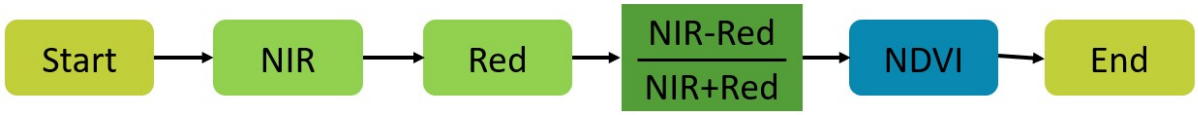
EXERCISE 1
Planning a Script



Introduction

This course is designed for users to learn the basics of either Python, Google Earth Engine (JavaScript), or R. To go through these exercises, you should **choose one** of these languages. Before writing a script there are several planning tactics and best practices to become familiar with, such as writing comments, constructing headers for your scripts, and creating flowcharts to aid in the planning of your scripts. Information in this exercise will help prepare you for concepts you will learn in later exercises.

Objectives

- Become familiar with the structure of flowcharts for planning a script
- Learn to write comments and create headers

Required Software

- Choose and install **one** software to use for this course
  - Python 2 (installed via ArcGIS)
  - Python 3 (installed via ArcPro)
  - RStudio
  - Google Earth Engine (no install required)

USDA Non-Discrimination Statement

In accordance with Federal civil rights law and U.S. Department of Agriculture (USDA) civil rights regulations and policies, the USDA, its Agencies, offices, and employees, and institutions participating in or administering USDA programs are prohibited from discriminating based on race, color, national origin, religion, sex, gender identity (including gender expression), sexual orientation, disability, age, marital status, family/parental status, income derived from a public assistance program, political beliefs, or reprisal or retaliation for prior civil rights activity, in any program or activity conducted or funded by USDA (not all bases apply to all programs). Remedies and complaint filing deadlines vary by program or incident.

Persons with disabilities who require alternative means of communication for program information (e.g., Braille, large print, audiotape, American Sign Language, etc.) should contact the responsible Agency or USDA's TARGET Center at (202) 720-2600 (voice and TTY) or contact USDA through the Federal Relay Service at (800) 877-8339. Additionally, program information may be made available in languages other than English.

To file a program discrimination complaint, complete the USDA Program Discrimination Complaint Form, AD-3027, found online at How to File a Program Discrimination Complaint and at any USDA office or write a letter addressed to USDA and provide in the letter all of the information requested in the form. To request a copy of the complaint form, call (866) 632-9992. Submit your completed form or letter to USDA by: (1) mail: U.S. Department of Agriculture, Office of the Assistant Secretary for Civil Rights, 1400 Independence Avenue, SW, Washington, D.C. 20250-9410; (2) fax: (202) 690-7442; or (3) email: program.intake@usda.gov.

USDA is an equal opportunity provider, employer, and lender.

Vermillion, M.; Scott, W.; April 21, 2021. Exercise 1: Planning a Script. GTAC. Salt Lake City, UT: U.S. Department of Agriculture, Forest Service, Geospatial Technology and Applications Center. 2 p.

Table of Contents

Part 1: Setting Up a New Script

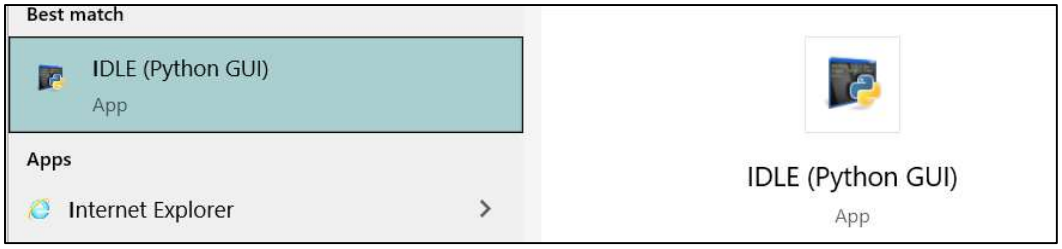
In all the three scripting environments covered in this course, you will learn how to write stand-alone scripts that can be saved, edited, and shared.

