Element locating

David LI

Selenium Webdriver 2个寻找元素方法

- driver.findElement
  - Return WebElement
- driver.findElements
  - Return List<WebElement>

- By ID
  -
  - driver.findElement(By.id("myID")
- By Name
  -
  - driver.findElement(By.name("myName")
- By class
  -
  - driver.findElement(By.className("myClass"))
- By tag name
  - <label ></lable>
  - driver.findElement(By. tagName("label"))
- By link text
  - My Link
  - driver.findElement(By. linkText("My Link"))
- By partial link text : driver.findElement(By.partialLinkText(<link text>))
  - My Partial Link
  - driver.findElement(By. partialLinkText("Partial"))

By CSS

- #Id
  -
  - driver.findElement(By.cssSelector("#myId")
- tagName
  - <input value="myText" />
  - driver.findElement(By.cssSelector("input")
- .className
  -
  - driver.findElement(By.cssSelector("input")
- Attribute
  -
  - driver.findElement(By.cssSelector("div[type='myType'][typeE='anotherType']")
  - driver.findElement(By.cssSelector("div[type]")
  - driver.findElement(By.cssSelector("div[type]")

- Not
  - driver.findElement(By.cssSelector("img:not([alt])"))
- ^=
  - Input[id^= ' ctrl']
- $=
  - input[id$='_userName']
- *=
  - Input[id*='userName']

Absolute and relative path

- Absolute
  - Html>body>p
- Relative
  - Html p

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>菜鸟教程(runoob.com)</title>
  </head>
  <body>
    <h1>我的第一个标题</h1>
    <p>我的第一个段落。</p>
  </body>
</html>
```

- *
  - Selects all elements
  - *[class='blue']
- div, p
  - Selects all elements and all <p> elements
- div p
  - Selects all <p> elements inside elements
- div > p
  - Selects all <p> elements where the parent is a element
- div + p
  - Selects all <p> elements that are placed immediately after elements
- p ~ ul
  - Selects every <ul> element that are preceded by a <p> element
- p:nth-child(2):start from 1
  - Selects every <p> element that is the second child of its parent
- p:first-child & p:last-child

By XPath : driver.findElement(By.xpath(<xpath query expression>))

- /:absolute path
- //:relative path
- TagName
- @Attribute
  - driver.findElement(By.xpath("//input[@id='username']"))
- And & Or
- //input[@type='submit'][@value='Login']
- //input[@type='submit‘ and @value='Login']
- //input[@type='submit‘ or @value='Login']

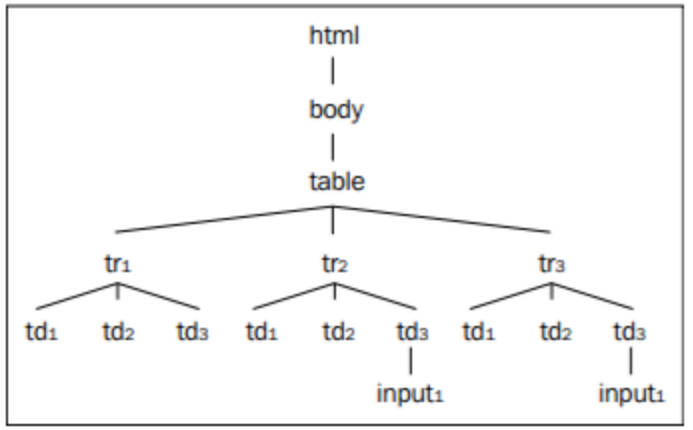
- Not
  - //div[not[@class=‘’myClass]]
- starts-with()
  - input[starts-with(@id,'ct rl')]
- ends-with()
  - input[ends-with(@id,'_use rName')]
- contains()
  - input[ends-with(@id,'_use rName')]
- Text()
  - My Text
  - //div[text()=‘My Text’]
  - //div[contains(text(),’My Text’)]
  - //div[contains(.,’My Text’)]
- Normalize-space: 去除字符串头尾的空格
  - My Text
  - //div[normalize-space(.)=’My Text’]

- //body[h1]
  - To get body that contains h1
- //body//h1
  - To get h1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>菜鸟教程(runoob.com)</title>
  </head>
  <body>
     <h1>我的第一个标题</h1>
     <p>我的第一个段落。</p>
   </body>
</html>
```

