

COURSE OVERVIEW

CSC 109 - Introduction to programming in Python - Summer 2023

Marcus Birkenkrahe

May 23, 2023

Contents

1 Who is your lecturer?	3
2 My first programming languages	3
3 Why Python? Why not?	5
4 Natural vs. programming languages	6
5 The importance of (understanding) infrastructure	9
6 What do you expect from this course?	11
7 What careers do our students have?	12
8 What will you do in this course?	12
9 How will you be evaluated?	14
10 Which tools are we going to use?	14
11 The Canvas LMS	16
12 The GitHub platform	18
13 IDLE (Integrated Development Learning Environment	18
14 Google Colaboratory IPython Notebooks	18
15 repl.it.com REPL + console + templates	24

16 DataCamp lessons + projects + workspace	26
17 Can we use AI coding assistants?	28
18 When are the first assignments due?	29
19 Will you need to buy any books for this course?	29
20 TODO Infinite skills	33
21 TODO Next / Any questions? / Review questions	33
22 TODO Practice: Python scripting infrastructure	35
23 Glossary	38
24 References	38



1 Who is your lecturer?



Figure 1: Photo of Poppy as a puppy

- Monty Python used to be one of my favorite shows:
 1. British humor (I went to school in London)
 2. Anarchic and irreverent (I was young once)
 3. MUDs (First online programming class)¹

2 My first programming languages

- My first programming languages:

¹As an alternative to Lisp (the original AI language and also the first choice for MUD programming), you can use Python (PyCon, 2020). Regarding the connection between Monty Python and MUD: see the mud scene from The Holy Grail (1975).



Figure 2: Texas Instruments TI-99/4 16-bit microcomputer, 1979

1. BASIC - Beginner's All-purpose Symbolic Instruction Code (1979)
 2. FORTRAN - FORMula TRANslator (1957)
 3. C++ (1989)
- How many programming languages are there today?

3 Why Python? Why not?

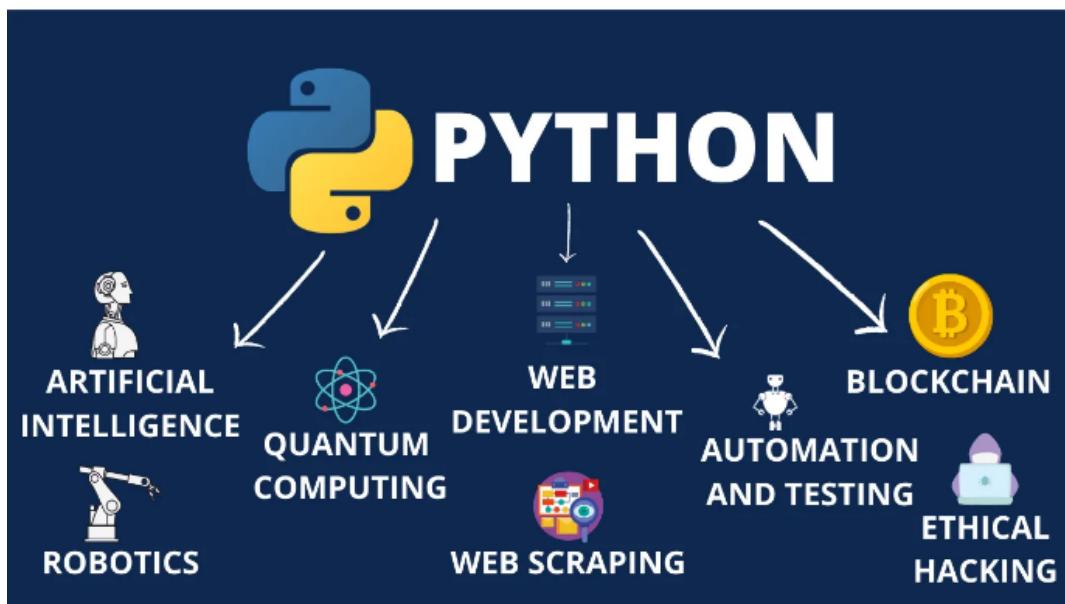


Figure 3: Why is Python so popular (Ozgon, 2021)

- WHY PYTHON?²
1. Readability and simplicity **
 2. Versatility and applicability **
 3. Large and active community
 4. Cross-platform compatibility
 5. Abundance of libraries and frameworks *

²These lists are the answers of ChatGPT (May 12 version). The number of * indicate my level of agreement.

6. Data science and machine learning ***
 7. Automation and scripting ***
 8. Career opportunities *
- WHY NOT PYTHON?
 1. Specific platform requirements **
 2. Performance-critical applications ***
 3. One-paradigm language (cp. R, Haskell, C/C++, SQL)
**
 4. Industry demand *
 5. You might just not like significant indentation *

4 Natural vs. programming languages

- C/C++ is like Latin



Figure 4: "C/C++ is like Latin"

- SQL is like English
- Lisp is like French
- R is like Italian
- Snap! is like Russian



Figure 5: "SQL is like English"



Figure 6: "Lisp is like French"



Figure 7: "R is like Italian"



Figure 8: "Snap! is like Russian"



Figure 9: "FORTRAN is like Hebrew"

- FORTRAN is like Hebrew
- HTML is like Gaelic



Figure 10: "HTML is like Gaelic"

- Python is like Spanish

5 The importance of (understanding) infrastructure

Some infrastructure that separates you from just "getting on" with it:

1. Network server / network
2. Keyboard / Screen
3. Operating system (OS)
4. OS shell / terminal / console
5. Python shell / console
6. File system
7. Compiler/interpreter
8. Editor

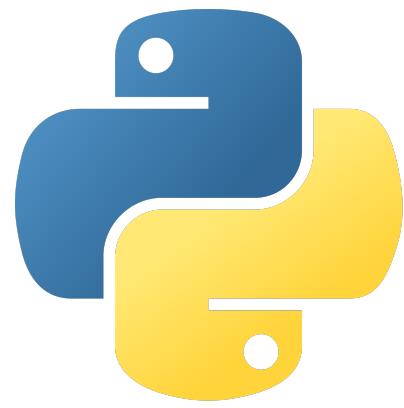


Figure 11: "Python is like Spanish"



Figure 12: Photo by Landon Martin on Unsplash

9. Middleware and meta data

10. Graphical user interface

(Most of these infrastructure components are written in C/C++.)

6 What do you expect from this course?



Figure 13: Dandelion in the grass, by Aaron Burden, @Unsplash

See also responses in the Google Forms survey.
What's your single most important personal goal?

What careers did our students have?



- Full stack developer @ OneBrief (military planning software)
- Cybersecurity @Sierra Nevada (aircraft engineering)
- Software engineer @Riot Games (League of Legends game company)
- Database specialist @DHS (Department of Homeland Sec)
- IT director @Bad Boy Mowers (High performance lawn mowers)

Figure 14: Careers of recent computer and data science graduates from Lyon College

7 What careers do our students have?

8 What will you do in this course?

- Part 1: Basics

1. Input and Output
2. Flow control
3. Functions
4. Lists and dictionaries

- Part 2: Applications

1. Strings and regular expressions
2. Webscraping and file manipulation
3. Working and visualizing tabular data
4. Machine learning: spam filter

- Part 3: Capstone Project, e.g.



Figure 15: Tad Dickerson (Bob's Flying Service) with AgNav system
(Flickr.com)

1. Program the Finch 2.0 robot (microbit) with Python - move / draw.
2. Setup and demonstrate machine learning on Google Coral Edge TPU.
3. Set my databot 2.0 up - extract data and visualize with Python.
4. Set up and program my Mycroft MKII (open source voice assistant)
5. Identify your own research problem and use Python to solve it
6. Program another 2D or 3D game in Python (e.g. with PyGame, using any of Sweigart's games in his 2016 book, or with Turtle).
7. Analyse and demo use of any of our course tools in your field of interest or expertise (GitHub, DataCamp, Colab, CLI or AI). E.g. complete and document (in Colab) an unguided DataCamp Python project (link to unguided relevant Python projects).
8. Create a literature (and practice) overview of Python in your field of professional or personal interest.
9. Something completely different related to introduction to programming (not necessarily with Python).
10. Take a topic from my GitHub issues list: [birkenkrahe/py/issues](#).

9 How will you be evaluated?

- All course requirements have deadlines (strict)
- Late submissions will be penalized (loss of points)
- No final exam but 4 weekly tests (graded)
- DataCamp/programming assignments are home assignments
- In-class practices are "literate programming" exercises
- You are expected to code along using Google Colab and/or IDLE

10 Which tools are we going to use?

1. Canvas (learning management system)
2. GitHub repository (all course materials except tests)

REQUIREMENT	UNITS	PPU	TOTAL	% of TOTAL
Programming assignments	4	25	100	25.
Class participation	20	5	100	25.
Multiple-choice tests	4	25	100	25.
Capstone project	1	100	100	25.
TOTAL			400	100.

Figure 16: From the course syllabus (Canvas, GitHub)

