

# 表达式：BybSelector

Beyebe项目·小蜜蜂

## 表达式：BybSelector

Beyebe项目·小蜜蜂

最简形态

执行顺序

CSS选择器

属性名

正则表达式的匹配与替换

执行顺序

RegexMatch与RegexReplace

IF检查

最简IF语法

执行顺序

IF检查の反选

IF检查の或

IF检查の与

范围选择

单一序号选择

全部合并

有始有终

有始无终

无始有终

单个BybSelector表达式的完全形态

执行顺序

多个BybSelector表达式の或

多个BybSelector表达式の与

---

## 最简形态

本表达式除了最简形态下的两个构成部分必不可缺外，其它所有构成部分均为可选项，仅在需要的时候才编写。

最简形态语法：

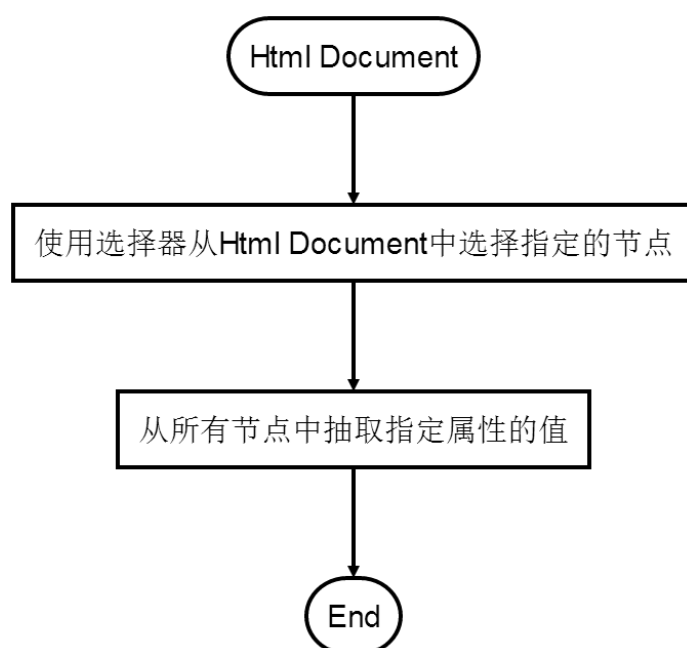
CSS选择器.属性名

最简形态分为“CSS选择器”和“属性名”两部分，使用英文句号连接。

**Demo：**

```
1. <a href="http://xxx.com">some text</a>
2.
3. 输入：
4.     a.href
5.
6. 输出：
7.     http://xxx.com
```

## 执行顺序



---

## CSS选择器

该部分满足[W3C Selectors Level 3](#) 标准，可直接参考该标准。

---

## 属性名

属性名分为“节点属性”和“对象属性”。

示例html节点：

```
1. <a href="http://xxx.com" data-from="step" id="link">some text</a>
```

### 节点属性

节点属性是指html节点内的属性，本表达式支持节点中所有的节点属性。在示例html节点中，该a节点包含如下节点属性：

1. href
2. data-from
3. id

### 对象属性

对象属性是指，在DOM ( Document Object Model ) 场景下，Document Element对象中的属性，本语法只支持两种对象属性：

1. text，对应element.innerText。
  2. html，对应element.outerHTML。
- 

## 正则表达式的匹配与替换

### 仅正则匹配：

## CSS选择器.属性名/RegexMatch

### Demo :

```
1. <a href="http://xxx.com?itemid=111&id=222">some text</a>
2.
3. 输入 :
4.     a.href/(?<=\bid=)\d+
5.
6. 输出 :
7.     222
```

