XPath

in Selenium WebDriver

What is XPath?

- XPath is defined as **XML path**.
- **It is a syntax or language for finding any element on the web page using XML path expression**.
- XPath is used to find the location of any element on a webpage using HTML DOM structure.

Types of X-path

- There are two types of XPath:

1) Absolute XPath

2) Relative XPath

Absolute XPath

- It is the direct way to find the element, but the disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.
- It begins with the single forward slash(/) ,which means you can select the element from the root node.
- Below is the example of an absolute xpath expression of the element shown in the below screen.
- Ex:
- /html[1]/body[1]/div[1]/table[1]/tbody[1]/tr[1]/td[2]/table[1]/tbody[1]/tr[4]/td[1]/table[1]/tbody[1]/tr[1]/td[2]/table[1]/tbody[1]/tr[2]/td[3]/form[1]/table[1]/tbody[1]/tr[4]/td[1]/table[1]/tbody[1]/tr[2]/td[2]/input[1]

Relative xpath

- For Relative Xpath the path starts from the middle of the HTML DOM structure.
- It starts with the double forward slash (//), which means it can search the element anywhere at the webpage.
- You can start from the middle of the HTML DOM structure and no need to write long xpath.

Ex:

```
//input[@name='userName']
```

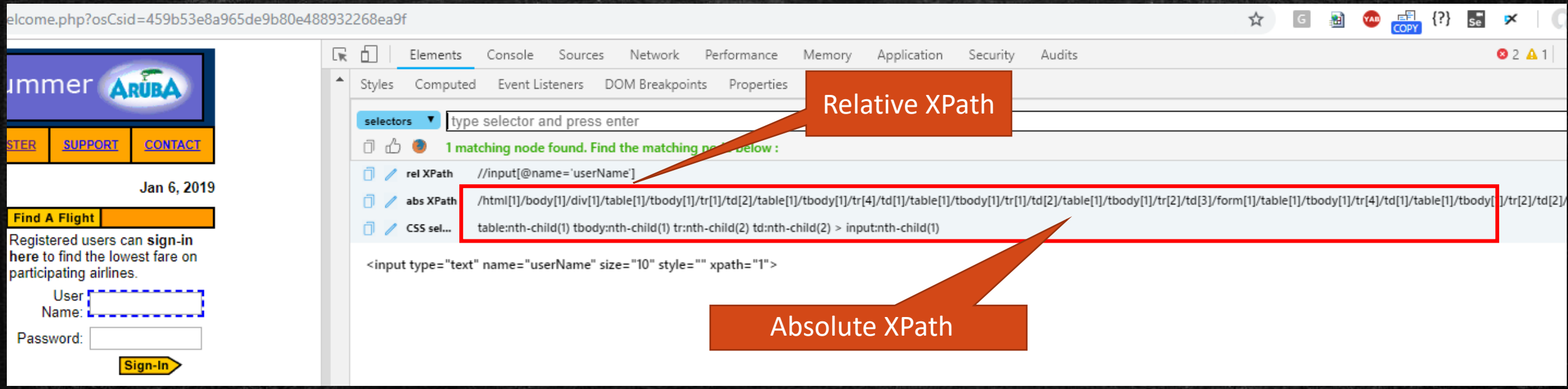
Syntax for Relative XPath

- XPath contains the path of the element situated at the web page. Standard syntax for creating XPath is.

```
Xpath=//tagname[@attribute='value']
```

- **// :** Select current node.
- **Tagname:** Tagname of the particular node.
- **@:** Select attribute.
- **Attribute:** Attribute name of the node.
- **Value:** Value of the attribute.

