

## CHAPTER 2

# Working with Python editors

## 2.1 Introduction

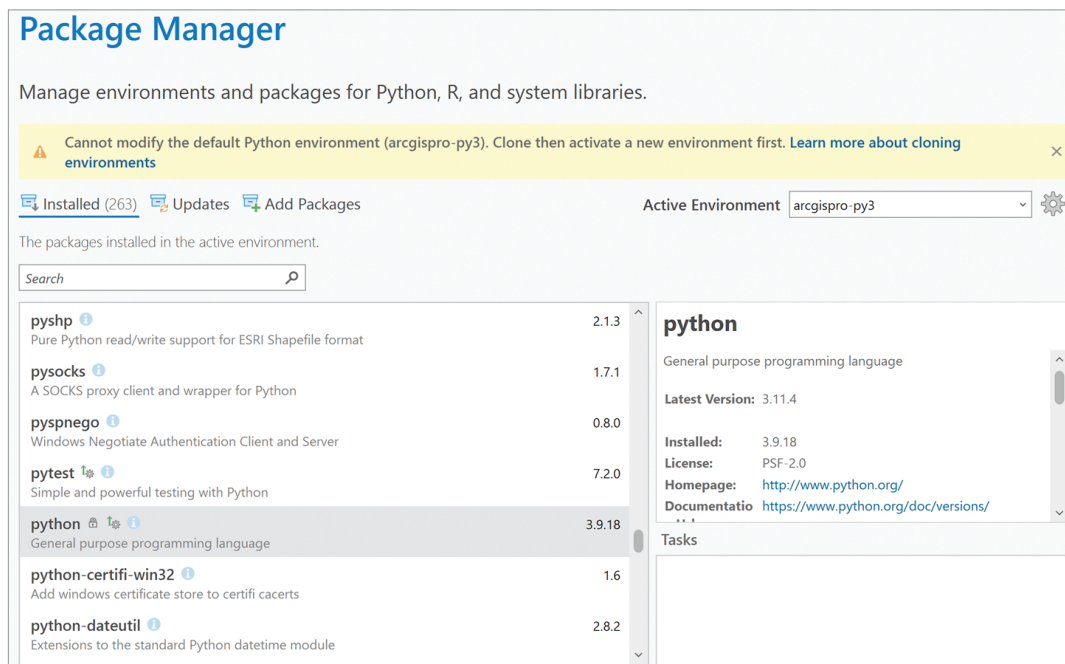
This chapter introduces the use of Python editors, or integrated development environments (IDEs), to write and test your scripts. Many different IDEs are used by the Python community, and experienced developers often have a strong personal preference for a specific editor. Although the choice of which IDE to use is important, Python code is the same regardless of which editor is used.

This chapter explains the ways you can run Python code and introduces several IDEs, including IDLE and PyCharm. Notebooks provide an alternative to traditional IDEs to write Python code and allow you to interact with data and maps directly in ArcGIS Pro. The Python window in ArcGIS Pro presents another way to write Python code directly within ArcGIS Pro without having to work in a separate application.

## 2.2 Python installation

Before examining how to work with Python code, it is useful to review how Python is installed. Python is installed as part of the ArcGIS Pro installation. This includes all the packages that are necessary to support common GIS-related workflows using Python. A **package** refers to functionality created by a third party that can be added to your Python installation. To make it easier to manage and install Python packages, the Python community uses **package managers**. ArcGIS Pro integrates a widely used package manager called **conda**.

The easiest way to use conda is from within ArcGIS Pro. Start ArcGIS Pro with a new project. Click Project from the top menu and then click Package Manager. This brings up a user-friendly interface to conda.



The Package Manager shows you the active environment and which packages are currently installed as part of that environment and allows you to update and add packages. Packages are managed using environments. An **environment** refers to a particular version of Python that is installed and the packages that come with it. They are referred to as **virtual environments** because they replicate what a different installation on a different computer would look like but they reside on the same computer. They are also referred to as **Python environments**. The default environment is called `arcgispro-py3`. When you first install ArcGIS Pro, this is the only environment available to use. This environment comes with more than 200 of the most widely used Python packages for GIS-related workflows.

