Table of Contents

[Files 3](#_Toc105937832)

[C:\Users\lauri\OneDrive\Työpöytä\Learn\_RF 3](#_Toc105937833)

[Links 3](#_Toc105937834)

[Unique Locators Section 4 3](#_Toc105937835)

[CSS & XPath Checker – Chrome 3](#_Toc105937836)

[Selenium IDE – Firefox 3](#_Toc105937837)

[Attributes: 3](#_Toc105937838)

[Use of Plugins: 3](#_Toc105937839)

[CSS locators 4](#_Toc105937840)

[With classes .class and with attributes type 5](#_Toc105937841)

[Xpath 5](#_Toc105937842)

[Xpath with \* 6](#_Toc105937843)

[Use Xpath with innertext 6](#_Toc105937844)

[Use Xpath with partial innertext 6](#_Toc105937845)

[Use Xpath with partial attribute value 6](#_Toc105937846)

[Locate element through its parent 6](#_Toc105937847)

[Opposite way – from Child to Parent 6](#_Toc105937848)

[Locate element through its siblings 7](#_Toc105937849)

[Locate element through parent | child | siblings 7](#_Toc105937850)

[FOR loop in with ROBOT 7](#_Toc105937851)

[Run IF 7](#_Toc105937852)

[Set Selenium Speed / sleep 7](#_Toc105937853)

[Set Selenium Implicit Wait 7](#_Toc105937854)

[Capture Snapshot 8](#_Toc105937855)

[Scroll Down – javascript 8](#_Toc105937856)

[Mouse with RF 8](#_Toc105937857)

[Keyboard Operations 8](#_Toc105937858)

[Wait commands with RF 8](#_Toc105937859)

[Section 14 9](#_Toc105937860)

[String Handling 9](#_Toc105937861)

[Fetch Substring 9](#_Toc105937862)

[Common String Functions 9](#_Toc105937863)

[List 9](#_Toc105937864)

[Tuple 9](#_Toc105937865)

[Dictionary (key, value) 10](#_Toc105937866)

[Functions 10](#_Toc105937867)

[Section 16: User Defined Keys Using Python Scripting 10](#_Toc105937868)

[Section17: Classes in Python 10](#_Toc105937869)

[Constructors 10](#_Toc105937870)

[Why and where we use constructors 10](#_Toc105937871)

[Modules 10](#_Toc105937872)

[Create Project Structure 11](#_Toc105937873)

[Difference between import and from-import 11](#_Toc105937874)

[Exception Handling in Python 11](#_Toc105937875)

[Section18: Read Excel Data (OpenPyXL -package) 11](#_Toc105937876)

[Section19: Work with JSON Data 11](#_Toc105937877)

[Json Path Basics 11](#_Toc105937878)

[Work with JSON 11](#_Toc105937879)

[Fetch and validate JSON response 12](#_Toc105937880)

[Section20: Advance Robot Skills 12](#_Toc105937881)

[Section21 Data Driven Testing 12](#_Toc105937882)

[Section22 Code Management using GitHub 12](#_Toc105937883)

# Files

# C:\Users\lauri\OneDrive\Työpöytä\Learn\_RF

# Links

# Unique Locators Section 4

### CSS & XPath Checker – Chrome

### Selenium IDE – Firefox

* <https://thetestingworld.com/testings/>

## Attributes:

<input class="field" name="fld\_username" required="" type="text" placeholder="myusername" value="" >Hello</input>