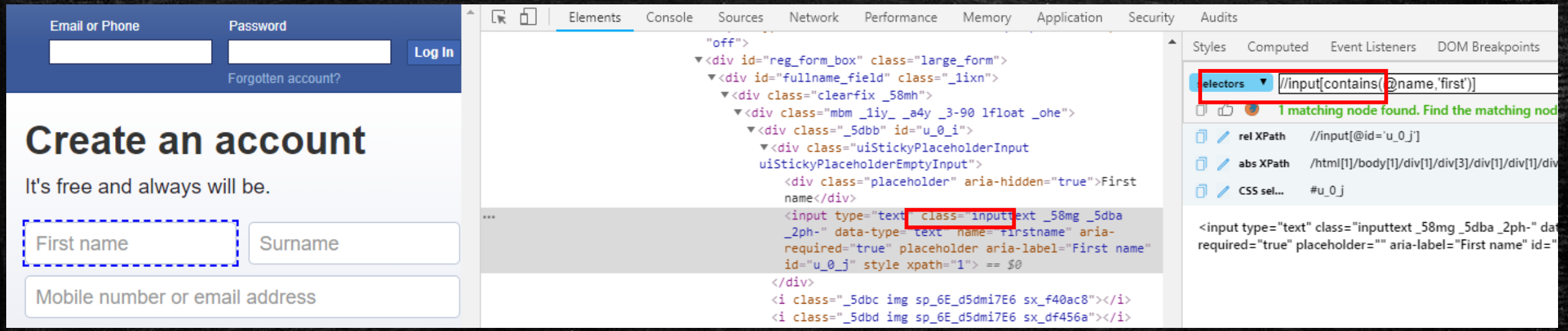
Absolute & Relative XPath



Contains()

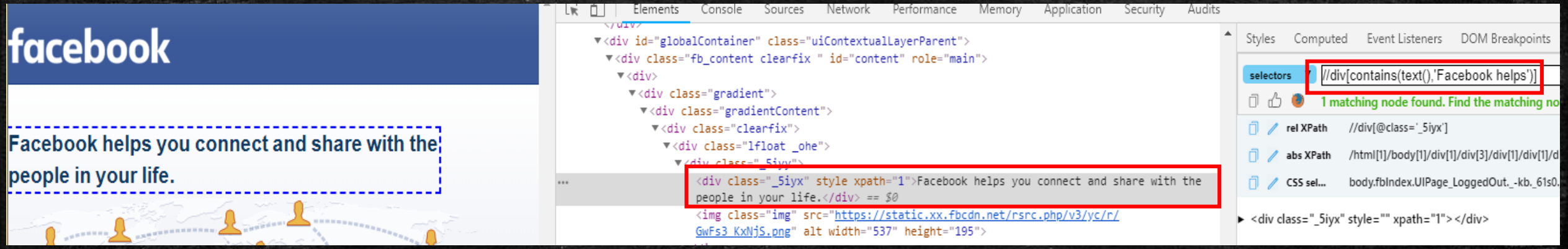
//input[contains(@name,'first')]

- Contains() is a method used in XPath expression. It is used when the value of any attribute changes dynamically, for example, login information.
- The contain feature has an ability to find the element with partial text