Specific tasks may require additional specialized packages, and you can add packages to an existing environment. The Package Manager gives you access to more than 9,000 specialized packages. However, the default environment `arcgispro-py3` cannot be modified and is kept in a pristine state. To add packages, therefore, you first need to clone the default environment. You can clone environments using the Package Manager. Cloning effectively creates a copy of the original Python installation that you can then modify by adding packages.

For this book, you do not need to clone environments or update or add packages. The `arcgispro-py3` environment contains all the functionality necessary to run the code in all the chapters. Details on cloning environments and managing packages using conda are covered in depth in chapter 6 of the *Advanced Python Scripting for ArcGIS Pro* book (Esri Press, 2024).

Although it is useful to review how Python is installed using the Package Manager, when you are just learning how to code using Python you typically do not need to make changes to this installation. The default environment consists of the correct version of Python (3.9.18 for ArcGIS Pro 3.2) and includes all the necessary packages for most tasks. When you update the software—for example, to ArcGIS Pro 3.3—the version of Python and the packages are automatically updated.

Finally, it is useful to consider where Python is installed. For a typical installation of ArcGIS Pro, Python is installed in the following folder:

```
C:\Program Files\ArcGIS\Pro\bin\Python
```

The default environment is installed in the following subfolder:

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3
```

These paths may be slightly different on your computer depending on how ArcGIS Pro was installed. The following sections of this chapter assume the paths referenced above. A common alternative path is

```
C:\Users\<username>\AppData\Local\Programs\ArcGIS\Pro\bin\Python
```

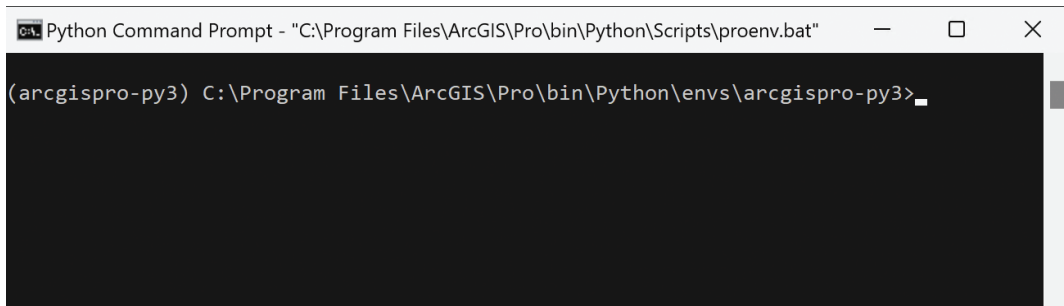
where <username> is the name of an individual user.

You can use File Explorer to confirm where ArcGIS Pro is installed on your computer and update the paths in the following sections accordingly.

## 2.3 Using a Python editor or IDE

A computer script is essentially a list of commands that can be run by a program or scripting engine. Scripts are usually plain text documents that have a specific file extension and contain instructions written in a scripting language. Most scripts can be opened and edited using a basic text editor. However, using a specialized script editor to work with a script provides additional functionality when you're writing the script and allows you to run the script as well.

You can work with Python in many ways. The most basic approach is to use the so-called **command line** or **command prompt**. If you have used other programming languages, you may be familiar with this type of interface. To access the Python command line in Windows, search for the application called Python Command Prompt. This application resides in the ArcGIS program group.

A screenshot of a Windows Command Prompt window titled "Python Command Prompt - \"C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\proenv.bat\"". The window has a black background and white text. The prompt shows the current directory as "(arcgispro-py3) C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3>".

