丁俊尧

XPath 选择器

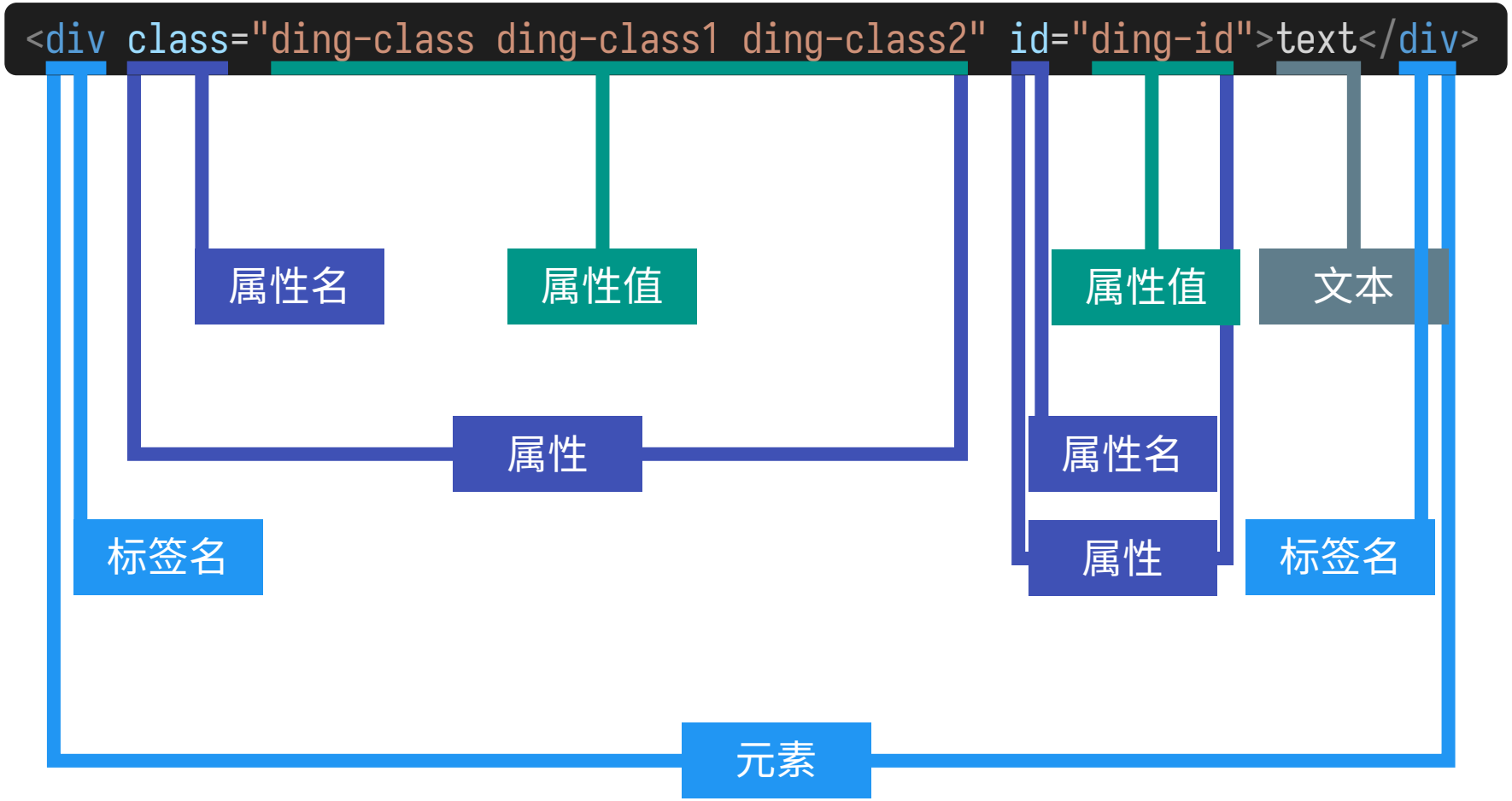
目录

XPath 简介

- 是一门在 XML 文档中查找信息的语言
- 可用来在 XML 文档中对元素和属性进行遍历
- HTML 可以视为一种不严格的 XML，所以也可以用 XPath 进行元素的查询

