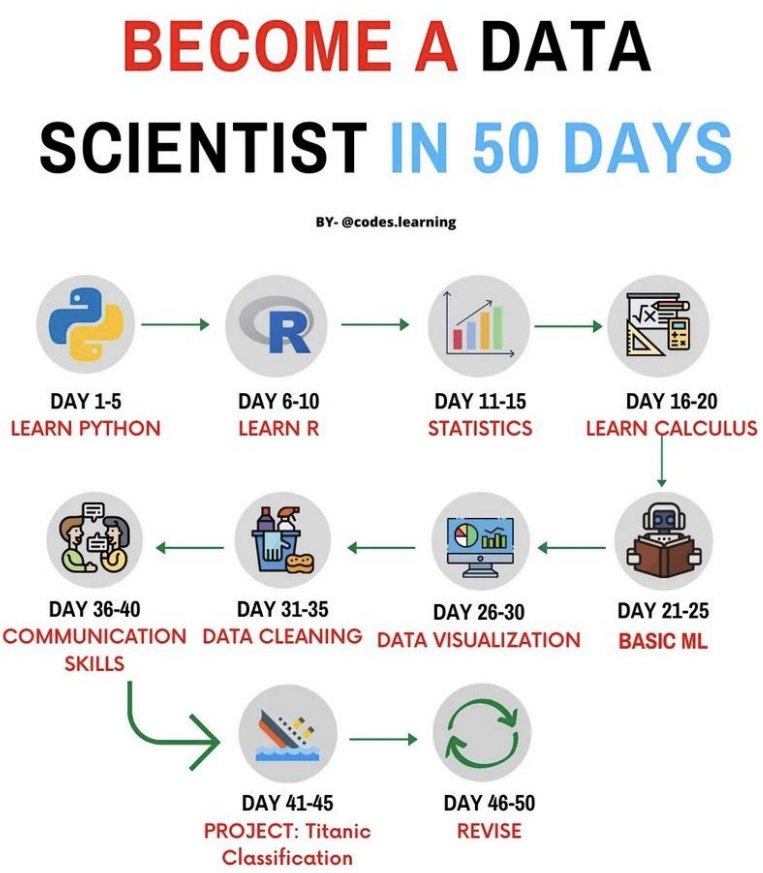
Agenda - Introduction to programming in Python

Lyon College, Summer school 2023

Marcus Birkenkrahe

May 24, 2023

Day 1: Entry survey analysis and course overview



☒ Go to Canvas (lyon.instructure.com)

☒ Open Zoom (lyon-edu.zoom.us/j/85828233316)

☒ Entry **survey** - review of your responses.

- What is the OS and why is it relevant to programming?
- What is the command line and why is it important?
- Which languages do you already know (10!): FORTRAN, Visual BASIC, SQL, R, Stata, Java, C#, Python,C,Lua
- Topics wanted: 1) **webscraping**, 2) **Machine learning**, 3) Cybersecurity[1]
- Tools: VS Code/Emacs IDEs known
- Terror: I will never show up drunk (in class or anywhere)
- Time: 120 min in class, 180 min outside of class (per day)
  1. class: 15 min check in and review
  2. class: 90 min lecture + practice
  3. class: 15 min summary, questions, outlook
  4. home: 30-90 min DataCamp + 60-120 min revision/assignments

☐ Course **overview** - assignments, grading, topics, platforms.

☐ **Practice:** a first (literate) Python program & platforms.

☐ First **assignment**: "Getting started in Python" @DataCamp.

In-class summary

- The terminal allows the user to communicate with the OS.
- Python comes in different versions.
- Python (and all apps) need to be built/compiled for an OS.
- Microsoft (and others) combines technical and commercial interests.
- Python [the interpreter] is written in C.
- There are (many) different layers of software between user and machine (which leads to greater usability/lower performance).

[1] Webscraping Excel manipulation and Machine Learning are already part of my plan, but cybersecurity is currently not included (except indirectly through the command line and string manipulation skills).

