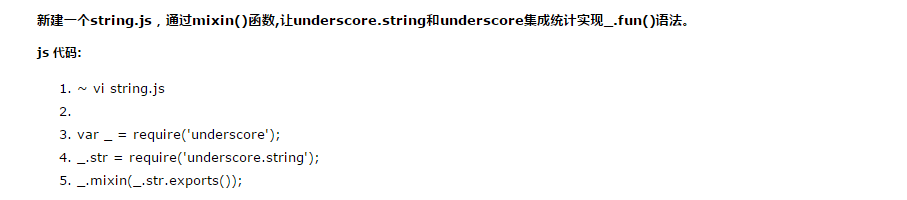
# Underscore关于函数

# 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>

## 首先引入underscore.string.js,



按照步骤才可以使用underscore.string.js

Html写法如下：



## 字符串数字格式化

### 转换任意位数浮点数

如：\_.numberFormat(n1, n2)



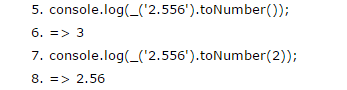
### 转换任意位数浮点数

如：\_.numberFormat(n1, n2,str1,str2)(参数1指的是数字，参数2指的是保留的小数位，参数3指的是小数点【省略默认小数点，可以改其他】，参数4指的是整数分割逗号【省略默认逗号，可改其他】)



### 四舍五入为整数或者为指定几位小数

如：\_.toNumber(n)(参数n存在，则四舍五入为n位小数，不存在，则四舍五入为整数)



## 字符串基础操作

### 1、匹配两个字符串集合，返回其不同的个数

如：\_.levenshtein(str1, str2)



### 2、首字母大写，强制转换首字母大写，其他字符串字母小写

如：\_.capitalize(str)//首字母大写

如：\_.capitalize(swapCase)//强制转换首字母大写，其他字符串字母小写

### 3、字符串集合截取为任意部分，返回数组

如：\_.chop(str，n)



### 4、字符串清除首尾空格，中间若有空格，则只是保留一个空格

如：\_.clean(str)

console.log(\_.clean(" 1 foo bar "));//1 foo bar

### 5、字符串转换数组

如：\_.chars(str)



### 6、字符串插入另外的字符串中

如：\_(str).insert(n,str)

console.log(\_('Hello ').insert(4, 'world'));//Hellworldo

### 7、带有‘\n’转义字符的字符串转化为数组

如：\_.lines(str)

console.log(\_.lines("Hello\nWorld\n2"));//["Hello", "World", "2"]

### 8、判断字符串集合后面和前面是否有某字符串

如判断后面：\_(str1).endsWith(str2)

console.log(\_("image.gif").endsWith("f"));//true

如判断前面：\_(str1).startsWith(str2)

console.log(\_("image.gif").startsWith("image"));//true

### 9、返回该字符串编码的下一个

如：\_(str).succ()

console.log(\_('a').succ());//b

## 字符串变换

### 1、字符串复制翻倍

带分割字符串的，如：\_.repeat(str1,n,str2)

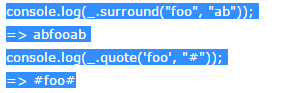
console.log(\_.repeat("foo", 3,"-"));//foo-foo-foo

不带分割字符串的，如：\_.repeat(str1,n)

console.log(\_.repeat("foo", 2));//foofoo

### 2、字符串前后缀插入指定字符串

如：\_.surround(str1,str2)或者\_.quote(str1,str2)



### 3、清除字符串的前后缀

不加额外参数，默认清除字符串前后双引号，如：\_.unquote(str)

console.log(\_.unquote('"foo"'));//foo

加额外参数，清除字符串前后指定的参数字符串，如：\_.unquote(str1,str2)或者\_.trim(str1,str2)

console.log(\_.unquote("9h9", "9"));//h



### 字符串去掉声调和上逗号，空格转换”-”

如：\_.slugify(str)

console.log(\_.slugify("Un éléphant à l'orée du bois"));//un-elephant-a-l-oree-du-bois

## 字符串替换截断

### 1、字符串替换截断

如：\_(str1).splice(n1,n2,str2)



### 2、清除字符串的前后空格

如：\_.trim(str)



### 3、指定字符串集合部分内容转换为省略号

如：\_(str1).truncate(n)

console.log(\_('Hello world').truncate(5));//Hello...

根据索引数，指定逗号后面的字符串转化为省略号，如：\_(str1).prune(n)

console.log(\_('Hello, world').prune(10));//Hello...

console.log(\_('He,llo, world').prune(10));//He,llo...

### 4、字符串转数组

不加参数，默认根据空格截取转数组，如：\_.words(str)



加参数，指定匹配字符截取转数组，如：\_.words(str1,/str2/)



### 5、字符串占位符

右占位，如：\_.rpad(str1/n1,n2,str2)(参数1指的是要被修改的字符串或者数字，参数2指的是字符串最终总共长度，参数3,指的是占位符)

console.log(\_.rpad(1, 8, '#'));//1#######

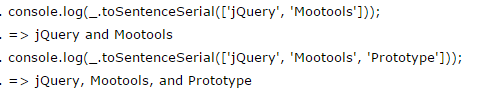
左占位，如：\_.lpad(str1/n1,n2,str2)(参数1指的是要被修改的字符串，参数2指的是字符串最终总共长度，参数3,指的是占位符)

console.log(\_.lpad("7577", 11, '\*'));//\*\*\*\*\*\*\*7577

## 字符串予语义处理

### 1、数组转化字符串，并且形成并列的语义关系

如：\_.toSentenceSerial(arr)



### 2、字符串所有单词首字母大写

a、根据空格截取，如：\_(str).titleize()



b、根据“-”符号截取，如：\_(str).camelize()



返转为小写的方法，如：\_(str).dasherize()



c、根据“\_”符号截取，如：\_(str).classify()



返转为小写的方法，如：\_(str).underscored()



d、去除“-”，“\_”,首单词首字母大写

