9.1

[Python](https://saturncloud.io/glossary/python) is an open source programing lanhuage that include increasing libraries that can not be installed by default with python as it will require a huge space then we use package manager to install packages , and the most common ones are PIP and CONDA

Pip is the default package manager for Python and is included by default with most Python installations.

Conda is a cross-platform package manager that can install packages for multiple languages, including Python. It was developed by Anaconda , Conda can also manage environments, which are isolated spaces where packages can be installed without interfering with each other.

**Pip vs Conda: Key Differences**

**1. Package Availability**

Pip installs packages from the Python Package Index (PyPI), which hosts a vast array of Python libraries. Almost any Python library can be installed using pip.

On the other hand, conda installs packages from the Anaconda distribution and other channels. While the number of packages available through conda is smaller than pip, conda can install packages for multiple languages and not just Python.

**2. Environment Management**

While pip can be used in conjunction with virtualenv to create isolated environments, conda has this feature built-in. Conda environments can have different versions of Python and other languages, making it a powerful tool for managing complex projects. Conda also creates a virtual environment, like virtualenv does.

**3. Binary Packages**

Conda installs binary packages, which means the packages include compiled code. This can make the installation process faster and more reliable, especially for packages with complex dependencies.

Pip, by contrast, often installs packages from source, which means the code is compiled during the installation process. This can be slower and more prone to errors, especially on Windows.

**4.easness**

pip can install anything from pypi in one command. conda requires three commands: skeleton, build, install and possibly more if that doesn't work. pip can install anything from github or source in one command. conda requires writing a "recipe", which is not easy

**When to Use Pip or Conda?**

Choosing guidelines as there is no right answer. it depends on the project needs and what works more efficient:

* Use pip if you are working with pure Python projects and need access to the vast array of libraries available on PyPI.
* Use conda if you are working with projects that use multiple languages, need different versions of Python, or require complex binary dependencies.

**Note** :   
In many cases, you can use both tools in the same project. For example, you can use conda to manage environments and install binary packages, and pip to install Python libraries that are not available through conda but Because Conda introduces a new packaging format, you cannot use pip and Conda interchangeably; pip cannot install the Conda package format. You can use the two tools side by side (by installing pip with conda install pip) but they do not interoperate either.

9.2

# How to create an automated task using Task Scheduler on Windows 10 :

This site helped me doing it step by step :<https://www.windowscentral.com/how-create-automated-task-using-task-scheduler-windows-10>

9.3

[Python packages are popular for creating RPA bots](https://research.aimultiple.com/python-rpa/) because they are easy to implement, compatible with other tools, and affordable.

RPA stands for Robotic Process Automation. It is a generic tool for creating specialized agents, or “bots,” that interact with GUI elements to complete repetitive, rules-based tasks.

Robotic process automation (RPA) is a productivity tool that allows a user to configure one or more scripts to activate specific keystrokes in an automated fashion.

#### Robotic process automation (RPA) can support a remote workforce & reduce costs as RPA software bots can be scheduled to run automatically, with an employee monitoring the progress from a visible dashboard. Moreover, the overall efficiency of [RPA can reduce back-office operations costs by 70%](https://www.fca.org.uk/publications/multi-firm-reviews/implementing-technology-change#:~:text=Over%2090%25%20of%20surveyed%20FS%20firms%20rely%20on%20legacy%20technology%20in%20some%20form%20to%20deliver%20their%20services).

#### Robotic process automation (RPA) can address system integration gaps

1. Robotic process automation (RPA) can tackle remote desktop automation
2. Robotic process automation (RPA) can automate what could be outsourced

## What does a Python RPA library do?

A [Python RPA](https://research.aimultiple.com/python-rpa/) library provides a set of tools and functions that make it easier to [implement RPA](https://research.aimultiple.com/rpa-implementation/) using the Python programming language for automating repetitive, time-consuming tasks like:

* Mouse automation
* Reading and writing to databases
* Copy-pasting information
* Filling out forms
* Moving files around, etc.

[Learn more about processes that are suitable for RPA automation.](https://research.aimultiple.com/rpa-processes/)

## What do Python libraries offer?

Python libraries offer the following:

### 1. User interface interactions

Some libraries provide GUIs for easy implementation of a Python script into a bot. These include:

* Clicking buttons
* Typing into fields
* Selecting items from dropdown menus
* Anything else that a human user might do

### 2. Web scraping

Some [RPA libraries provide tools for web scraping](https://research.aimultiple.com/rpa-web-scraping/), which involves extracting data from websites.

### 3. Decision making

More advanced RPA tools can include features for making decisions based on the data they interact with. This can involve things like sending an email if certain conditions are met, or choosing different actions based on the content of a document.

### 4. Integration

They often provide the ability to integrate with other software or applications. This can include things like reading and writing Excel files, interacting with email clients, or connecting to APIs.

