

Chapter 67: Symbols

Section 67.1: Basics of symbol primitive type

`Symbol` is a new primitive type in ES6. Symbols are used mainly as **property keys**, and one of its main characteristics is that they are *unique*, even if they have the same description. This means they will never have a name clash with any other property key that is a `symbol` or `string`.

```
const MY_PROP_KEY = Symbol();
const obj = {};

obj[MY_PROP_KEY] = "ABC";
console.log(obj[MY_PROP_KEY]);
```

In this example, the result of `console.log` would be `ABC`.

You can also have named Symbols like:

```
const APPLE = Symbol('Apple');
const BANANA = Symbol('Banana');
const GRAPE = Symbol('Grape');
```

Each of these values are unique and cannot be overridden.

Providing an optional parameter (*description*) when creating primitive symbols can be used for debugging but not to access the symbol itself (but see the `Symbol.for()` example for a way to register/lookup global shared symbols).

Section 67.2: Using `Symbol.for()` to create global, shared symbols

The `Symbol.for` method allows you to register and look up global symbols by name. The first time it is called with a given key, it creates a new symbol and adds it to the registry.

```
let a = Symbol.for('A');
```

The next time you call `Symbol.for('A')`, the *same symbol* will be returned instead of a new one (in contrast to `Symbol('A')` which would create a new, unique symbol that happens to have the same description).

```
a === Symbol.for('A') // true
```

but

```
a === Symbol('A') // false
```

Section 67.3: Converting a symbol into a string

Unlike most other JavaScript objects, symbols are not automatically converted into a string when performing concatenation.

```
let apple = Symbol('Apple') + ''; // throws TypeError!
```

Instead, they have to be explicitly converted into a string when necessary, (for example, to get a textual description

of the symbol that can be used in a debug message) using the `toString` method or the `String` constructor.

```
const APPLE = Symbol('Apple');  
let str1 = APPLE.toString(); // "Symbol(Apple)"  
let str2 = String(APPLE);    // "Symbol(Apple)"
```