EGM722 – Programming for GIS and Remote Sensing

Week 1, Part 1: Why programming?

Week 1 Outline

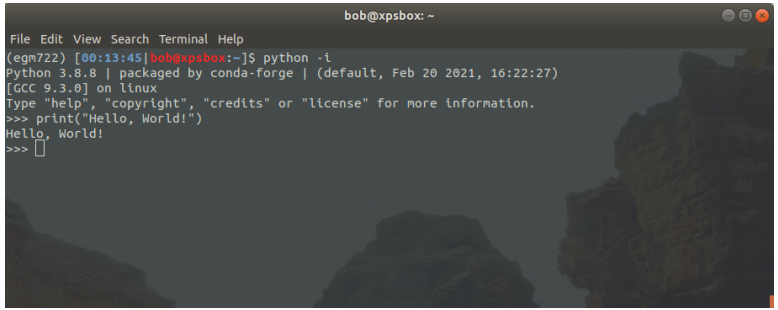
1. A brief introduction to programming using python
2. Built-in types
3. Controlling Flow
4. Functions
5. A brief introduction to git

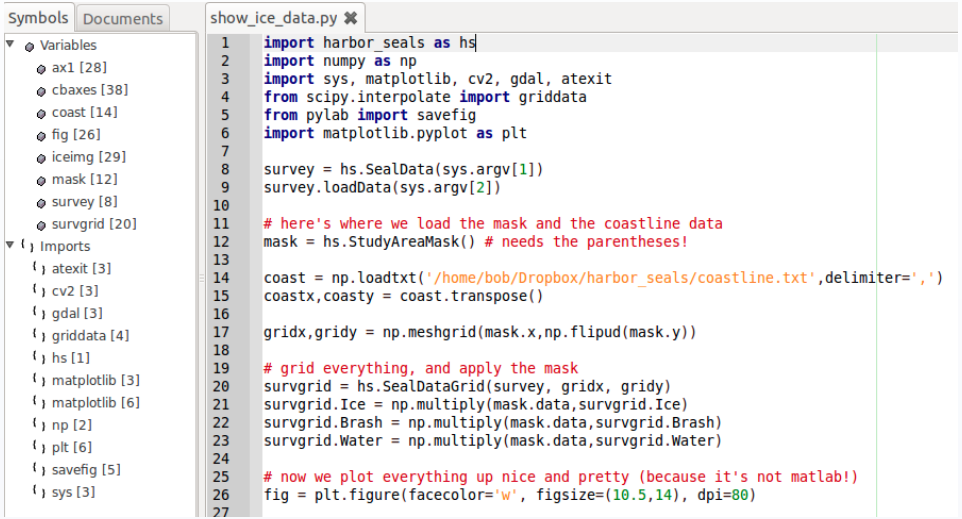
Why programming?

- In GIS and Remote Sensing, we often repeat the same tasks over and over
  - e.g., processing new/different data for different areas
- These tasks often form a workflow
  - What happens when you need to re-do a single step in the middle or beginning of the workflow?
  - What happens when someone else needs to do the same steps that you did?
  - What happens if we're lazy and don't want to do the same tasks over and over?
- Computers: really good at repeating the same tasks over and over and over and over and over and over and over and over and over and…
  - But (for now), we have to give them instructions: programming

What is python?

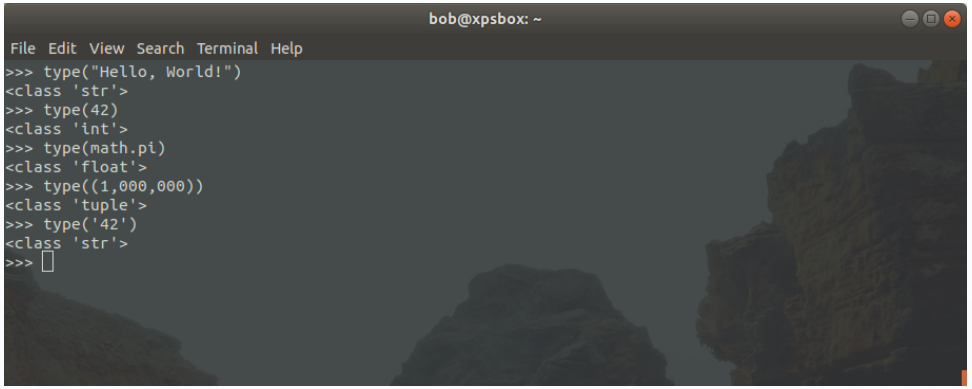
- An interpreted, high-level language
- Python interpreter:
  - Reads code
  - Translates it
  - Executes it
- Run in two ways:
  - Interactive mode
  - Script mode