准备

元素、属性与文本

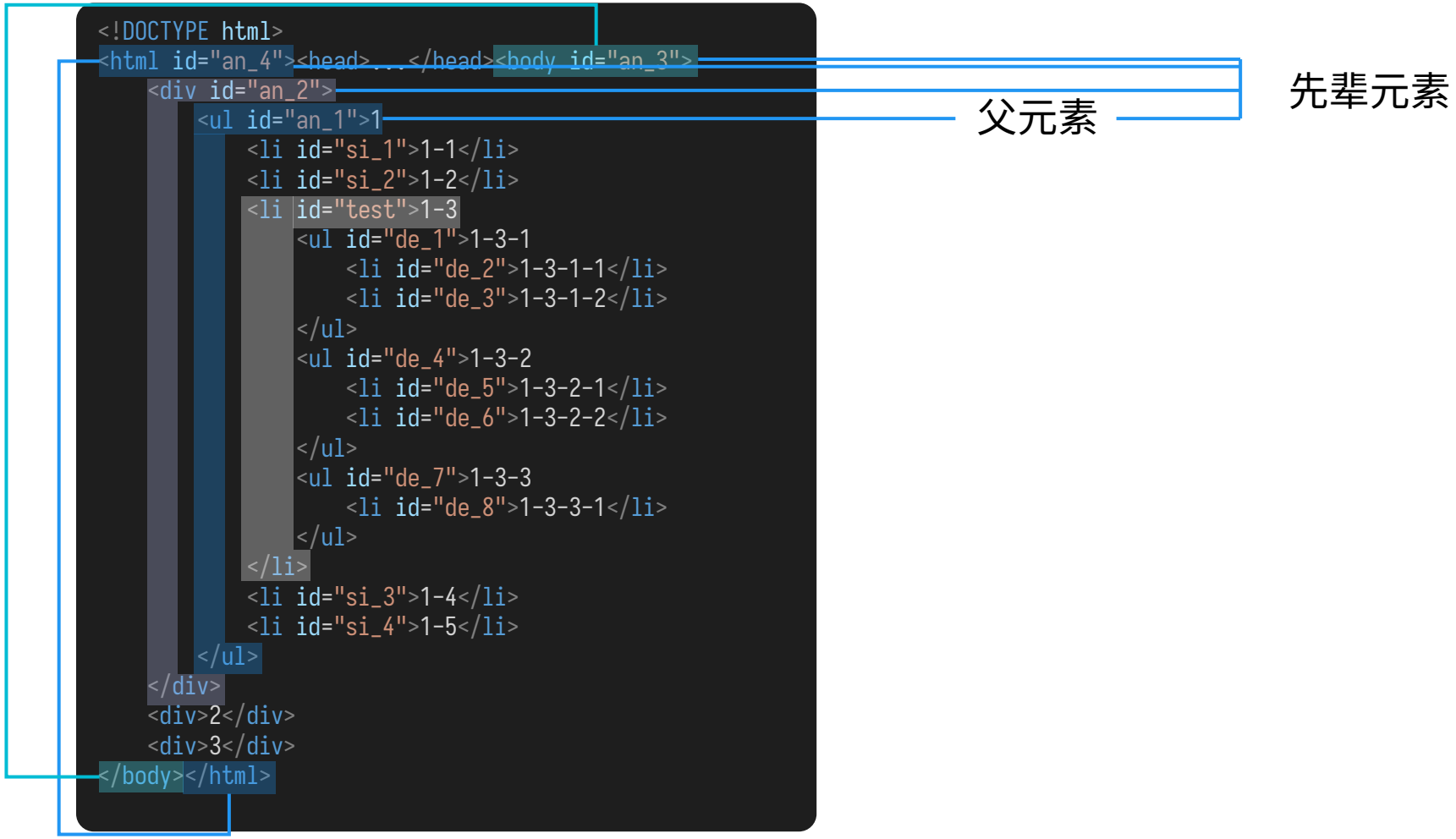


元素间的关系

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

以该元素为例

元素间的关系——父元素、先辈元素




```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

元素间的关系——子元素、后代元素

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

子元素

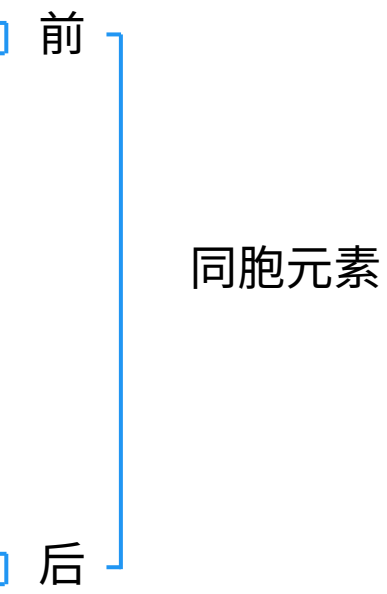
子元素

子元素

后代元素

元素间的关系——同胞元素

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```



基本选取元素方式

绝对路径与相对路径

- / 开头为绝对路径，相对于整个 HTML 代码而言
- 否则为相对路径，只能在特定情况下使用

```
a = driver.find_element(By.XPATH, '//div[@id="content_left"]')
b = a.find_elements(By.XPATH, 'div[not(@tpl="recommend_list")]//h3')
```

标签、属性、下标与参数

◆**标签**直接写标签名即可

◆表示任意标签用 *

◆在标签后加上方括号 []，方括号内写这个元素的**属性**

◆在属性中写上数字，表示选取第几个元素（**下标**从1开始）

◆在**参数**名前加上 @，写在属性里，表示选取具有该参数的元素

◆@参数名=值 表示选取参数为某个值的元素

```
div
p
/*
div[3]
div[@id]
li[@id="test"]
```

文本值用什么包裹

◆文本值用单引号、双引号包裹都可以，只要统一就行，看个人习惯

◆如果你要将其应用于编程，最好用你包裹字符串用的引号不一样的引号，免去转义的麻烦

```
driver.find_element(By.XPATH, '//div[@id="content_left"]')
driver.find_element(By.XPATH, "//div[@id='content_left']")
```

子元素、后代元素

◆**子元素**：元素/子元素

◆**后代元素**：元素//后代元素

◆以 // 开头可以简略 /html/body/...

```
/html/body/div[1]/ul/li[@id="test"]
//li[@id="test"]
//div[@id="an_2"]//li[@id="test"]
```

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

父元素

◆..

```
//li[@id="test"]/../li[1]
```

◆XPath 和文件路径的表达方式很像——

所以有时候我们就叫“XPath 路径”

◆. 表示自身

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

运算与常用函数

基本运算

◆加减乘除 `a+b` `a-b` `a*b` `a div b`

◆取余 `a mod b`

◆比较 `a=b` `a!=b` `a<b` `a>b` `a<=b` `a>=b`

◆逻辑 `a and b` `a or b` `not(a)`

