**[Installation instructions](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html" \l "toc-entry-220)**

These instructions cover installing [Robot Framework](https://robotframework.org/) and its preconditions on different operating systems. If you already have [Python](http://python.org/) installed, you can install Robot Framework using the standard package manager [pip](https://pip.pypa.io/):

pip install robotframework

[**Python installation**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-228)

[Robot Framework](https://robotframework.org/) is implemented using [Python](http://python.org/), and a precondition to install it is having Python or its alternative implementation [PyPy](https://pypy.org/) installed. Another recommended precondition is having the [pip](https://pip.pypa.io/) package manager available.

Robot Framework requires Python 3.6 or newer. If you need to use Python 2, [Jython](http://jython.org/) or [IronPython](http://ironpython.net/), you can use [Robot Framework 4.1.3](https://github.com/robotframework/robotframework/blob/v4.1.3/INSTALL.rst). The forthcoming Robot Framework 7.0 will require [Python 3.8 or newer](https://github.com/robotframework/robotframework/issues/4294).

[**Installing Python on Windows**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-230)

On Windows Python is not available by default, but it is easy to install. The recommended way to install it is using the official Windows installers available at [http://python.org](http://python.org/). For other alternatives, such as installing from the Microsoft Store, see the [official Python documentation](https://docs.python.org/3/using/windows.html).

When installing Python on Windows, it is recommended to add Python to [PATH](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#configuring-path) to make it and tools like pip and Robot Framework easier to execute from the command line. When using the [official installer](https://docs.python.org/3/using/windows.html#windows-full), you just need to select the Add Python 3.x to PATH checkbox on the first dialog.

To make sure Python installation has been successful and Python has been added to PATH, you can open the command prompt and execute python --version:

C:\*>python --version*

Python 3.9.4

If you install multiple Python versions on Windows, the version that is used when you execute python is the one first in PATH. If you need to use others, the easiest way is using the [py launcher](https://docs.python.org/3/using/windows.html" \l "launcher):

C:\*>py --version*

Python 3.9.4

C:\*>py -3.10 --version*

Python 3.10.2

[**Configuring PATH**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-233)

The [PATH environment variable](https://en.wikipedia.org/wiki/PATH_(variable)) lists directories where commands executed in a system are searched from. To make using Python, [pip](https://pip.pypa.io/) and Robot Framework easier from the command line, it is recommended to add the Python installation directory as well as the directory where commands like pip and robot are installed into PATH.

When using Python on Linux or macOS, Python and tools installed with it should be automatically in PATH. If you nevertheless need to update PATH, you typically need to edit some system wide or user specific configuration file. Which file to edit and how depends on the operating system and you need to consult its documentation for more details.

On Windows the easiest way to make sure PATH is configured correctly is setting the Add Python 3.x to PATH checkbox when [running the installer](https://docs.python.org/3/using/windows.html#the-full-installer). To manually modify PATH on Windows, follow these steps:

1. Find Environment Variables under Settings. There are variables affecting the whole system and variables affecting only the current user. Modifying the former will require admin rights, but modifying the latter is typically enough.
2. Select PATH (often written like Path) and click Edit. If you are editing user variables and PATH does not exist, click New instead.
3. Add both the Python installation directory and the *Scripts* directory under the installation directory into PATH.
4. Exit the dialog with Ok to save the changes.
5. Start a new command prompt for the changes to take effect.

[**Running pip command**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-235)

Typically you use pip by running the pip command, but on Linux you may need to use pip3 or even more Python version specific variant like pip3.8 instead. When running pip or any of its variants, the pip version that is found first in [PATH](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#configuring-path) will be used. If you have multiple Python versions installed, you may need to pinpoint which exact version you want to use. This is typically easiest done by running python -m pip and substituting python with the Python version you want to use.

To make sure you have pip available, you can run pip --version or equivalent.

Examples on Linux:

$ pip --version

pip 20.2 from ... (python 3.8)

$ python3.10 -m pip --version

pip 20.2 from ... (python 3.10)

Examples on Windows:

C:\*> pip --version*

pip 20.2.2 from ... (python 3.9)

C:\*> py -m 3.10 -m pip --version*

pip 20.2.1 from ... (python 3.10)

In the subsequent sections pip is always run using the pip command. You may need to use some of the other approaches explained above in your environment.

[**Installing and uninstalling Robot Framework**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-236)

The easiest way to use pip is by letting it find and download packages it installs from the [Python Package Index (PyPI)](https://pypi.org/project/robotframework), but it can also install packages downloaded from the PyPI separately. The most common usages are shown below and [pip](https://pip.pypa.io/) documentation has more information and examples.

*# Install the latest version (does not upgrade)*

pip install robotframework

*# Upgrade to the latest stable version*

pip install --upgrade robotframework

*# Upgrade to the latest version even if it is a pre-release*

pip install --upgrade --pre robotframework

*# Install a specific version*

pip install robotframework==5.0

*# Install separately downloaded package (no network connection needed)*

pip install robotframework-5.0-py3-none-any.whl

*# Install latest (possibly unreleased) code directly from GitHub*

pip install https://github.com/robotframework/robotframework/archive/master.zip

*# Uninstall*

pip uninstall robotframework

[**Verifying installation**](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#toc-entry-238)

To make sure that the correct Robot Framework version has been installed, run the following command:

$ robot --version

Robot Framework 5.0 (Python 3.8.10 on linux)

If running these commands fails with a message saying that the command is not found or recognized, a good first step is double-checking the [PATH](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#configuring-path) configuration.

If you have installed Robot Framework under multiple Python versions, running robot will execute the one first in [PATH](https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#configuring-path). To select explicitly, you can run python -m robot and substitute python with the right Python version.

$ python3.10 -m robot --version

Robot Framework 5.0 (Python 3.10.2 on linux)

C:**\>**py -3.10 -m robot --version

Robot Framework 5.0 (Python 3.10.3 on win32)