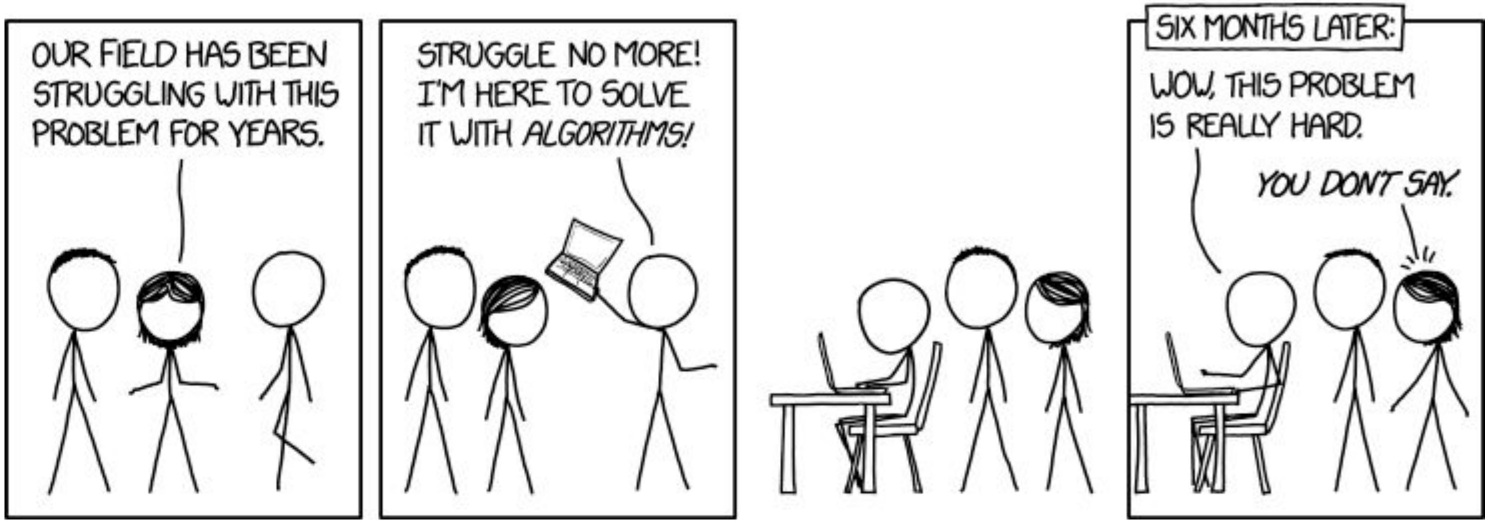
GEOS F436/636 Beyond the Mouse

Christine (Chris) Waigl
University of Alaska Fairbanks – Fall 2018
Week 1: Instruction (Aug 28)


OUR FIELD HAS BEEN STRUGGLING WITH THIS PROBLEM FOR YEARS.
STRUGGLE NO MORE! I'M HERE TO SOLVE IT WITH *ALGORITHMS!*
SIX MONTHS LATER:
WOW, THIS PROBLEM IS REALLY HARD.
YOU DON'T SAY.

Today's program

- Was is programming?
- Why do we want to learn how to program … as scientists
- What is a computer? What is a programming language?
- What does it mean to approach a problem computationally
- A first look at MATLAB and our environment

Why don't we just learn computer science?

