**Libraries**

**Selenium**

**SeleniumLibrary**

SeleniumLibrary is a web testing library for Robot Framework.

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>

**XPath**

XPath란 XML Path Language를 의미합니다.

XPath는 XML 문서의 특정 요소나 속성에 접근하기 위한 경로를 지정하는 언어입니다.

XPath는 W3C 표준 권고안으로, XSLT와 XPointer에 사용될 목적으로 만들어졌습니다.

또한, XML DOM에서 노드를 검색할 때에도 사용할 수 있습니다.

특징

1. 1. XPath는 XML 문서를 탐색하기 위해 경로 표현식(path expression)을 사용합니다.
2. 2. XPath는 수학, 문자열 처리 등을 하기 위한 표준 함수 라이브러리를 내장하고 있습니다.
3. 3. XPath는 W3C의 표준 권고안인 XSLT에서 가장 중요한 부분 중 하나입니다.

**XPath를 이용한 웹에서 데이터 수집**

<https://www.youtube.com/watch?v=l0QNyBBWY50&ab_channel=%EA%B3%BD%EA%B8%B0%EC%98%81>

* 21:00 까지 play

(본 영상은 R을 이용한 방법이나, 절차를 활용하여 시도해 볼 필요성은 충분히 있음

beautifulSoup, 또는 다른 방법으로 페이지를 읽어 이를 file로 저장하는 방법)

XPath를 이용하여 데이터 수집하기 위해서는

1. 해당 웹 페이지를 읽어 들여 html 문서로 저장
2. XML parsing
3. XPath를 이용하여 웹 페이지 상의 원하는 데이터에 접근하기 위해서는 html 계층 구조를 이해/파악하여야 함 🡪 web browser의 요소(Elements) 탐색기를 이용하여web browser의 콘텐츠와 소스 코드의 대응관계를 쉽게 파악 가능
4. 콘텐츠를 특정하기 위해 tag를 지정하는데, unique하게 특정하기 위하여 상위 tag를 함께 이용

Doc=*<html file>*, Path=*”//article//h4”* 🡪 html file 내 “…/article/…/h4” 의 의미

Doc=*<html file>*, Path=*”//article[@class=’content’]//h4”* 🡪 html file 내 <article class=’content’>인 article/…/h4” 의 의미

**XPath 경로 표현식 (path expression) 이해 필요**

**XPath 표현식 (Expression)**

#### 위치 경로(location path)

#### 위치 경로란 XML 문서의 각 노드의 위치를 지정하기 위한 XPath 표현식

| **경로 연산자** | **설명** |
| --- | --- |
| 노드 이름 | 해당 '노드 이름'과 일치하는 모든 노드를 선택함. |
| / | 루트 노드부터 순서대로 탐색해 나감. |
| // | 현재 노드의 위치와 상관없이 지정된 노드에서부터 순서대로 탐색해 나감. |
| . | 현재 노드를 선택함. |
| .. | 현재 노드의 부모 노드를 선택함. |
| @ | 속성 노드를 선택함. |

#### 경로 표현식 (path expression)

#### XPath에서는 노드 선택을 위해 경로 표현식(path expression)을 사용

#### 문법

#### 검색방향::노드테스트[필터표현식]

#### 1. 검색 방향 : 현재 노드를 기준으로 노드를 검색할 방향을 전달합니다.

#### 2. 노드 테스트 : 검색 방향에 존재하는 해당 노드를 검색합니다.

#### 3. 필터 표현식 : 검색된 노드셋(node-set)에서 특정 노드나 특정 값을 포함하는 노드를 선택합니다.

#### 필터 표현식 (filter expression)

#### XPath에서는 특정 노드나 특정 값을 포함하는 노드를 선택을 위해 필터 표현식(filter expression)을 사용

#### 필터 표현식은 언제나 대괄호([]) 안에 표현.

#### XPath 함수

#### XPath 함수는 선택된 노드 셋을 평가하여 조건에 맞는 노드를 시퀀스 형태로 반환

#### 임의 문자 기호 (wild card)

|  |  |
| --- | --- |
| 임의 문자 기호 | 설명 |
| \* | 어떠한 요소 노드와도 일치함 |
| @\* | 어떠한 속성 노드와도 일치함 |
| node() | 어떠한 종류의 어떤 노드와도 일치함 |
| text() | 어떠한 텍스트 노드와도 일치함 |

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **Dfjdhfj** | **djhfdjf** |
|  |  |  |  |

**MiPlatform에서의 어려움**: MiPlatform은

xml 형식으로 <Window> tag 안에 <script> 가 혼재된 상태로 Browser로 다운됨

페이지 이동시 URL 등 정보가 표시 안됨

그리고, 어떤 페이지는 source 접근이 안됨

또한, 화면 상에서 오른쪽 버튼이 동작 안함

**Training course and Certifications** - <https://robocorp.com/docs/courses>

(중요: 실체를 이해하는 데 필요 과정)

Completing the 1) [Beginners' course](https://robocorp.com/docs/courses/beginners-course) will grant you the Robocorp Level I certificate! If you want more challenge, try the Level II certificate course, 2) [Build a robot](https://robocorp.com/docs/courses/build-a-robot)! Learn even more by taking the level III certificate course, 3) [Work data management](https://robocorp.com/docs/courses/work-data-management). View [all Robocorp courses](https://robocorp.com/docs/courses) to learn more.