使用正则表达式对抽取到的属性做正则匹配，最终返回正则表达式匹配到的内容。

在最简语法的基础上，使用 “/” 连接RegexMatch部分。

正则替换：

## CSS选择器.属性名/RegexMatch→RegexReplace

使用正则表达式执行文本的替换，最终返回正则表达式处理过的内容。

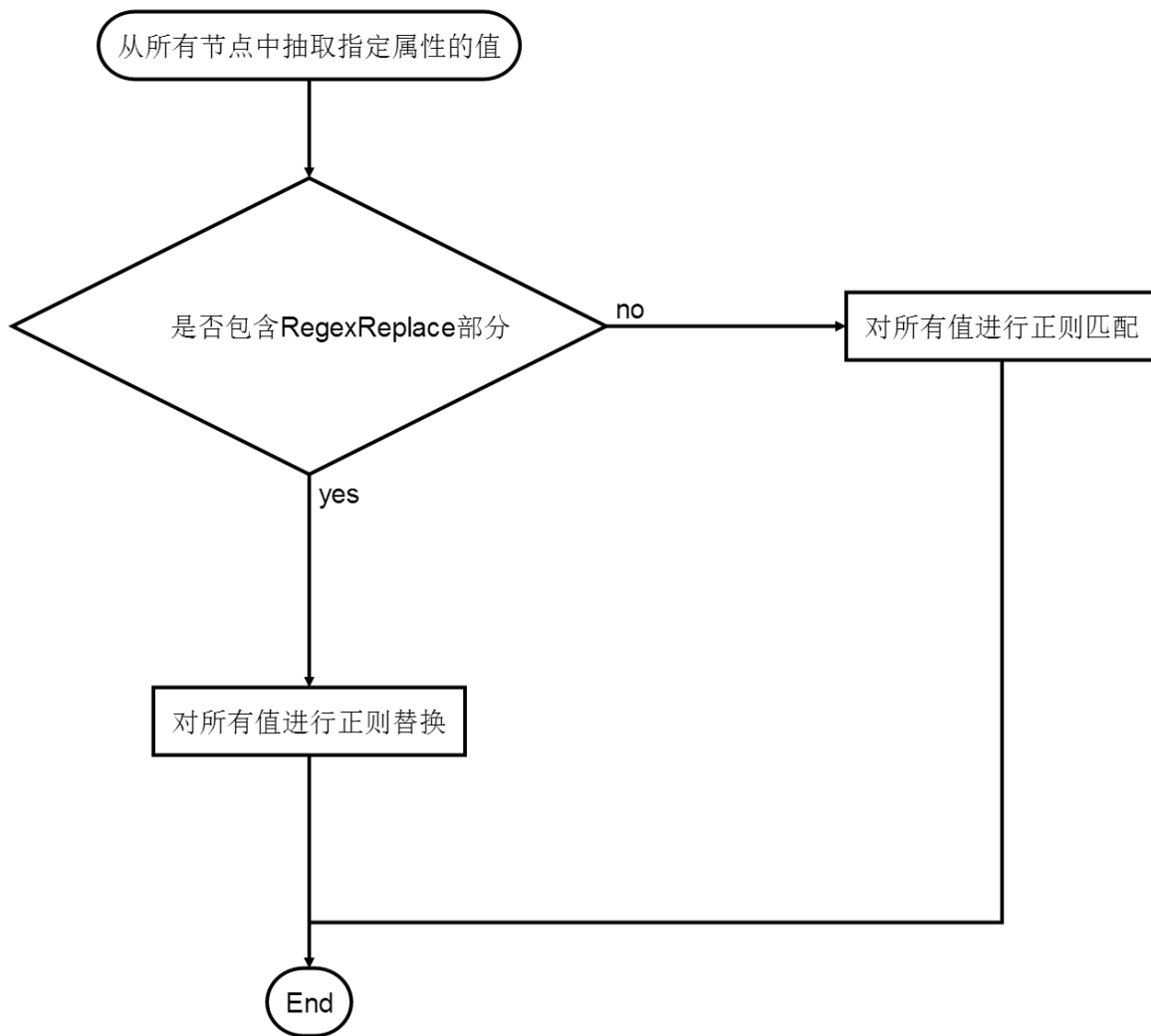
RegexReplace使用 “→” 连接在RegexMatch后面。

### Demo :

```
1. <a href="http://xxx.com?itemid=111&id=222">some text</a>
2.
3. 输入 :
4.     body a.href/(?<=\bid=)\d+→http://item.yyy.com/$1.html
5.
6. 输出 :
7.     http://item.yyy.com/222.html
```