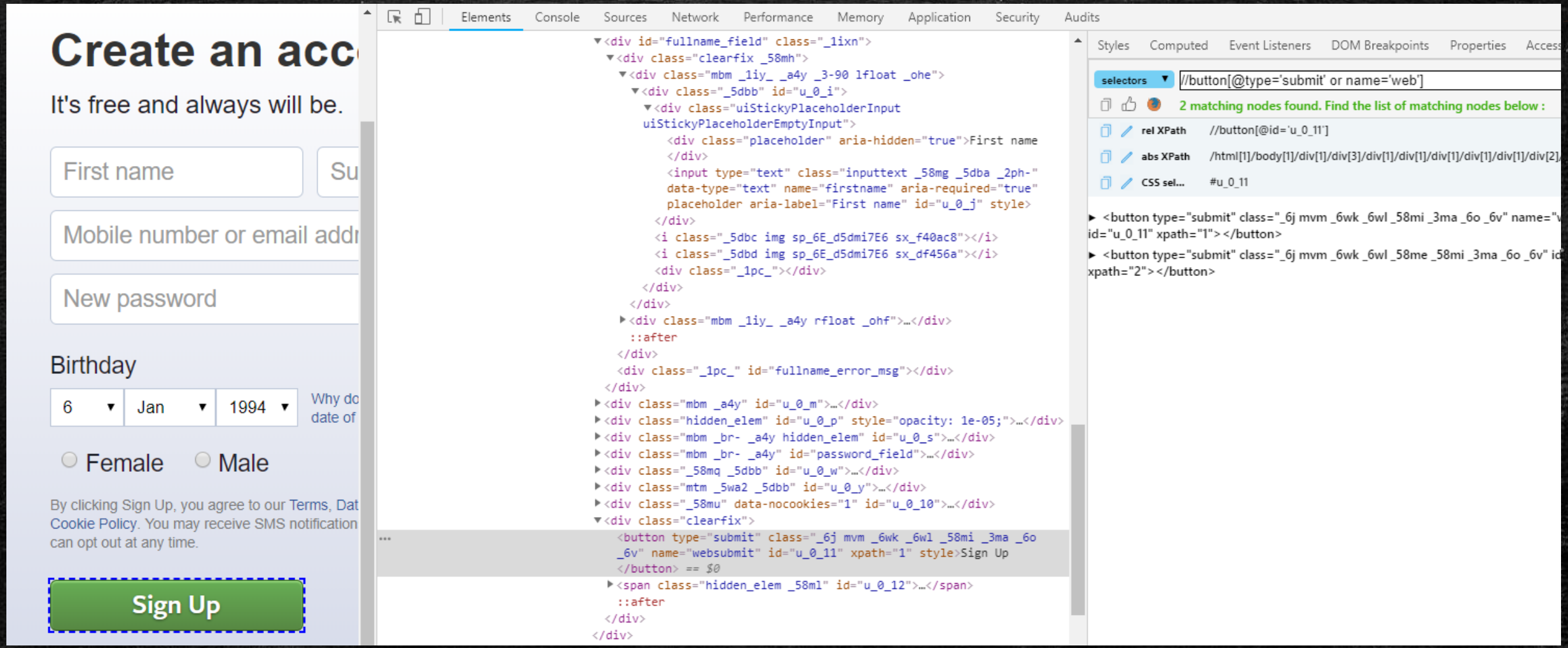
Contains() with text()

//div[contains(text(),'Facebook helps')]