Tags, Attributes, Inner Text

## Use of Plugins:

* <https://www.facebook.com/reg/>
* <https://fi-fi.facebook.com/>

name=email, pass

id=email,

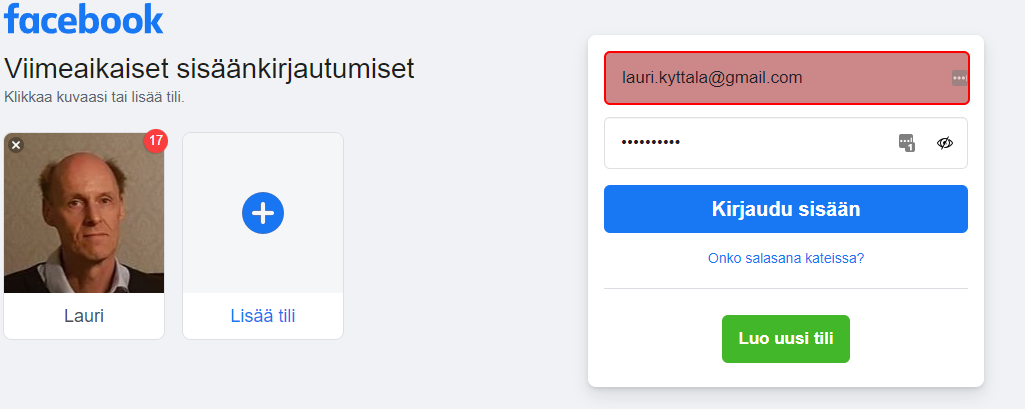
link=Onko sinulla jo tili?

## CSS locators

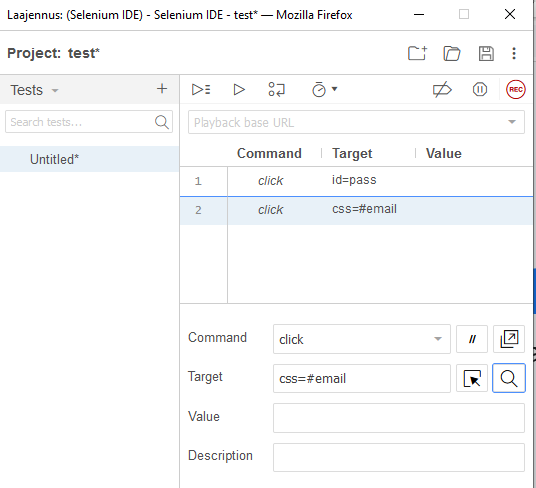
<https://fi-fi.facebook.com/>

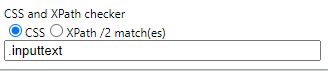
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti



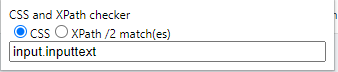
***with firefox:***





.inputtext

or



input.inputtext

[type=’text’]

input[type='text']

input#pass[type='password'] / #pass[type='password'] / #pass

## With classes .class and with attributes type

e.g. (.inputtext[type=’email’])

#email

#pass

## Xpath

<https://thetestingworld.com/testings/>

//input[@name='fld\_username']

//input[@name='firstname' or @aria-label='Etunimi']

//select[@name='country' or @class='last country']

//select[@name='birthday\_day' and @title='Päivä']

## Xpath with \*

//\*[@\*='birthday\_day' and @\*='Päivä']

//input[@type='text'] or //\*[@type='text']

## Use Xpath with innertext

<https://www.facebook.com/reg/>

//div[text()='Luo uusi käyttäjätili']

//a[@aria-label='Onko sinulla jo tili?']

## Use Xpath with partial innertext

<https://www.facebook.com/reg/>

//div[contains(text(),'uusi käyttäjätili')]

## Use Xpath with partial attribute value

<https://www.facebook.com/reg/>

//input[contains(@type,'pass')]

## Locate element through its parent

<https://www.amazon.com/>

//table[@class='navFooterMoreOnAmazon']/tbody/tr[1]/td[1]

<https://thetestingworld.com/testings/>

//ul[@class='tabs blue']/li[2]/label

## Opposite way – from Child to Parent

* didn’t find example



## Locate element through its siblings

//input[@id='tab2']/following-sibling::label

(following or preceeding)

## Locate element through parent | child | siblings





# FOR loop in with ROBOT

    FOR    ${i}    IN RANGE    1    10

        log to console  ${i}

    END

    @{List1}    create list    Hello    22    23.23    World!    Audio

    FOR    ${i}    IN    ${List1}

        log to console  ${i}

    END

# Run IF

    ${var}=    set variable    NO

    run keyword if    '${var}'=='YES'    log to console    Value found

    run keyword if    '${var}'=='NO'    log to console    Value Not found

# Set Selenium Speed / sleep

    Set Selenium Speed    2s

sleep 5s

# Set Selenium Implicit Wait

    set selenium implicit wait    10s

# Capture Snapshot

capture page screenshot    ./Snapshots/TC1.png

# Scroll Down – javascript

    execute javascript    window.scrollTo(0,1000)