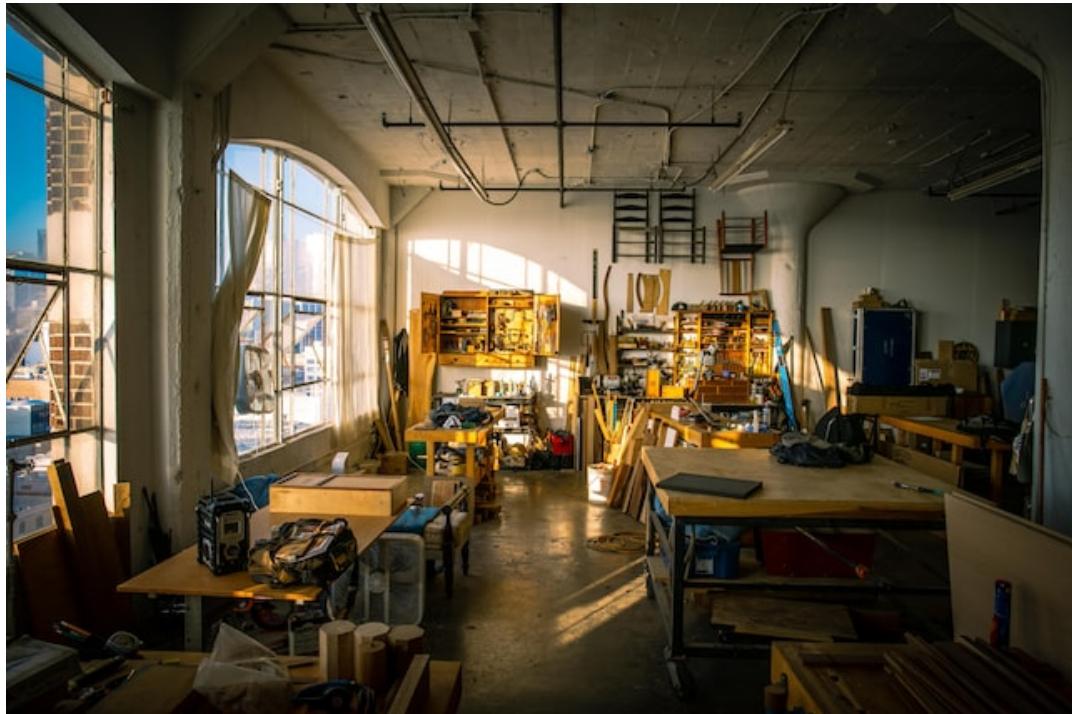


Figure 17: Unsplash, workshop

3. Google Colaboratory (online interative notebook / Literate IDE)
4. Command line interface interpreter - (shell / REPL at repl.it.com)
5. Python Interactive Development Learning Environment (IDLE)
6. DataCamp (data science online learning for Python, R, SQL)
7. AI (OpenAI ChatGPT, Google Bard, GitHub CoPilot w/editor)

ANY OF THESE TOOLS CAN BE THE BASIS OF YOUR COURSE PROJECT!

11 The Canvas LMS



The screenshot shows the Canvas LMS interface. On the left is a dark sidebar with yellow highlights for 'Dashboard' and 'Syllabus'. The main content area has a white header with 'CSC 109 01 > Syllabus' and a 'Immersive Reader' button. The header also includes '2022-2023 - Summer Term'. The main content is titled 'Course Syllabus' and describes the course as 'Introduction to Programming with Python' offered by 'Lyon College, Summer term I 2023, CSC 109 Derby Science Center Room 209, Mon-Fri 1pm-3pm'. It has sections for 'Overview' and 'Objectives', with detailed descriptions of the course goals and content.

Figure 18: Canvas

- Course announcements (notifications via email)
- Assignments for submission and online tests

The screenshot shows a learning management system interface with a sidebar on the left containing icons for Account, Dashboard, Courses, Calendar, Inbox, History, Commons, Help, and a back arrow.

The main area displays the course "DSC 305 Machine Learning spring 2023".

At the top right are buttons for "+ New" and "+ Filters". Below them is a search bar and a "Filters" button.

The page lists several questions:

- Multiple Answer | Question Data storage questions**
Which of the following questions are relevant when making **data storage** decisions?
This section includes edit, copy, delete, and collapse buttons.
- Multiple Choice | Question**
Complete the code ??? to return the output
x <- list("a" = 2.5, "b" = TRUE, "c" = c(1,2,3))
x[??]
This section includes edit, copy, delete, and collapse buttons.
- Output:**
: \$c
: [1] 1 2 3
This section includes edit, copy, delete, and collapse buttons.
- Multiple Choice | Question Confusion matrix**
Traffic sign data were used to train a kNN model. The vector `actual` contains the original (labeled) signs, the
This section includes edit, copy, delete, and collapse buttons.

Figure 19: Canvas

- Zoom links (participate remotely/cloud recording)
- Course links (GitHub, GDrive Whiteboard, DataCamp invite link)
- Gradebook (weekly updates)

12 The GitHub platform

- GitHub is the most popular software development platform.
- The 'Git' in the name is a version control software by Torvalds
- It also contains Gists (code snippets), Issues, and CoPilot (AI)
- Here for you: all course materials, and the general course FAQ
- No registration required but recommended (watch/fork/collaborate)

13 IDLE (Integrated Development Learning Environment)

- Built-in Python shell for interactive experimentation
- Includes file editor and "Turtle" graphics
- Local and private to your computer
- No registration required

More information:

- Video: What is Python IDLE and How to Use the IDLE Shell (6')
- Tutorial: Getting Started with Python IDLE + video lessons (25')
- Interactive Shell also available as Google Chrome extension

14 Google Colaboratory IPython Notebooks

- Free interactive (literate) notebook: text + code + output
- Upload to Lyon GDrive and/or GitHub/Gist
- Linux Docker container (virtual OS)

The screenshot shows the GitHub dashboard for user [birkenkrahe](#). At the top, there is a search bar and navigation links for Pulls, Issues, Codespaces, Marketplace, and Explore. Below the header, there is a pinned profile picture of Marcus Birkenkrahe. The main content area includes:

- Pinned:** A grid of five pinned repositories:
 - [org](#) (Public): FAQ, diary, organizational stuff for all my courses @Lyon. Stars: 2, forks: 2.
 - [ds2](#) (Public): Advanced introduction to data science (DSC 205) at Lyon College, Spring 2023. Stars: 1.
 - [ml](#) (Public): Machine learning course in R at Lyon College, Batesville AR, spring 2023.
 - [tm](#) (Public): Digital Humanities - text mining with R, Lyon College, CSC 105, spring 2023.
 - [cc](#) (Public): Introduction to programming in C and C++ at Lyon College, 2023. Stars: 1.
- Customize your pins:** A section for pinning repositories.
- Contributions:** A chart showing contributions in the last year, with a legend for 'Less' (light green), 'More' (dark green), and 'Learn how we count contributions'.
- Activity overview:** A summary of recent activity, including contributions to repositories like [birkenkrahe/admin](#), [birkenkrahe/org](#), and [birkenkrahe/os420](#).