---

## 执行顺序



---

## RegexMatch与RegexReplace

该正则表达式的`match`和`replace`均为标准微软正则表达式语法，可参照[MSDN](#)。

---

## IF检查

### 最简IF语法

`["subString"? "valueTrue": "valueFalse"]` CSS选择器.属性名

IF检查用于对最后抽取到的数据做检查，并根据检查结果对数据做修改。

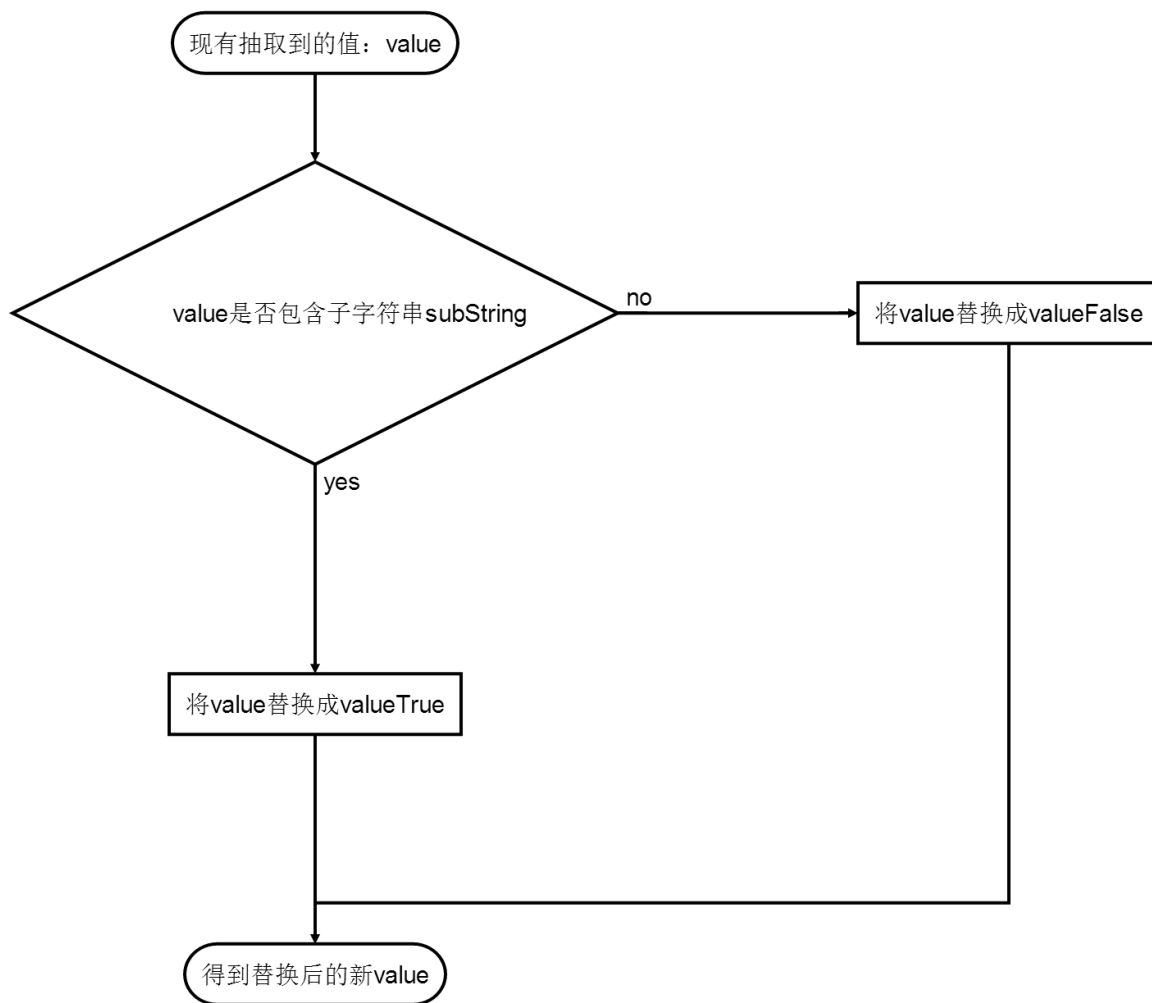
IF表达式的执行优先级低于正则处理，也就是说，如果表达式中存在正则处理，那么IF检查将是对正则处理过的数据进行检查。

### Demo：

```
1. <a href="http://jd.com">京东自营</a>
2.
3. 输入：
4.     ["自营"? "true": "false"]a.text
5.
6. 输出：
7.     true
```

---

## 执行顺序



## IF检查の反选

[!"subString"? "valueTrue": "valueFalse"]CSS选择器.属性名

IF检查支持反选检查，即从之前的检查是否包含变成检查是否不包含，实现方式是在subString前添加英文感叹号“!”。

**Demo :**

1. `<a href="http://jd.com">京东自营</a>`
- 2.

```
3.  输入 :
4.      ["!自营"?true:"false"]a.text
5.
6.  输出 :
7.      false
```

## IF检查の或

`["subString1" | "subString2" | "subString3" ... ?"valueTrue":"valueFalse"]`CSS选择器.属性名

