

```
// 本节只讨论数据结构的可遍历性
// JavaScript 遍历相关的内容如下:
// for 语句
// for...in... 语句
// for...of... 语句
// in 操作符
// ... 扩展操作符
// Object.getOwnPropertyNames()
// Object.getOwnPropertySymbols()
// Object.keys()
// 以及 Array.prototype 上的一些数组的遍历方法
```

// 想了解 JavaScript 遍历, 要从三个角度来分析, 在此之前, 你需要掌握 property descriptor 的知识:

```
// 1.纯粹的逻辑遍历
```

```
let arr = [58, 59, 60, 61];
let arrayLike = {
  2: '222',
  6: '666',
  12: '1122',
  length: 8
}
```

```
for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
```

```
// 依次打印 58, 59, 60, 61
```

```
for (let i = 0; i < arrayLike.length; i++) {
  console.log(arrayLike[i]);
}
```

```
// 依次打印 undefined, 222, undefined, 666, undefined
```

```
// 这种遍历方式, 实际上是根据数据结构的特点和 for 语句配合, 这是最基本的遍历方式
```

```
// 能不能遍历, 怎么遍历, 都根据自己写的 for 语句逻辑有关
```

```
// 2.依赖 property descriptor 的遍历:
```

```
// 因为对象类型的数据结构特点, 无法使用 for 语句来进行遍历, 因此, JavaScript 提供了一种语法糖 for...in...
```

```
let s = Symbol();
let o = {
  foo: 'foo value',
  bar: 'bar value',
  [s]: 'Symbol value'
}
```

```
Object.defineProperty(o, 'baz', {
  value: 'baz value'
});
```

```
for (let key in o) {
  console.log(key);
}
```

```
// 依次打印 foo, bar
```

```
// 可以看出来, for...in... 语法只能遍历 enumerable 为 true 的非 Symbol 自有属性, 至于原型链上的属性将不会被遍历
```

```
// for...in... 语句是一种语法糖, 里面的 in 操作符单独也可以使用, 如下案例:
```

```
console.log('foo' in o); // true
console.log('bar' in o); // true
```

```
console.log('baz' in o); // true
console.log(s in o); // true
console.log('toString' in o); // true
// in 操作符是比较特殊，无论是否是自有属性，无论是否是可便利的，无论是否是 Symbol 类型，都可以被 in 操作符判断出来
// 属性的可见性对 for...in... 操作符和 in 操作符完全不同，虽然都有 in，并且都是作用在属性上的，但是实际上是并不相关的东西
// 新规范中提供的 Object.keys() 方法，与 for...in... 的特性非常契合，因此可以猜测，[Object.kyes()] 方法本就存在，并为 for...in... 语句提供服务，只在新规范中开放成了可见方法
// 下面把 for...in... 的语法糖打开，对于 Object.keys() 方法的实现，请看该方法的同名文档：
for (let i = 0; i < Object.keys(o).length; i++) {
    console.log(Object.keys(o)[i]);
}

// 3.依赖于 Symbol.iterator 遍历器接口的遍历
// ... 扩展运算符，for...of... 依赖的都是遍历器接口，至于能不能遍历或者如何遍历，完全依赖于遍历器接口的逻辑

// 其他：
// Object.getOwnPropertyNames() 用来遍历非 Symbol 的自有属性，忽略属性的 enumerable 值
// Object.getOwnPropertySymbols() 用来遍历 Symbol 属性，忽略属性的 enumerable 值
// Object.keys() 用来遍历 enumerable 值为 true 和非 Symbol 的自有属性
```