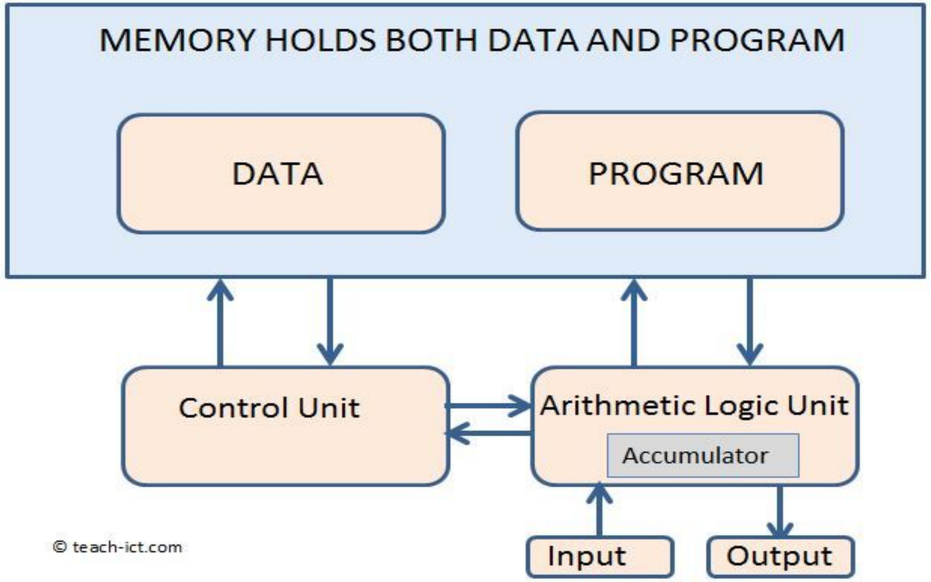
- Computer science is the science of what computers can do
- Programming is the skill, or art, to get the computer to do what we want

(Think of the difference between Natural Resources Management and Biology. Between GIS and Geography.)

What is a computer, and where does the programming language come in?

Source: Wikipedia

This model is called von Neumann architecture (after John von Neumann, 1903-1957)



http://teach-ict.com/2016/GCSE_Computing/AQA_8520/3_4_computer_systems/344_systems_architecture/von_neumann/miniweb/pg2.htm

Via the programming language we communicate with the computer.

There are many types of programming languages. The distinctions come with trade-offs.

- Interpreted vs. compiled (speed of execution vs. ease of development)
- Proprietary vs. open-source (price and community vs. paid customer support… sometimes!)
- Imperative vs. declarative vs. functional (oh, boy!)

