

Array.of()

将一系列值转换成数组

```
let items = new Array(2) ;  
console.log(items.length) ; // 2  
console.log(items[0]) ; // undefined  
console.log(items[1]) ;
```

当使用单个数值参数来调用 `Array` 构造器时，数组的长度属性会被设置为该参数。如果使用多个参数(无论是否为数值类型)来调用，这些参数也会成为目标数组的项。数组的这种行为既混乱又有风险，因为有时可能不会留意所传参数的类型。

```
let items = new Array(1, 2) ;  
console.log(items.length) ; // 2  
console.log(items[0]) ; // 1  
console.log(items[1]) ; // 2
```

`Array.of`基本上可以用来替代`Array()`或`new Array()`，并且不存在由于参数不同而导致的重载，而且他们的行为非常统一。

Array.from()

`Array.from`接受三个参数，但只有`input`是必须的：

input：你想要转换的类似数组对象和可遍历对象

map：类似于数组的`map`方法，用来对每个元素进行处理，将处理后的值放入返回的数组

context：绑定`map`中用到的`this`

只要是部署了`iterator`接口的数据结构，`Array.from`都能将其转为数组：

```
let div = document.querySelectorAll('div') ;  
let divArr = Array.from(div) ;  
divArr.forEach(function (item) {  
    console.log(item.textContent) ;  
});  
console.log(Array.from([1, 2, 3], function (item) { return item*2; })))
```