1. **Setup**

**환경 준비**

<https://koreapy.tistory.com/1129> 를 참조, python3.9를 삭제하고, conda가 제공하는 가상 환경으로 구성

1. ~~Python 3.9 설치 (3.9.13, 위치: c:\python\python39)~~ 🡪 삭제했음
2. VS Code 설치 (April 2022 (version 1.67))
3. Conda 설치 (4.12.0, 위치: c:\users\LnY\anaconda3)
   * 가상환경 ‘pyenv39’ 생성하였음 (기본 ‘base’도 사용 가능한 듯)
   * $> conda activate pyenv39 혹은 base 사용 시 $> conda activate
   * $> conda deactivate
   * $> conda env list
   * $> conda create -n *venvName* python=3.9
   * $> conda update –all
   * $> conda env remove -n *venvName*
   * Ctrl+Shift+P 하여 “Python: select interpreter” 선택 🡪 pyenv39 또는 base 선택
4. VS Code에 Robocorp extension 설치 (Robocorp code, Robot Framework Language Server)

**Default Structure**

* conda.yaml
  + controlling environment where the robots run.
  + 포함되는 package 정보
* robot.yaml
  + 실행 command 정의 및 위치
  + 2 mandatory parts
  + At least one task defined, and that must have either command, shell or robotTaskName defined.
  + Artifacts output path must be defined.
* tasks.robot 또는 task.py
  + 실행 action

1. [**Robocorp extension**](https://marketplace.visualstudio.com/items?itemName=robocorp.robocorp-code)**for**[**Visual Studio Code**](https://code.visualstudio.com/)

<https://robocorp.com/docs/developer-tools/visual-studio-code/overview>

With the [**Robocorp extension**](https://marketplace.visualstudio.com/items?itemName=robocorp.robocorp-code)**for**[**Visual Studio Code**](https://code.visualstudio.com/), you can create, run, and debug robots, publish to Control Room, read secrets from a Control Room vault, use work items, locate UI elements in browsers and desktop applications using a visual inspector, and enjoy other powerful features.

The [**Robot Framework Language Server**](https://marketplace.visualstudio.com/items?itemName=robocorp.robotframework-lsp)**extension** provides code completion, syntax validation and highlighting, code formatting, and other powerful robot development features.

**\*\*\* Settings \*\*\***

* Documentation, possibly some libraries, references to other files

\*\*\* Settings \*\*\*

Documentation Template robot main suite

\*\*\* Tasks \*\*\* -- suite

Minimal task -- task

Log Done -- keyword + argument

Open the intranet site

\*\*\* Keywords \*\*\*

Open the intranet site -- New keyword

**\*\*\* Task \*\*\***

* Define tasks
* “Minimal task” is the name of the task.

**Keyword** (=Function)

* Provide the robot with specific skills
* Input Text *locator* *text*

**Library**

* Teach the robot new skills by making new keyword available to them
* added in the \*\*\* Settings \*\*\* section of your .robot file.

**Template**

* Extended
* Python
* Standard

1. **Course I: Beginner’s course**

**Robot Framework**

<https://robocorp.com/docs/courses/beginners-course/collecting-the-results>

다양한 종류의 locator(예: id, name, xpath 등등)가 있는데, 이를 사용하는 방법

[How to find user interface elements using locators in web applications](https://robocorp.com/docs/development-guide/browser/how-to-find-user-interface-elements-using-locators-in-web-applications)

투비소프트 XPlatform 테스트 시 꼼꼼히 검토할 필요 있음

(참조 자료: <http://docs.tobesoft.com/admin_guide_xplatform_ko#1977446c60481709>)

대상 object를 right-click 🡪 검사 🡪 (DevTools 표시되고, object 부분 하이라이트) 🡪 right click 🡪 copy 🡪 Copy element, Copy selector, Copy XPath, Copy full XPath 등을 확인

Css. Div 등의 tag를 이용하여 정보 수집

We have a div element. It has class and role attributes. Since there are no id or name attributes to work with, we decide to use the [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) classes to target it. The sales-summary class seems like a good candidate because then it will be clear in our script what we are taking a screenshot of. Because we use CSS to locate the element, our locator will start with the css: prefix. There are multiple types of locators: id, name, xpath, depending on the [strategy](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Explicit%20locator%20strategy) used.

The css:div.sales-summary locator means: Using the CSS strategy, find me a div element that has the sales-summary CSS class.