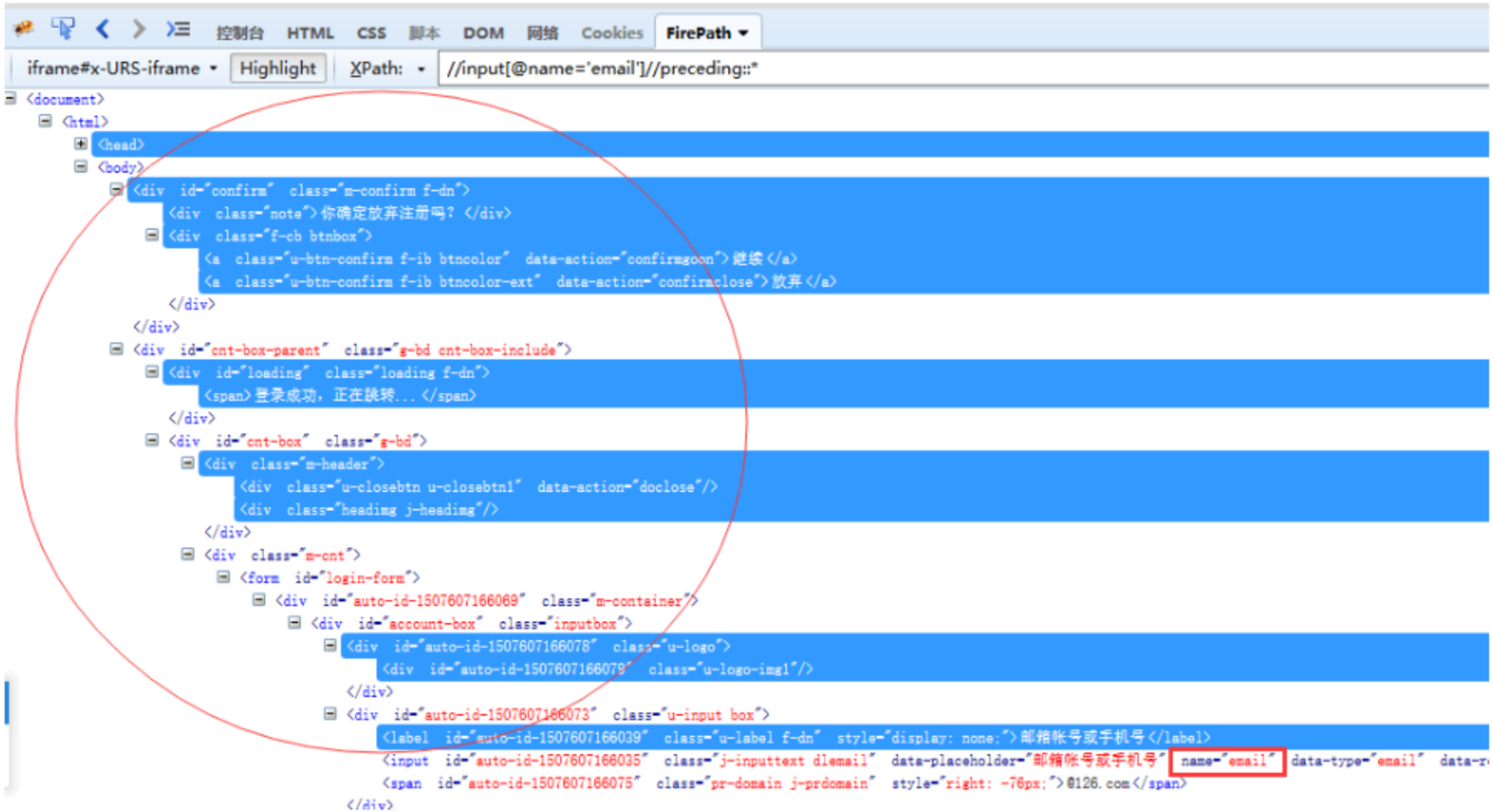
axis

- Ancestor：选择当前节结点所有的父类元素，包括祖先元素
  - //td[text()='Product 1']/ancestor::table
- Descendant：选择当前节点所有子元素
  - //table/descendant::td/input
- Following：选择当前元素结束标签后的所有元素
  - //td[text()='Product 1']/following::tr
- Preceding：选择文档中当前节点的开始标签之前的所有节点
  - //td[text()='$150']/ preceding::tr
- following- sibling：选择当前元素后的兄弟元素
  - //td[text()='Product 1']/following- sibling::td
- preceding- sibling：选择当前节点之前的所有同级节点
  - //td[text()='$150']/ preceding-sibling::td



