

TypeScript 3.2 is now available. [Download \(/#download-links\)](#) our latest version today!

Documentation ▼

Symbols

Introduction

Starting with ECMAScript 2015, `symbol` is a primitive data type, just like `number` and `string`.

`symbol` values are created by calling the `Symbol` constructor.

```
let sym1 = Symbol();

let sym2 = Symbol("key"); // optional string key
```

Symbols are immutable, and unique.

```
let sym2 = Symbol("key");
let sym3 = Symbol("key");

sym2 === sym3; // false, symbols are unique
```

Just like strings, symbols can be used as keys for object properties.

```
let sym = Symbol();

let obj = {
  [sym]: "value"
};

console.log(obj[sym]); // "value"
```

Symbols can also be combined with computed property declarations to declare object properties and class members.

```
const getClassNameSymbol = Symbol();

class C {
  [getClassNameSymbol]() {
    return "C";
  }
}

let c = new C();
let className = c[getClassNameSymbol()]; // "C"
```

Well-known Symbols

In addition to user-defined symbols, there are well-known built-in symbols. Built-in symbols are used to represent internal language behaviors.

Here is a list of well-known symbols:

Symbol.hasInstance

A method that determines if a constructor object recognizes an object as one of the constructor's instances. Called by the semantics of the instanceof operator.

Symbol.isConcatSpreadable

A Boolean value indicating that an object should be flattened to its array elements by `Array.prototype.concat`.

Symbol.iterator

A method that returns the default iterator for an object. Called by the semantics of the for-of statement.

Symbol.match

A regular expression method that matches the regular expression against a string. Called by the `String.prototype.match` method.

Symbol.replace

A regular expression method that replaces matched substrings of a string. Called by the `String.prototype.replace` method.

Symbol.search

A regular expression method that returns the index within a string that matches the regular expression. Called by the `String.prototype.search` method.

Symbol.species

A function valued property that is the constructor function that is used to create derived objects.

Symbol.split

A regular expression method that splits a string at the indices that match the regular expression. Called by the `String.prototype.split` method.

Symbol.toPrimitive

A method that converts an object to a corresponding primitive value. Called by the `ToPrimitive` abstract operation.

Symbol.toStringTag

A String value that is used in the creation of the default string description of an object. Called by the built-in method `Object.prototype.toString`.

Symbol.unscopables

An Object whose own property names are property names that are excluded from the 'with' environment bindings of the associated objects.

