

Unofficial Language Installers

A collection of unofficial script installers for various programming languages.

- [Install Python Using Pyenv](#)
- [Install Node.js](#)
- [Install Golang](#)
- [Install Ruby](#)
- [Install Rust](#)

Install Python Using Pyenv

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Customers are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff. Use at your own risk.

pyenv lets you easily switch between multiple versions of Python. It is simple, unobtrusive, and follows the UNIX tradition of single-purpose tools that do one thing well.

At a high level, pyenv intercepts Python commands using shim executables injected into your PATH, determines which Python version has been specified by your application, and passes your commands along to the correct Python installation.

For more information, visit the [pyenv GitHub repository](#).

Pyenv and Python3 Installation Script

- Connect to your Ultra.cc service via SSH, see guide [here](#).
- Once connected, execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Python-Installer/main.sh)
```

- Select the Python version to install by typing `1`, `2`, `3`, `4` or `5` and then press `ENTER`
 - Wait while the script download and install the necessary files
- To load Python with pyenv in your shell, execute the following command:

```
source ~/.profile
```

- Check which Python binary you are currently using by executing `which python`.
 - The below output is what you should see.

```
ultradocs@pollux: ~$ which python
/home/ultradocs/.pyenv/shims/python
```

Pyenv Usage Guide

In this section, you will learn some basic usage of pyenv. If you are looking for [pip](#) usage instructions, see [this guide](#).

List Installed Python Versions

- Execute the following SSH command:

```
pyenv versions
```

- Assuming that you used the script to install python 3.9, the output will look like this:

```
ultradocs@pollux: ~$ pyenv versions
  system
* 3.9.13 (set by /home/ultradocs/.pyenv/version)
ultradocs@pollux: ~$
```

- The `*` beside `3.9.13` means that pyenv has set it as the global python version.

Install other Python versions using Pyenv

- Execute the following SSH command:

```
pyenv install --list | grep -E " 3.[8-9].*| 3.10.*"
```

- It will list all versions of python from version 3.8 to version 3.10 available for install via pyenv.
- Example output:

```
ultradocs@pollux: ~$ pyenv install --list | grep -E " 3.[8-9].*| 3.10.*"
 3.8.0
 3.8-dev
 3.8.1
 3.8.2
 3.8.3
 3.8.4
 3.8.5
 3.8.6
 3.8.7
 3.8.8
 3.8.9
 3.8.10
 3.8.11
 3.8.12
 3.8.13
 3.9.0
 3.9-dev
 3.9.1
```

```
3.9.2
3.9.4
3.9.5
3.9.6
3.9.7
3.9.8
3.9.9
3.9.10
3.9.11
3.9.12
3.9.13
3.10.0
3.10-dev
3.10.1
3.10.2
3.10.3
3.10.4
```

```
ultradocs@pollux: ~$
```

- You can install any one of them easily with the following SSH command:

```
pyenv install -v <version number>
```

For example:

```
pyenv install -v 3.10.4
```

- Since I have also installed python 3.10.4, my `pyenv versions` output looks like this:

```
ultradocs@pollux: ~$ pyenv install -v 3.10.4
ultradocs@pollux: ~$ pyenv versions
  system
  3.10.4
* 3.9.13 (set by /home/ultradocs/.pyenv/version)
ultradocs@pollux: ~$
```

- Notice that pyenv is still using 3.9.13 as the global python version.
- This can be changed with the following SSH command:

```
pyenv global <version>
```

For example:

```
pyenv global 3.10.4
```

- Now pyenv will set python 3.10.4 as the global version.
- Example output:

```
ultradocs@pollux: ~$ pyenv global 3.10.4
ultradocs@pollux: ~$ pyenv versions
  system
* 3.10.4 (set by /home/ultradocs/.pyenv/version)
  3.9.13
ultradocs@pollux: ~$ python -V
Python 3.10.4
ultradocs@pollux: ~$
```

Uninstall a Python version using Pyenv

- Again, list your installed python versions and uninstall using the SSH command given below:

```
pyenv uninstall <version>
```

Example:

```
pyenv uninstall 3.9.13
```

- Example output:

```
ultradocs@pollux: ~$ pyenv versions
  system
* 3.10.4 (set by /home/ultradocs/.pyenv/version)
  3.9.13
ultradocs@pollux: ~$ pyenv uninstall 3.9.13
pyenv: remove /home/ultradocs/.pyenv/versions/3.9.13? [y|N] y
pyenv: 3.9.13 uninstalled
ultradocs@pollux: ~$
```

- We basically upgraded to python 3.10.4 and removed python 3.9.13.

Uninstall Pyenv and the Python version(s) installed by it

- Execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Python-Installer/main.sh)
```

Manual Uninstallation

- Execute the following command:

```
rm -rf ~/.pyenv
```

- Open `~/.profile` for editing with the following SSH command:

```
nano ~/.profile
```

- Remove the following lines from it:
 - `export PYENV_ROOT="$HOME/.pyenv"`
 - `export PATH="$PYENV_ROOT/bin:$PATH"`
 - `eval "$(pyenv init --path)"`
- Use the key combination `Ctrl + O` and press `Enter` to save the changes.
- Then, use the key combination `Ctrl + X` to exit the nano text editor.
- Load your new `~/.profile` with the following SSH command:

```
source ~/.profile
```

Install Node.js

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Clients are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff.
Use at your own risk.

This script will install Node.js on your Ultra.cc slot. Included in the installation is also Node Version manager (NVM) and Node Package Manager (NPM)

Installation

- Connect to your Ultra.cc slot via SSH, see guide [here](#).
- Once connected, execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Node-Installer/main.sh)
```

- Type `1` and press `ENTER` to proceed with the installation
- Done!

Uninstallation

- Execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Node-Installer/main.sh)
```

- Type `2` and press `ENTER` to proceed with uninstallation
- Done!

Install Golang

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Clients are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff. Use at your own risk.

This script will install Golang on your Ultra.cc service. Go is an open-source and multipurpose programming language which is fast and efficient.

Installation

- Connect to your Ultra.cc service via SSH, see guide [here](#).
- Once connected, execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Golang-Installer/main.sh)
```

- Type `1` and press `ENTER` to install.
- Next, pick the version you want to install by typing `1`, `2`, `3`, `4` or `5`.
- Press `ENTER` to confirm.
- Done!

Uninstallation

- Execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util/LanguageInstaller/Golang-installer/main.sh)
```

- Type `2` and press `ENTER` to proceed with uninstallation
- Done!

Install Ruby

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Clients are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff. Use at your own risk.

This script will install Ruby on your Ultra.cc slot. Ruby is an interpreted, high-level, general-purpose programming language which supports multiple programming paradigms. It was designed with an emphasis on programming productivity and simplicity.

Installation

- Connect to your Ultra.cc slot via SSH, see guide [here](#).
- Once connected, execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Ruby-Installer/main.sh)
```

- Type `1` and press `ENTER` to confirm the installation.
- Wait for installer to finish compiling. This may take a while.
- Done!

Uninstallation

- Execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Ruby-Installer/main.sh)
```

- Type `2` and press `ENTER` to proceed with uninstallation.
- Done!

Install Rust

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Clients are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff. Use at your own risk.

This script will install Rust on your Ultra.cc slot. Rust is a multi-paradigm and general-purpose programming language. It was designed with an emphasis on performance, type safety and concurrency.

Installation

- Connect to your Ultra.cc service via SSH, see guide [here](#).
- Once connected, execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Rust-Installer/main.sh)
```

- Type `1` and press `ENTER` to install.
- Wait for installer to finish. This may take a while.
- Once the installation has finished, execute the following command:

```
source ~/.bashrc
```

- To check if the installation was successful, execute `rustc --version` and you should see the below output in your terminal.

```
ultradocs@spica: ~$ rustc --version  
rustc 1.75.0 (82e1608df 2023-12-21)
```

- Done!

Uninstallation

- Execute the following command:

```
bash <(wget -qO- https://scripts.ultra.cc/util-v2/LanguageInstaller/Rust-Installer/main.sh)
```

- Type `2` and press `ENTER` to proceed with uninstallation.
- Done!