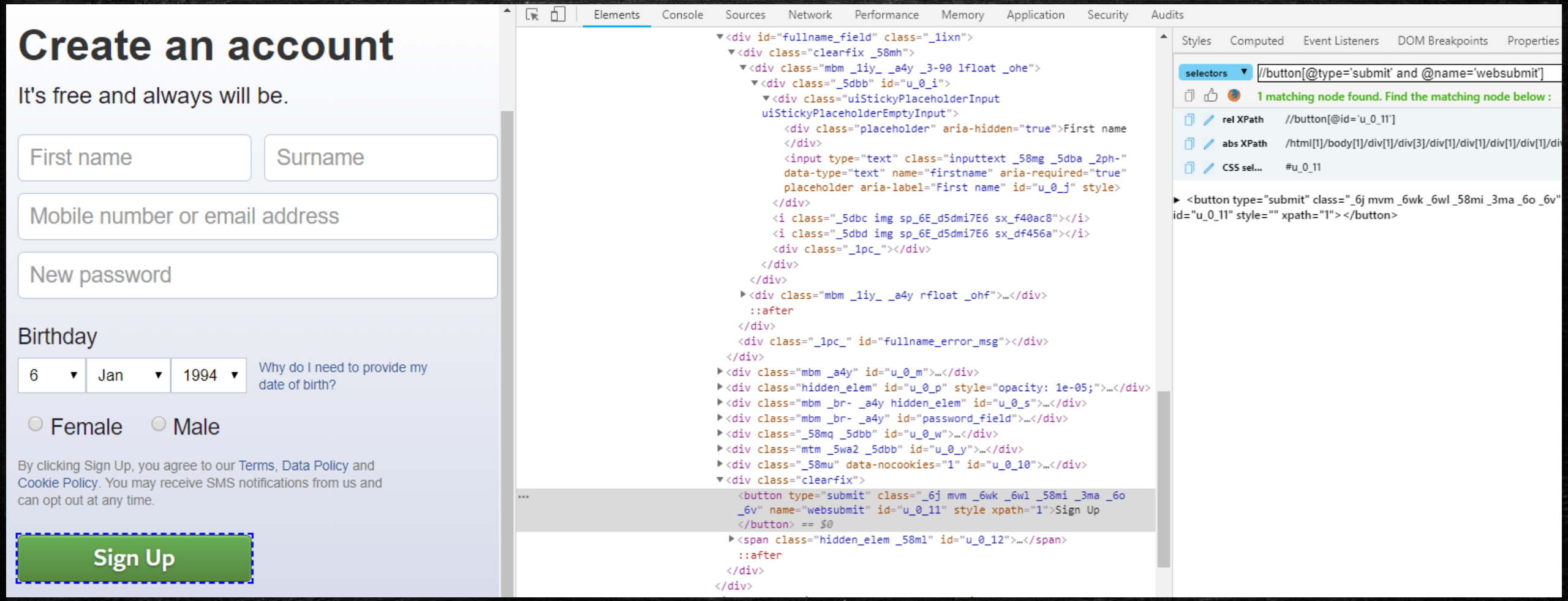
Using OR

//button[@type='submit' or name='web']



Using AND

//button[@type='submit' and @name='websubmit']

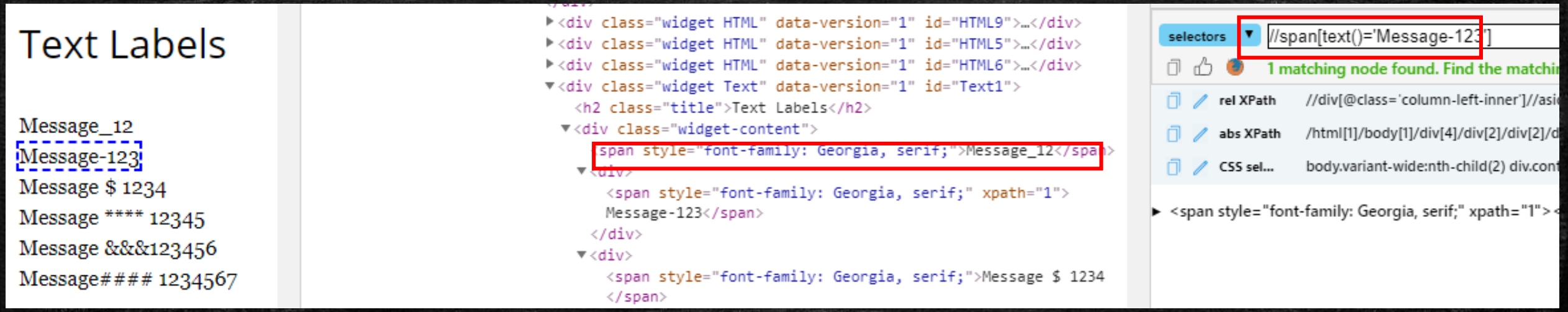


Start-with()

//span[starts-with(text(),'Message')]