```
//li[not(@index div @alt < 6)]
```

```
<div>
  <ul>
    <li index="2" alt="1">1</li>
    <li index="12" alt="2">2</li>
    <li index="24" alt="2">3</li>
    <li index="d" alt="e">4</li>
    <li index="2" alt="f">5</li>
    <li index="e" alt="1">6</li>
    <li index="2" alt="1">7</li>
    <li index="e" alt="e">8</li>
    <li index="2">9</li>
    <li index="e">10</li>
    <li alt="1">11</li>
    <li alt="e">12</li>
    <li>13</li>
  </ul>
</div>
```

选取第几个元素

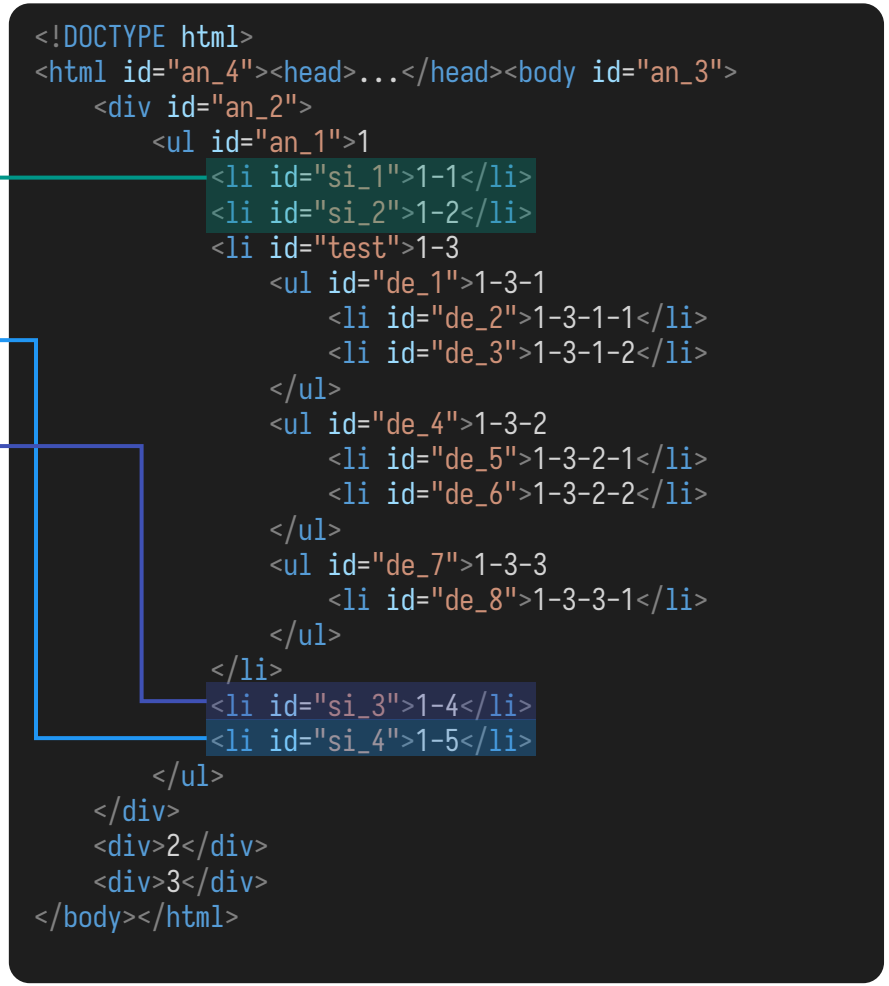
◆最后一个元素：last()

◆元素序号：position()

```
//li[@id="test"]/../li[last()]
```

```
//li[@id="test"]/../li[last()-1]
```

```
//li[@id="test"]/../li[position()<3]
```



选取多个路径

◆使用 | 来连接多个路径

```
//ul[@id="de_1"] | //li[@id="si_3"]
```

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

文字 - 1

◆使用 text() 来表示文字

◆在 Web 自动化测试时，较常用于定义属性

```
//li[text()="1-3-3-1"]
```

◆该函数取的文本不包括后代元素的文本，且如果有换行、空格和缩进，也会算进去

◆如果换行和缩进之间被其他元素隔开，也会被分隔为两个部分

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

文字 - 2

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

