EGM722 – Programming for GIS and Remote Sensing

Week 2, Part 1: Using other people’s code

Week 2 Outline

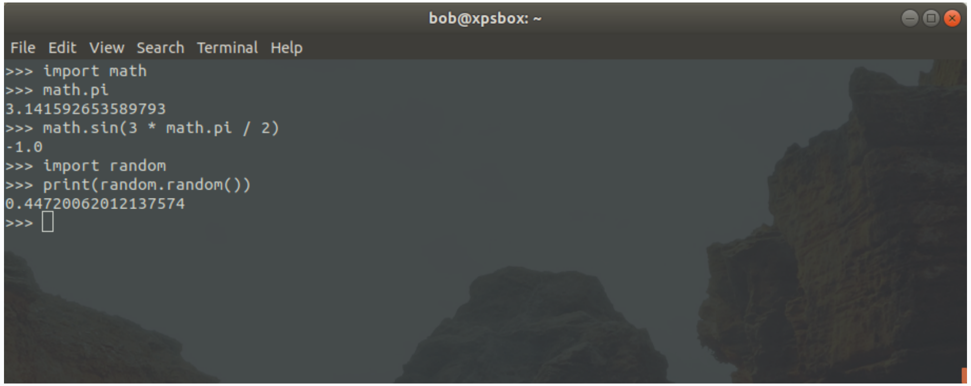
1. Using other people’s code
2. Help and documentation
3. Errors and debugging
4. Classes and objects

Standard Library

- In addition to the built-in functions, python ships with standard library (link)
- Includes tools for math-related functions, random numbers, multiprocessing, etc.
- These are not automatically available – we have to tell python to use them

Modules

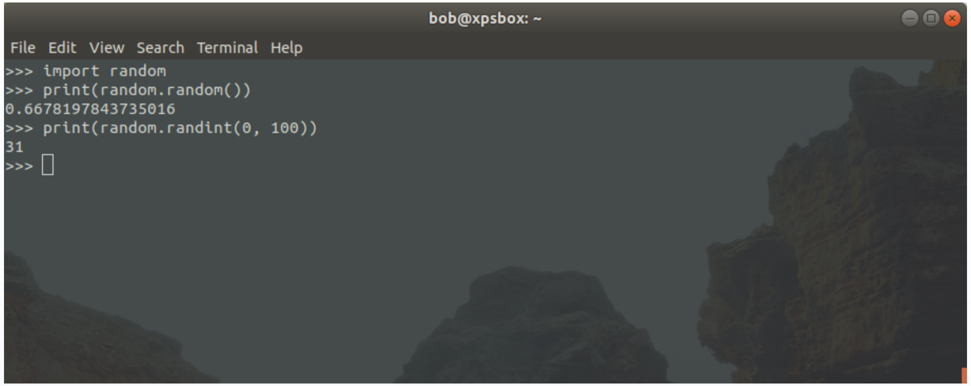
- Python provides ability to import and use modules
- Module: a file that contains a collection of functions and classes
- e.g., math, random, geopandas, arcpy



```
>>> import math
>>> math.pi
3.141592653589793
>>> math.sin(3 * math.pi / 2)
-1.0
>>> import random
>>> print(random.random())
0.44720062012137574
>>> 
```

