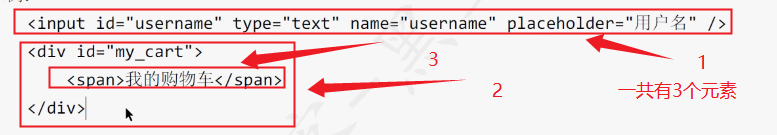
**如何进行元素定位**

元素：由标签头+标签尾+标签头和标签尾包括的文本内容



元素信息：指元素的**标签**以及元素的**属性**

元素的层级结构：指**元素之间相互嵌套**的层级结构

元素定位：是通过**元素的信息**或**元素的层级结构**来进行定位

**浏览器开发者工具介绍**

浏览器开发者主要用来**查看元素的信息**，同时也可以查看**接口的相关信息**

浏览器开发者不需要安装，浏览器自带

浏览器开发者工具的启动：

1）直接按F12,不区分浏览器

2）通过右键的方式来启动浏览器开发者工具

**元素定位（8种）**

1. id定位
2. name定位
3. class\_name定位
4. tag\_name定位
5. link\_text定位
6. partail\_link\_text定位
7. xpath定位
8. css定位

**ID定位**

通过元素的id属性值来进行元素定位（在html标准规范中id值是唯一的）

定位方法：find\_element\_by\_id(id)#id参数表示的是id的属性值

注意： 元素要有id属性

**name定位**

通过元素name属性值进行元素定位，name属性值在HTML页面中是可以重复的

说明：元素要有name属性

定位方法：find\_element\_by\_name(name)#name参数表示的是name的属性值

**Class\_name定位**

通过元素的class属性值进行元素定位，class属性值是可重复的

说明：元素必须要有class属性

定位方法：find\_element\_by\_class\_name(class\_name)#class\_name参数表示的是class的**其中一个属性值**

**tag\_name**

通过元素的标签名称进行定位，在同一个HTML页面当中，相同标签元素会有很多

这种定位元素的方式不建议大家在工作当中使用

定位方法：find\_element\_by\_tag\_name(tag\_name)#tag\_name表示的是元素的标签名称

如果有重复的元素，定位到的元素默认都是第一个元素

**link\_text定位**

通过超链接的全部文本信息进行元素定位，主要用来定位a标签

定位方法：find\_element\_by\_link\_text(link\_text)#link\_text参数代表的是a标签**全部**文本内容

**Partial\_link\_text定位**

通过超链接的局部文本信息进行定位，主要用来定位a标签

定位方法：find\_element\_partial\_link\_text(partial\_link\_text)

**局部包全部->partial\_link\_text包含了link\_text的功能**

**定位一组元素**

1）定位一组元素的方法：find\_element**s**\_by\_id(id)

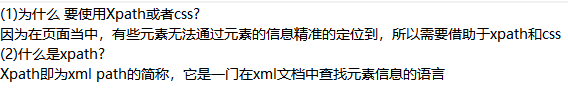
2）定位一组元素返回的值是一个列表

3）可以通过下标来使用列表中的元素

4）下标是从0开始



**Xpath定位**



1. 路径定位

定位方法：find\_element\_by\_xpath(xpath)

**绝对路径**：表达式是以/html开头,元素的层级之间是以/分隔

相同层级的元素可以下标，**下标是从1开始**

需要列出元素所经过的所有层级元素，工作当中，一般不使用绝对路径

如:/html/body/div/fieldset/form/form/p[1]/input

**相对路径**：匹配任意层级的元素，是以//tag\_name或者//\*开头

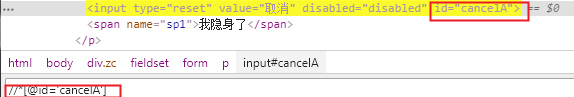
也可以使用下标，**下标是从1开始**

如://p[5]/button

1. 元素属性定位

//或者//tag\_name //\*[@attribute=’value’]#attribute表示的是元素的属性名,value表示的是元素

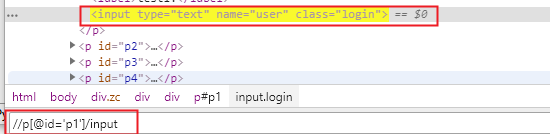
对应属性值



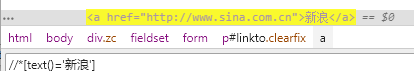
1. 属性与逻辑结合

//\*或者//tag\_name //\*[@attribute1=’value1’and @attribute2=’value2’]

1. 属性与层级结合



5、Xpath扩展(定位文本的全部内容)



**css定位**

css是可以用来在selenium中定位元素的

css定位元素的方法：find\_element\_by\_css\_selector(css\_selector)#css\_selector表示的是css选择器表达式

**css定位策略**

1. id选择器

表达式:#id #表示通过元素的id属性进行元素选择

1. class选择器

表达式：.class #.表示通过元素的class属性进行元素选择，class表示的class属性的其中一个属性值

1. 元素选择器

通过元素标签名称来选择元素 表达式：tag\_name 不推荐使用

1. 属性选择器

通过元素的属性来选择元素 表达式：[attribute=’value’]#attribute表示的是属性名称 value表示的是属性值

注意：

**如果使用的是class属性，需要带上class的全部属性值**

5、层级选择器

说明：根据元素的父子关系来选择

格式1：elements>elements 通过elements1来定位elements2，并且elements2必须为elements1子元素

说明：隔代层级关系选择器

格式2：elements elements 通过elements1来定位elements2，并且elements2是elements1的后代元

**元素操作**

点击操作：element.click()

输入操作:element.send\_keys(‘value’)#element表示的是元素对象，value表示的是要输入的内容

清除操作:element.clear() element表示的是元素对象，将输入框里面的内容全部清除

**浏览器常用操作**

1）**Maximize\_window**()最大化浏览器窗口------>模拟浏览器最大化按钮

2）**Set\_window\_size(width,height)**设置浏览器窗口大小---->设置浏览器宽、高

3）**Set\_window\_position(x,y)**设置浏览器窗口位置

X,y是一个坐标点，通过此坐标点确定浏览器最左上角的位置，以此确定浏览器在屏幕上

4）**Back()后退**---->模拟浏览器后退按钮

5）**Forward()前进**--->模拟浏览器前进按钮

6）**Refresh()刷新**--->模拟浏览器F5刷新

7）**Close()关闭当前窗口**---->模拟点击浏览器关闭按钮

8）**Quit()关闭浏览器驱动对象**---->关闭所有程序启动的窗口

9）**title获取页面title**

10）**Current\_url获取当前页面url**