<**div** class="alert alert-dark **sales-summary**" role="alert">

<div><span>Active sales people:</span>...</div>

</div>

${sales\_results\_html}= Get Element Attribute id:sales-results outerHTML

We create a variable (${sales\_results\_html}=). We store into it what we get out of the [Get Element Attribute](https://robocorp.com/docs/libraries/rpa-framework/rpa-browser-selenium/keywords#get-element-attribute) keyword. We pass two arguments to that keyword: the first is the locator for the element (id:sales-results); the second is the name of the attribute of the element we want to get.

**Running the robot in Control Room**

Control Room is very powerful and can support very complicated workflows.

To set this all up, we will have to go through these steps:

* Logging into Control Room (<https://id.robocorp.com/login>)
* Creating a new Organization for RobotSpareBin Industries
  + [Organizations](https://robocorp.com/docs/control-room/administration/organizations-and-workspaces) allow you to group users, workspaces, processes, assistants, and robots.
* Creating a new Workspace
* Creating a new Robot
* Uploading the robot code
* Creating a Process (the "assembly line", remember?)
* Adding the Step to the Process and setting it up
* Executing and scheduling the process.

**Other options for running the robot**

Running the robot on **1) your local machine** with Visual Studio Code, or **2)** [**RCC**](https://github.com/robocorp/rcc), or **3) in Control Room** are not the only options.

* + **RCC** (Robot Creation Command) is a set of tooling that allows you to create, manage, and distribute Python-based self-contained automation packages - or 'robots' as we call them.

If your robot needs to be run on a physical (or a virtual) machine to access, for example, desktop applications, you can use our applications to that:

* + Use **4)** [**Robocorp Workforce Agent**](https://robocorp.com/docs/control-room/configuring-workforce/overview)**for unattended** cases where the robot can work in the background without human intervention.
  + Use **5)** [**Robot Assistants**](https://robocorp.com/docs/control-room/operating-assistants)**for attended** cases where the robot can work together with humans to complete automated tasks!

**The same robot in Pure Python and with Playwright**

(<https://robocorp.com/docs/courses/beginners-course/python-robot>)

Pure Python

* + In conda.yaml, define dependencies and the initialization command:

node.js, rpaframework and robotframework-browser

* + robot.yaml:

robot.yaml configuration will handle executing the robot (ex. task.py):

* + python program:

may use the Playwright-based Browser library or Selenium library for the browser automation

Playwright

* + MicroSoft의 새로운 오픈 소스 자동화 테스트 도구

1. **Course II:** [**Build a robot**](https://robocorp.com/docs/courses/build-a-robot)

<https://robocorp.com/docs/courses/build-a-robot>

1. **Course III: Work Data Management**
2. **Project 방법론**
3. Software robot delivery process

* The initial idea of automating something 💡
* Evaluating whether the automation is technically feasible 🤔
* Calculating the return on investment (cost vs. benefits) 💰
* Documenting the process ✏️
* Implementing & testing the robot ⌨️
* Launching the robot to production 🚀
* Maintaining the robot ⚙️

1. Tips for finding potential processes to be automated

* What is most of the people in the organization working on? More people -> more work -> more tasks to automate.
* Ideas for automation can come from the person completing the process or someone else involved in the process.
* You can try [process mining](https://en.wikipedia.org/wiki/Process_mining) to help with analyzing and identifying potential processes with automation potential.
* Ideation workshops can be a valuable source of great automation ideas. Gather people together an let your imagination run wild!
* You might have an idea for a process that would improve efficiency but can not be completed even manually at the moment. Sometimes automation enables you to start creating entirely new processes instead of just automating existing ones.
* See Figure 12. "Distribution of use cases by complexity of implementation and benefit realized across functions and sectors" in the excellent [RPA use cases research report by Capgemini Research Institute](https://www.capgemini.com/wp-content/uploads/2018/10/Automation-Use-Cases_Digital1.pdf) (PDF file).
* Collect all the possible automation ideas in a backlog. Then decide which process makes sense to start with: depending on your use case, it could be the most straightforward process to automate or the one you think will have the most significant ROI. Do not just pick the first one that you find: having good first experiences is essential, both for people and for organizations! 🙂

1. Write Process Definition Document

* What is the process used for, and what is its end goal?
* What are the steps involved in the process? Are there any decision points?
* Who is in charge of the process?
* By whom and how often is the process executed?
* What are the systems involved in the process?
* Does the user need special authorization or roles to run the process?
* What are the possible problems and exceptions that can happen during the procedure? How does the user handle those?
* What is process flow?

1. **Keymap**

* Open the Command Palette by pressing Shift-Command-P (macOS) or Ctrl+Shift+P (Windows).
* (Optional): Type run robot. This will find the command you need.
* Press Enter to run the Robocorp: Run Robot command.
* Toggle (hide/show) the left panel: Command-B (macOS) or Ctrl+B (Windows)
* Toggle (hide/show) the bottom panel: Command-J (macOS) or Ctrl+J (Windows)
* Zoom out: Command and - (macOS) or Ctrl and - (Windows)
* Zoom in: Command and + (macOS) or Ctrl and + (Windows)