Figure 20: My GitHub dashboard - close to 4,000 commits in 2022 (100,000 lines of code?)

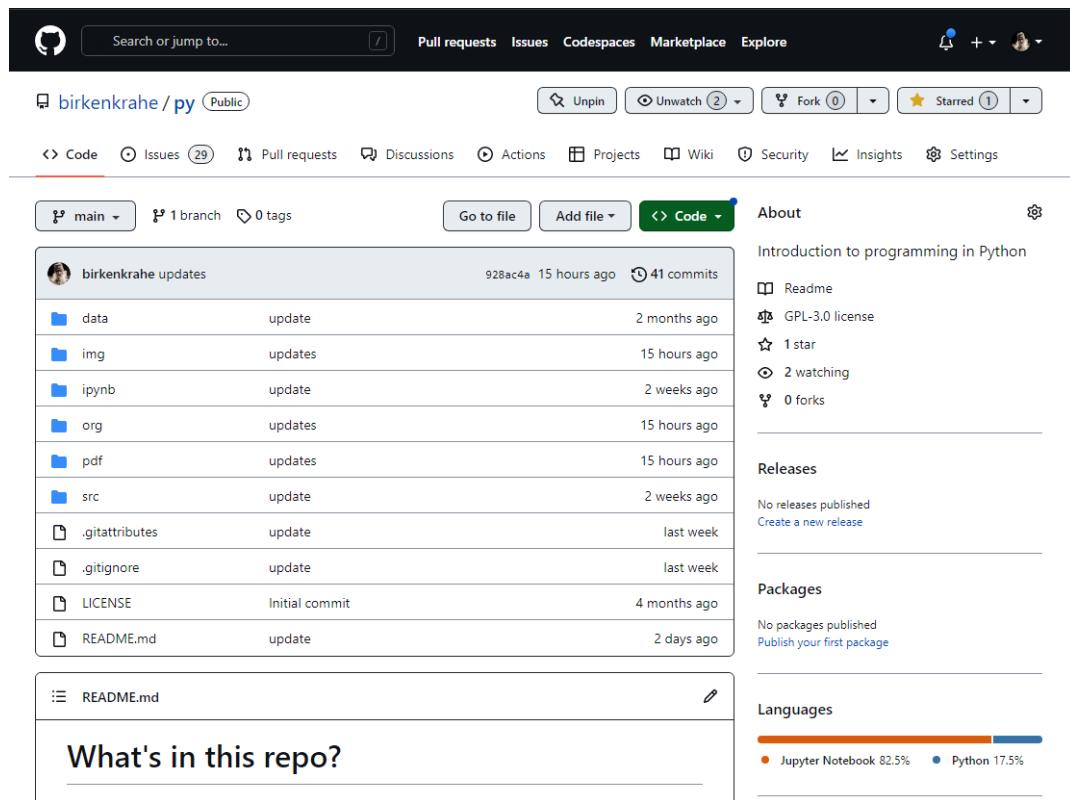


Figure 21: My GitHub dashboard - close to 4,000 commits in 2022 (100,000 lines of code?)

