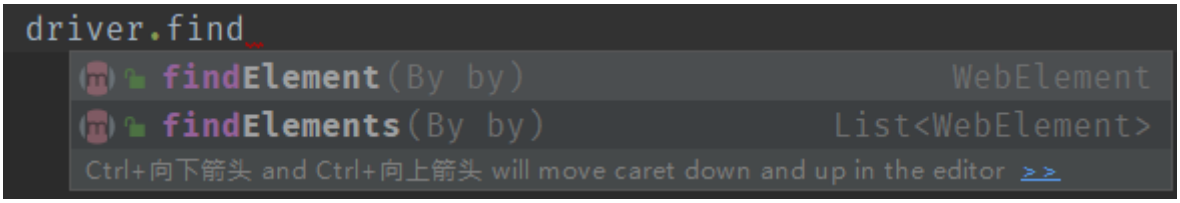
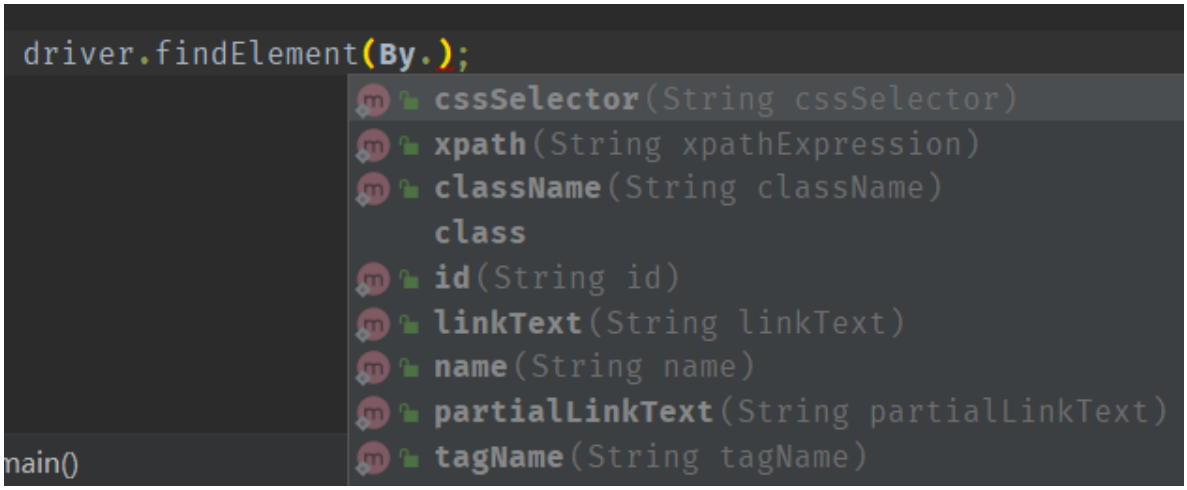
1. driver.findElement(By.)方法

findElement和findElements



class By

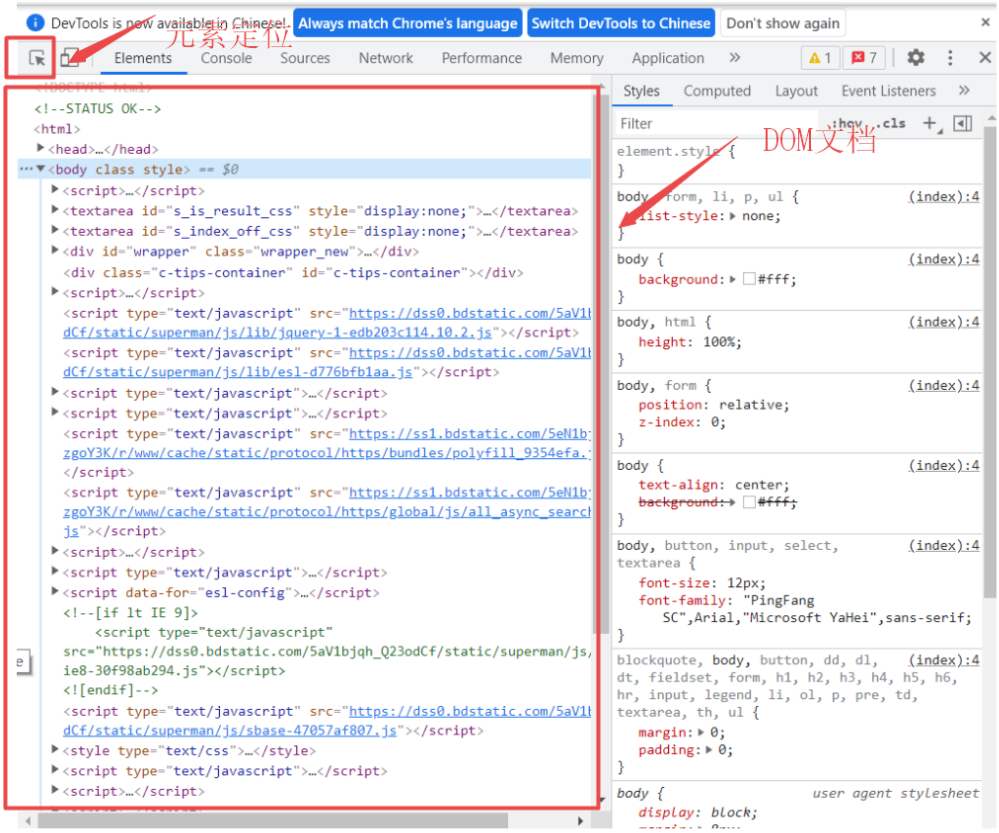



```
public static By id(String id) { return new By.ById(id); }

public static By linkText(String linkText) { return new By.ByLinkText(linkText); }

public static By partialLinkText(String partialLinkText) { return new By.ByPartialLinkText(partialLinkText); }

public static By name(String name) { return new By.ByName(name); }

public static By tagName(String tagName) { return new By.ByTagName(tagName); }

public static By xpath(String xpathExpression) { return new By.ByXPath(xpathExpression); }

public static By className(String className) { return new By.ByClassName(className); }

public static By cssSelector(String cssSelector) { return new By.ByCssSelector(cssSelector); }
```

