Cheat Sheet – Symbols

What Symbols Are

Symbols are a new primitive type in ES6. Basically, a Symbol is a unique ID. However, you don’t see the ID (like 120) but you only got your symbol which represents the ID.

```
let symbol = Symbol(‘only for debugging, this is not the ID!’);
```

As a symbol represents a unique ID, each symbol is unique. This means, that the following comparison will resolve to false:

```
let symbol1 = Symbol(‘only for debugging, this is not the ID!’);
let symbol2 = Symbol(‘only for debugging, this is not the ID!’);
if (symbol1 == symbol2) { … } // false
```

The only exception make symbols created via Symbol.for():

```
let ageSymbol = Symbol.for('age');
```

These symbols are registered in a global symbol registry and therefore the following comparison will return true:

```
let ageSymbol1 = Symbol.for('age');
let ageSymbol2 = Symbol.for('age');
if (ageSymbol1 == ageSymbol2) { … } // true
```

Where would you use Symbols?

As they are unique symbols are useful as object keys.

```
let obj = {
    [symbol]: 22,
    [symbol2]: 'symbol assigned'
};
console.log(obj[symbol]); // prints 22
```

You can be sure that this fieldname hasn’t been taken yet.

Built-in Symbols

There are some built-in symbols you may utilize to overwrite default behaviors of JavaScript. This is also called meta-programming (i.e. changing parts of the language/ its behavior).

```
class Person {
}
let person = new Person();


Person.prototype[Symbol.toStringTag] = 'Person Class';
let person = new Person();
console.log(person); // prints [object Person Class]
```

More information may be found here: https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Symbol