Setting up a Python Environment

Download Python

Before setting up a Python Environment, you need to install Python. This can be done by downloading the latest version from the web here. Simply follow the downloader steps and accept the defaults when unsure.

Once Python is installed on your machine, there are a few options to setting up an environment. The most common options are listed below

- Option 1: Anaconda
- Option 2: ArcGIS Pro
- Option 3: venv

Option 1: Anaconda

"Conda is a powerful command line tool for package and environment management" (conda.io). Creating and managing Python Environments via Anaconda is straightforward; it has a good user interface, plenty of documentation, and lots of helpful tools that make it easy to manage environments.

Download Anaconda

To get started, you first need to install conda via the instructions here. There are three ways to download conda, but my suggestion is to install with Anaconda Distribution. Anaconda Distribution is a user interface that uses the conda tool in the background.

Create Environment

After installing Anaconda Distribution, open up the Anaconda Navigator app via the start menu. Next, click `Environments` on the left sidebar. This is where you create new and manage existing Python environments. By default, the `base (root)` environment should already be selected. This environment is sufficient for basic Python tasks and can be used. However, in many cases creating a new environment may be beneficial.

To create a new environment click `Create` near the bottom and then choose a name for the environment. Click `Create` again, and Anaconda will set up the environment with the defaults.

Advanced Method

Alternatively, you can create environments via the terminal and the conda tool.

Open terminal and type the following:

```
conda create --name <my-env>
```

When conda asks you to proceed, type `y`:

```
proceed ([y]/n)?
```

For additional instructions, please refer to the conda documentation.

Activate Environment

In order to use the environment you create, you must activate it. This can be done in the `Anaconda Navigator` by clicking on the environment name in the environment list. A green play button will display to the right of the name of the activated environment.

Advanced Method

If you choose to activate an environment via the terminal, type the following:

```
conda activate <my-env>
```

Install Packages

Usually upon creating a new environment, you will want to download new packages that are not found in the default. To do this, first activate the environment. In the `Anaconda Navigator` under the `Envrionments` tab, you can install common packages via the search tool in the top right corner. First adjust the dropdown to search for `Not Installed` or `All`. Next, type in the name of the package you wish to install. If the package shows up, click the check mark next to the name and select `Apply`. The package will then be installed in the environment. If the name of the package does not appear, you will have to follow the instructions in the Advanced Method.

Advanced Method

To install new packages via the terminal, type the following:

```
conda activate <my-env>
pip install <package-name>
```

Option 2: ArcGis Pro

A default python is downloaded alongside the installation of ArcGis Pro as well as a default python environment, `arcgispro-py3`. This environment is used for running stand-alone scripts in ArcGis Pro and includes all the Python libraries used by ArcGis Pro as well as a few additional packages.

ArcGis Pro provides a method to creating new environments within its interface. Upon opening ArcGis Pro, navigate to the `Settings` menu. Click on `Package Manager` located in the sidebar. In the `Package Manager` you have the options to create new environments, activate existing environments, and install new packages.

Create Environment

To create a new environment, click the gray gear next to the `Active Envrionment` dropdown (located in the top right corner). A new window will appear with the default `arcgispro-py3` environment listed. Click the stacked paper icon to the right of the `arcgispro-py3` environment name. Determine a name for the environment and click `Ok`. This will *clone* the environment, thus creating a new environment with a stable base setup.

Activate Environment

To activate an environment, simply select the environment name in the `Active Environment` dropdown.

Install Packages

To install new packages, first activate the environment. Then click the `Add Packages` button. Search for the package you wish to install. Click on the package and then click the `Install` button.

If the package you wish to install isn't available in the list, you will have to download via the terminal. Before opening terminal, find the location on your computer where the environment is located (to see this, simply click the gray gear button -- the path will display underneath the environment name). The path may look something like this: `C:\Users\<user>\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone`. In terminal type the following:

```
cd <environment-path>\Scripts
pip install <package-name>
```

For more information about using ArcGis Pro to manage Python Environments, please refer to the ArcGis Pro documentation.

Option 3: venv

venv is a virtual environment tool that is useful for managing separate package installations for different projects. It is mainly used for creating an isolated environment within your workspace directory.

Create Environment

To create a virtual environment open terminal and type the following:

```
cd <project-directory>
python -m venv .venv
```

Usually .venv is used as the name of the virtual environment, but feel free to change the name to something else.

Activate Environment

Next, activate the environment through the terminal and type the following:

```
.venv\Scripts\activate
```

Install Packages

To install packages, type the following in terminal:

```
python -m pip install <package-name>
```

For more information about using venv to manage Python Environments, please reference the venv documentation.

Troubleshooting

Ensure Python is Installed

To ensure python is installed, open terminal and type the following:

```
python --version
```

If no version number appears, the python installation may have had issues. Or, you could be in the wrong directory (i.e. on the D drive instead of the C drive).

Ensure pip is Installed

To ensure pip is installed, open terminal and type the following:

```
python -m pip --version
```

If no version number appears, open terminal and type the following:

```
python -m pip install --upgrade pip
```

Ensure conda is Installed (Option 1)

To ensure conda is installed, open terminal and type the following:

```
conda --version
```

If no version number appears, the python installation may have had issues. Or, you could be in the wrong directory.