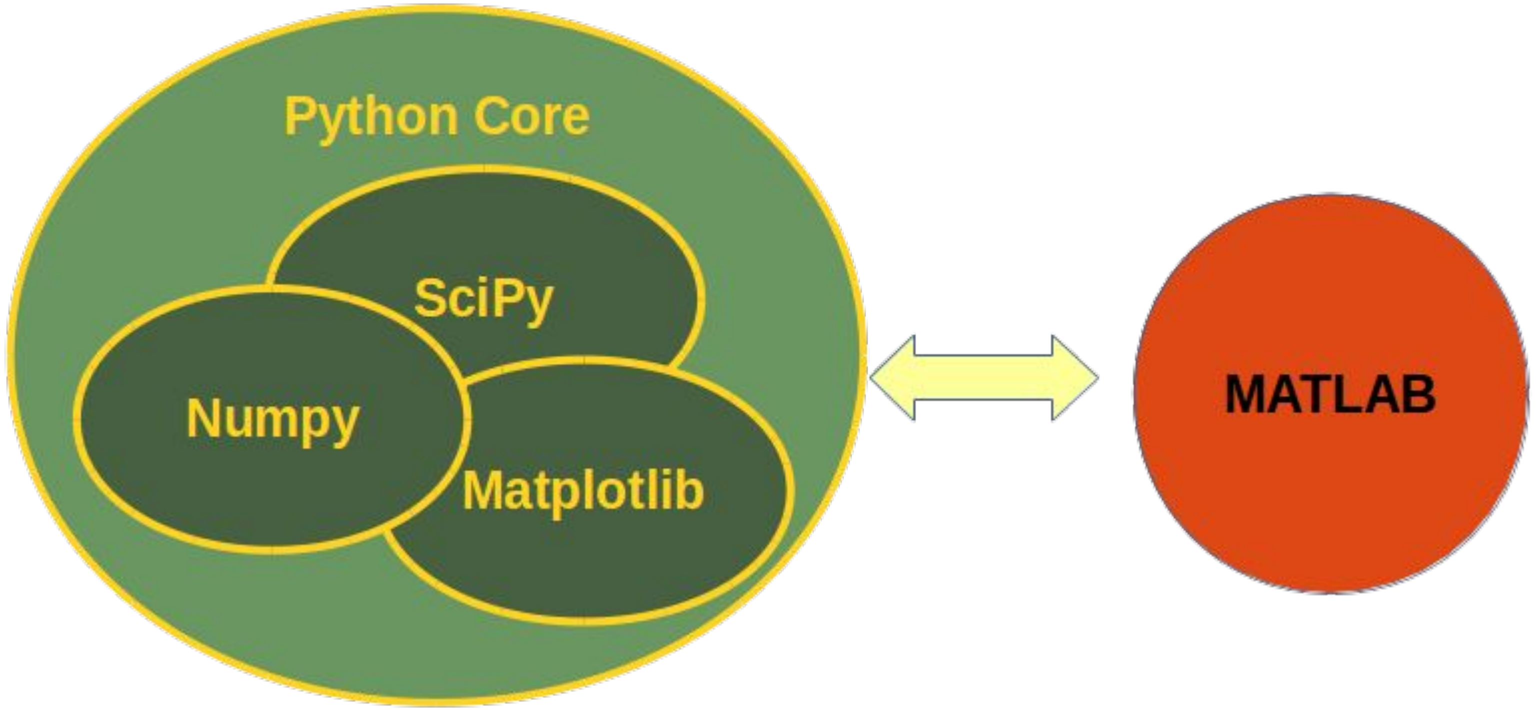
Examples

- C (and C++): Imperative, compiled, open-source OR proprietary
- Python: Imperative, interpreted, open-source
- MATLAB: Imperative, interpreted (with JIT = just-in-time egine), proprietary
- Python also offers JIT engines! For example Numba

How do we select a good programming language?

The most commonly used programming languages in the geospatial sciences are: Python, MATLAB, R.

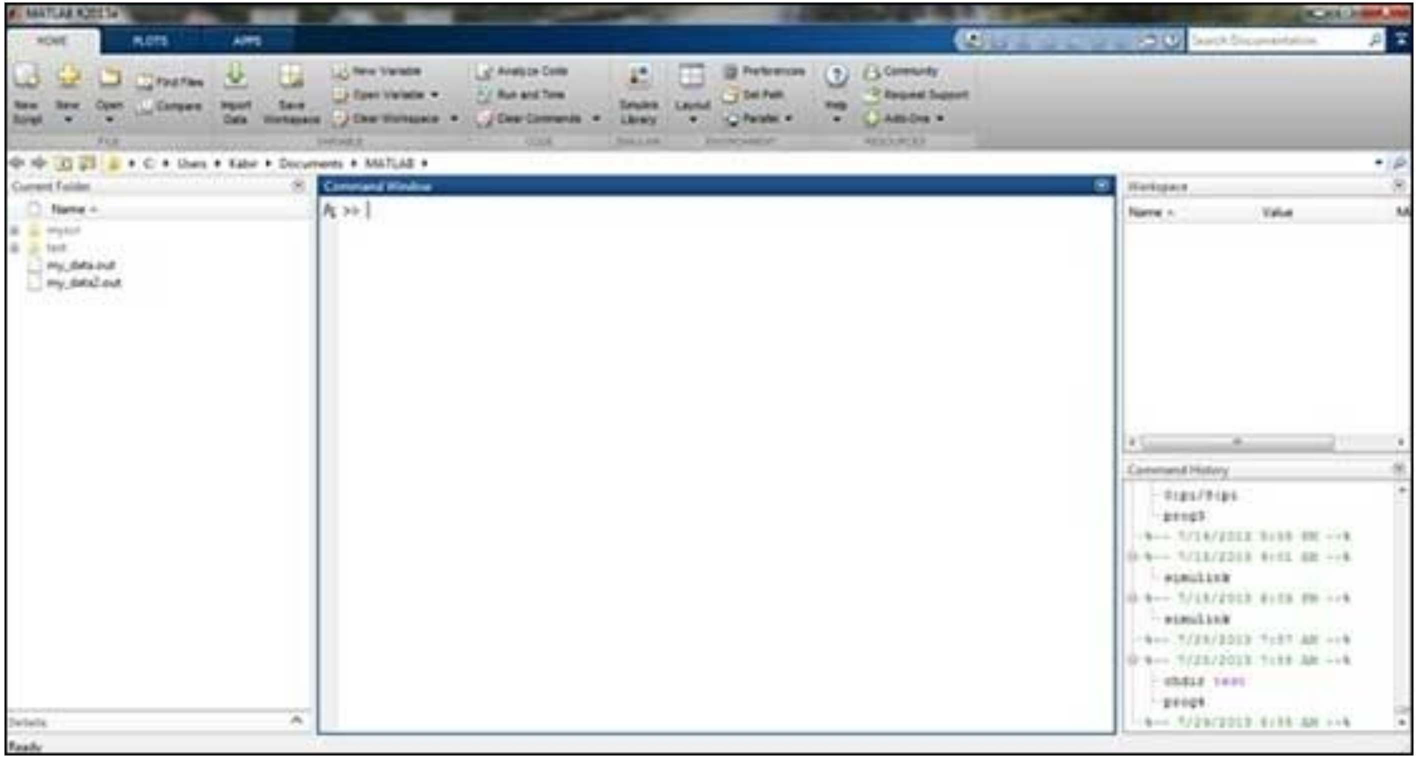
MATLAB is proprietary and more monolithic (less modular) than Python



