Python (https://cloud.google.com/python/) Guides

Setting up a Python development environment

This tutorial shows how to prepare your local machine for Python (https://python.org) development, including developing Python apps that run on Google Cloud.

If you already have a development environment set up, see Python and Google Cloud (https://cloud.google.com/python) to get an overview of how to run Python apps on Google Cloud.

Objectives

- Install the latest version of Python.
- Use `venv` to isolate dependencies.
- Install an editor (optional).
- Install the Cloud SDK (optional).
- Install the Cloud Client Libraries for Python (optional).
- Install other useful tools.

Installing Python

Python's installation instructions vary by operating system. Follow the guide for the operating system you're running in your development environment, MacOS, Windows, or Linux.

MACOS WINDOWS LINUX

macOS includes a version of Python by default and uses it for its own purposes. To avoid interfering with macOS, we recommend creating a separate development environment and installing the latest version of Python. To install Python, use homebrew (https://brew.sh/).

1. To use homebrew to install Python packages, you need a compiler, which you can get by installing Xcode's command-line tools (https://developer.apple.com/library/content/technotes/tn2339/_index.html).

   ```
   xcode-select --install
   ```

2. Install homebrew by following the instructions on the homebrew homepage (https://brew.sh/), and then use homebrew to install Python as follows:

   ```
   brew install python
   ```

3. After the installations are complete, verify that Python 3 is available as `python` and `python3`, and that `pip` is also installed.

   To verify that Python is available, run the following command:

   ```
   python --version
   ```

   The output shows the version. You can learn about the latest version of Python homebrew in the Homebrew Python Formulae (https://formulae.brew.sh/formula/python) page.

   To verify that pip is available, run the following command:

   ```
   pip --version
   ```

   The outputs show the versions. You can learn about the latest version of pip in the pip Release Notes (https://pip.pypa.io/en/stable/news/).

   Homebrew installs the latest versions of Python available on your platform via Homebrew. The version numbers in the outputs may be different from the latest official releases of Python (https://www.python.org/downloads/).

Using venv to isolate dependencies

venv (https://docs.python.org/3/library/venv.html) is a tool that creates isolated Python environments. These isolated environments can have separate versions of Python packages, which allows you to isolate one project's dependencies from the dependencies of other projects.

We recommend that you always use a per-project virtual environment when developing locally with Python.

1. Use the **venv** command to create a virtual copy of the entire Python installation. This tutorial creates a virtual copy in a folder named **venv**, but you can specify any name for the folder.

   MACOS WINDOWS LINUX

   ```
   cd your-project
   python -m venv venv
   ```

2. Set your shell to use the **venv** paths for Python by activating the virtual environment.

   MACOS WINDOWS LINUX

   ```
   source venv/bin/activate
   ```

3. Now you can install packages without affecting other projects or your global Python installation:

   ```
   pip install google-cloud-storage
   ```

   If you want to stop using the virtual environment and go back to your global Python, you can deactivate it:

   ```
   deactivate
   ```

You can read more about **venv** in the venv docs (https://docs.python.org/3/library/venv.html).

Note: If you're using Anaconda (https://docs.anaconda.com/anaconda/user-guide/tasks/install-packages/), follow the instructions on their website.

Installing an editor

To develop Python apps, you need an editor. Here are a few of the more popular editors (in no particular order):

- Sublime Text (https://www.sublimetext.com/) by Jon Skinner
- Atom (https://atom.io/) by GitHub
- PyCharm (https://www.jetbrains.com/pycharm/) by JetBrains

Installing the Cloud Client Libraries for Python

The Cloud Client Libraries for Python (https://github.com/googleapis/google-cloud-python#google-cloud-python-client) is how Python developers integrate with Google Cloud services like Datastore and Cloud Storage. To install the package for an individual API like Cloud Storage, use a command similar to the following:

```
pip install --upgrade google-cloud-storage
```

Installing the Cloud SDK

The Cloud SDK (https://cloud.google.com/sdk) is a set of command-line tools for Google Cloud. It contains **`gcloud`**, **`gsutil`**, and **`bq`**, which you can use to access Compute Engine, Cloud Storage, BigQuery, and other products and services from the command line. You can run these tools interactively or in your automated scripts.

What's next

- Learn more about Python on Google Cloud (https://cloud.google.com/python).
- Deploy a Python app to App Engine (https://cloud.google.com/python/quickstarts).
- Browse the documentation for Google Cloud products (https://cloud.google.com/products).

Last updated January 15, 2020.