2. HTML/CSS/JS元素介绍

f12的用法

![image-20211006081755665](C:\Users\26237\AppData\Roaming\Typora\typora-user-images\image-20211006081755665.png




```
<div id="divID" name="divName" class="div-class" href="divHref" my-attr="my-value">
    文本内容
<div/>
```

3. XPath语法介绍

选取节点

表达式	描述
nodename	选取此节点的所有子节点。
/	从根节点选取（取子节点）。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置（取子孙节点）。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

路径表达式	结果
bookstore	选取 bookstore 元素的所有子节点。
/bookstore	选取根元素 bookstore。注释：假如路径起始于正斜杠(/)，则此路径始终代表到某元素的绝对路径！
bookstore/book	选取属于 bookstore 的子元素的所有 book 元素。
//book	选取所有 book 子元素，而不管它们在文档中的位置。
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置。
//@lang	选取名为 lang 的所有属性。

谓语

路径表达式	结果
/bookstore/book[1]	选取属于 bookstore 子元素的第一个 book 元素。
/bookstore/book[last()]	选取属于 bookstore 子元素的最后一个 book 元素。
/bookstore/book[last()-1]	选取属于 bookstore 子元素的倒数第二个 book 元素。
/bookstore/book[position()<3]	选取最前面的两个属于 bookstore 元素的子元素的 book 元素。
//title[@lang]	选取所有拥有名为 lang 的属性的 title 元素。
//title[@lang='eng']	选取所有 title 元素，且这些元素拥有值为 eng 的 lang 属性。
/bookstore/book[price>35.00]	选取 bookstore 元素的所有 book 元素，且其中的 price 元素的值须大于 35.00。
/bookstore/book[price>35.00]//title	选取 bookstore 元素中的 book 元素的所有 title 元素，且其中的 price 元素的值须大于 35.00。

选取未知节点

通配符	描述
*	匹配任何元素节点。
@*	匹配任何属性节点。
node()	匹配任何类型的节点。

路径表达式	结果
/bookstore/*	选取 bookstore 元素的所有子元素。
//*	选取文档中的所有元素。
//title[@*]	选取所有带有属性的 title 元素。

XPath 轴

轴名称	结果
ancestor	选取当前节点的所有先辈（父、祖父等）。
ancestor-or-self	选取当前节点的所有先辈（父、祖父等）以及当前节点本身。
attribute	选取当前节点的所有属性。
child	选取当前节点的所有子元素。
descendant	选取当前节点的所有后代元素（子、孙等）。
descendant-or-self	选取当前节点的所有后代元素（子、孙等）以及当前节点本身。
following	选取文档中当前节点的结束标签之后的所有节点。
following-sibling	选取当前节点之后的所有兄弟节点
namespace	选取当前节点的所有命名空间节点。
parent	选取当前节点的父节点。
preceding	选取文档中当前节点的开始标签之前的所有节点。
preceding-sibling	选取当前节点之前的所有同级节点。
self	选取当前节点。

例子	结果
child::book	选取所有属于当前节点的子元素的 book 节点。
attribute::lang	选取当前节点的 lang 属性。
child::*	选取当前节点的所有子元素。
attribute::*	选取当前节点的所有属性。
child::text()	选取当前节点的所有文本子节点。
child::node()	选取当前节点的所有子节点。
descendant::book	选取当前节点的所有 book 后代。
ancestor::book	选择当前节点的所有 book 先辈。
ancestor-or-self::book	选取当前节点的所有 book 先辈以及当前节点（如果此节点是 book 节点）
child::*/child::price	选取当前节点的所有 price 孙节点。

