XPath 与 CSS Selector 的差别（原作者义甬君，有删改）

上一篇文章中总结了 web 自动化测试中用 xpath 来进行元素定位的方法，那么 CSS Selector 又是什么鬼？
其实，对于使用者来讲，哪个用的习惯就选哪个。总的来说，XPath 更强大，而 CSS 选择器通常语法比较简洁，运行速度更快些。

Target	CSS 3	XPath
所有元素	*	//*
所有的 P 元素	p	//p
所有的 p 元素的子元素	p > *	//p/*
根据 ID 获取元素	/转义/#id	//*[@id= ‘id’]
根据 Class 获取元素	.class	//*[contains(@class, ‘class’)]
拥有某个属性的元素	*[title]	//*[@title]
所有 P 元素的第一个子元素	p > *:first-child	//p/*[0]
所有拥有子元素 a 的 P 元素	无法实现	//p[a]
下一个兄弟元素	P + *	//p/following-sibling::*[0]

由于 Selenium 使用 **xpath 定位时采用遍历页面**的方式，在性能上采用 CSS 选择器的方式更优。Xpath 虽然性能指标较差，但是在浏览器中有比较好的插件支持，定位元素比较方便，对于性能要求严格的场景，可考虑通过 xpath 改写 css 的方式进行替换，比较以下 2 种方式：

```
XPATH 地址: id('index')/div[2]/div/ul/div/li[2]/a

CSS 选择器地址: #index div:nth-of-type(2) div ul li:nth-of-type(2) a

或者： #index>div:nth-of-type(2)>div>ul>li:nth-of-type(2)>a
```

实战积累：

- “>”表示直接子节点，“ ”空格表示任意子节点

```
css=div.resource-list>div:nth-child(1)>div:nth-child(1) div.progress
```

- 非隐藏

```
css=div:not(.hidden) div.resource-buttons
```

- 部分匹配

```
css=.dijitReset.dijitInline.dijitButtonText:contains("Join Event")

css=a[href*='/scheduledDeployment/45d67'] 包含了

css=a[href^='/scheduledDeployment/45d67'] 以 xxx 开头的
```

```
css=a[href$='/scheduledDeployment/45d67'] 以 xxx 结尾的
```

- 如果 class 里带的空格，用.来代替空格如：

```
<button class="x-btn-text module_picker_icon">...

可以这样写： css=button.x-btn-text.module_picker_icon
```

- 子节点和兄弟节点

```
div+p: 紧接在 <div> 元素之后的所有 <p> 元素 p~ul: 前面有 <p> 元素的每个 <ul> 元素 p:nth-of-type(2): 第二个 <p> 元素 p:nth-last-child(2): 父元素的倒数第二个 <p> 子元素
```

番外篇：为什么要自己写 xpath 或 css path

有人说，现在有了 Selenium IDE 自动录制脚本，我们根本不需要会 xpath 和 css，但我要说的是虽然 Selenium IDE 会帮助自动识别元素，但其录制的脚本非常不稳定，回放失败率极高：

以 dojo 框架实现的动态页面为例，主要问题：

1. dojo 框架所给出的各个控件并不会像传统的 html 标签一样，一旦代码书写完成就有固定的 html id 和 html 结构，dojo 会自动为每个所使用的控件创建一系列节点，并随机给每个控件一个"控件类别+顺序号"的 html id，这样就给书写 selenium 用例带来了麻烦.
2. dojo 框架提供的控件使用的事件机制无法用普通的 Selenium IDE 录制来完成，必须给以加工才可以使用，否则无法触发实际控件的动作。
3. dojo 作为一个 ajax 工具包，当然使用了 ajax 框架，所以 web 的行为具有异步特征，在进行事件的模拟（类似单击下拉框进行选择时）由于一般此种控件的数据源会在服务器处获得由此会产生一个 ajax 调用，在此时直接执行单击动作会无法找到相应节点（因为数据还木有返回，节点当然就无法生成），从而产生一个 element not found 的错误而导致测试出错.