当value中包含任意subString即返回valueTrue，否则返回valueFalse。

### Demo：

```
1.  <a href="http://jd.com">本商品由 海外购 提供售后服务</a>
2.  <a href="http://jd.com">本商品由 美的家电 提供售后服务</a>
3.  <a href="http://jd.com">本商品由 京东超市 提供售后服务</a>
4.
5.  输入 :
6.      ["海外购","京东超市"?true:"false"]a.text
7.
8.  输出 :
9.      ["true","false","true"]
```

## IF检查の与

`["subString1" & "subString2" & "subString3" ... ?"valueTrue":"valueFalse"]`CSS选择器.属性名

当value中如果包含所有subString即返回valueTrue，否则返回valueFalse。

IF检查支持对多个subString做“或”和“与”的多条件检查，但是或模式和与模式**不能混**



用，但是或模式和与模式中均能任意使用反选。

## Demo：

```
1. <a href="http://jd.com">本商品由 海外购 提供售后服务</a>
2. <a href="http://jd.com">本商品由 美的家电 提供售后服务</a>
3. <a href="http://jd.com">本商品由 京东超市 发货，并由 海外购 提供售后服务</a>
4.
5. 输入：
6.     ["海外购"&"京东超市"? "true": "false"]a.text
7.
8. 输出：
9.     ["false", "false", "true"]
```

---

## 范围选择

由于选择器能够选择到多个节点，所以最后得到的属性的值被存放在数组对象中。

本表达式支持通过序号的设定，保留指定范围的数组元素。

范围选择的执行优先级最低，低于正则处理和IF检查。

**注意：这里的数组第一个元素序号为1，不为0。**

---

## 单一序号选择

CSS选择器.属性名[n]

仅保留数组中第n位元素。

注意：单一序号选择允许n为负数，当n为负数时，n会被修正为“数组长度 + n”，即仅保留数组中倒数第n（绝对值）的元素。

## Demo :

```
1.  <ul>
2.    <li>首页</li>
3.    <li>家用电器</li>
4.    <li>空调</li>
5.    <li>壁挂式</li>
6.  </ul>
7.
8.  输入 :
9.    ul li.text[2]
10.
11.  输出 :
12.    家用电器
13.
14.  -----
15.
16.  输入 :
17.    ul li.text[-2]
18.
19.  输出 :
20.    空调
```

## 全部合并

CSS选择器.属性名[0]

当指定的序号为0时，会返回一个所有元素的合并值，各元素之间以中文分号“；”连接。

## Demo :

```
1.  <ul>
2.    <li>首页</li>
3.    <li>家用电器</li>
4.    <li>空调</li>
5.    <li>壁挂式</li>
6.  </ul>
7.
```

```
8.  输入：
9.      ul li.text[0]
10.
11.  输出：
12.      首页；家用电器；空调；壁挂式
```

---

## 有始有终

### CSS选择器.属性名[from,to]

保留数组中序号大于等于from，小于等于to的元素。

- \* 当from小于1时，from会被修正为1。
- \* 当from大于数组长度时，from会被修正为数组长度。
- \* 当to小于0时，to会被修正为“数组长度 + to”。
- \* 当to大于数组长度时，to会被修正为数组长度。

tips：由上可知，可以通过对to设置负数来实现选择从from到倒数第to（绝对值）个元素的目的。

### Demo：

```
1.  <ul>
2.      <li>首页</li>
3.      <li>家用电器</li>
4.      <li>空调</li>
5.      <li>壁挂式</li>
6.  </ul>
7.
8.  输入：
9.      ul li.text[2,3]
10.
11.  输出：
12.      ["家用电器", "空调"]
13.
14.  -----
15.
```

```
16.  输入 :
17.      ul li.text[2,-2]
18.
19.  输出 :
20.      ["家用电器","空调"]
```

---

## 有始无终

CSS选择器.属性名[from,]

当不填写to时，to会被修正为数组长度，也就是会选择到最后一个元素位置。

**Demo :**

```
1.  <ul>
2.      <li>首页</li>
3.      <li>家用电器</li>
4.      <li>空调</li>
5.      <li>壁挂式</li>
6. </ul>
7.
8.  输入 :
9.      ul li.text[2,]
10.
11.  输出 :
12.      ["家用电器","空调","壁挂式"]
```

---

## 无始有终

CSS选择器.属性名[,to]