A. Open and Save a New File

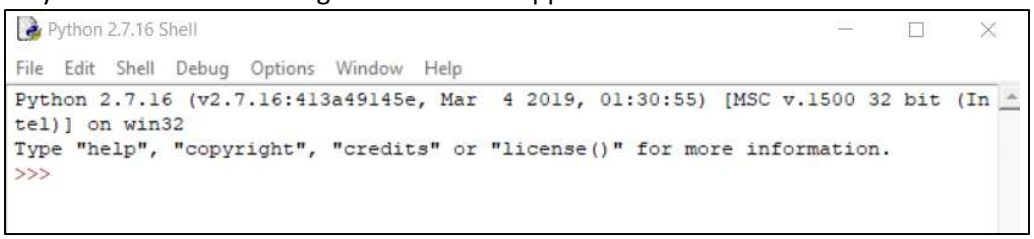
In this course we will be saving scripts to use again later. To save a script, scroll down and follow the steps below specific to your scripting environment.

1. Python 2 (ArcGIS)

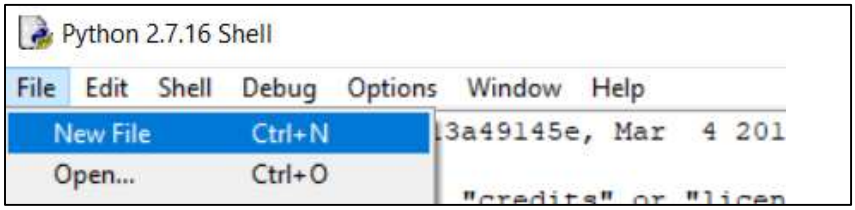
i. Open an IDLE window by navigating to the lower left-hand corner in the 'Type here to search' then type 'IDLE'. Click on the IDLE icon that appears



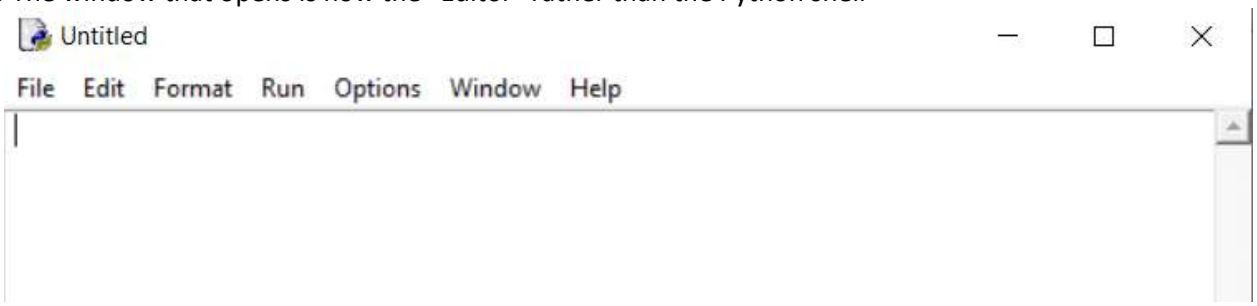
ii. A Python shell like the image below should appear



iii. Click **File** and the select **New File**



iv. The window that opens is now the "Editor" rather than the Python shell

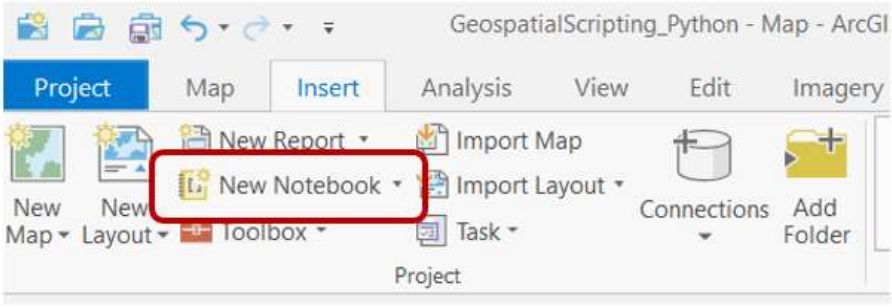


To save it as a script and reuse it later, click **File**, then **Save**. Save the script as "GeospatialScripting_ex1.py". This must be saved with the .py extension to run as a Python script