Import statements

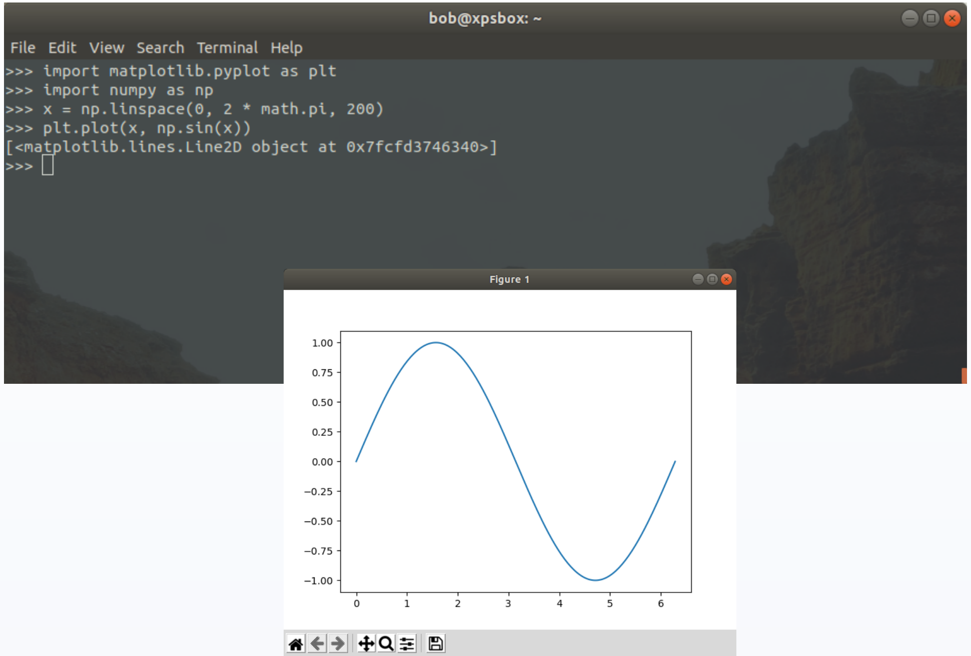
- To import a whole module, use import statement:
  - e.g., import random
- Can now use all functions/classes contained in random:
  - e.g., random.randint()
- In scripts, recommended to limit import statements to single module at a time



```
>>> import random
>>> print(random.random())
0.6678197843735016
>>> print(random.randint(0, 100))
31
>>> 
```

Aliasing

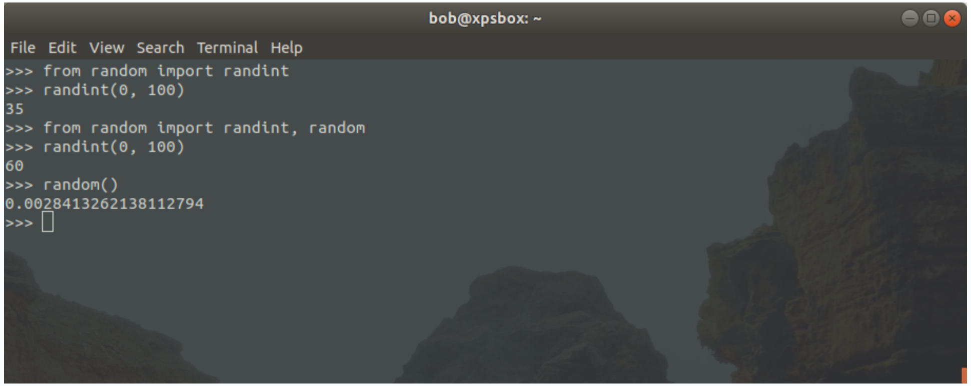
- Sometimes, module names are too long
- Shorten these using an alias:
  - import matplotlib.pyplot as plt
  - import numpy as np
- Use the aliased name in script, interpreter:
  - plt.plot(x, np.sin(x))



```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = np.linspace(0, 2 * math.pi, 200)
>>> plt.plot(x, np.sin(x))
[<matplotlib.lines.Line2D object at 0x7fcfd3746340>]
>>> 
```

Import from

- If we only want a single function (or submodule), use from:
  - from random import randint
- Can separate imports using commas:
  - from random import randint, random
- Can also use *, but this is **not recommended**



```
>>> from random import randint
>>> randint(0, 100)
35
>>> from random import randint, random
>>> randint(0, 100)
60
>>> random()
0.0028413262138112794
>>> 
```

Summary

- Python comes with a number of standard modules for a variety of functions
- To use these, or any modules/packages beyond built-in functions, we must import them
- Import statements tell the interpreter:
  - What module(s) or functions to import
  - What to call them
- Avoid using * imports, as it makes it difficult to keep track of variable and function names