```
> $x('//li[@id="test"]/text()')[0].textContent
"1-3\n                "
> $x('//li[@id="test"]/text()')[1].textContent
"\n                "
> $x('//li[@id="test"]/text()')[2].textContent
"\n                "
> $x('//li[@id="test"]/text()')[3].textContent
"\n            "
> $x('//li[@id="test"]/text()')
▶ (4) [text, text, text, text]
> $x('//li[@text="1-3"]')
▶ []
>
```

包含

◆contains(a, b) 函数表示在 a 中包含 b

◆现在需要选取 class 属性带有 ding-class 的 div 元素

```
//div[@class="ding-class"]
```

```
<div class="ding-class">text</div>
<div class="ding-class bing-class">text</div>
<div class="bing-class ding-class ping-class">text</div>
<div class="ading-class">text</div>
<div class="ding-class-1 x-class">text</div>
```

```
//div[contains(@class, "ding-class")]
```

```
<div class="ding-class">text</div>
<div class="ding-class bing-class">text</div>
<div class="bing-class ding-class ping-class">text</div>
<div class="ading-class">text</div>
<div class="ding-class-1 x-class">text</div>
```

连接

◆concat(a, b, ...) 函数表示连接 a、b...

◆精准选取 class 参数

```
//div[contains(concat(" ", @class, " "), " ding-class ")]
```

```
<div class="ding-class">text</div>
<div class="ding-class bing-class">text</div>
<div class="bing-class ding-class ping-class">text</div>
<div class="ading-class">text</div>
<div class="ding-class-1 x-class">text</div>
```

开头

◆starts-with(a, b) 函数表示 a 以 b 开头

轴

轴的用法

之前的元素/轴::标签[条件或下标]

◆表示之前的元素的轴中，符合条件或下标的标签

◆Web 自动化测试中常用的轴如右表：

轴	含义	相当于
parent	父元素	..
ancestor	先辈元素	
ancestor-or-self	先辈元素与自身	
child	子元素	
descendant	后辈元素	//
descendant-or-self	后辈元素与自身	
preceding	之前的元素	
following	之后的元素	
preceding-sibling	之前的同胞元素	
following-sibling	之后的同胞元素	

轴的示例

```
//li[@id="test"]/preceding-sibling::*
```

```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

轴与下标

◆如果涉及到下标，**根据该元素的位置，从近到远排序**，而不是按照代码中的顺序排序

```
//li[@id="test"]/preceding-sibling::*[1]
```