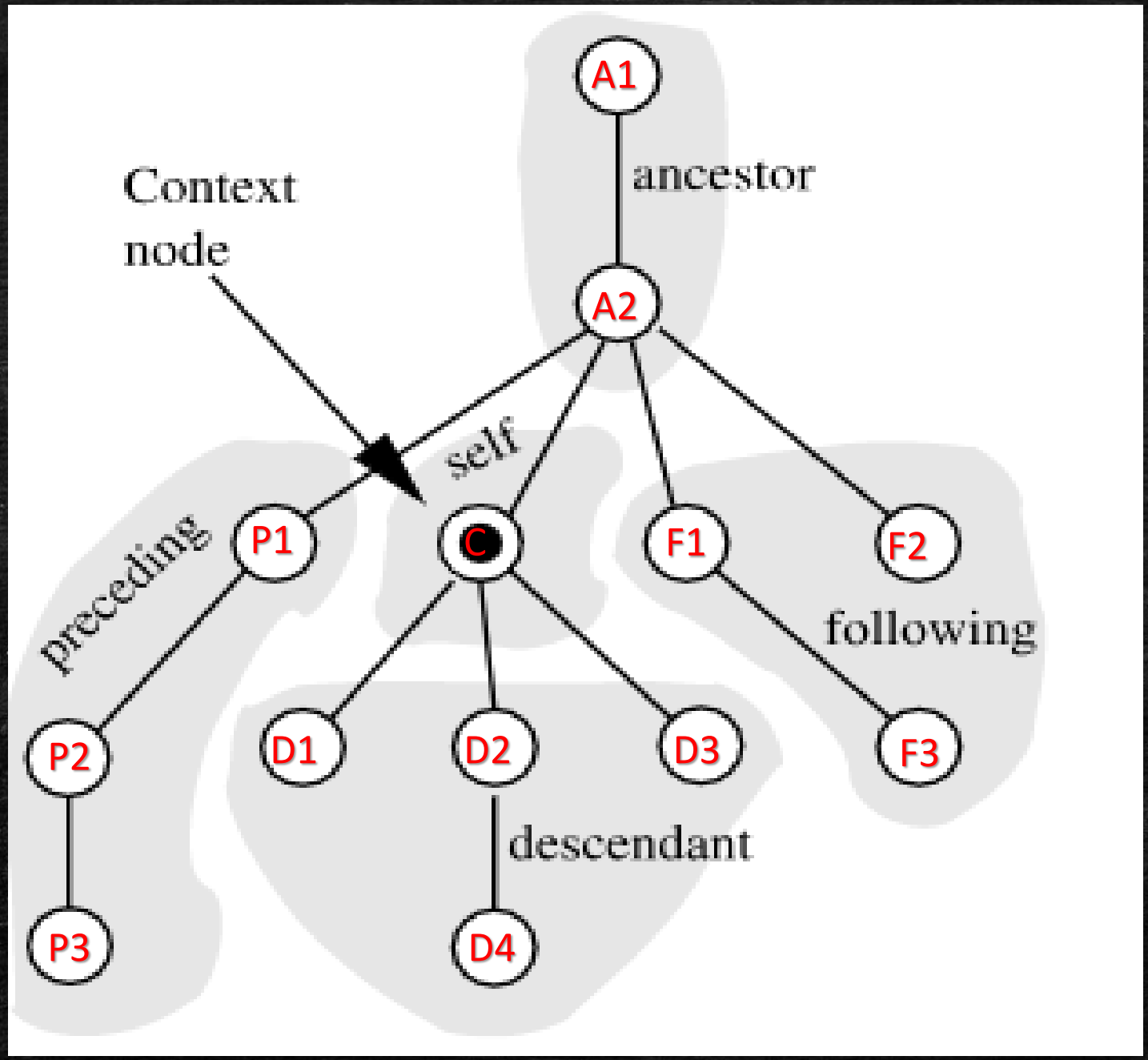
text()

//span[text()='Message-123']



XPath axes methods

- XPath methods are used to find the complex or dynamic elements.
  - Ancestor
  - Child
  - Parent
  - Preceding
  - Following
  - Self
  - Descendant



Ancestor

- selects all ancestors element (grandparent, parent, etc.) of the current node

//employee/ancestor::*	Select all ancestor node of the employee node.
//ancestor::name	Select all ancestor of the name node in context node.

Child

- select child of the context node.

child::*	Select All child nodes of the context node.
child::employee	Select all child elements of employee node.