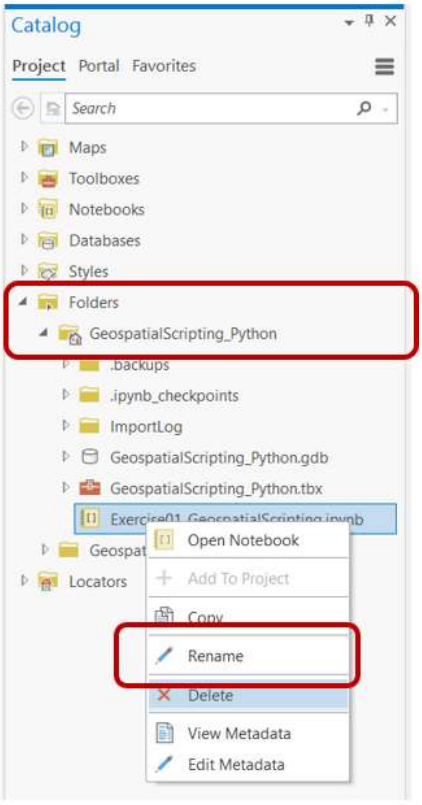
2. Python 3 (ArcPro)

i. Launch ArcPro and save the project as "IntroGeospatialScripting"

ii. Open a Jupyter Notebook from within ArcPro by navigating to the **Insert** tab then select **New Notebook**



iii. Rename the notebook by navigating to **View** > **Catalog Pane** in the **Catalog Pane** drop down the **Folders** option, then select the project home folder to view your new notebook



Right-click on the notebook and select **Rename**

Rename the notebook to **Exercise 1**

iv. Save the notebook by hitting the **Save** button in the upper left-hand corner or by pressing **cntrl+s**

v. Read and review 5 Tips to Get You Started With Jupyter Notebooks

vi. Here is the Jupyter Notebook User Interface

3. **R**

   i. Open RStudio

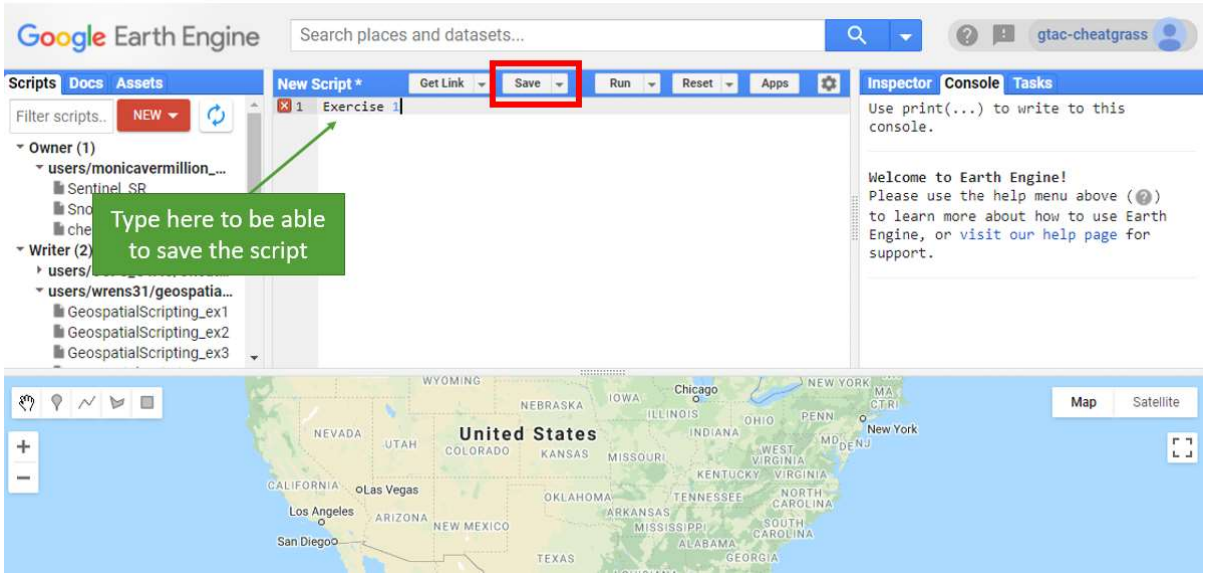
   ii. Click File, then New File, then R Script