```
Python Command Prompt - "C:\Program Files\ArcGIS\Pro\bin\Python\Scripts\proenv.bat"
(arcgispro-py3) C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3>
```

On a typical computer, this is not the only Python command prompt. For example, if you installed ArcGIS Desktop 10.x, you also have a command prompt in the ArcGIS program group called Python (command line), which works with Python 2.7. If you installed other programs that use Python, you may have additional command prompts. When working with ArcGIS Pro, you should use the specific command prompt installed with the software, which ensures that you have the correct version of Python that works with ArcGIS Pro and that you have access to the necessary packages.

Notice that the first line in the command prompt starts with (arcgispro-py3), which is the default Python environment that is installed as part of ArcGIS Pro. As the name suggests, this environment is specific to using ArcGIS Pro and Python 3.

Following the environment is the full path where the specific environment is installed on your local hard drive. This path will vary depending on how ArcGIS Pro is installed. The path is the default working directory when running Python; in many cases, you will want to change it to a custom directory of your choice.

Although this interface provides full access to all of Python's functionality, it provides limited support for writing and testing scripts. So instead of using the command prompt interface, it is typically more productive to use a Python editor. A **Python editor** has a menu-driven interface and tools for organizing and testing Python scripts to make working with Python easier.

A Python editor is also referred to as an **integrated development environment (IDE)**. The terms "Python editor" and "Python IDE" are often used interchangeably. Many IDEs exist, including several open-source and commercial software applications. Some IDEs are developed for specific platforms, such as Windows, macOS, or Linux, whereas others are designed to work with one or more specific programming languages, such as Java or a .NET language.

A comprehensive list of Python editors can be found on the Python wiki page (<https://wiki.python.org/moin/PythonEditors>). The editor you use is largely a matter of preference, and experienced Python programmers have their favorites. Python syntax remains the same for different editors, so you don't need different scripts to work with different editors.

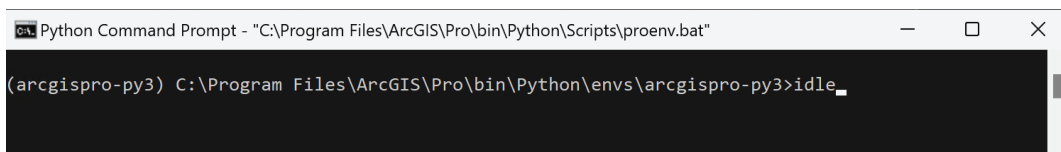
The two IDEs discussed in the rest of this chapter are IDLE and PyCharm. These editors were chosen because they represent a range of functionality, from very basic to relatively sophisticated. However, all the code in this book can be used by any IDE that is properly configured to work with ArcGIS Pro.

In addition to IDLE and PyCharm, this chapter also discusses the use of Notebooks and the Python window to write and run Python code. These two approaches are not considered IDEs, but they provide some of the same functionality.

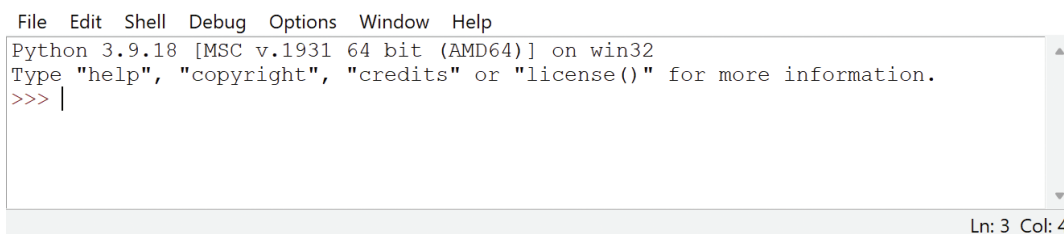
## 2.4 Using IDLE

The default IDE that comes with any installation of Python is the **Integrated DeveLopment Environment (IDLE)**. Notice how the *L* is capitalized. IDE is the general term used for Python editors, whereas IDLE is the specific editor that is provided as the default application to work with Python scripts. IDLE is basic, but because it is the default editor that is installed with Python, it is good to become familiar with it. The screen captures of Python code you saw in chapter 1 used IDLE. Any computer that has Python installed will have IDLE installed, so you can count on it being available. IDLE is also often referred to as the **Python Shell**. “Shell” is a general computing term to describe a user interface, as opposed to the command line interface.

To access IDLE, type the `idle` command in the Python command prompt. You type it at the blinking cursor at the end of the line following the `>` symbol.



Then press the Enter key. This command brings up IDLE.

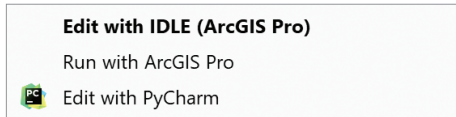


Notice how the name of the application at the top is **IDLE Shell 3.9.18**. The application is commonly referred to as both IDLE and Shell, and you will see these references in some of the menu options. Because you started IDLE from the `arcgispro-py3` environment, the version of Python being used is the one that installs with ArcGIS Pro. At the time of this writing, it is version 3.9.18 for ArcGIS Pro 3.2, but this will change with future releases of the software.

Because it is somewhat inconvenient to launch IDLE by first starting the command prompt, you can also create a shortcut for IDLE on your desktop or add it to the ArcGIS Pro program group. The path to run IDLE is as follows for a typical installation of ArcGIS Pro:

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\Lib\idlelib\idle.bat
```