Parent

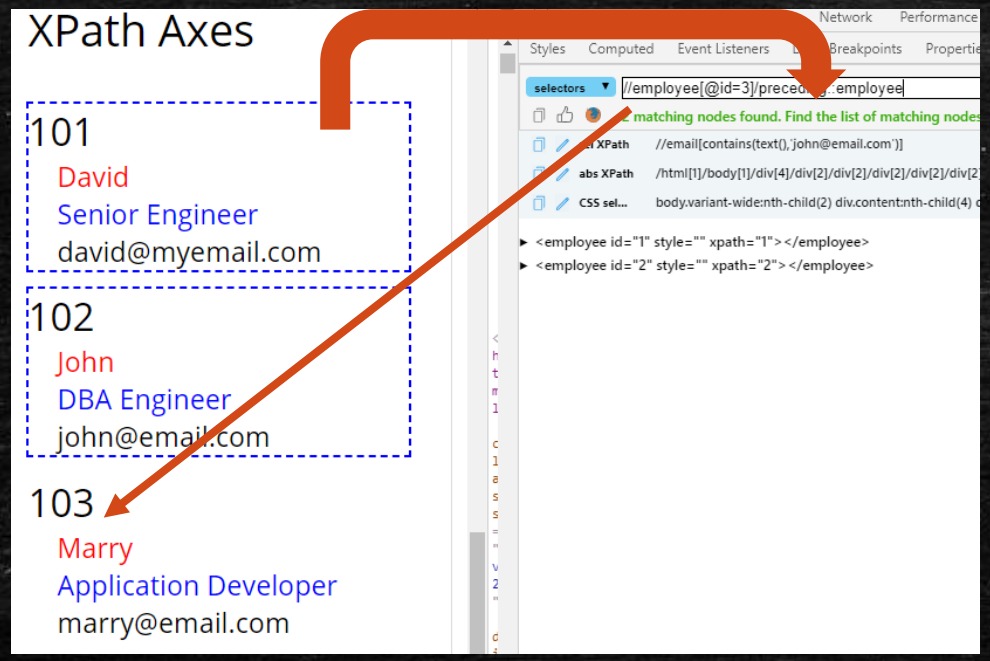
- select the parent node of the context node.

//name/parent::*	Select parent node of the 'name' context node.
//email/parent::employee	Return result node if employee node is parent node of the context node, otherwise no node found.

Preceding

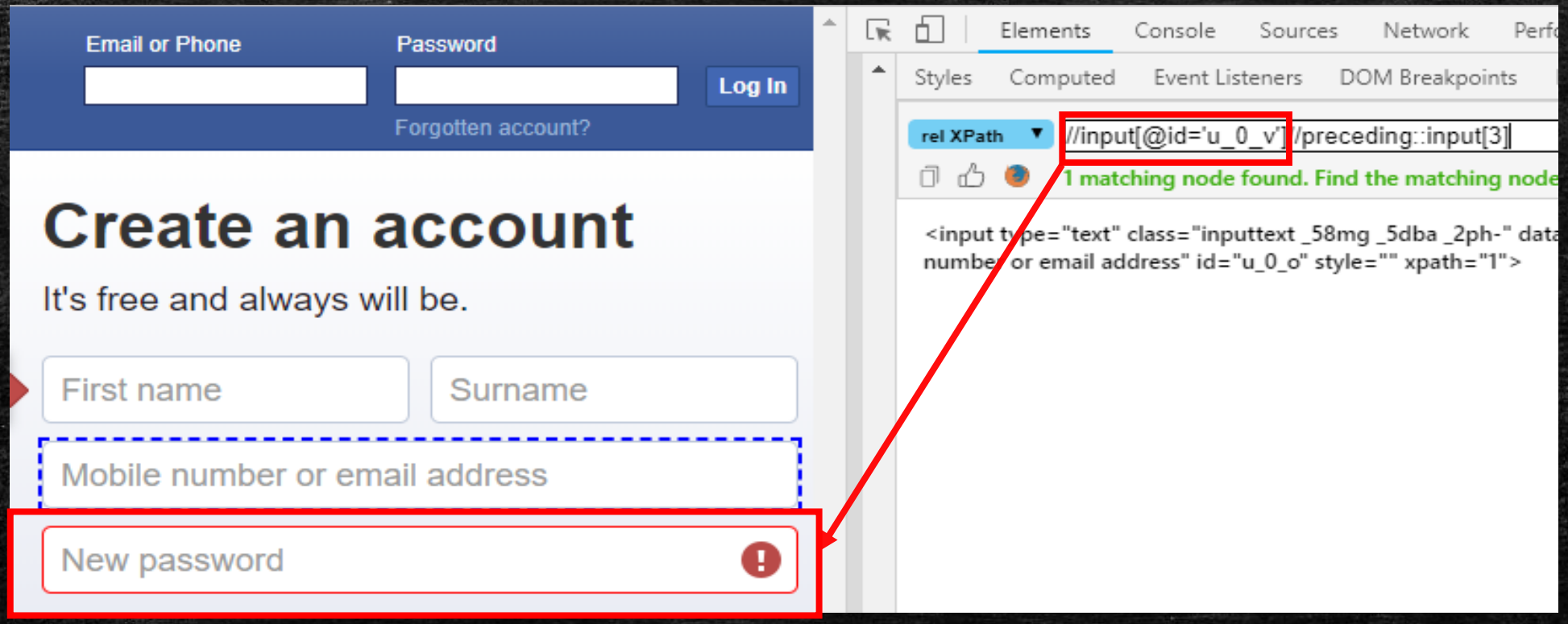
- select all nodes before the context node, excluding attributes node or namespace

