Chapter 5: Jupyter Notebooks

# Objectives

* Understand iPython development patterns
* Get an overview of Jupyter Notebooks
* Review where Jupyter Notebooks exist in the ArcGIS ecosystem
* Build a Notebook effectively showcasing the results of an analytical exercise

# Introduction

Writing a Python script can be a daunting task if you’re not entirely sure of all the steps you’re going to want to do. All the patterns we’ve discussed up to this point involve being very thoughtful about our process and our solution. There are other development patterns out there, though, that will let us write and test our code one step at a time.

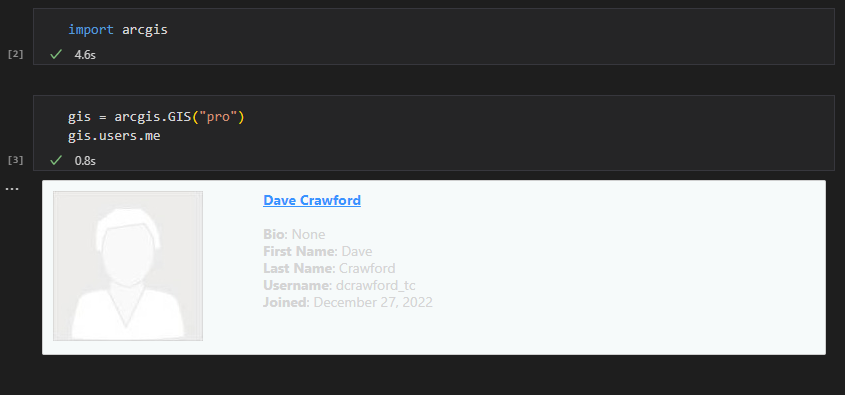
In this chapter, we’ll look at some interactive Python development patterns and how we can use them with specific regard to spatial data and automation. First, we’ll introduce some basic concepts with interactive terminals. Then we’ll dive into Jupyter Notebooks as a development environment and documentation tool. Finally, we’ll explore how Jupyter Notebooks fit into the ArcGIS ecosystem and write a polished notebook to showcase the results of our analysis.

# Interactive Terminals

Depending on which development environment you work in, you may see some variation of the phrase “Interactive Terminal” or “Interactive Window” in your options. Using an interactive terminal is a bit different than writing your script and running the whole thing. An interactive terminal allows you to write and execute smaller blocks of code and interrogate the results.

Many interactive development environments (IDEs) incoroporate an interactive iPython terminal. iPython is an open-source interactive shell that allows the user to have a more back-and-forth writing experience. In a practical sense, it allows you to run blocks of code and gives you back the results. The blocks of code in this case are called “cells”.

Interactive iPython terminals were designed with data scientists in mind and have some handy tools that we’ll explore in our exercise. One of the big features that comes with iPython is a concept called “magic”. iPython has plenty of built-in magics that can apply to either a single line or an entire cell. You can also write your own magic.



<insert 05-01>

<alt>An example of an interactive window in Visual Studio Code.</alt>

Figure 5-1. Visual Studio Code’s interactive window executing code and displaying results.

# Jupyter Notebooks

So now that we’ve got our heads wrapped around iPython, there’s one more thing we need to cover in this chapter before we get our hands dirty and start writing some code. We’re going to be using Jupyter Notebooks in our exercises. Jupyter Noteboks are special open-source development environment where a Python kernel is running behind the scenes and all the interaction takes place in a web browser. Because all the interactivity is built into a web browser, Jupyter Notebooks can have lots of great interactive widgets and in-line documentation. Jupyter Notebooks are excellent environments for early-stage script development or exploratory data analysis.

Similarly to iPython interactive terminals, Jupyer Notebooks consist of series of cells. Each cell consists of a user input and (depending on what you write) an output returned from the Python kernel.

# Tutorial 1-1: Open up a Jupyter Notebook

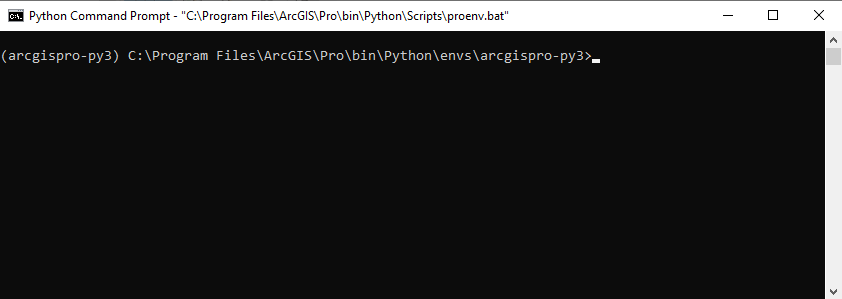
Now we’re almost ready to get writing. We’ll see later in the chapter that there are easier ways to open up a Jupyter Notebook, but it’s worth learning how to do it the hard way. We’ll learn a little bit about creating a Notebook from the command line.

## Start a Jupyter Notebook from the Command Prompt

The Python Command Prompt is a tool that comes with ArcGIS Pro, but you could do this same thing with any command prompt that can access an Anaconda environment that has Jupyter Notebooks installed.

