interweb

Generated on 15/11/2026 00:12

KNOWLEDGE BASE · PUBLIC DOCUMENT

Python in cPanel: virtual environments and pip

Python & Other Languages

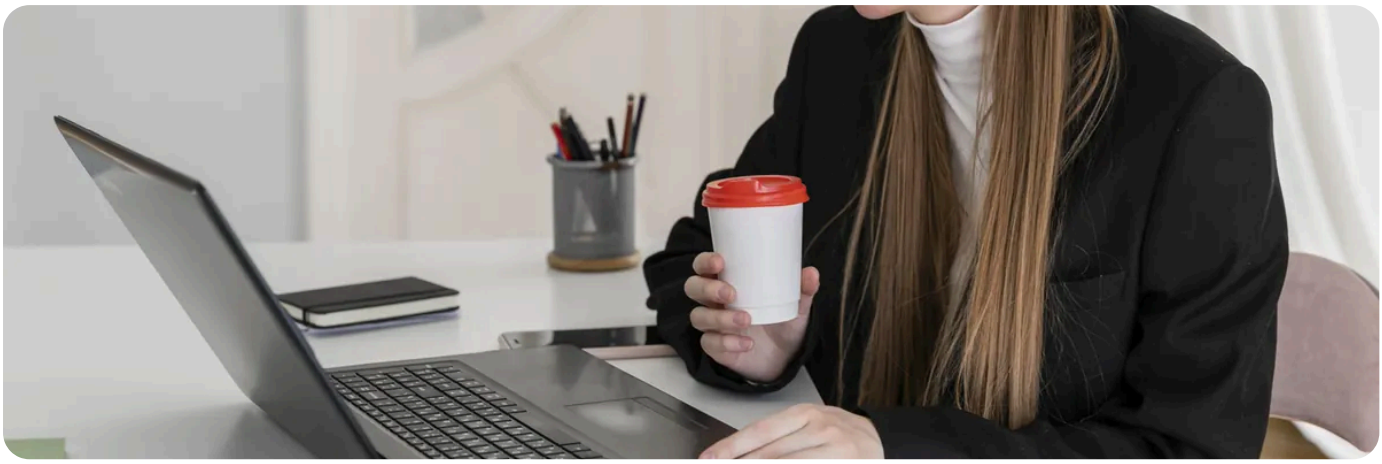
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-880	v2026.10.10	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/880

Python in cPanel: virtual environments and pip

Python & Other Languages · Created on **October 10, 2026** · 3 min read



In cPanel, Python runs through **"Setup Python App"**, which offers Python 2.7 and the 3.3 to 3.13 versions. Each application has **its own virtual environment**, created by the tool, and that is where `pip` should install libraries. The golden rule: **use the application's environment, do not create another one**, and always install with that environment's `pip`. For a new application pick a current 3.x version; Python 2.7 is no longer maintained.

Creating the application and installing libraries

1. **Open "Setup Python App"** in cPanel (Software section) and choose "Create Application".
2. **Fill in the essentials:** the Python version, the **application root** (a folder inside your account, outside the site's public folder), the application **address**, the startup file (`passenger_wsgi.py` by default) and the entry point (`application` by default). Save.
3. **Write a** `requirements.txt` in the application root with one library per line, for instance `flask`. On a computer of your own, generate it with `pip freeze > requirements.txt`.
4. **Install.** On the application screen, in the configuration files part, name the `requirements.txt` and use the **"Run Pip Install"** button.
5. **Restart** the application. Passenger only loads new libraries after a restart.

Using pip in the terminal

If the account has a terminal (it depends on the account; see the **cPanel Terminal**), the application screen shows at the top the **command to enter the virtual environment**: a `source` of an `activate` file inside a `virtualenv` folder. Copy it exactly. Once inside:

```
python --version
pip --version
pip install -r requirements.txt
pip list
```

`pip --version` should show a path inside the application's environment. If it shows another, you are not inside.

Error	What it means	What to do
`ModuleNotFoundError: No module named ...`	The library is not in the environment the application uses.	Install with the application environment's `pip` (or "Run Pip Install") and restart.
`Permission denied` **when installing**	You are trying to install outside the environment, on the system.	Enter the application environment first. Never use `sudo` on a shared account.
Build error (`gcc` **, "Python.h", "could not build wheel")**	The library needs compiling and tools or headers are missing.	Look for a pure Python alternative (to talk to MySQL, for example, `PyMySQL` instead of `mysqlclient`) or a version that ships a precompiled package.
The application uses an old library version	Another version was installed on top, or `requirements.txt` does not pin versions.	Pin them: `flask==X.Y.Z` with the version you tested.
The Python version changed and nothing works	The environment is new and empty.	Run "Run Pip Install" again on the new version.

Changing the Python version creates a new, empty environment.
The old version's libraries do not carry over: reinstall everything from `requirements.txt` . And never delete the environment folder by hand; use the tool, which knows what it connected to Passenger.

And on a VPS?

On a VPS with Ubuntu 22.04 LTS, you make the virtual environment by hand, per project:

```
sudo apt install python3-venv python3-pip
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
deactivate
```

On newer versions of Ubuntu and Debian, `pip` refuses installs outside a virtual environment (the message is "externally-managed-environment"): that is what the environment is for. In services (systemd, PM2, Docker), always point at the Python **inside** `.venv` .

Pin the versions

with `pip freeze > requirements.txt` in the environment that works, and keep the file with the code. To get a web application running, see **Flask or FastAPI**, and for Django, **getting a Django application running**. Database connections are in **connecting to your database from Python and Node.js**.

NEED A HAND?

The install button gives an error you do not understand, or the Python version you need is not listed? Tell us the domain and the message.

Open a support ticket →

SEE ALSO

- **Running a Flask or FastAPI application** →
- **Getting a Django application running** →
- **Which technologies run here: PHP, Node, Python, Perl, games and the limits** →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-880 · v2026.10.10 · Generated on 15/11/2026 00:12