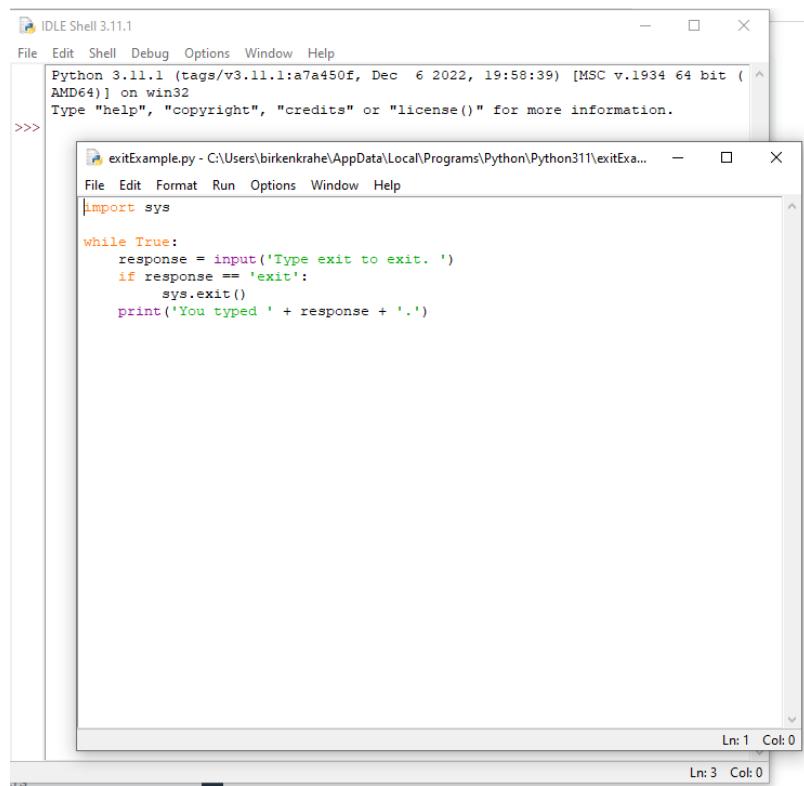


Figure 22: IDLE shell, file editor and turtle demo

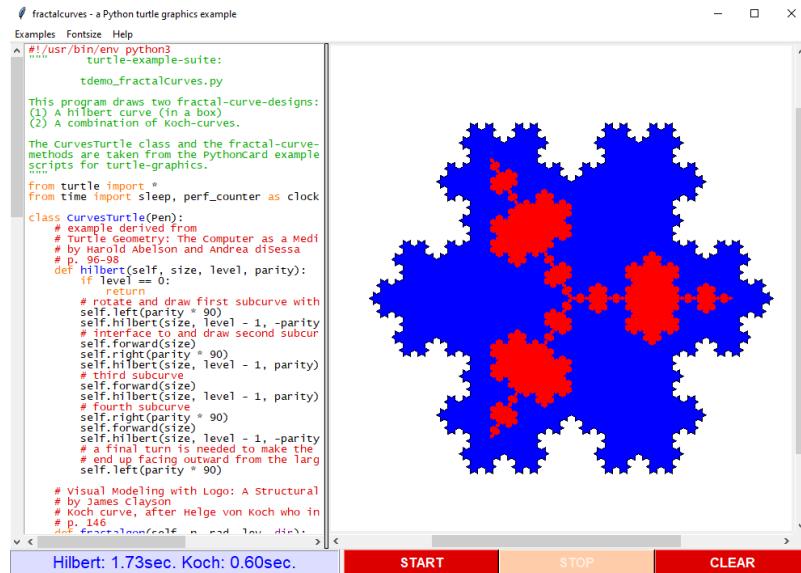


Figure 23: IDLE shell, file editor and turtle demo

The screenshot shows the Google Colab dashboard. The top navigation bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". The main content area starts with a "Welcome to Colab!" message and a video thumbnail. Below this is a section titled "What is Colab?", which explains that Colab is an interactive environment for writing and executing Python code. It lists benefits such as zero configuration required, access to GPUs free of charge, and easy sharing. A note for students, data scientists, and AI researchers follows, along with a link to the introduction. The sidebar on the left contains a "Table of contents" section with links to "Getting started", "Data science", "Machine learning", "More Resources", and "Featured examples".

Figure 24: colab.research.google.com dashboard and overview

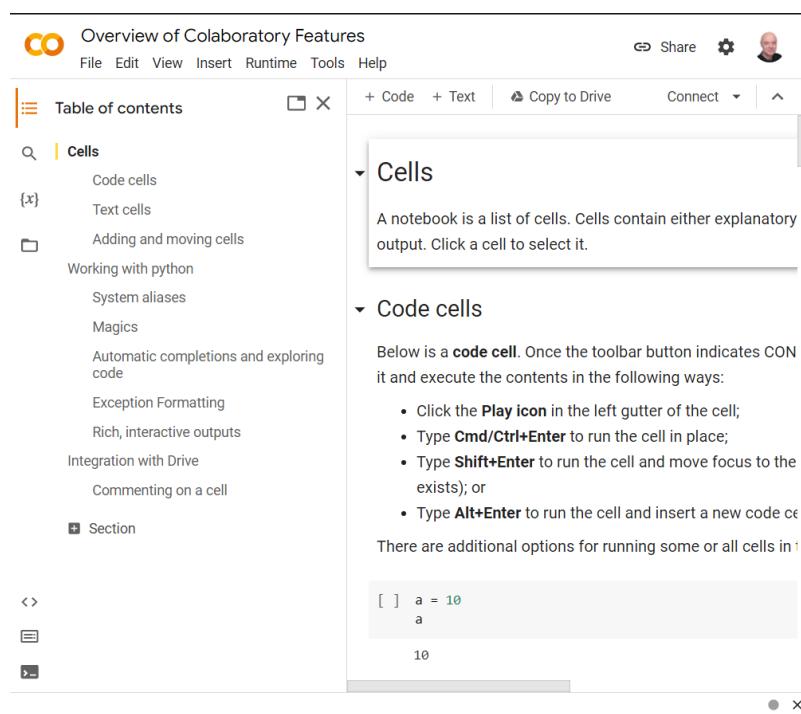


Figure 25: colab.research.google.com dashboard and overview

- Access to shell only with upgrade
- Primary collaborative tool in this course
- Registration required to save your notebooks for submission
- Online alternative to Emacs + Org-mode or VS Code cloud (vscode.dev)

