# Underscore关于集合,数组,对象的方法

# 参数类型参考

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **数据类型** | **数组类型** | **字符串类型** | **数字类型** | **对象类型** | **函数类型** | **元素选择器** | **未定义类型** | **空类型** | **无参数类型** | **事件类型** | **或者** | **所有参数** |
| **参数指向** | Arr/arr | Str/str | N/n | Obj/obj | Fn/fn | Ele/ele | underfind | null |  | event | / | param |

## 详细内容参考：https://www.cnblogs.com/zzsdream/p/6843864.html

## 一、关于筛选过滤

### 筛选特定对象数组，返回对象数组(查询含有该对象属性值的对象集合，数组返回)

如：\_.where(arr,obj)

### 过滤集合中符合条件的元素。返回匹配元素数组

如：\_.filter(arr,fn)

### 过滤集合中符合条件的元素。返回第一个元素

如：\_.find(arr,fn)

### 过滤集合中不符合条件的元素。返回匹配元素数组(和filter相反)

如：\_.reject(arr,fn)

### 提取一个集合（数组对象）里指定的属性值,返回数组

如：\_.pluck(arr,str)

### 提取数组对象某属性的最大值，或者取数组的最大值

如：\_.max(arr,fn)或者\_.max(arr)

### 提取数组对象某属性的最小值，或者取数组的最小值

如：\_.min(arr,fn)或者\_.max(arr)

### 提取集合里面第一个值

如：\_.first(arr/str)

### 提取集合里面最后一个值

如：\_.last(arr/str)

### 提取集合里面除了最后n个的集合，返回数组

如：\_.initial(arr/str,n)

### 提取集合里面除了前面n个的集合，返回数组

如：\_.rest(arr/str,n)

### 提取范围数集合，返回数组

如1、：\_.range(n1,n2)//得到的数组范围为n1--(n2-1)

例子：

console.log(\_.range(1,10));//[1, 2, 3, 4, 5, 6, 7, 8, 9]

如2、：\_.range(n1,n2,n3)//得到的数组范围为=>算法：n2>n1+(n3\*i),i是从零开始的正整数

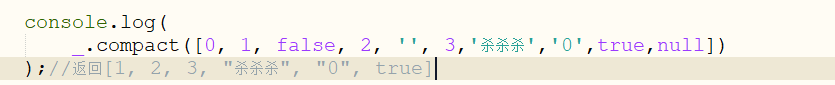
例子：

console.log(\_.range(0, 10, 2.5));//=>[0, 2.5, 5, 7.5]

### 13、 ****数组去除空值，0和null****

如：\_.compact(arr)

例子：



### 14、 ****数组降维（将一个嵌套多层的数组(嵌套可以是任何层数)转换为只有一层的数组）****

如：\_.**flatten**(arr/[arr])

### 15、 ****数组去掉指定元素(可去掉多个)****

如：\_.**without**(arr,n/str/bool...)

### 15、 ****数组合并去重 （并集，所有数组元素合并一个数组，并且去重）****

如：\_.union(arr,arr,arr...)

console.log(\_.union([1, 2, 3], [101, 2, 1, 10], [2, 1]));

=> [ 1, 2, 3, 101, 10 ]

### 16、 ****数组取相交（交集，取相同的元素，重新生成一个数组）****

如：\_.union(arr,arr,arr...)

console.log(\_.intersection([1, 2, 3], [101, 2, 1, 10], [2, 1]));

=> [ 1, 2 ]

### 16、 ****数组取差（差集，筛选第一个数组里面所有与其他数组里面不相等的元素，重新组成一个数组）****

如：\_.difference(arr,arr,arr...)

console.log(\_.difference([1, 2, 3, 4, 5], [5, 2, 10]));

=> [ 1, 3, 4 ]

### 17、 取唯一****（去重）****

如：\_.uniq(arr)

console.log(\_.uniq([1, 2, 1, 3, 1, 2,true,true]));

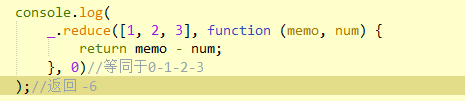
=> [1, 2, 3, true]

## 关于集合值运算

### 1、数组集合运算(第三参数和数组所有值递归运算)

如：\_.reduce(arr,fn)