Homework

- Register with GitHub, DataCamp, replit.com
- Get access to Canvas for materials and for Zoom
- "Infinite skills" exercise

Day 2: Python scripting infrastructure practice I

(A PDF and an Org-mode version of this exercise are on GitHub)

Infinite skills - infinite resources

Canvas announcement

- A math app that you cannot cheat with - an honor math app
- A resource matching app: e.g. match labor supply with demand (really efficiently), possibly including a market and negotiation engine
- A game where the world generates itself as soon as you reach the end of it including quest lines, NPCs
- An Internet Wayback machine without the snags where you can retrieve a version of a website fast and very granularly
- A script that can reach into SQL and pull values out of a column to monitor daily changes in the columns, e.g. money

My infinite skills Python projects:

1. An application that lets students know which remote or in-person data science internships are currently available in the USA.
2. An application that predicts cancer risk from medical images with no false negatives (i.e. images that show cancer but were not labeled as such).
3. An application that shows me the (publicly available) syllabi of all courses (world-wide) with the title and content of my upcoming courses.
4. An application that provides real-time insight into Lyon College data, especially retention, grade average, finance, incoming students, alumni vs. visitor data.
5. An application that finds funding opportunities, pre-fills them with available data (publications, personal, location) and identifies next steps.
6. An application based on a speech assistant coupled with a conversational chat bot that submits: early alerts, grades, summaries after each class as announcements.

Day review (and test preview)

We review yesterday's material. These questions will resurface in the weekly multiple-choice tests, i.e. you can use this to prepare.

1. What's a computer terminal?

   - A window to communicate with the Operating System (OS)
   - A command line interface (CLI)
   - A place to enter shell commands (like `cd`)

2. What are three strengths and three weaknesses of Python?

STRENGTH	WEAKNESS
Easy to learn and to read	Performance (speed)
Data science applications	Significant indentation
Automation and small scripts	Platform requirements

3. What language is the Python interpreter written in?

Figure 1: Family packing fruit, Sydneay 1911 (Flickr.com)

Answer: in the C programming language (which allows programming 'close to the machine' (making full use of its resources).

4. What are we going to use Schoology for in this course?

   Answer: for nothing. We use the Canvas learning management system, GitHub, DataCamp, replit.com, IDLE and Google Colaboratory.

5. What's our policy regarding the use of AI coding assistants?

   Answer: it's allowed as long as you reference and document your use.

6. What does Google Colab provide but Python's IDLE does not?

   Answer: An interactive notebook

7. What's the difference between a compiled language (for example C) vs. an interpreted language (for example Python or R)?

- Compiler: Human-readable source code is compiled to a machine-readable file that needs to be executed for results.
- Interpreter: source code is read and executed line by line at runtime, and no executable file is produced.
- Both compiler and interpreter are specific to the underlying computer architecture.

Practice

Figure 2: RAAF crew woring on newly arrived Lockheed aircraft (1940)

- Find Python on your computer
- Find Python for download to your PC on the web
- Starting and using Google Colab, IDLE, DataCamp and replit.com
- Understand the "literate programming concept"
- Understand when we're using Colab + IDLE + replit.com
- Options for creating and running Python scripts

- Creating and running our first ever Python script

In-class summary - what did we discuss today?

Figure 3: Women spectators at wrestling, Sydney stadium 1940 (Flickr.com)

- See also the whiteboard photos from the class session in GDrive.
- Practice file as PDF (class handout)
- Windows' Command Line Interface: finding and opening the terminal, prompt, absolute/relative path, root and file tree, CLI commands and options; `cd`, `python`, `ls`, `whoami`.
- Python's IDLE: interactive shell `>>>`, getting `help()` and finding documentation; Python keywords, symbols and modules (libraries).
- Python's file editor: printing a greeting in Python (hello world) with `print`. Default arguments. Comments (`#`), `.py` file types, debugger and traceback. Keyboard interrupt (`CTRL + c`).
- Literate programming: weaving documentation and tangling source code from a literate file that contains text + code + output. With Emacs: one document with code in < 44 languages.

- Google's Colab: code and text cells, scratch cell, GDrive backup, GitHub gist export, auto-completion, interactive notebooks with IPython (`.ipynb`), `matplotlib` and `numpy` modules. `import`, lineplot with `pyplot`. Colab dashboard. Markdown with `#` (headline).

Homework: first assignment

Figure 4: Max Williams photographed by Ian Grimes, 1942 (Flickr.com)

- The problem: word count.
- Requirements:
  1. Google Colab (platform for submission)
  2. String manipulation (function `str.count`)
  3. Getting input (function `input`)
  4. Storing variables with `=`
  5. Printing numeric output as string with `str`
  6. Bonus: nesting functions

Day 3: Infrastructure practice II: DataCamp, REPL

Day 4: Expressions, Variables, Statements

Day 5

Day 6

Day 7

Day 8

Day 9

Day 10

Day 11

Day 12

Day 13

Day 14

Day 15

Day 16

Day 17

Day 18

Day 19

Day 20

Day 21

Day 22

Day 23: Project presentations

Day 24: Project presentations / Summary / Outlook