//employee[@id=3]/preceding::employee	Select all nodes (with child nodes) before the context node.



Preceding...

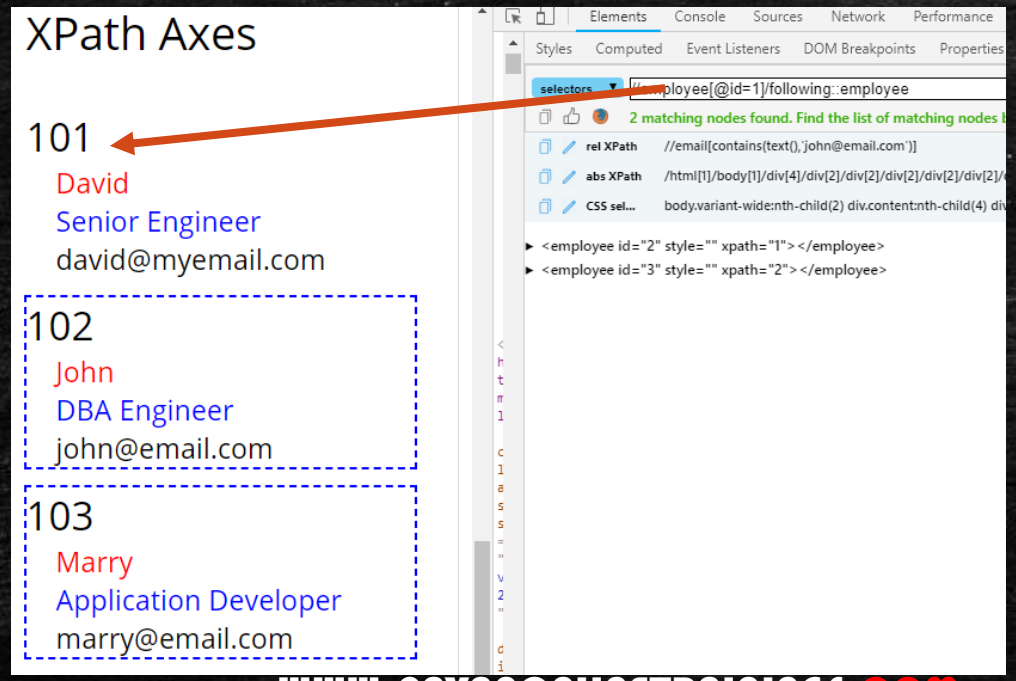
//input[@id='u_0_v']//preceding::input[3]