15 repl.it.com REPL + console + templates

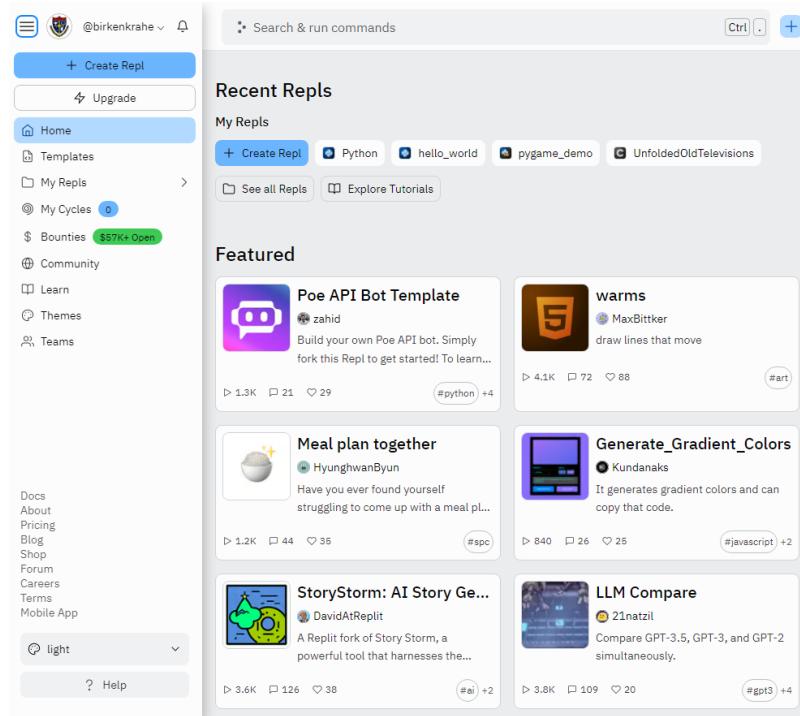


Figure 26: repl.it.com dashboard for Python

- Free REPL (Read-Eval-Print-Loop) + Python console + Shell
- Templates for many programming languages to choose from
- Cool data science and AI project templates
- Easy to share code publicly

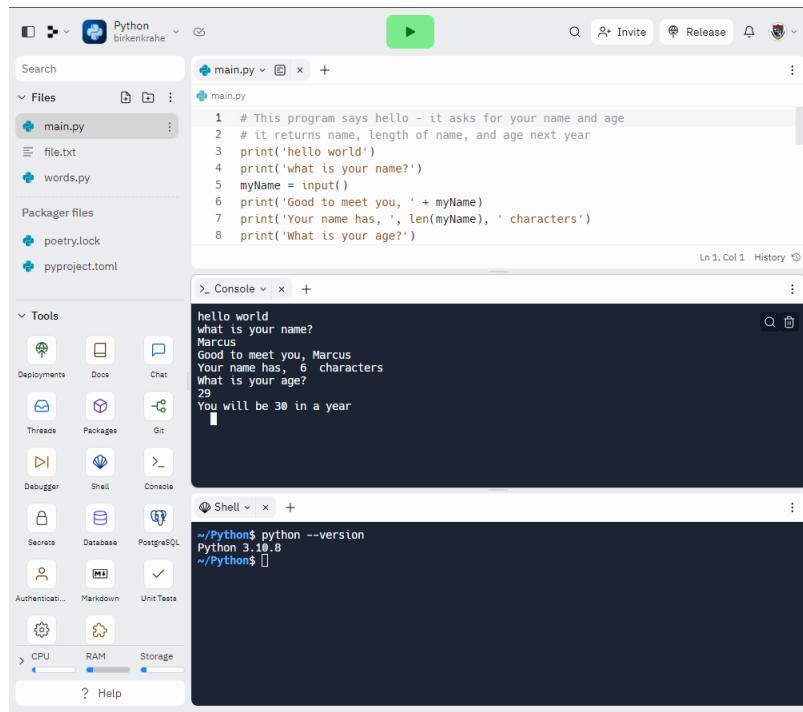


Figure 27: replit.com dashboard for Python

- Alternative to IDLE (though can be slow to respond at times)
- Registration required to create your own REPL

16 DataCamp lessons + projects + workspace

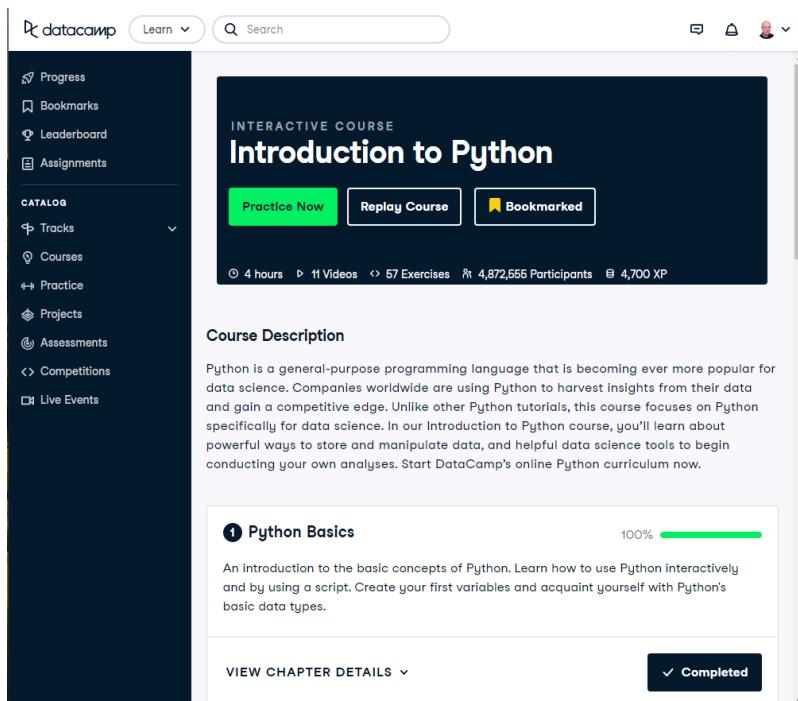


Figure 28: DataCamp dashboard

DataCamp:

- Free online data science learning platform (Python, R, SQL)
- Each course is 3-4 hours long and leads to a certificate
- Many guided and unguided projects (ca 1 hour per project)
- Mobile app and practice exercises for each course
- Jupyter Lab available as part of DataCamp workspaces
- Registration **with Lyon email** required

