**Dev Guide**

Robot Framework provides a higher level of abstraction and will make your life easier.

And, because Robot Framework is based on Python, you can also [easily combine the two approaches](https://robocorp.com/docs/development-guide/robot-framework/how-to-use-custom-python-libraries-in-your-robots).

Python을 사용하는 경우, RPA framework을 사용하는 경우

(둘 모두 경험이 쌓이면 기 개발 환경은 재 사용하고, 이로 인해 개발 공수가 적게 필요한 것은 배제한 상황에서)

(또 기능적으로 excel, web, pdf 등 다양한 업무환경에서)

|  |  |
| --- | --- |
| Python을 사용하는 경우 | RPA framework을 사용하는 경우 |
| 각 업무에 필요한 API 를 개발자가 숙달해야 하고,  여기서 API라 함은 primitive한 기능으로, 한 개 업무 기능을 구현하기 위하여는 여러 개의 primitive api 호출이 필요 | 어느 정도 많이 사용하는 기능을 keyword 로 이미 정의하여 놓아, 개발자는 직관적으로 필요 기능 파악이 용이하고,  이 하나의 keyword는 여러 개의 primitive API 호출과 동일한 기능을 수행 |
| 업무 자동화에는 Web, Excel, PDF, 등등의 기능이 많이 사용되고, python에서 필요한 기능을 위한 다양한 종류의 library의 api 호출에 대한 숙달 필요 |  |
| 개발 뿐 아니라 유지보수에서도 상대적으로 많은 공수가 필요 | 코드의 가독성이 python 코드보다 좋아 개발자 변경 시 적응이 빠르리라 예상 |
|  | 필요한 기능이 제공 안되는 경우도 발생 예상 |

Python, RPA Framework이 서로 장단점이 있으나, 이 둘이 기능적으로 서로 배타적이지 않고, 가령 RPA framework에서 제공하지 않는 기능은 python으로 개발하여 연동할 수 있으므로, 상호 보완적이라 할 수 있고,

조금 다른 비유이지만, 솔루션 vendor들이 많이 사용하는 용어 중에 “Build vs. Buy”라는 것이 있었죠.

많은 인력, 시간을 들여서 build할 것이냐, 좋은 솔루션을 buy하여 빠른 시일에 구현하고 업무에 빨리 활용할 것이냐? 의 선택이었는데, 요즘의 비즈니스 환경을 보면 빨리 활용하고, 시간이 조금 지나 업무 환경이 바뀌면 다시 구현하는 것이 좋다는 것이죠.

타 RPA 솔루션과의 비교는 (마케팅 관점이 아닌, 개발자 관점에서는)

(아직 전체 기능을 모두 사용해 보지 않은 상태에서 판단이라 왜곡된 정보가 있을 수 있으나,)

|  |  |
| --- | --- |
| 타RPA 솔루션 | Robocorp 솔루션 |
| 사용자가 사용하는 화면을 직접 클릭하여 작업. 따라서 RPA 개발은IT 개발 경험이 없어도 어느 정도 가능하다고 함 | Web 경우에도 화면을 클릭하는 것이 아닌 tag 정보를 파악하여 이에 대한 정보를 가져오는 방식이라 개발 경험 등이 필요 |

IT 개발 경험이 없는 인력(현업인력)도 RPA 개발이 가능하다고 하지만, 현실은 결국 개발자를 고용하여 진행해야 하는 상황에서 개발자 cost는 동일하다고 하면, 결국 license 비용에서 큰 차이가 날 것이고(이건 마케팅 관점인데…), 개발자 career path를 고려하면 Robocorp 솔루션은 개발자 경력에 python, selenium, ~~ 업무 자동화 등으로 한 줄 이력이라도 추가할 수 있어 조금이라도 더 개발자에게 유리한 솔루션이 아닐까 생각되죠.

**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. **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)