Which programming language we choose depends on many factors!
There is no absolutely best programming language, just best for a certain purpose.
Community of users is often as important as the properties of the language.

For example, some people say that while a program in FORTRAN or C (compiled) runs in 1h, it may take 1 week to write, while a program in Python or MATLAB may take 1h to write, but runs 1 day.

⇒ You may reach your goal faster in a slower language!



```
>> |
```

```
figs/figs
prog3
%-- 7/14/2013 5:55 PM --%
%-- 7/15/2013 8:01 AM --%
simulink
%-- 7/15/2013 4:59 PM --%
simulink
%-- 7/25/2013 7:37 AM --%
%-- 7/25/2013 7:44 AM --%
chdir test
prog4
%-- 7/29/2013 6:55 AM --%
```

Let's play!

```
>> 1 + 1

>> 1 - 4 + 10 / 2      (<-- what does this teach us?)

>> ans + 2             (<-- what is "ans"?)

>> 1 + 2;              (<-- what does the semicolon do?)

>> ans                 (<-- what do you expect?) '’
```

These things are called *literals.* They come in *types*. Between them are *operators*.

Types and literals and variables

Try this:

```
>> class(ans)
```

Does the answer surprise you?

"class()" is a MATLAB *function*. Type "help class" to find out what it does.

"ans" is a *variable.* You can assign variables with the = *operator.*

You can compare variables with the == *operator*.

What can we do with variables?

("ans" was a *variable.*)

We can assign variables. And compare variables. Try these steps, one by one:

```
>> test_var == 1
>> test_var = 1
>> test_var == 1
>> test_var == 5
>> test_var >= 5
```

Some programming on paper

How would you find the largest of three numbers?

Let's discuss!

Then let's watch a video:

https://www.mathworks.com/videos/solving-a-sudoku-puzzle-using-a-webcam-68773.html

What two steps do you expect the developer to address?

Let's try to diagram the steps the Sudoku solver requires.

Don't forget to use "save your workspace".

You can save your workspace in a .mat file. And load it next time!

BRING YOUR OWN USB STICK.