   iii. A new R script will open in the window above the console in RStudio. To save the script click File, then Save. Save the script as "GeospatialScripting_ex1.R". This must be saved with the ".R" extension to run as an R script

4. **Java Script**

   i. Navigate to an Earth Engine Code Editor window

   ii. Under the "New Script" there must be something typed to save the script. Type something under "New Script" then save it!



   iii. Click **Save** in the center column on the upper half of the window

   iv. Give the script the name, "GeospatialScripting_ex1 " then click **Ok**

B. Comments

Comments in a script are important. Not to the computer, in fact the computer doesn't use them at all, but to yourself or other humans who may read your scripts later. Comments are sections that you type into your code that are ignored by the machine. Think of them as writing a note in the margin of a book. You haven't changed the story, just left a reminder for yourself or someone else. This can be useful to explain, in English, what a complicated line of code does both for other people and yourself!

Just like most commands written in code, the syntax to create comments is language specific. In all three languages, comments are preceded by the comment symbol.

1. In the scripting window you opened in Part A, begin by typing the comment, "This is a comment" in your script. A table showing the different syntax over languages is shown below.

Language	Environment	Symbol	Code
Python	IDLE or ArcPro	#	`# this is a comment`
R	RStudio	#	`# this is a comment`
JavaScript	Google Earth Engine	//	`// this is a comment`

C. Writing a Header

Headers are important for script maintenance and documentation. Headers should include information such as the author of the script, its purpose, and the date of last modification.

1. Delete the comment you made in the previous section using the backspace button
2. Type the comment symbol then 'Author: <your name>'

Language	Environment	Code
Python	IDLE or ArcPro	`# Author: <your name>`
R	RStudio	`# Author: <your name>`
JavaScript	Google Earth Engine	`// Author: <your name>`

3. Add a comment with the current date to the next line
4. On the next line, create another comment with the purpose of the script. You can write something like, "Intro to geospatial scripting, exercise 1".

Get into the habit of including a header at the beginning of each script you create. Especially if you're planning on passing along the script to someone else, or if you may not use the script again for a long time. The author, date, and purpose are always a good start, but be sure to include any pertinent information you or another user may need to know about the script in the future.

Part 2: Planning a Script

Writing a script can get complicated quickly, so it is important to have a plan before you dive in. One great way to plan a script is to create a flowchart.

This part of the exercise walks you through creating flowcharts to see how processes in a script are executed. If you would like to follow along and create your own flow charts as you go you can draw your own in Microsoft PowerPoint or Microsoft word, or use a convenient drawing tool such as Draw io

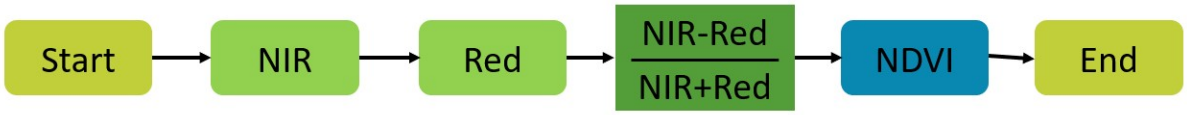
Of course, you can also draw the flowcharts by hand on a sheet of paper. The flowcharts here are drawn for you, so you can simply read along.

A. Visualizing a Process

The process, or the flow, of a script can be easily seen working in a flowchart. This makes it easy to see what you need to write into a script and identify potential errors in a script before you code them.