例子：



## 关于数组排序

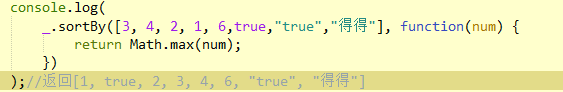
### 1、通过函数名调用函数运行:通过函数名调用内置函数运行,二维数组排序

如：\_.invoke([arr],’sort’)

### 2、通过函数名调用函数运行:通过函数名调用内置函数运行,二维数组排序

如：\_.sortBy(arr,fn)

例子：

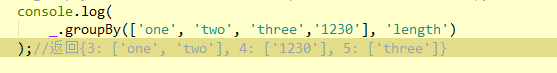


## 关于集合分组

### 1、把一个集合分组成多个集合(根据集合元素值的长度来分),返回对象

如：\_.groupBy(arr,’length’)

例子：

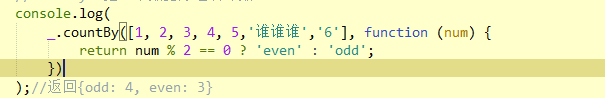


## 关于集合计数

### 1、'把一个数据分组后计数(分单双)，返回对象计数'

如：\_.countBy(arr,fn)

例子：



### 2、****得到集合中元素个数****

如：\_.**size**(arr/obj/str)

## 关于集合随机数

### 1、随机打乱一个数据内部顺序，返回数组

如：\_.shuffle(arr/str)

### 2、 ****返回范围内的随机数****

如：\_.random(n1,n2)

## 关于集合遍历

### 1、****对集合循环操作****

如：\_.**each**(arr,fn)

### 1、 ****对集合以map方式遍历，产生一个新数组****

如：\_.**map**(arr,fn)

## 关于数组索引

### 1、 排序****求索引区间值****

如：\_.sortedIndex(arr,n/str)

例子：

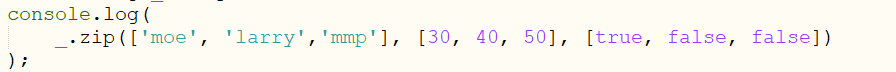


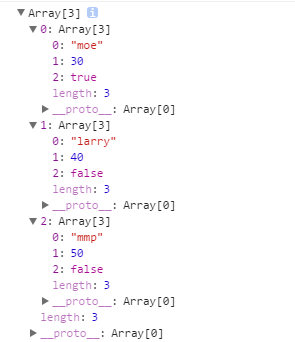
## 关于数组重组

### 1、 ****合并多个数组中的元素****

如：\_.zip(arr,arr,arr...)

例子：

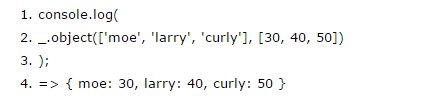




### 2、 ****把数组转换成对象****

如：\_.object(arr1,arr2)//arr1元素的值作为新对象的属性，arr2的值作为新对象的属性值

例子：



## 关于对象键值对操作

### 1、 取对象的属性重组为数组

如：\_.keys(obj)

例子：

var obj = {one: 1, two: 2, three: 3,four:true,five:false,six:'mmp',seven:null,eight:''};

console.log(\_.keys(obj));//["one", "two", "three", "four", "five", "six", "seven", "eight"]

### 2、 取对象的属性值重组为数组

如：\_.values(obj)

例子：

var obj = {one: 1, two: 2, three: 3,four:true,five:false,six:'mmp',seven:null,eight:''};

console.log(\_.values(obj));//[1, 2, 3, true, false, "mmp", null, ""]