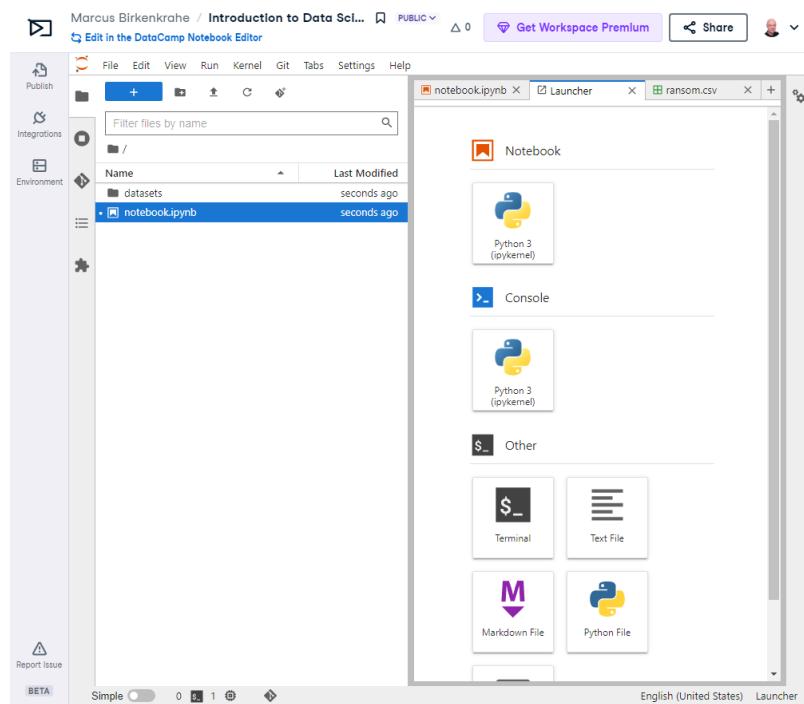


Figure 29: DataCamp dashboard

Alternatives worth checking out (first course is free):

- freeCodeCamp.org, esp. Harvard's CS50 (video and tutorials only)
- dataquest.io (looked at the first lesson - more demanding)
- codeacademy.com (very beginner friendly, programmer-focused)
- w3schools.com (low entry, nice integrated workspace)

17 Can we use AI coding assistants?



Figure 30: Hieronymus Bosch, The Concert in the Egg/Ship of Fools (1561)

- Yes, you may as long as you document it meticulously and diligently
- I frequently use OpenAI ChatGPT, and I tried Google Bard
- GitHub CoPilot only works as an editor extension (haven't tried it)
- Google Colab will soon introduce Google Codey
- AI assistance will take lots of time if you use them well

My typical workflow looks like this:

1. My question: "How can I get π in Python?"
2. ChatGPT spits out explanation with code snippet
3. I check the text and the code
4. I create an exercise or two around the question
5. If I received chatbot help, I reference the software

18 When are the first assignments due?

- The first home assignment, first test and in-class assignments are due a week from now, on Monday, 28 May, by 12 pm (so that I can check 'em).
- All DataCamp assignments are due by Friday, June 23rd (so that I can grade them): Three courses to be completed in your own time.

19 Will you need to buy any books for this course?

Nope. The main textbook that I used is available to read online for free: Al Sweigart, Automate the Boring Stuff with Python (4e) (2020), there's also a course by the author on Udemy (\$12.99 until tomorrow).

You could just read (a) book(s) to learn Python (that's what I do) but you do need to put in the time to reflect, refactor and code anyway.

The Python online documentation recommends both books and courses.

My typical workflow with programming books:

1. I start reading a book chapter (paper copy).

A screenshot of a dark-themed code editor. At the top left is a user icon of a man's face. Next to it is the text "how can I get Pi in python". Below the header is a green icon of a circular swirl. To its right is a block of Python code:

```
python
import math

pi_value = math.pi
print(pi_value)
```

At the top right of the code block is a "Copy code" button. Below the code block is the word "Output:" followed by the output of the code: "3.141592653589793". This output also has a "Copy code" button at the top right.

By importing the `math` module and accessing the `pi` constant, you can assign the value of Pi to the variable `pi_value`. Finally, when you print `pi_value`, you will get the value of Pi (approximately 3.141592653589793).

Figure 31: Hieronymus Bosch, The Concert in the Egg/Ship of Fools (1561)



Figure 32: Four monks by Claudio Rinaldi (1852-1909)