Although no program shortcut is created for IDLE, File Explorer has a built-in shortcut. Browse to a folder in which one or more of your Python scripts are located. When you right-click on a Python script—i.e., a file with a .py extension—it brings up various options to work with the file, including Edit with IDLE (ArcGIS Pro). This option appears only if ArcGIS Pro is installed on your local computer.



If you have another installation of Python or another Python editor, additional options may be available. For example, if you have ArcMap 10.x installed, an option called Edit with IDLE (ArcGIS Desktop) is available. This option opens the version of IDLE installed with ArcMap 10.x—i.e., Python 2.7.18.4 when using ArcMap 10.8.2. There may also be an option called Edit with IDLE, which is typically associated with a separate installation of Python apart from ArcGIS Pro or ArcMap 10.x.

Clicking Edit with IDLE (ArcGIS Pro) opens the correct version of IDLE to work with ArcGIS Pro. The script opens in a script window. To launch the Shell, click Run > Python Shell.

This built-in shortcut is helpful because it uses the active environment in ArcGIS Pro. By default, this is the `arcgispro-py3` environment, but if the ArcGIS Pro application is using a different environment, that environment will be used when launching IDLE in this manner. This feature is helpful because other shortcuts (e.g., entries in the program group or desktop shortcuts) are specific to one environment only. However, from within IDLE, it is not immediately clear which environment is being used. You must open ArcGIS Pro and check the active environment.

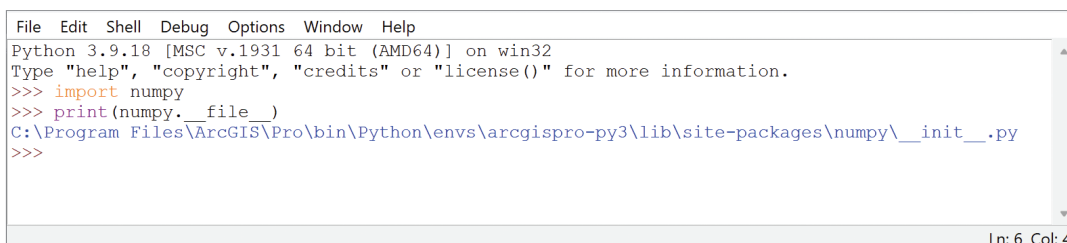
Alternatively, to check the environment being used from within IDLE, run the following code in the IDLE Shell:

```
>>> import numpy
>>> print(numpy.__file__)
```

Note that there are two underscores before and after `file`. This code prints the location where the files for one of the packages reside—in this example, the NumPy package. This location indirectly tells you the environment being used. When using the default environment, the result looks something like this:

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages\numpy\__
init__.py
```

In IDLE, the result is shown in the figure.



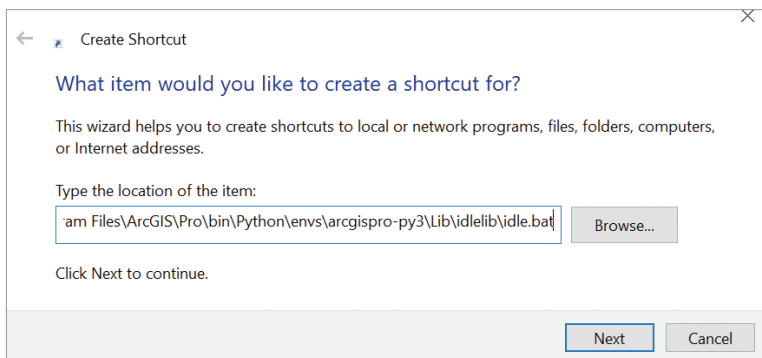
```
File Edit Shell Debug Options Window Help
Python 3.9.18 [MSC v.1931 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy
>>> print(numpy.__file__)
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages\numpy\__init__.py
>>>
```

Ln: 6 Col: 4

The result confirms that the environment being used is arcgispro-py3.

Although IDLE is installed as part of the ArcGIS Pro installation, no program shortcut is created because every environment has a different version of IDLE. Therefore, be careful when using any shortcuts you create for IDLE because they are specific to one environment.

To add this shortcut in Windows, right-click on your desktop, and click New > Shortcut. On the dialog box, enter the path to run IDLE. If you are not sure that is the correct location, you can also browse to the file.



Click Next, and in the next dialog box, enter a meaningful name such as IDLE for Pro. This step creates a shortcut with a generic icon by default.



You can replace the generic icon with the actual IDLE icon. Right-click the shortcut, click Properties, and on the Shortcut tab, click Change Icon. Browse to the idle.ico file, which is typically located in the following location:

C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\Lib\idlelib\Icons\idle.ico.

This file replaces the icon with the actual IDLE icon.



You can now double-click the icon to start IDLE.

**Note:** *You have several alternatives to launching IDLE. For example, an `idle.exe` file is located in `C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\Scripts`. This file also launches IDLE but keeps a terminal window running in the background, which can be distracting.*

Every Python installation comes with its own version of IDLE. If you have ArcGIS Desktop 10.x installed, you typically will have a shortcut under the ArcGIS program group called IDLE (Python GUI). If you are using ArcMap 10.8.2, and you start IDLE using this shortcut, you will see that it uses Python 2.7.18.4. To work with ArcGIS Pro, you should use only the version of IDLE that is installed with ArcGIS Pro in the location referenced previously. Using this version of IDLE ensures that you have the correct version of Python and have access to the packages needed to work with ArcGIS Pro.

Descriptions of the various menu items of IDLE can be obtained from the help menu. On the IDLE Shell menu bar, click Help > IDLE Help to view these descriptions. More information on IDLE can be found at <https://www.python.org/idle>.

Notice that the last line in the IDLE Shell window starts with the symbol `>>>`. This notation is the **prompt** of the interactive Python interpreter. This prompt signals where you can type code and press Enter, and the **interactive interpreter** carries out your command.

Are you ready for your first line of Python? At the prompt in IDLE, type the following text after the prompt:

```
>>> print("Hello World")
```

When you press Enter, the following output appears:

```
Hello World
>>>
```

What's happening here? When you press Enter, the interactive Python interpreter reads the input command, printing `Hello World` to the next line, and gives you a new prompt on the following line, waiting for the next input. The term **print** here refers to writing text to the screen, not sending a file to an external printer.

You have now seen why Python is called an **interpreted programming language**. When you finish typing your commands and press Enter, these commands are interpreted and immediately carried out.