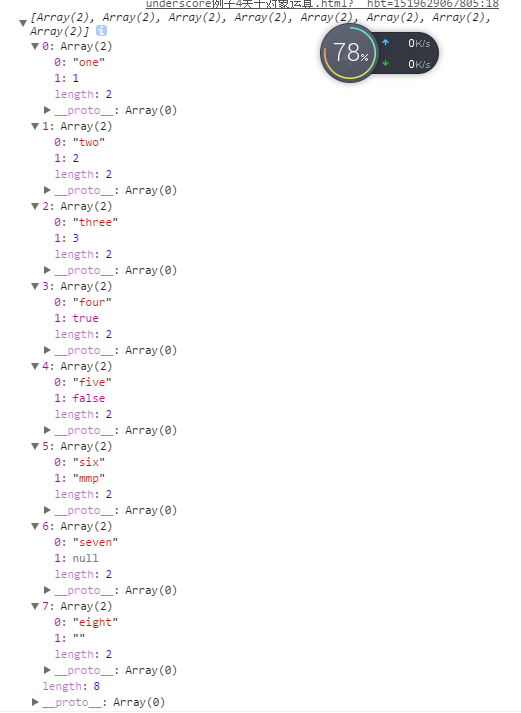
### 3、 ****把对象转换成[key,value]数组（二维数组）****

如：\_.pairs(obj)

例子：

var obj = {one: 1, two: 2, three: 3,four:true,five:false,six:'mmp',seven:null,eight:''};

console.log(\_.pairs(obj));



### 4、 对象键值互换

如：\_.invert(obj)

例子：

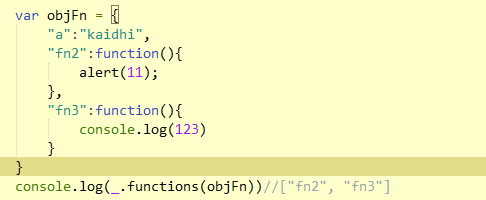
var obj = {one: 1, two: 2, three: 3,four:true,five:false,six:'mmp',seven:null,eight:''};

console.log(\_.invert(obj));//Object {1: "one", 2: "two", 3: "three", true: "four", false: "five", mmp: "six", null: "seven", "": "eight"}

### 5、 ****返回对象的所有方法名****

如：\_.functions(obj)

例子：



### 6、 多个****对象合并为新的对象，如果多个对象中属性相同，最后面的对象属性覆盖前面的（去重）****

如：\_.extend(obj,obj...)

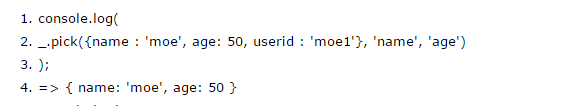
### 7、 多个****对象合并为新的对象，如果多个对象中属性相同，最前面的对象属性覆盖前面的（去重）****

如：\_.defaults(obj,obj...)

### 8、 ****返回一个对象的副本，保留指定的属性****

如：\_.pick(obj,str...)

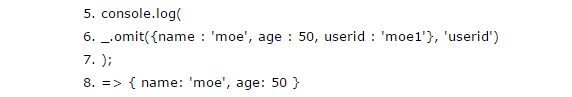
例子：



### ****返回一个对象的副本，去掉指定的属性****

如：\_.omit(obj,str...)

例子：



## 十一、关于判断

### 1、判断数组、对象，字符串元素是否有某个值，返回布尔值

如：\_.contains(arr/obj/str,n/str)

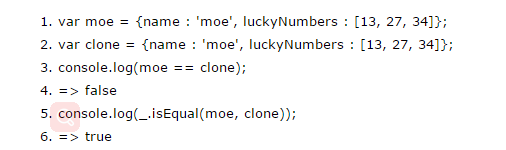
### 2、判断对象是否有某属性，返回布尔值

如：\_.has(obj,str)

### 3、****判断两个对象否是值相等****，返回布尔值

如：\_.**isEqual**(obj1,obj2)

例子：



### 4、判断是否为空（true:空字符串，空对象，空数组，0，n,str,bool,null,NaN）

如：\_.isEmpty(param)

### 5、判断是否为数字

如：\_.isFinite(param)

### 6、判断是否为数组，对象，函数是否有参数，函数，字符串，数字，布尔值，日期对象，非数字，空，未定义

1. console.log(\_.isArray([1,2,3]));
2. console.log(\_.isObject({}));
3. console.log((function(){ return \_.isArguments(arguments); })(1, 2, 3));
4. console.log(\_.isFunction(console.log));
5. console.log(\_.isString("moe"));
6. console.log(\_.isNumber(8.4 \* 5));
7. console.log(\_.isBoolean(true));
8. console.log(\_.isDate(new Date()));
9. console.log(\_.isNaN(NaN));
10. console.log(\_.isNull(null));
11. console.log(\_.isUndefined(undefined));