XPath 运算符

运算符	描述	实例	返回值
|	计算两个节点集	//book | //cd	返回所有拥有 book 和 cd 元素的节点集
+	加法	6 + 4	10
-	减法	6 - 4	2
*	乘法	6 * 4	24
div	除法	8 div 4	2
=	等于	price=9.80	如果 price 是 9.80，则返回 true。如果 price 是 9.90，则返回 false。
!=	不等于	price!=9.80	如果 price 是 9.90，则返回 true。如果 price 是 9.80，则返回 false。
<	小于	price<9.80	如果 price 是 9.00，则返回 true。如果 price 是 9.90，则返回 false。
<=	小于或等于	price<=9.80	如果 price 是 9.00，则返回 true。如果 price 是 9.90，则返回 false。
>	大于	price>9.80	如果 price 是 9.90，则返回 true。如果 price 是 9.80，则返回 false。
>=	大于或等于	price>=9.80	如果 price 是 9.90，则返回 true。如果 price 是 9.70，则返回 false。
or	或	price=9.80 or price=9.70	如果 price 是 9.80，则返回 true。如果 price 是 9.50，则返回 false。
and	与	price>9.00 and price<9.90	如果 price 是 9.80，则返回 true。如果 price 是 8.50，则返回 false。
mod	计算除法的余数	5 mod 2	1

XPath 函数

名称	说明
text()	标签内文本内容
fn:contains(string1,string2)	如果 string1 包含 string2，则返回 true，否则返回 false。例子：contains('XML','XM')结果：true
fn:starts-with(string1,string2)	如果 string1 以 string2 开始，则返回 true，否则返回 false。例子：starts-with('XML','X')结果：true
fn:ends-with(string1,string2)	如果 string1 以 string2 结尾，则返回 true，否则返回 false。例子：ends-with('XML','X')结果：false
fn:substring	返回 string2 在 string1 中出现之前的子字符串。例子

名称	说明
fn:substring-before(string1,string2)	返回 string2 在 string1 中出现之前的子字符串。例子：substring-before('12/10','/')结果：'12'
fn:substring-after(string1,string2)	返回 string2 在 string1 中出现之后的子字符串。例子：substring-after('12/10','/')结果：'10'
fn:matches(string,pattern)	如果 string 参数匹配指定的模式，则返回 true，否则返回 false。例子：matches("Merano", "ran")结果：true

https://www.w3school.com.cn/xpath/xpath_functions.asp

4. CSS选择器语法介绍

id 选择器和class选择器

选择器	例子	例子描述
.class	.intro	选取所有 class="intro" 的元素。
#id	#firstname	选取 id="firstname" 的那个元素。
*	*	选取所有元素。
element	p	选取所有 元素。
element,element,..	div, p	选取所有 元素和所有 元素。

组合选择器

选择器	示例	示例描述
element element	div p	选择 元素内的所有 元素。
element>element	div > p	选择其父元素是 元素的所有 元素。
element+element	div + p	选择所有紧随 元素之后的 元素。
element1~element2	p ~ ul	选择前面有 元素的每个 元素。

伪类/伪元素选择器

选择器	示例	示例说明
:checked	input:checked	选择所有选中的表单元素
:disabled	input:disabled	选择所有禁用的表单元素
:empty	p:empty	选择所有没有子元素的p元素
:enabled	input:enabled	选择所有启用的表单元素
:first-of-type	p:first-of-type	选择的每个 p 元素是其父元素的第一个 p 元素
:in-range	input:in-range	选择元素指定范围内的值
:invalid	input:invalid	选择所有无效的元素
:last-child	p:last-child	选择所有p元素的最后一个子元素
:last-of-type	p:last-of-type	选择每个p元素是其母元素的最后一个p元素
:not(selector)	:not(p)	选择所有p以外的元素
:nth-child(n)	p:nth-child(2)	选择所有 p 元素的父元素的第二个子元素
:nth-last-child(n)	p:nth-last-child(2)	选择所有p元素倒数的第二个子元素
:nth-last-of-type(n)	p:nth-last-of-type(2)	选择所有p元素倒数的第二个为p的子元素

