

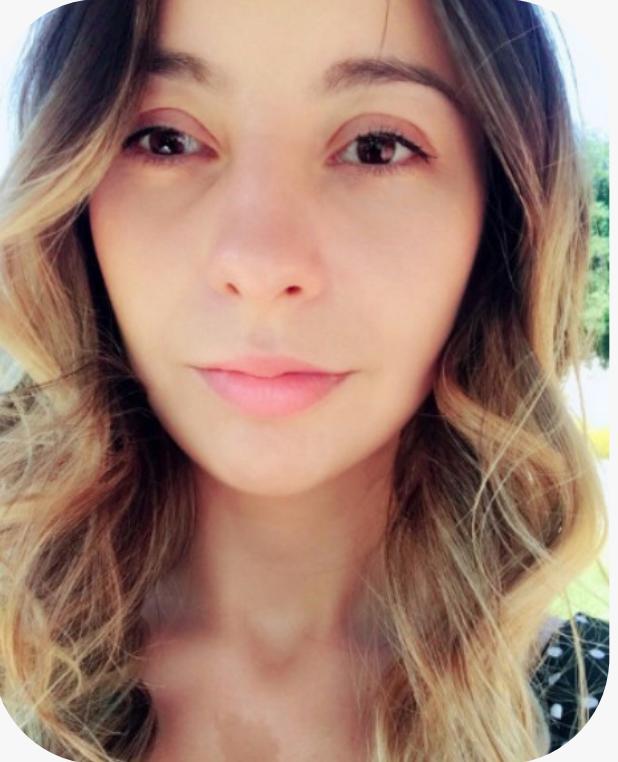


Lesson 2. QA Automation

- First test case creation → Locators (XPATH)
- Browser launch, find_element, send_keys,
clear, click commands

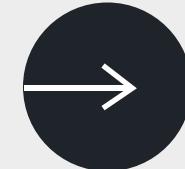
Teacher

careerist



Lana Levinsohn

8+ years in all kinds of manual
and automated testing



[Linkedin](#)

Selenium

careerist



→ What is Selenium?

Selenium is a suite of tools for browser automation

→ What is Selenium WebDriver?

A collection of language specific bindings to drive a browser
W3C WebDriver:

<https://www.w3.org/TR/webdriver/>

→ What can Selenium do?

Selenium automates browsers
(<https://www.selenium.dev/>)

→ What is Selenium IDE?

A Chrome and Firefox add-on that will do simple record-and-playback of interactions with the browser

Locators. Types:

careerist

By ID – ids of the elements should be unique for the page; this is the most efficient and fast way to locate an element

By XPATH – locates an element using DOM structure of the page. This is the type of locator that is always available for an element, even more, there are usually various possible XPATHs

By CSS Selector – CSS selectors are another alternative of finding elements. This is useful for locating items that have a unique style on the page

By Class – class name locator finds the element which matches the value specified in the attribute name “class”

By Name – if there is no id to use in HTML code, another preferred way to locate the web element is ‘name’ attribute

By TagName – this kind of locator is used to find the element matching the specified Tag Name

By LinkText – if there are any link on the page, it's very easy to locate them, but make sure, there is one unique link on that web page.

By Partial LinkText – this locator works the same way as the LinkText one, but it gives user the opportunity to handle more dynamic links from the pages, as the element can be find by a partial link text

Locators. XPATH

careerist

To test XPATH in a browser, use `$x()` command in your browser console.

When writing XPATH expression we use forward slash '/'. The first '/' represents beginning of the html tree which is called as root. After every '/' we should specify tagName of the immediate child element only.

There are 2 main different types of path:

- 1 **Absolute Path**
This one is the complex Path of the element starting from the root. 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.
For example: /html/body/div/div/input/div/div/div[3]/input[2]
- 2 **Relative Path**
In this case 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. This one reduce the length of Path expression. It's much better to use it, because in the moment when any changes will be introduced to the source code, our Xpath will still be valid.
For example: //[@class='actions']//*[text()='testing']*

Locators. XPATH. Syntax

careerist

XPath contains the path of the element at the web page.

Standard syntax for a XPath:

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.

[Source](#)

Commands

careerist

- **get('https://url')** - opens a web page
- **find_element()** - searches for an element, throws Exception if element not found
- **click()** - clicks
- **send_keys()** - inputs keys into a field
- **clear()** - clears input field
- **quit()** - exits browser

Homework 2

careerist

Homework 2
description and tasks

Questions?



careerist

Take your career
to the next level

advisors@careerist.com

www.careerist.com