

Symbol、BigInt

JS原始类型 (3)



奇舞团 前端基础课



讲师 | 月影



Symbol

ES6及之后的版本中引入的新原始数据类型。Symbol可以创建唯一标识。

- 作为对象的key
- Symbol.for
- Symbol.keyFor

```
const id1 = Symbol('foo');
const id2 = Symbol('foo');

console.log(id1 === id2); // false

const foo = Symbol.for('foobar'),
      bar = Symbol.for('foobar');

console.log(foo === bar); // true

const foo = Symbol.for('foobar');

console.log(Symbol.keyFor(foo)); // foobar
```


私有属性

新的语言标准中private field有些不理想，所以也可以采用Symbol来定义私有属性。

```
const size = Symbol('size');

class Collection {
  constructor() {
    this[size] = 0;
  }

  add(item) {
    this[this[size]] = item;
    this[size]++;
  }

  static sizeOf(instance) {
    return instance[size];
  }
}
```

内置Symbol

ES6内置了一些有用的Symbol，可以用来控制对象的一些内部行为。

- Symbol.iterator
- Symbol.toPrimitive
- Symbol.toStringTag

```
class Path {
  constructor() {
    this._points = [];
  }
  add( ... points) {
    this._points.push( ... points);
  }
  *[Symbol.iterator]() {
    yield *this._points;
  }
  get length() {
    return this._points.length;
  }
}

const path = new Path();
path.add([1, 1], [1, 2], [2, 3]);
console.log(path.length);

for(let point of path){
  console.log(point); // [1, 1], [1, 2], [2, 3]
}
```


BigInt

BigInt是JavaScript新的原始类型，可以精确表示大于 $2^{53}-1$ 的整数。

- BigInt字面量：数字+n
- BigInt运算：不能与Number直接进行运算
- 显式类型转换：BigInt与Number相互转换

```
const a = Number.MAX_SAFE_INTEGER + 1;  
const b = a + 1;  
const c = BigInt(a) + 1n;
```

```
console.log(a, b, c);  
// 9007199254740992  
// 9007199254740992  
// 9007199254740993n
```

```
const d = Number.MAX_VALUE;  
const e = d * 2;  
const f = BigInt(d) * 2n;
```

```
console.log(d, e, f);  
// 1.7976931348623157e+308  
// Infinity  
// 3595386 ... 9716736n
```

BigInt 内置方法

BigInt提供了两个内置方法可以转换为width位的有符号或无符号BigInt值。

- BigInt以计算机补码表示负数
- BigInt.asIntN转换为对应的有符号整数
- BigInt.asUintN转换为对应的无符号整数

```
const a = -1n;  
const b = BigInt.asUintN(16, a);  
console.log(b); // 65535n  
  
const c = BigInt.asIntN(16, b - 1n);  
console.log(c); // -2n
```


总结

- Symbol 和 Symbol.for
- Symbol 实现私有属性
- 内置 Symbol
- BigInt 与 Number
- BigInt.asIntN 和 BigInt.asUintN

谢谢观看！