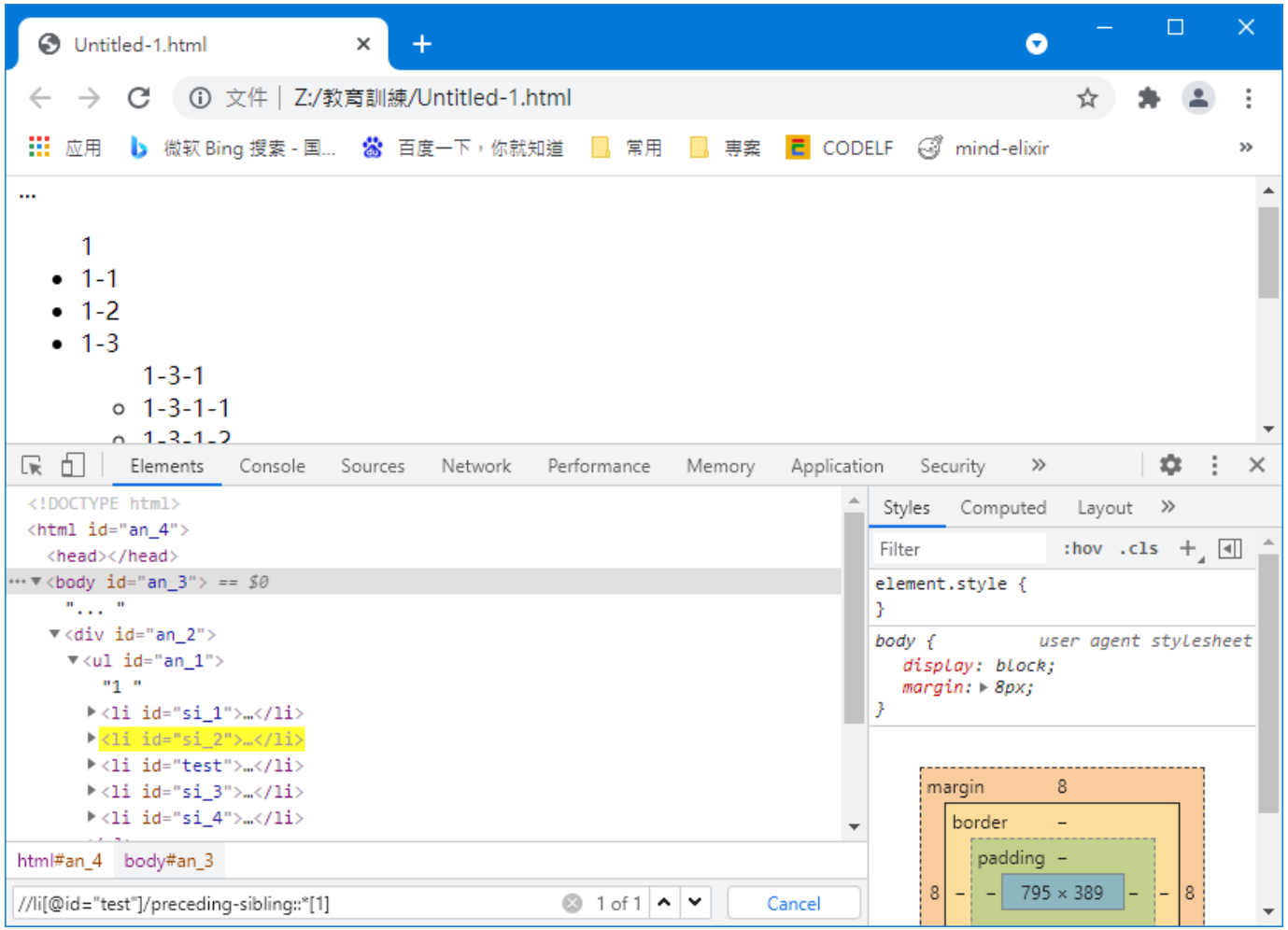
```
<!DOCTYPE html>
<html id="an_4"><head>...</head><body id="an_3">
    <div id="an_2">
        <ul id="an_1">1
            <li id="si_1">1-1</li>
            <li id="si_2">1-2</li>
            <li id="test">1-3
                <ul id="de_1">1-3-1
                    <li id="de_2">1-3-1-1</li>
                    <li id="de_3">1-3-1-2</li>
                </ul>
                <ul id="de_4">1-3-2
                    <li id="de_5">1-3-2-1</li>
                    <li id="de_6">1-3-2-2</li>
                </ul>
                <ul id="de_7">1-3-3
                    <li id="de_8">1-3-3-1</li>
                </ul>
            </li>
            <li id="si_3">1-4</li>
            <li id="si_4">1-5</li>
        </ul>
    </div>
    <div>2</div>
    <div>3</div>
</body></html>
```

书写选择器文本的诀窍

善用工具

◆浏览器的开发人员工具能够轻松调试选择器文本

◆仔细观察层级结构和关系



避免使用自动生成、层级过多、多处使用下标的嵌套

◆由于 Web 框架可能会变动层级和元素，这样的查询语句很容易查询不到想要的元素

◆建议使用一些标志性的特征来定位

```
//*[@id="post-XPath"]/div[1]/main/div[2]/div/div[2]/div/
```

```
//h3[@id="root-node"]/../div[@class="body"]/p
```

谢谢

2021年6月27日