1. Below is an example of how to calculate a vegetation index (in this case, NDVI). You can read more about NDVI here. To calculate an NDVI image, all you need is a Near-Infrared (NIR) band and a red band. Observe the flowchart below.

2. The yellow start box is where the process begins. This process has two inputs, the NIR band and the red band, which are shown in the light green boxes. In a script, these inputs will be variables, which you will learn about in the next exercise.

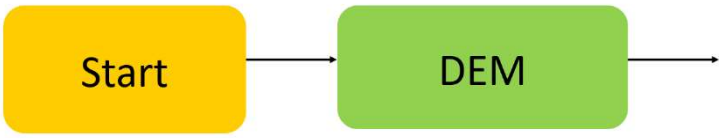


The inputs go into the equation in the dark green box and the NDVI as the output in the blue box. This is a simple example, so you probably don't need a flowchart but as code gets more complicated, programmers use flowcharts to organize the flow of the script before they begin writing any code. You might also hear this referred to as pseudocode!

B. Conditional Flowcharts

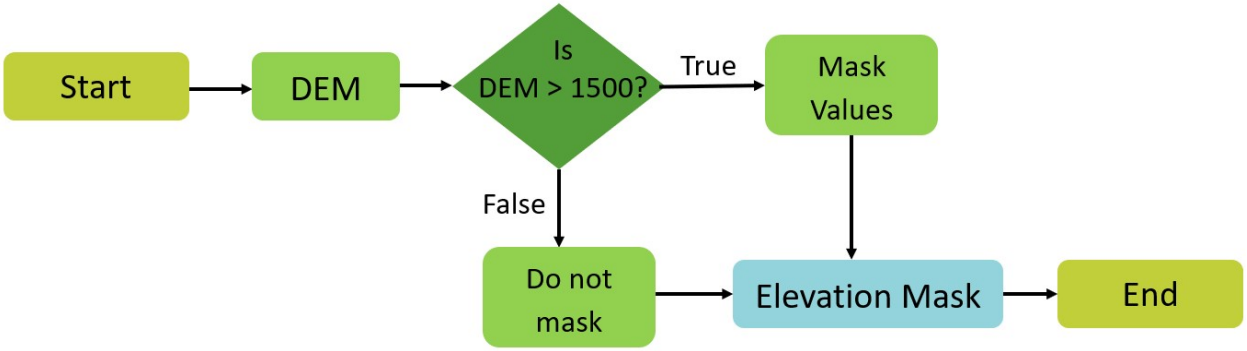
Conditional statements are pieces of code that will only run if some condition fulfils a value (true or false). This section will help visualize a conditional statement.

1. Here we're making an elevation mask from a Digital Elevation Model (DEM). What are the inputs here?



In this case the only input is a DEM, which is a raster with elevation values. Now say you want to mask out values above 1500 meters

2. In the flow chart below, you can visualize the process of a conditional statement in a script as it evaluates a true or false value (whether a value is above 1500 meters or not). The condition is in the dark green diamond



Learning to plan scripts this way is a fantastic tool to make efficient scripts when you begin scripting. It becomes especially important when you begin writing longer, more complex scripts. It will help you identify potential errors in your script before you even get started, which will save time and effort down the road.

As you continue through the remaining exercises in this course there will be flowcharts like this to help you see how processes are executed in a script.

Part 3: Notes and Other Resources

A. Planning Scripts

1. Programiz: Flowchart Programming
2. Code Avengers: Planning Flowcharts

B. General Coding Resources

1. Software Carpentry
   i. Programming with R
   ii. Programming with Python
2. Earth Lab
3. Python Documentation
4. ESRI Python for ArcPro
5. Five Tips to Get Your Started with Jupyter Notebook
6. Automated Mapping using Python in ArcGIS Pro
7. Google Earth Engine Beginner's Cookbook
8. GEE and Python Youtube

Congratulations! In this exercise you learned how to set up a new script by saving it and including a header. You also learned how to begin planning a script by writing out a process in a flow chart. In the next exercises you will learn how to begin writing these processes into a program.