Following

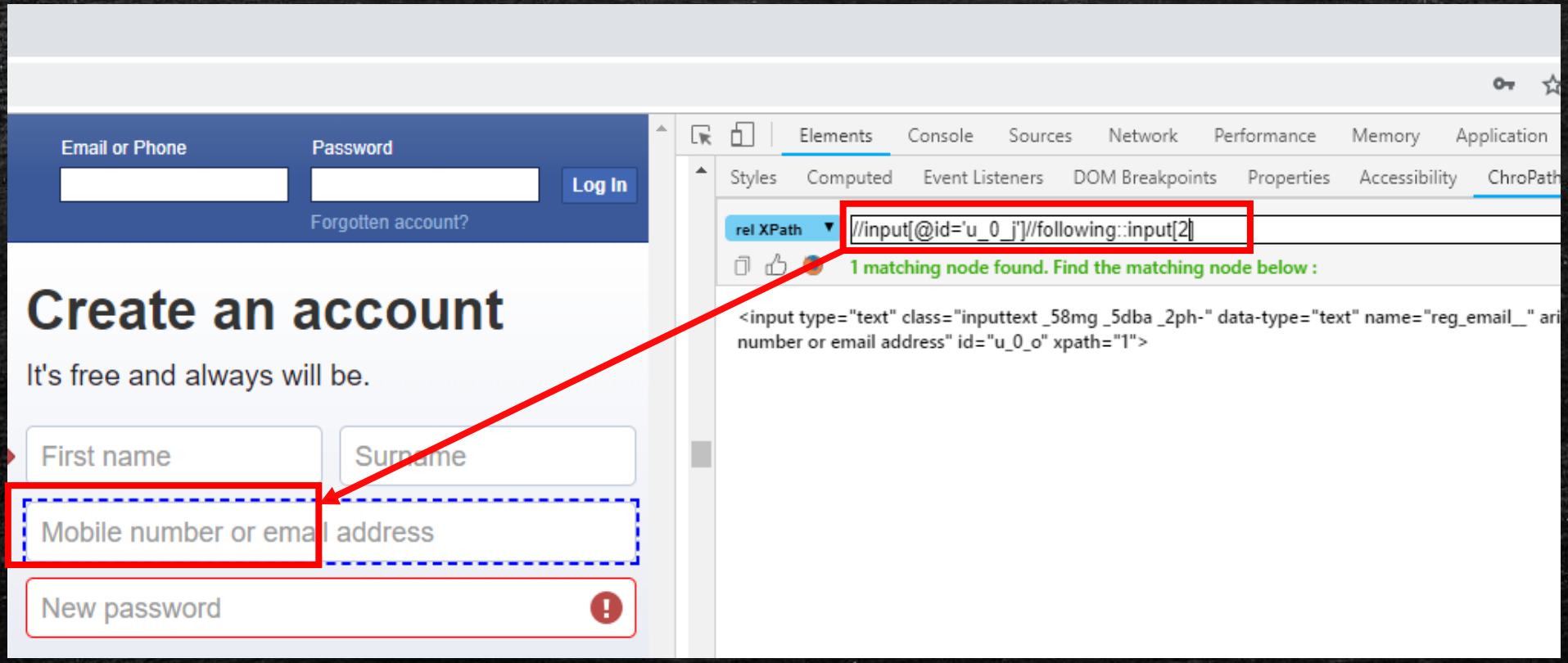
- select all nodes after the context node, excluding attributes node or namespaces node.

//employee[@id=1]/following::employee	Select all nodes (with child nodes) after the context node.



Following..

//input[@id='u_0_j']//following::input[2]



Self

- Selects the current node ‘name’

  //name/self::*

Descendant

- Selects all descendants (children, grandchildren, etc.) of the current node

//descendant::employee	Select all descendant of the employee node in context node.