "CEO"
// 1: "CTO"
// 2: "CIO"
// length: 3
```

Now we finally got the array containing all the values of our symbols.

Keep this all in mind as we move onto another quiz and coding challenge.