选择器	示例	示例说明
:nth-of-type(n)	p:nth-of-type(2)	选择所有p元素第二个为p的子元素
:only-of-type	p:only-of-type	选择所有仅有一个子元素为p的元素
:only-child	p:only-child	选择所有仅有一个子元素的p元素
:optional	input:optional	选择没有"required"的元素属性
:out-of-range	input:out-of-range	选择指定范围以外的值的元素属性
:read-only	input:read-only	选择只读属性的元素属性
:read-write	input:read-write	选择没有只读属性的元素属性
:required	input:required	选择有"required"属性指定的元素属性
:root	root	选择文档的根元素
:target	#news:target	选择当前活动#news元素(点击URL包含锚的名字)
:valid	input:valid	选择所有有效值的属性
:link	a:link	选择所有未访问链接
:visited	a:visited	选择所有访问过的链接
:active	a:active	选择正在活动链接
:hover	a:hover	把鼠标放在链接上的状态
:focus	input:focus	选择元素输入后具有焦点
:first-letter	p:first-letter	选择每个 元素的第一个字母
:first-line	p:first-line	选择每个 元素的第一行
:first-child	p:first-child	选择器匹配属于任意元素的第一个子元素的 元素
:before	p:before	在每个 元素之前插入内容
:after	p:after	在每个 元素之后插入内容
:lang(*language*)	p:lang(it)	为 元素的lang属性选择一个开始值

属性选择器

选择器	例子	例子描述
[*attribute*]	[target]	选择带有 target 属性的所有元素。
[*attribute=value*]	[target=_blank]	选择带有 target="_blank" 属性的所有元素。
[*attribute~=value*]	[title~=flower]	选择带有包含 "flower" 一词的 title 属性的所有元素。
[*attribute|=value*]	[lang|=en]	选择带有以 "en" 开头的 lang 属性的所有元素。
[*attribute^=value*]	a[href^="https"]	选择其 href 属性值以 "https" 开头的每个 元素。
[*attribute$=value*]	a[href$=".pdf"]	选择其 href 属性值以 ".pdf" 结尾的每个 元素。
[*attribute**=value*]	a[href*="w3school"]	选择其 href 属性值包含子串 "w3school" 的每个 元素。

5. 参考代码

```
        System.setProperty("webdriver.chrome.driver",
"src/main/resources/chromedriver.exe");// chromedriver服务地址
        WebDriver driver = new ChromeDriver(); // 新建一个WebDriver 的对象，但是new
的是谷歌的驱动
        String url = "http://www.baidu.com";
        driver.get(url); // 打开指定的网站
        driver.navigate().to(url); // 打开指定的网站

        //根据id查询
        WebElement e1 = driver.findElement(By.id("s-top-left"));
        System.out.println(e1.getText());
        //根据name查询
        WebElement e2 = driver.findElement(By.name("wd"));
        System.out.println(e2.getTagName());
        //根据link的文字查询
        WebElement e3 = driver.findElement(By.linkText("地图"));
        System.out.println(e3.getText());
        List<WebElement> e4 = driver.findElements(By.partialLinkText("地"));
        for(WebElement e: e4){
            System.out.println(e.getText());
        }
        //根据标签查询
        List<WebElement> e5 = driver.findElements(By.tagName("ul"));
        for(WebElement e: e5){
            System.out.println(e.getText());
        }
        //根据class查询
```

```
        System.out.println();
        List<WebElement> e6 = driver.findElements(By.className("hotsearch-
item"));
        for(WebElement e: e6){
            System.out.println(e.getText());
        }
        //根据xpath查询
        WebElement e7 = driver.findElement(By.xpath("//*[@id=\"s-top-
left\"]/a[1]"));
        System.out.println(e7.getText());
        WebElement e8 =
driver.findElement(By.xpath("/html/body/div[1]/div[1]/div[5]/div/div/form/span[2
]/input"));
        System.out.println(e8.getTagName());
        WebElement e9 =
driver.findElement(By.xpath("/html/body/div[1]/div[1]/div[5]/div/div/form/span[2
]/input/parent::*"));
        System.out.println(e9.getTagName());
        WebElement e10 = driver.findElement(By.xpath("//a[contains(text(), \'学
\')]"));
        System.out.println(e10.getText());
        //根据css查询
        WebElement e11 = driver.findElement(By.cssSelector("#s-top-left > a:nth-
child(5)"));
        System.out.println(e11.getText());
        WebElement e12 = driver.findElement(By.cssSelector("#hotsearch-refresh-
btn > span"));
        System.out.println(e12.getText());
```

6. 参考运行截图

信息: Detected dialect: W3C
新闻hao123地图直播视频贴吧学术更多
input
地图
地图

1美贸易代表:将与中方坦率对话热
4新能源车高速服务区排队4小时充电
2滞留伊犁旅客:有超市部分货架空了
5女孩看完长津湖回家尝冻土豆
3#脸书旗下应用宕机超6小时#
6景区山顶拥堵 游客套塑料袋御寒

1美贸易代表:将与中方坦率对话热
4新能源车高速服务区排队4小时充电
2滞留伊犁旅客:有超市部分货架空了
5女孩看完长津湖回家尝冻土豆
3#脸书旗下应用宕机超6小时#
6景区山顶拥堵 游客套塑料袋御寒
新闻
input
span
学术
视频
换一换