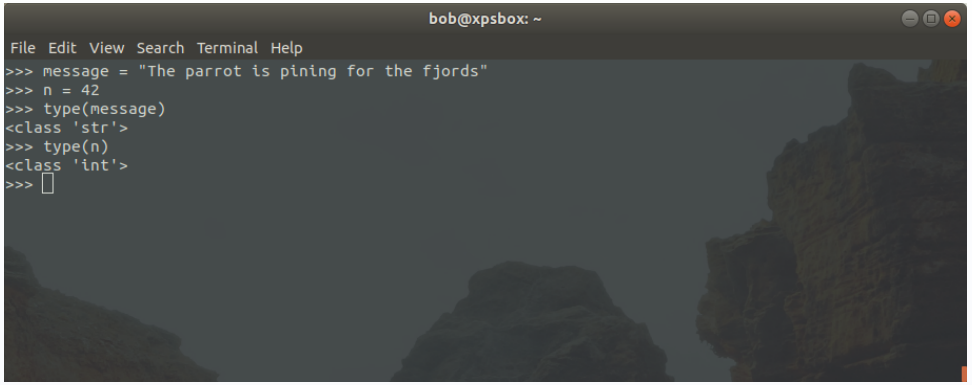
The world is filled with objects

- Python is an object-oriented programming language
- Object: the basic “thing” that python works with
- Objects have:
  - type
  - properties
  - methods



Variables

- A variable is a name that refers to an object
- “Like a box in the computer’s memory where you can store a single value” (Swiegart, 2020)
- If we want to save values to use later, have to store them in a variable
- To create variables, use assignment statements
  - NB: ‘=’ does not mean ‘equal to’!



Naming variables

- Good practice:
  - Choose meaningful names
  - Names must begin with a letter
  - Can contain underscores
- Bad practice:
  - Names cannot contain illegal characters (@, !, etc.)
  - Cannot be a protected keyword (and, for, if, etc.)
  - Try not to overwrite built-in types/classes (list, int, etc.)

Operators

- We use operators to perform some kind of computation
- Examples:
  - +: addition (concatenation for strings and lists)
  - -: subtraction
  - *: multiplication (also works for strings, lists)
  - /: division
    - In python 3, normal division (e.g., 4/5 = 0.8)
    - In python 2, floor division (e.g., 4/5 = 0)
    - python 2 is (mostly) gone now, unless you use ArcMap
  - **: exponentiation (NB: ^ is a separate operator)
  - %: modular arithmetic

Order of operations

- Python follows PEMDAS:
  - Parentheses
  - Exponentiation
  - Multiplication/Division
  - Addition/Subtraction
- Operators with same precedence are evaluated left to right

- 2*(3 – 1)
- (1+1)**(5-2)
- 2**1+1
- 3*1**3
- 2*3-1
- 6+4/2

Expressions and Statements

- An expression is a combination of objects, variables, and operators:
  - 42
  - x
  - x+42
- A statement is a unit of code the interpreter can execute
  - Assignment statements (e.g., x = 42)
  - return, pass statements

Summary

- Programming is giving a computer instructions to execute tasks
- Python: one language we can use to give a computer instructions
- Python uses objects to carry out computations and other tasks

Additional resources

- Automate the Boring Stuff with Python (2nd ed.)
- Automate the Boring Stuff with Python [youtube]
- Learnpython.org
- Beginner's Guide to Python [python.org]
- Python documentation (3.8.8) [python.org]