1. Open the “Python Command Prompt” from your Start Menu

When you open the Python Command Prompt, you’ll see a command prompt with two components. On the left, in parentheses, is your current Anaconda environment. In the case of the screenshot below, the environment is ArcGIS Pro’s default environment “arcgispro-py3”. To the right of the environment is a folder path. This folder path is a location on your computer, probably the default location of the ArcGIS Pro environment.



<insert 05-02>

<alt>The Python Command Prompt showing the environment and local path</alt>

1. Change the directory of the command prompt

This is something people often forget to do before they start Jupyter Notebooks. It’s a good idea to change our directory here because the file navigation within the Jupyter Notebook interface is limited. Let’s navigate to the path for Chapter 5 in the sample data you downloaded. In my case, the path is C:\Top20Python\Chapter05. We’ll use a handy command called “cd” which stands for “Change Directory”. Run the following code in the command prompt (using your folder path).

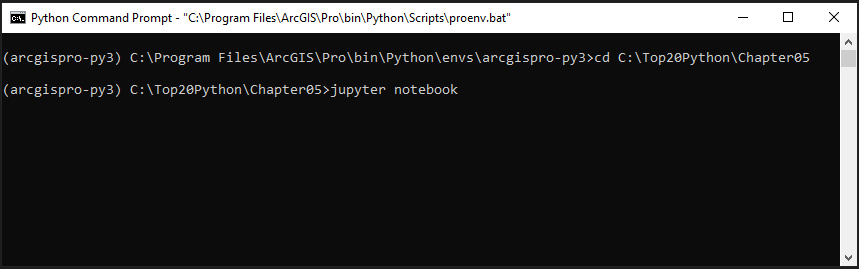
cd C:\Top20Python\Chapter05

You should notice now that the directory or path of the command prompt has changed. This is beneficial going forward because now we can open Jupyter Notebooks in this directory.

1. Open Jupyter Notebooks

This is going to be easy. Once you’ve changed the directory of your prompt to the folder you want to open Jupyter Notebooks in, you can just use the following line of code to open Jupyter Notebooks.

jupyter notebook



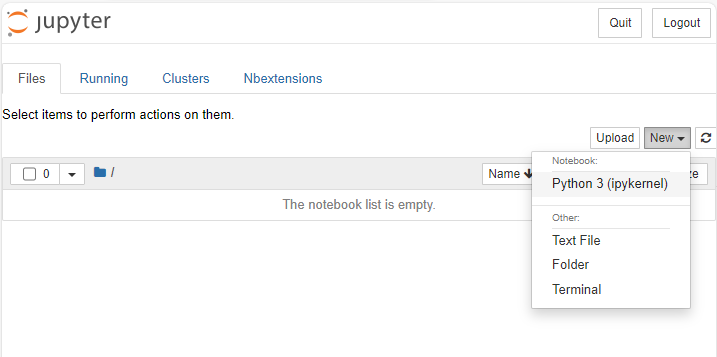
<insert 05-03>

<alt>Python Command Prompt showing the first three steps opening Jupyter Notebooks</alt>

After you run the last line of code, Jupyter Notebooks should start up and open a browser window. If not, check the text that gets displayed in the command prompt. Jupyter should display a URL that you can enter to access the Notebooks interface.

1. Create a new Notebook

In the web browser Jupyter Notebooks interface, find the drop down that says New then choose Python 3 (ipykernel).



<insert 05-04>

<alt>Jupyter Notebooks interface highlighting the new notebook dialogue.</alt>

A new tab will open in your browser that has a Jupyter Notebook.

### Explore some properties of Notebooks

Before we get into some more in-depth code, let’s explore some of the special properties of Notebooks. These little things can be helpful in your future development.

1. Execute the code in a cell

This may be a little different than the development you’re used to if you haven’t used an interactive terminal before. Now that you have a notebook open, you can start writing code. Let’s start by importing a package in the first cell. Type the following code in the first cell

import arcgis

Now you can execute that code by either finding the **Run** button or clicking **Shift+Enter**. While the code is executing, you’ll see an asterix to the left of the cell. Afterward, there will be a number indicating the order in which you executed the code. At this point, there should be some text to the left of your first cell that says “In [1]” indicating that you executed that cell first.

You may have noticed that there is no corresponding “Out” to our input. This is because we imported a package and didn’t return anything. Let’s write some code that actually returns something now. First we’ll create a GIS object. Then we’ll return that object so that we get a display.

gis = arcgis.GIS(“pro”)

gis.users.me

This should return a nicely formatted description of the account you’re currently using in ArcGIS Pro (similar to Figure 05-01).

1. Explore the doc-strings and source code for a function

This is one of the handiest things that Jupyter Notebooks allows you to do. If you’ve worked with Python packages before, you may have had to do a lot of consulting of package documentation. In a Notebook, you can directly reference the documentation in the code of a package as you’re writing. Let’s explore by writing the code

arcgis.GIS?

