Array.prototype.reduceRight()

`reduceRight()` 方法接受一个函数作为累加器（accumulator）和数组的每个值（从右到左）将其减少为单个值。

> PS: 与 `Array.prototype.reduce()` 的执行方向相反

```
let flattened = [
    [0, 1],
    [2, 3],
    [4, 5]
].reduceRight((a, b) => {
    return a.concat(b);
}, []);

// flattened is [4, 5, 2, 3, 0, 1]
```

对于从左至右遍历的相似方法请参阅 `Array.prototype.reduce()`.

语法

```
arr.reduceRight(callback[, initialValue])
```

参数

`callback`

一个回调函数，用来操作数组中的每个元素，可接受四个参数：

`previousValue`

上一次调用回调的返回值，或提供的 `initialValue`

`currentValue`

当前被处理的元素

`index`

当前处理元素的索引

array
调用 reduce 的数组

initialValue
可作为第一次调用回调 callback 的第一个参数

返回值

执行之后的返回值

描述

reduceRight 为数组中每个元素调用一次 callback 回调函数，但是数组中被删除的索引或从未被赋值的索引会跳过。回调函数接受四个参数：初始值（或上次调用回调的返回值）、当前元素值、当前索引，以及调用 reduce 的数组。

可以像下面这样调用 reduceRight 的回调函数 callback：

```
array.reduceRight(function(previousValue, currentValue, index, array) {
    // ...
});
```

首次调用回调函数时， previousValue 和 currentValue 可以是两个值之一。如果调用 reduceRight 时提供了 initialValue 参数，则 previousValue 等于 initialValue， currentValue 等于数组中的最后一个值。如果没有提供 initialValue 参数，则 previousValue 等于数组最后一个值， currentValue 等于数组中倒数第二个值。

如果数组为空，且没有提供 initialValue 参数，将会抛出一个 TypeError 错误。如果数组只有一个元素且没有提供 initialValue 参数，或者提供了 initialValue 参数，但是数组为空将会直接返回数组中的那一个元素或 initialValue 参数，而不会调用 callback。

该函数的完整执行过程见下例：

```
[0, 1, 2, 3, 4].reduceRight(function(previousValue, currentValue, index,
    return previousValue + currentValue;
});
```

回调将会被调用四次，每次调用的参数及返回值如下：

	previousValue	currentValue	index	array	return value
first call	4	3	3	[0,1,2,3,4]	7
second call	7	2	2	[0,1,2,3,4]	9
third call	9	1	1	[0,1,2,3,4]	10
fourth call	10	0	0	[0,1,2,3,4]	10

`reduceRight` 返回值是最后一次调用回调的返回值（ `10`）。

如果提供了一个 `initialValue` 参数，则结果如下：

```
[0, 1, 2, 3, 4].reduceRight(function(previousValue, currentValue, index,
    return previousValue + currentValue;
}, 10);
```

	previousValue	currentValue	index	array	return value
first call	10	4	4	[0,1,2,3,4]	14
second call	14	3	3	[0,1,2,3,4]	17
third call	17	2	2	[0,1,2,3,4]	19
fourth call	19	1	1	[0,1,2,3,4]	20
fifth call	20	0	0	[0,1,2,3,4]	20

`reduceRight` 返回值为 20。

示例

例子：求一个数组中所有值的和

```
var total = [0, 1, 2, 3].reduceRight(function(a, b) {
    return a + b;
});
// total == 6
```

例子：扁平化（flatten）一个元素为数组的数组

```
var flattened = [[0, 1], [2, 3], [4, 5]].reduceRight(function(a, b) {
    return a.concat(b);
}, []);
// flattened is [4, 5, 2, 3, 0, 1]
```

例子：reduce 与 reduceRight 之间的区别

```
var a = ['1', '2', '3', '4', '5'];
var left  = a.reduce(function(prev, cur)      { return prev + cur; });
var right = a.reduceRight(function(prev, cur) { return prev + cur; });

console.log(left);  // "12345"
console.log(right); // "54321"
```

兼容性旧环境（Polyfill）

`reduceRight` 被添加到 ECMA-262 标准第 5 版，因此它在某些实现环境中可能不被支持。把下面的代码添加到脚本开头可以解决此问题，从而允许在那些没有原生支持 `reduceRight` 的实现环境中使用它。

```
// Production steps of ECMA-262, Edition 5, 15.4.4.22
// Reference: http://es5.github.io/#x15.4.4.22
if ('function' !== typeof Array.prototype.reduceRight) {
  Array.prototype.reduceRight = function(callback /*, initialValue*/) {
    'use strict';
    if (null === this || 'undefined' === typeof this) {
      throw new TypeError('Array.prototype.reduce called on null or undef
    }
    if ('function' !== typeof callback) {
      throw new TypeError(callback + ' is not a function');
    }
    var t = Object(this), len = t.length >>> 0, k = len - 1, value;
    if (arguments.length >= 2) {
      value = arguments[1];
    } else {
      while (k >= 0 && !(k in t)) {
        k--;
      }
      if (k < 0) {
```

```
        throw new TypeError('Reduce of empty array with no initial value
      }
      value = t[k--];
    }
    for (; k >= 0; k--) {
      if (k in t) {
        value = callback(value, t[k], k, t);
      }
    }
    return value;
  };
}
```

规范

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) Array.prototype.reduceRight	ST Standard	Initial definition. Implemented in JavaScript 1.8
ECMAScript 2015 (6th Edition, ECMA-262) Array.prototype.reduceRight	ST Standard	
ECMAScript Latest Draft (ECMA-262) Array.prototype.reduceRight	LS Living Standard	

浏览器兼容性

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes)	3.0 (1.9)	9	10.5	4.0

相关链接

- `Array.prototype.reduce()`