AND/OR FIND A NICE WORKSTATION TO RETURN TO

Some answers to thinking questions

- You can use the MATLAB Command Window like a calculator. Operator precedence applies just like you learned in school.
- If you leave the semicolon off the end of a statement, the calculated value will be assigned to a variable called "`ans`" and printed to the Command Window.
- If you end a statement with a semicolon, nothing will be printed to the Command Window (at least not without a command from you!), but the variable "ans" is available and can be used in calculations.
- Numbers are by default floating-point numbers, not integers. But they are displayed without a decimal points if the there are only zeros after the decimal point.

More answers!

- You can use the function `class` to find out the type of a variable. A variable that displays with the value 1 could be of type (class) 'double' 'int16', 'int32', 'int64' (these are integers) or 'logical' (Boolean, values 1 and 0, or true and false).
- We will learn more about floating-point numbers ('float' or 'double') later, but for the moment you just need to know that floating-point numbers is how we can store numbers that are not integers, such as rational or real numbers (always approximately, thus "double precision"). If you divide a float by a float, the result will be another float (if it's a valid division). If you divide an integer by an integer, the result will be forced to be an integer.

Information about MATLAB

- You can always ask for help. Type

  ```
  help unknowncommand
  ```

  … and follow the instruction

- All variables that you have created are available for you to inspect in the Workspace part of your IDE. You can delete them using the command `clear`.
- You can also delete old output from your Command Window using the command `clc`
- Try `help clear`!

… cont'd …

- Make sure you understand the difference between the operator `=` and `==`:

  `=` assigns a variable, ie, it sets the value of the variable to the left. For example: `>> a = 2`

  `==` compares the left-hand side and the right-hand side. It returns true (1) or false (2) depending on whether the two sides are identical.

- In the lab, we will encounter our first *arrays.* These are sequences of variables (or literals). Character arrays are created with single quotation marks: `'Hello world!'.` In general, arrays are created in many ways, One is to use square brackets: `[1, 2, 3]`

MATLAB scripts and functions

Most of the time, your MATLAB code will exist in a script, which consists of one or several files containing code. You'll still have the (very useful!) Command Window available while you're working on a script.

You have already called functions. For example `class` is a function. Or `sin` or `log`. You can, and should (why?[*]) create your own functions. In MATLAB, except in the newest version, typical user-provided functions each have their own file.

[*] Why? Because, remember, programming is about breaking a problem down into smaller steps and solving the smaller, simpler problem to build a solution for the larger problem. Functions solve one small problem (usually), and can be reused as often as you need them.

The "largest of three numbers" exercise

Your task is to figure out how to structure a program that calculates which of three numbers is the largest, You do this by breaking the task down into smaller steps,

The critical questions to ask you are:

- What do I need to know? (= What are the inputs?)
- What should the output be?
- What tools do I need to accomplish the task?
- What are the successive steps that get me there?

The first step is to assign the three numbers to variables. It is easier to re-use variables and solve the problem for *any* three numbers, not just specific ones.

MATLAB conditionals: A first look at `if`

Another insight that became clear when thinking about the problem was that a program sometimes (often!) has parts that are only relevant if some condition si true. This phenomenon, or structural element, is called "branching" or sometimes "conditional statements". In MATLAB, the syntax (=way of expressing) a conditional is, in a basic form:

```
if somecondition-is-true
    do-this-thing
else
    do-some-other-thing
end
```

Optional reading

- Hahn & Valentine ch. 1.1.1, 1.1.2, 1.1.3, and 3.1 **or** Attaway, ch 1.1 to 1.4
- R. Grapenthin, "Computer Programing for Geosciences: Teach Your Students How to Make Tools". https://eos.org/opinions/computer-programing-for-geosciences-teach-your-students-how-to-make-tools (by the original instructor of this course)
- P. Guo, "Why Scientists and Engineers Must Learn Programming". https://cacm.acm.org/blogs/blog-cacm/166115-why-scientists-and-engineers-must-learn-programming/fulltext