## What are the top Python RPA libraries?

### 1. Robot Framework

* **Pros:** Robot Framework is [open-source](https://research.aimultiple.com/open-source-rpa/) and keyword-driven. It makes creating test cases and automation scripts simple and intuitive. Robot Framework supports Python code and Java, with strong capabilities for web, API, database, and GUI testing.
* **Cons:** The keyword-driven approach might be insufficient for complex applications. It also lacks some high-level features a commercial Python package provides, such as:
  + Low-code formatting
  + Advanced reporting and analytics
  + Built-in orchestrator
  + Advanced error-handling
  + Community support

### 2. Pywinauto

* **Pros:** Pywinauto’s pure-Python solution can automate repetitive tasks on Windows without relying on screen resolution or mouse cursor positioning, which improves reliability. Pywinauto also doesn’t require a separate installation, thus streamlining the process.
* **Cons**: It’s unsuitable for operating systems besides Windows and can only interact with web applications. It doesn’t support scheduling and managing bots at scale. And it may not be as feature-rich as other libraries, by lacking image file recognition and advanced reporting.

### 3. Selenium

* **Pros:** Selenium can carry out regression, functional, and load testing by integrating with tools like TestNG and JUnit. It works on a variety of web browsers, like Firefox, Edge, Safari, etc.
* **Cons:** Selenium’s only use case is web automation and command line. It doesn’t support non-web tasks and might be overkill for simple ones, like creating Excel files. It also requires a good understanding of web technologies, like HTML, CSS, and XPath selectors.

### 4. TagUI for Python (RPA for Python)

* **Pros:** TagUI for Python is a simple yet powerful tool for RPA, enabling a wide range of automation tasks. It uses a straightforward syntax to emulate user behavior, making it easy to get started with.
* **Cons:** Although it’s easy to use, for complex automation tasks, the simplified syntax might be limiting. Its simplicity also means that it might not offer as many features as other, more complex tools like a built-in optical character recognition engine.

### 5. BotCity

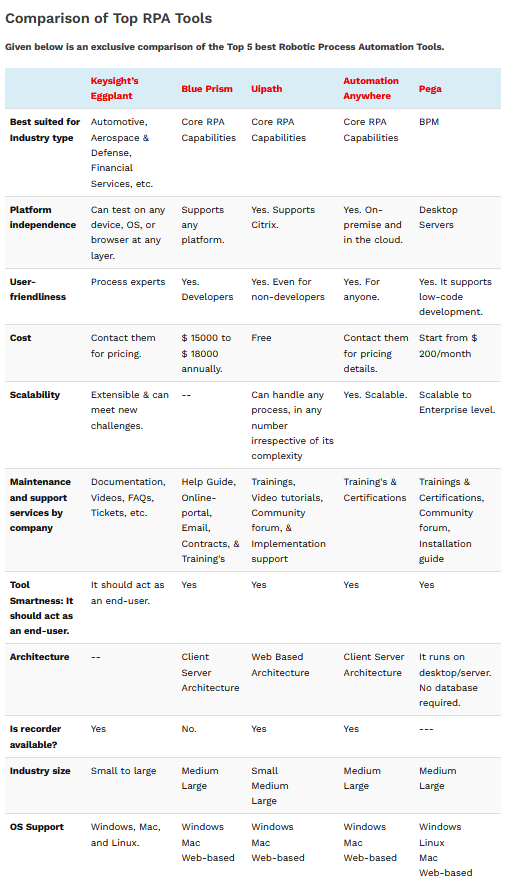
* **Pros:** BotCity is easy to use for people with limited programming skills, and has open source libraries for automating tasks and business processes, like to write excel files and data entry, across a various operating systems
* **Cons:** As an open-source tool, support for BotCity mainly comes from the community. There might be limited resources for troubleshooting and issue resolution compared to commercial RPA solutions that come with dedicated support.

### 6. Automagica

* **Pros:** Automagica’s ready-to-use activities make it easier to create robotic process automation bots to automate tasks like:
  + Web automation
  + [Desktop automation](https://research.aimultiple.com/rda/)
  + Keyboard and mouse movements automation
  + Working with files and directories, and more.
* **Cons:** Automagica’s ease of use might limit its functions, such as:
  + Lack of flexibility for complex tasks that don’t fit neatly into predetermined formats, like those involving data manipulation
  + Limited lower-level control over cases that deal with complex systems and unusual edge cases
  + No built-in orchestration and control room for managing, controlling, and deploying bots across an organization

### Top RPA Tools

### For more details : <https://www.softwaretestinghelp.com/robotic-process-automation-tools/#Top_Robotic_Process_Automation_RPA_Tools>



References :

RPA:

<https://www.softwaretestinghelp.com/robotic-process-automation-tools/#Top_Robotic_Process_Automation_RPA_Tools>

<https://pythonrpa.org/>

<https://research.aimultiple.com/python-rpa/>

<https://research.aimultiple.com/python-rpa-library/>

<https://research.aimultiple.com/rpa/#rpa-tools>

<https://www.simplilearn.com/tutorials/rpa-tutorial/python-rpa>

pip vs conda :

<https://saturncloud.io/blog/pip-vs-conda-a-guide-to-managing-python-packages-for-data-scientists/>

<https://stackoverflow.com/questions/20994716/what-is-the-difference-between-pip-and-conda>

<https://www.anaconda.com/blog/understanding-conda-and-pip>

<https://medium.com/analytics-vidhya/understand-conda-and-pip-9e5c67da47cc>

<https://pythonspeed.com/articles/conda-vs-pip/>

<https://www.anaconda.com/blog/understanding-conda-and-pip>