当不填写from时，from会被修正为1。

**Demo :**

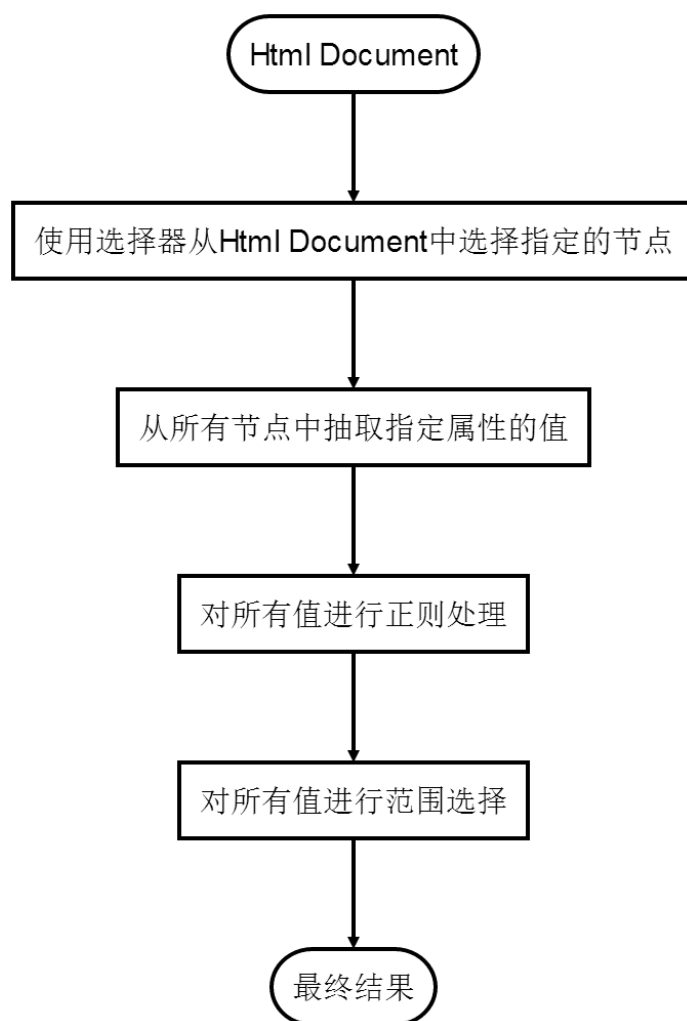
```
1. <ul>
2.   <li>首页</li>
3.   <li>家用电器</li>
4.   <li>空调</li>
5.   <li>壁挂式</li>
6. </ul>
7.
8. 输入：
9.     ul li.text[,2]
10.
11. 输出：
12.     ["首页", "家用电器"]
13.
14. -----
15.
16. 输入：
17.     ul li.text[,-2]
18.
19. 输出：
20.     ["首页", "家用电器", "空调"]
```

---

## 单个BybSelector表达式的完全形态

```
["subString"? "valueTrue": "valueFalse"]CSS选择器.属性名
[from,to]/RegexMatch→RegexReplace
```

### 执行顺序



## 多个BybSelector表达式的或

BybSelector1 || BybSelector2 || BybSelector3 ...

当多个BybSelector表达式以“||”连接时，会优先使用BybSelector1进行数据抽取，当BybSelector1抽取不到数据时，会尝试使用BybSelector2进行数据抽取，直到取到任意多个数据或者所有BybSelector用完。

### Demo :

```
1. <div>
2.   <p>原价：<span id="originalPrice">99.8</span></p>
3.   <p>现价：<span id="currentPrice">88.8</span></p>
```

```
4.     </div>
5.
6.     输入：
7.         #originalPrice.text || #currentPrice.text
8.
9.     输出：
10.        99.8
```

---

## 多个BybSelector表达式の与

```
BybSelector1 && BybSelector2 && BybSelector3 ...
```

当多个BybSelector表达式以“&&”连接时，会依次使用所有BybSelector进行数据抽取，并将所有结果依次拼接，最终返回所有BybSelector抽取结果的合集。

注意：BybSelector表达式的或模式和与模式**不能混用**。

### Demo：

```
1.     <div>
2.         <p>原价：<span id="originalPrice">99.8</span></p>
3.         <p>现价：<span id="currentPrice">88.8</span></p>
4.     </div>
5.
6.     输入：
7.         #originalPrice.text && #currentPrice.text
8.
9.     输出：
10.        ["99.8", "88.8"]
```