# Mouse with RF

    # open context menu    xpath://span[contains(text(),'VIDEOS')]

    # double click element    xpath://a[text()='Login']

    # mouse down    xpath://a[text()='Login']

    # mouse up    xpath://a[text()='Login']

    mouse over    xpath://span[contains(text(),'VIDEOS')]

# Keyboard Operations

    press key    name:username    lasse

    press key    xpath://button[@type='submit']  \\13

    # ascii value ofenter key

# Wait commands with RF

wait until page contains

wait until page contains element

wait until element contains

wait until element visible

wait until element enable

# Section 14

* For loop with final range
* For loop with starting and final range
* For loop with increment
* For loop with decrement
* For loop with list
* Break Statement
* Continue Statement

## String Handling

* Concatenation
* + and \*

## Fetch Substring

* Fetch substring by given index
* Start and end index both
* Only start index
* Only end index

## Common String Functions

* Len
* Capitalize
* Upper
* Lower
* Lstrip
* Rstrip
* Strip
* Replace
* Find
* Split

## List

* Len
* Cmp
* Concatenate

## Tuple

* Cannot increase, or change values

## Dictionary (key, value)

* Key and value pair
* Key must be unique

## Functions

def takeInput3(a=100, b=10, e=7):

    c=a-b

    print("Subtraction of values: " + str(c))

takeInput3(15)

# Section 16: User Defined Keys Using Python Scripting

* Without argument
* With argument but no return value
* Argument and return value

# Section17: Classes in Python

## Constructors

* Special type of method
* Created with \_\_init()\_\_, first argument is always self
* Automatically called when object is created
* Can take arguments
* Can’t return any value
* Constructors are used for initialization

## Why and where we use constructors

* e.g. database usage (opening connection only once)

## Modules

* can be define as python files
* can have executable code, module, functions, classes
* classes can also have functions, properties (variables) and constructors

## Create Project Structure

* with modules
* with directories
* classes and executable code

## Difference between import and from-import

* use import statement
* use from module import class

## Exception Handling in Python

* try, except, finally

# Section18: Read Excel Data (OpenPyXL -package)

* pip install openpyxl

# Section19: Work with JSON Data

* JavaScript Object Notation a lightweight data interchange format
* JSON is a syntax for storing and exchanging data
* Data is placed in JSON in the format of Key and Value pair
* Value could be an array
* Value can have further key-value (Call it object)
* {
* "Name":"Testing world",
* "Age": 23,
* "PhoneNumber": "+358 644 231"
* }

## Json Path Basics

* How to use simple Json path (<https://jsonpath.com/>) e.g. $.Address.Number.Mobile
* How to write complex Json path with array
* How to write complex Json path with Object

## Work with JSON

* Parse dictionary to JSON
* Parse JSON to dictionary

## Fetch and validate JSON response

* Send request to API
* Parse response to JSON
* Validate by JSON Path
* <https://reqres.in/api/users?page=2>

# Section20: Advance Robot Skills

## Element Locators

* Using element locators
* TC\_002\_Start\_Close\_Browser.robot is working

## BDD format test cases (Gergin)

* Example
* \*\*\* Test Cases \*\*\*
* Test Case in BDD Format
* Given Start Browser and Maximize
* When Create Folder at Runtime    robot12    test
* Then Concatenate Username and Password    lasse    123456

## Record Test Cases

* Install robotcorder to chrome
* Might be the case you have to update locators
* Save the recorded file, and sections and libraries are added – copy saves file to new robot test file
* Scan scans whole page
* with settings (add sleep, check page contains)

# Section21 Data Driven Testing

* <https://thetestingworld.com/testings/>

# Section22 Code Management using GitHub