following::* 表示当前节点后的所有节点

preceding::* 表示当前节点前的所有节点



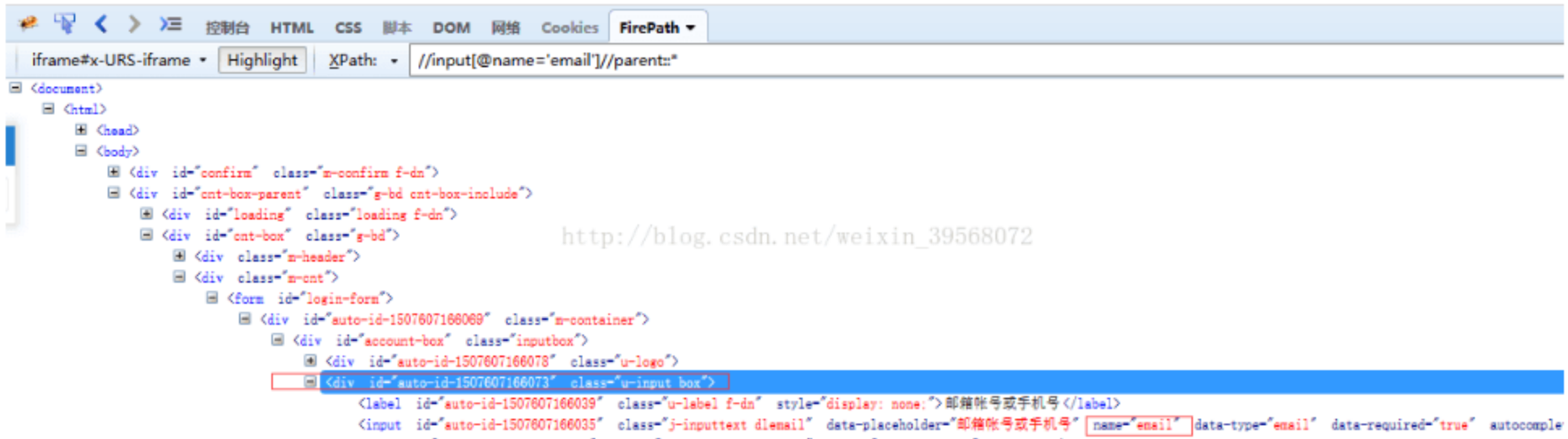
following-sibling::* 表示当前节点后的所有同级节点

preceding-sibling::* 表示当前节点前的所有同级节点

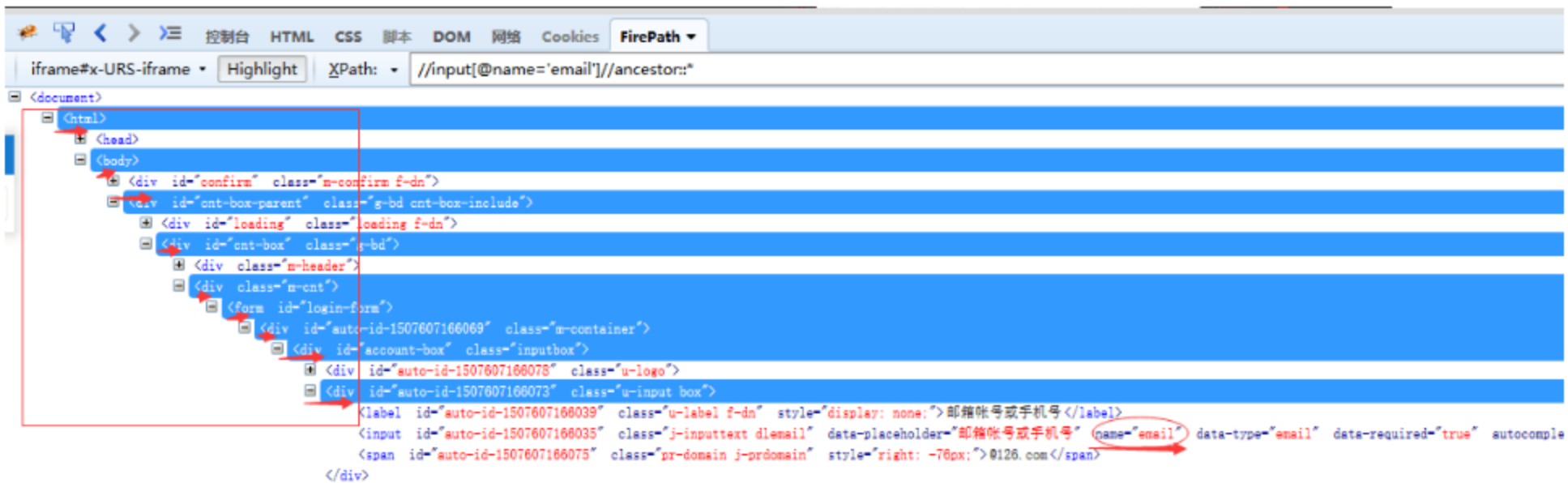
preceding-sibling::* 表示当前节点前的所有同级节点

child::* 表示当前节点的所有子节点

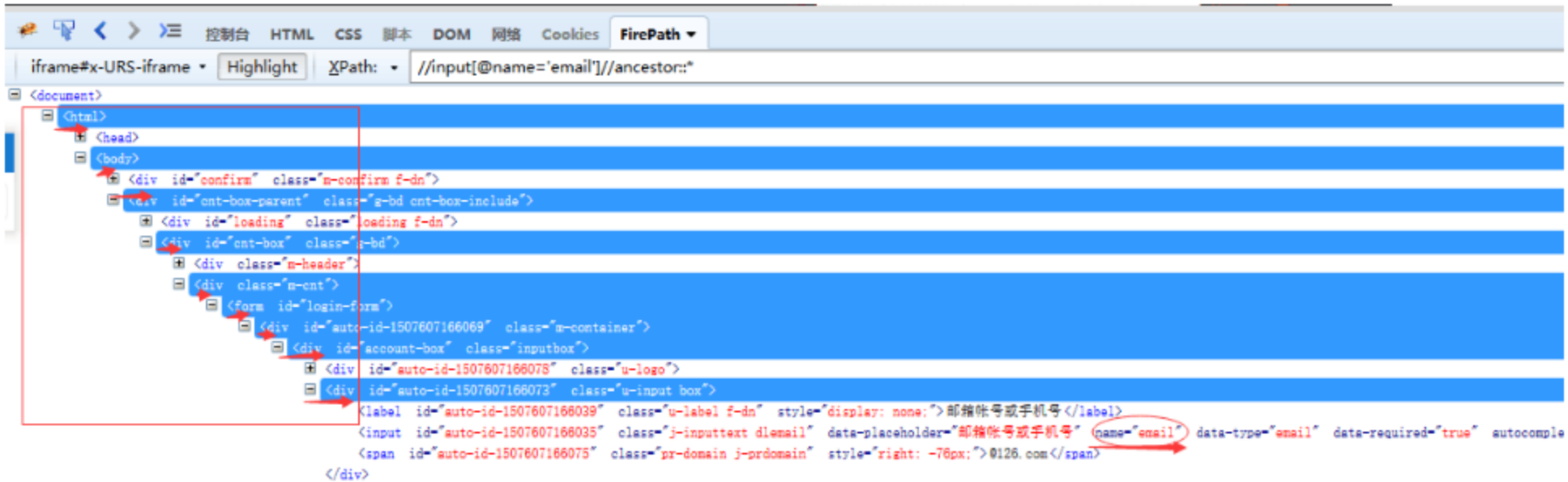
parent::* 表示当前节点的所有父节点



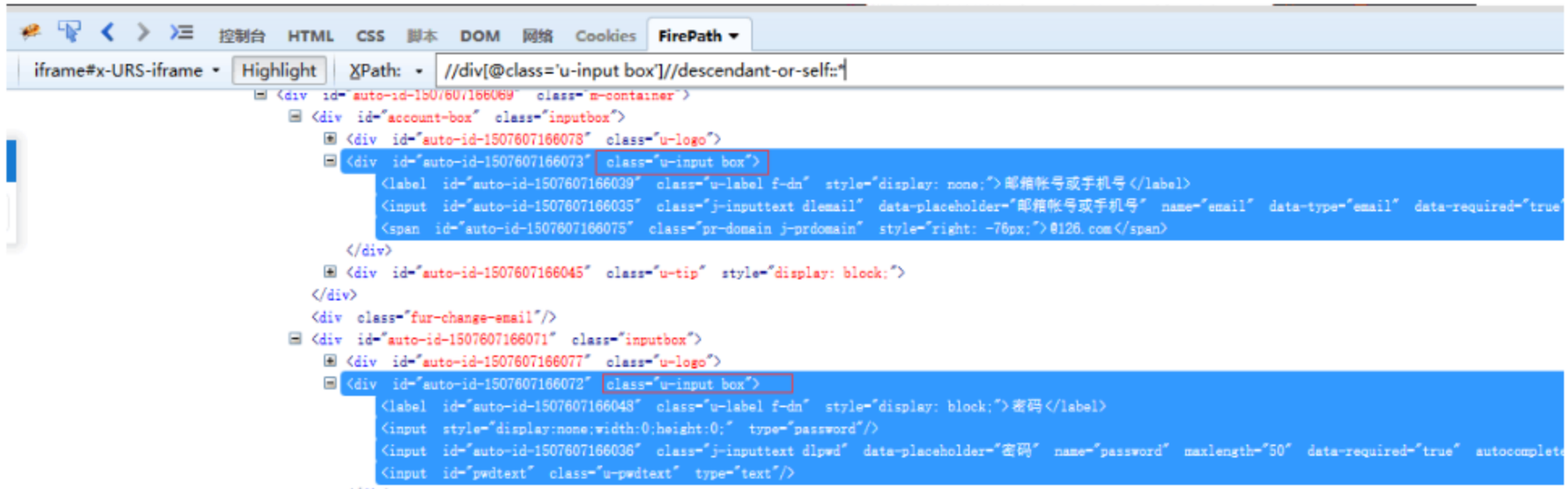
