

GIS TUTORIAL FOR PYTHON SCRIPTING 1ST EDITION

Pdf Free



•
•

Author: David W Allen
ISBN: 9781589483569

Gupta, P. Yadav A Python toolkit for visualizing greenhouse gas emissions at sub-county scales. Michael Wohlstadter, Lubna Shoaib, ... J. Fishman Environ. Topics from this paper. Citation Type. Has PDF. Publication Type. More Filters. View 2 excerpts. Research Feed. The tool dialog box has a lot of options for inputs, outputs, and analysis settings, reflecting the nature of the Huff Model.

A tool Help file provides detailed descriptions of how the tool works, including the inputs, model formula, and outputs, as illustrated in the diagram

that is part of the Help file and shown in the figure. All the code for this tool is provided as a single Python script, part of which is shown in the figure. Given the complexity of the Huff Model, the Python script is quite lengthy, at over lines of code. So as soon as you become familiar with what these elements are, you can start actually reading and understanding some of the more complex scripts. Once you learn the basics, from the chapters in this book, of how to use Python for writing scripts, you will find one of the best ways to keep learning Python scripting is to work with existing code. Using example code can also speed up the process of writing your own scripts. A computer script is essentially a list of commands that can be run by a certain program or scripting engine.

Scripts are usually just plain text documents that have a specific file extension and contain instructions in a particular scripting language. Most scripts can be opened and edited using a basic text editor. You can work with Python in a number of ways. The most basic approach is to use the so-called command line. If you have used other programming languages, you may be familiar with this type of interface. So instead of using the command line interface, it is typically much more productive to use a Python script editor. A Python editor has a menu-driven interface and tools for organizing and testing Python scripts to make working with Python easier. Python editors are also known as integrated development environments IDEs.

There are many different ones, including several open source and commercial packages. To a large degree, the editor you use is a matter of preference, and experienced Python programmers all have their favorite ones. Python syntax remains the same for different editors, which is one of the strengths of Python. GUI stands for graphic user interface. IDLE is also known as the Python shell. Descriptions of the various menu items can be obtained from the Help menu. This is the prompt of the interactive Python interpreter. It is where you can type code and press Enter, and the interactive interpreter will carry out your command. Are you ready for your very first line of Python?

When you press Enter, the interactive Python interpreter reads the input command, prints the string Hello World to the next line, and gives you a new prompt on the following line, waiting for the next input. The term printing here refers to writing text to the screen. You have now seen why Python is called an interpreted programming language. When you are finished typing your commands and press Enter, the commands are interpreted and immediately carried out. What if you type something else that does not make sense to the interactive Python interpreter? For example: You will also notice something else about the interactive Python interpreter. As soon as you type some text, it is given a color based on the nature of the input. For example, as soon as you type the word print, the word turns orange. Similarly, the string Hello World turns green. This is a way for the interactive Python interpreter to show how the input is being interpreted—in this case, orange is for statements and green is for strings.

This is called syntax highlighting and is a helpful means of error checking as you write code. When you are learning the basics of Python syntax, it is useful to work directly in the interactive Python interpreter. You get immediate results that way, and you can keep going with new lines of code without worrying about having to save your work. However, when you are ready to write slightly more complex, multiline code, it will be more beneficial to write it as a script you can save. So remember that code written to the interactive Python interpreter is not meant to be saved. Now you can see how writing a script differs from writing code in the interactive Python interpreter. This opens a new window called Untitled. It is a script window, and there is no prompt. A script needs to be run as a program for the command to be executed. Before you can run a script, however, it needs to be saved.

The file extension. Now it is time to run the script. The string Hello World is printed to the interactive Python interpreter. When you are coding, it is useful to have both the interactive Python interpreter and your script s open at the same time. If you want to try out something very quickly or check the syntax of a particular line of code, you can use the interactive Python interpreter.

The script window contains the actual lines of code you want to save and keep working on. Occasionally, you can test your script by producing results that are printed to the interactive Python interpreter. One widely used Python editor on the Windows platform is PythonWin. For the remainder of this book, it is assumed that you have access to PythonWin. The syntax, however, is the same, independent of the editor. The major differences between editors lie in how Python scripts are created, organized, and tested, although the syntax is the same. The corresponding exercise chapter, Exercise01, available online on the companion book resources page, includes instructions on how to install PythonWin. The basic PythonWin interface is shown in the figure. A new script window appears that can be resized so that both your interactive window and the script window are visible.

In the example that follows, the script is saved as hellowin. In the script window, enter the following line of code: The result is printed to the Interactive Window. Notice that the syntax highlighting in PythonWin is slightly different. About the Author Dr. Paul A. His teaching includes courses in introductory GIS, spatial analysis and modeling, spatial statistics, and GIS programming. His research focuses on the robustness of spatial analytical techniques in the areas of crime analysis, environmental science, public health, spatial ecology, and water resources.

He has published over 50 journal articles and book chapters. He lives with his family in Vancouver, British Columbia, Canada. Table of Contents Preface Acknowledgments Chapter 1. Introducing Python Chapter 2. Working with Python editors Chapter 3. Learning Python language fundamentals Chapter 5. Geoprocessing using Python Chapter 6.

[Early Bird Special!!! and 174 Other Signs That You Have Become a Senior Citizen download book](#)

[\(geheime\) liefde van zonen pdf](#)

[Theorie leren bromfiets download pdf, epub](#)

[Introduction to Physical Anthropology 15th edition pdf, epub, mobi](#)