

Extractor使用方法:

三种基本的单一选择器:

class选择器

```
$( ".className" ,htmlContent);
```

例如: `$( ".infolist_tab" ,htmlContent);`

id选择器

```
$( "#tagId" ,htmlContent);
```

例如: `$( "#title" ,htmlContent);`

标签选择器

```
$( "tagName" ,htmlContent);
```

例如: `$( "table" ,htmlContent);`

标签选择器附加属性筛选条件

```
$( "tagName[name=value][name=value]..." ,htmlContent);
```

例如: `$( "table[cellspacing=0][class=infolist_tab]" ,htmlContent);` //可以无限多的标签属性限制

但是标签属性限制还可以加上附加条件

<code>name^=value</code>	value位于此属性的值的起始
<code>name*=value</code>	value包含在此属性值中
<code>name\$=value</code>	value位于此属性的值的末尾

例如: `$( "table[cellspacing^=0][class*=infolist_tab]" ,htmlContent);` //可以无限多的标签属性限制

多(两个)选择器控制(这两个选择器可以是任何一种上面的单一选择器)

子孙多选择器:

```
$( "selector1 selector2" ,htmlContent);
```

例如: `$( "table[class=infolist_tab] tr" ,htmlContent);` 选取第一个选择器符合的全部标签后,依次查找这些标签的子孙标签,选出符合第二个选择器的标签,并返回他们。

```
$( ".infolist_tab tr" ,htmlContent);  
$( ".infolist_tab #title" ,htmlContent);
```

直接儿子多选择器：

`$( "selector1 > selector2" ,htmlContent);` //只遍历直接子标签，不遍历孙子等更多辈分的子标签

例如：`$( "table[class=infolist_tab] > tr" ,htmlContent);` 选取第一个选择器符合的全部标签后，依次查找这些标签的子孙标签，选出符合第二个选择器的标签，并返回他们。

筛选器控制：

支持以下筛选器：

列出支持的filter

`eq(index)` 只返回下标为...index的，index为Integer类型，从0开始

`lt(index)` 只返回下标小于...index的，同上

`gt(index)` 只返回下标大于...index的，同上

`contains(text)` 只返回text包含...text的，String类型

`empty` 只返回html为空的

`even` 只返回下标为偶数的,0,2,4,6,8,...

`first` 只返回第一个

`last` 只返回第二个

筛选器以这种方式附加在任何单一选择器上：

例如：`$( "table:eq(2)" ,htmlContent);`

`$( "selector1:even selector2:last" ,htmlContent);`