ancestor::* 表示当前节点的祖父节点



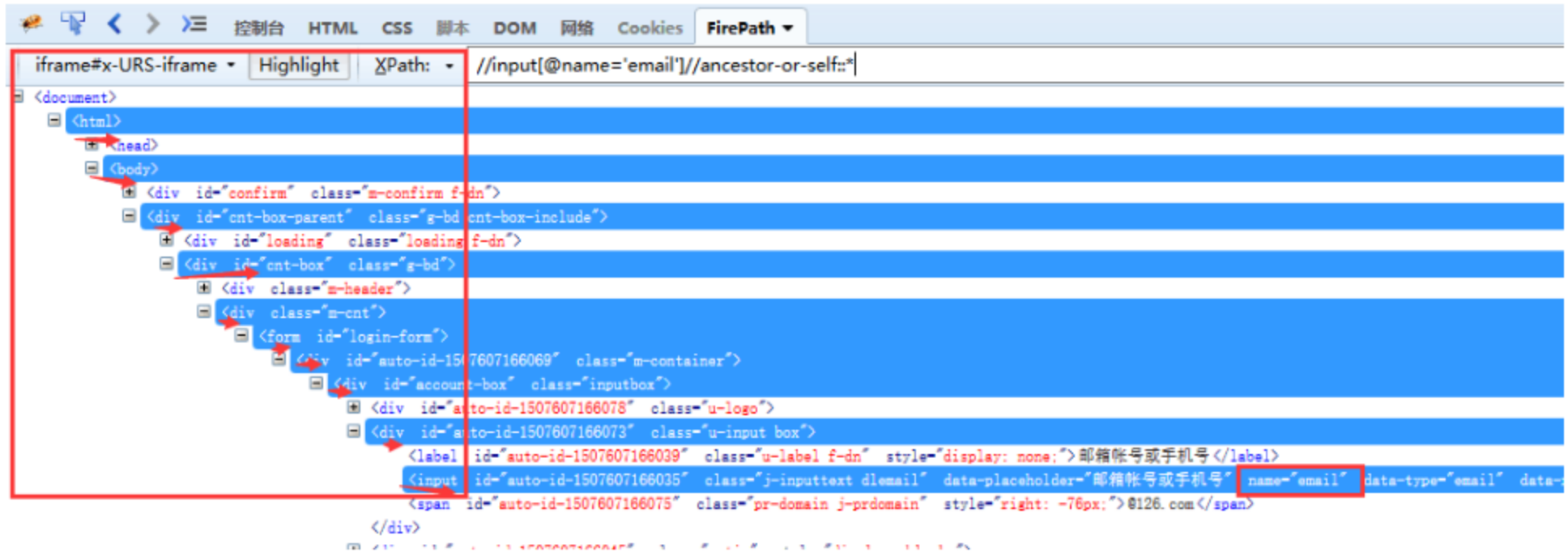
self::* 表示当前节点的自身元素



descendant-or-self::* 表示当前节点及他们的后代元素



ancestor-or-self::* 表示当前节点及它的祖先节点



Driver.findElements

```
package com.example.tests;
import static org.junit.Assert.*;
import java.util.*;
import org.junit.*;
import org.openqa.selenium.*;
import org.openqa.selenium.ie.InternetExplorerDriver;
public class Selenium2 {
WebDriver driver = new InternetExplorerDriver();
JavascriptExecutor jse = (JavascriptExecutor)driver;
@Test
public void tableTest() {
    driver.get
    ("http://www.w3school.com.cn/html/html_tables.asp");
    //首先得到所有tr的集合
    List<WebElement> rows =
        driver.findElements(By.cssSelector(".dataintable tr"));
       //验证表格的行数
    assertEquals(11,rows.size());
        //打印出所有单元格的数据
    for (WebElement row : rows) {
    //得到当前tr里td的集合
        List<WebElement> cols =
            row.findElements(By.tagName("td"));
        for (WebElement col : cols) {
        System.out.print(col.getText());//得到td里的文本
        }
        System.out.println();
    }
       driver.close();
}
```

Any Questions?