When you execute this code, Jupyter will bring up the documentation string (doc-string) for the GIS class in a window in the bottom of your browser tab.

On your own.

If you run the code from Step 6 but add an additional question mark, Jupyter will return all the source code for the GIS object.

# Notebooks in the ArcGIS Ecosystem

There are several places where you can use Notebooks in Esri software. In this section, we’ll explore a few and highlight the benefits of each.

## Notebooks in ArcGIS Pro

Notebooks were incorporated into ArcGIS Pro at version 2.5. Since then, they’ve integrated with a lot of the familiar components you may already be working with. You can create a new notebook in ArcGIS Pro by clicking the **New Notebook** button on the **Insert** toolbar of the ribbon at the top of the user interface. You can also create and them through the **Catalog View or Pane**.

The Notebook environment in ArcGIS Pro has some special features. Your Notebooks can interact with maps and reference layers by name. This environment also has access to the **arcpy** package, making it ideal for developing geoprocessing workflows.

## Notebooks in ArcGIS Online

This is one of the most interesting places to use Notebooks in ArcGIS. Because all of the infrastructure and

# Tutorial 1-1: Design labels for your map

You will create labels for your map based on a variety of attributes and set the labels to turn on and off depending on how zoomed in you are. Keeping the map from getting cluttered with text is an important part of cartography.

## Set up an ArcGIS Pro project

You will open the project and familiarize yourself with it.

1. Browse to Chapter1\Tutorials, and open Tutorial1-1.aprx.
2. Go to the West Village bookmark.
3. Review the Attribute Table for the Streets feature class.

This bookmark will zoom you to the area of interest.

Bookmarks are a useful tool for sharing locations of interest and presenting a particular view.



<insert 01-02>

<alt>Snapping tool options with snapping turned on and default settings highlighted.</alt>

Graphical user interface, text, application

Description automatically generated

<insert 01-03>

<alt>Label Class group with Class set to Class 1, Label Features In This Class check box checked, Label unchecked, and Field set to ArrFName.</alt>

1. On the Imagery tab, in the Analysis group, click Function Editor.
2. Right-click the Jersey\_Streets layer, and click Attribute Table.



<insert 01-04>

<alt>First five records for the X\_Coor and Y\_Coor fields in the attribute table.</alt>

## Project your layer

The Sidewalks\_2021 feature class has an unknown coordinate system. You will specify the desired coordinate system with the Input Coordinate System parameter and create a new projected feature class with the Project tool.

Note: You will frequently encounter unknown coordinate systems in your data. You should learn the three or four most commonly used coordinate systems in your industry to most effectively specify the most suitable system.

1. On the Analysis tab, in the Geoprocessing group, click Tools.
2. In the Geoprocessing pane, type Project in the search box.
3. Click the Project tool.
4. In the Project tool pane, apply the following settings:

* For Input Dataset Or Feature Class, select JC\_Assault\_Post.
* For Output Dataset Or Feature Class, type Post\_Assault.
* For Output Coordinate System, select NAD\_1983\_StatePlane\_New\_Jersey\_FIPS\_2900\_Feet.

Tip: The parameter for Geographic Transformation will autopopulate based on the tool’s determination of the best transformation, so you will rarely need to modify this.

Graphical user interface, text, application

Description automatically generated

<insert 01-05>

<alt>Project tool pane with parameters added. Geographic Transformation set to NAD\_1983\_To\_WGS\_1984\_1.</alt>

**Make “Mistakes” Now**

Try picking a projection from another geographic location, for example from Antarctica. Use your AOI bookmark to see if the projection shifted your points. It’s a good practice to check your projected features to catch if anything went wrong. Remove the incorrectly projected feature class from you Contents Pane and delete it from your Catalog. Deleting data can’t be undone, so you should be sure you want to, in this case we’ll delete it since we definitely don’t need that bad data cluttering up our project.

On your own

Project the remaining layers in your map using the skills you just learned. Think about how this may change the acreage you are investigating.

This image is a result of steps 6-8.

Map

Description automatically generated

<insert 01-06>

<alt>ArcGIS StoryMap choropleth map of the US with state boundaries. States are in shades of green for population, with graduated purple points for income level. The three largest points are shown in California, New York, and Maryland.</alt>

Editor- Please check the name of this Esri product, ArcGIS StoryMap

# Take it to the next level

You aren’t stuck picking from the set of stock vector graphics, you can use a picture marker of your own making/finding. Here’s how!

# Summary

In this chapter, you learned . . .

## Exercise workflow

1. Add the ACLED data for the African continent.
2. Examine the dataset’s attributes.
3. Follow preconfigured tasks to perform the following:
4. Limit the layer’s definition so that it shows only conflicts in a single country during specific years.
5. Rename and symbolize the defined layer.
6. Select only those conflicts classified as “violence against civilians.”
7. Create a new task to perform the following:
8. Summarize the number of fatalities within the selected set of features.

**Information at your fingertips**

**Resources**

# User Story

References

Fernando Pérez, Brian E. Granger, *IPython: A System for Interactive Scientific Computing*, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: https://ipython.org