Array.prototype[@@iterator]()

@@iterator属性和 `values()` 属性的初始值均为同一个函数对象

语法

```
arr[Symbol.iterator]()
```

返回值

数组的 **iterator** 方法，默认情况下与 `values()` 返回值相同

示例

使用 `for...of` 循环进行迭代

```
var arr = ['w', 'y', 'k', 'o', 'p'];
// 您的浏览器必须支持for...of循环
// 以及let —— 将变量作用域限定在 for 循环中
for (let letter of arr) {
  console.log(letter);
}
```

另一种迭代方式

```
var arr = ['w', 'y', 'k', 'o', 'p'];
var eArr = arr[Symbol.iterator]();
console.log(eArr.next().value); // w
console.log(eArr.next().value); // y
console.log(eArr.next().value); // k
console.log(eArr.next().value); // o
console.log(eArr.next().value); // p
```

规范

规范名称	规范状态	备注
ECMAScript 2015 (6th Edition, ECMA-262) Array.prototype[@@iterator]()	ST Standard	首次定义
ECMAScript Latest Draft (ECMA-262) Array.prototype[@@iterator]()	LS Living Standard	

浏览器兼容性

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	38	36 (36) [1]	未实现	25	未实现

[1] From Gecko 17 (Firefox 17 / Thunderbird 17 / SeaMonkey 2.14) to Gecko 26 (Firefox 26 / Thunderbird 26 / SeaMonkey 2.23 / Firefox OS 1.2) the `iterator` property was used (bug 907077), and from Gecko 27 to Gecko 35 the `"@@iterator"` placeholder was used. In Gecko 36 (Firefox 36 / Thunderbird 36 / SeaMonkey 2.33), the `@@iterator` symbol got implemented (bug 918828).

相关链接

- `Array.prototype.keys()`
- `Array.prototype.entries()`
- `Array.prototype.forEach()`
- `Array.prototype.every()`
- `Array.prototype.some()`
- `Array.prototype.values()`