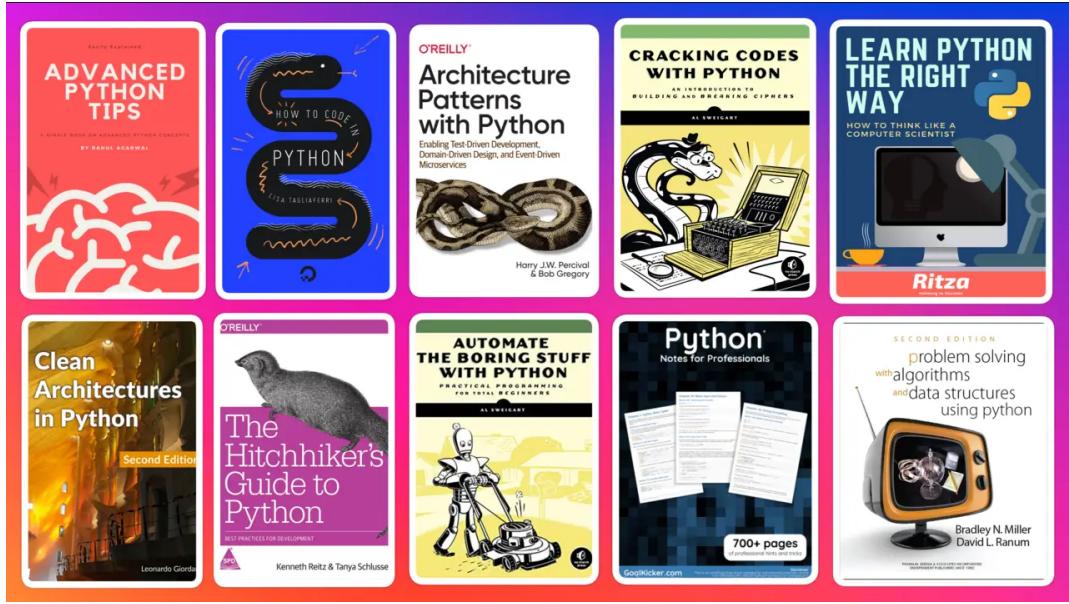


Figure 33: Python book titles

2. I get antsy to try things out and throw the book away.
3. I code on my own until my fingers bleed.
4. I get lost and must return to the book.
5. I vow to stay with the book and open the digital version.
6. `continue` with (2). No `break`.
7. Years later, I find the book in my library.
8. I realize that my copy is completely out of date.
9. I order the latest edition as paper and digital copy.
10. `continue` with (1).

20 TODO Infinite skills

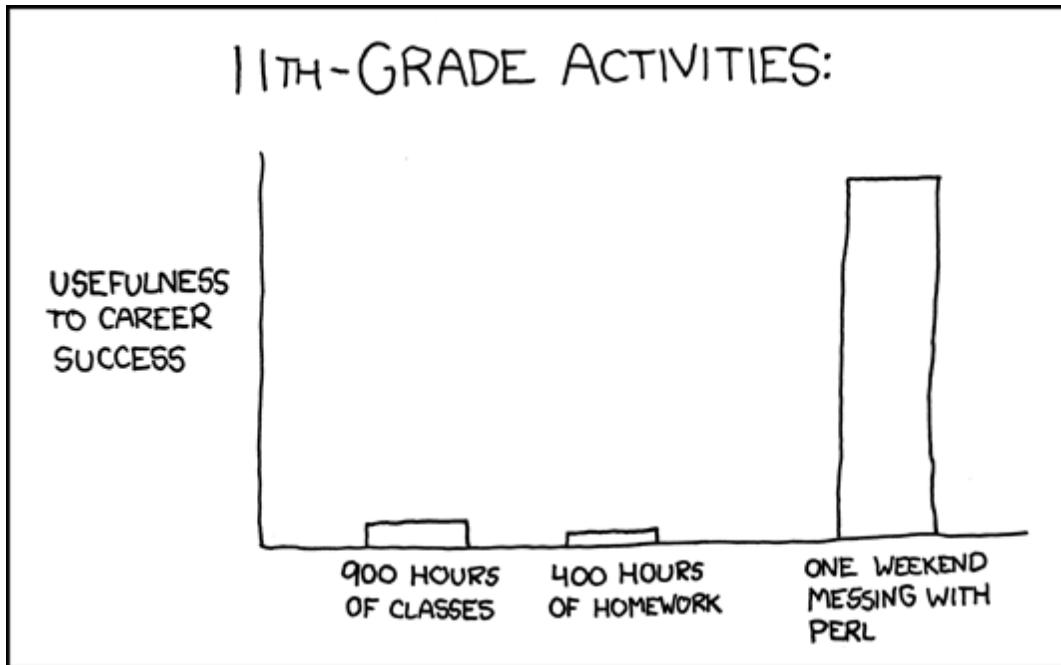


Image: xkcd (Randall Munroe, robotics engineer at NASA)

»... And the ten minutes striking up a conversation with that strange kid in homeroom sometimes matters more than every other part of high school combined.«

- **Infinite Skills:** come up with three programs you would create if you had "infinite programming skills" and if you could build anything you wanted **using a[ny] computer** and the **Python programming language**.
- One example from everyone! Now!

21 TODO Next / Any questions? / Review questions

- Next up: Literate Python scripting infrastructure basics!
- Before we go on:

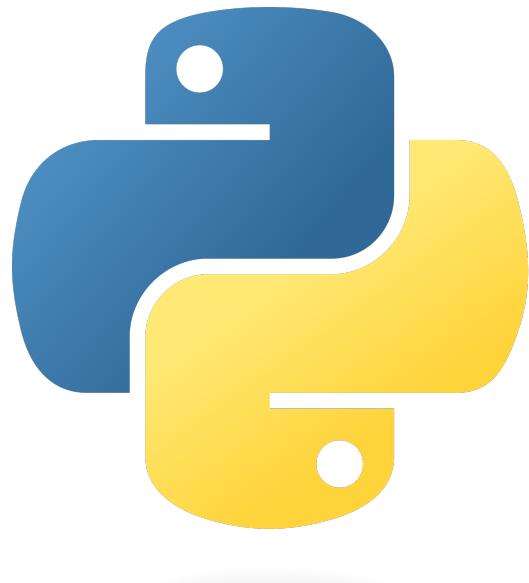


Figure 34: Introduction to literate Python scripting infrastructure



Figure 35: Introduction to literate Python scripting infrastructure

1. What's the difference between "Python", "python3", and "IPython"?
 - "**Python**" is the name for the programming language
 - "**python3**" is the executable for Python's version 3
 - "**IPython**" is a Python shell or console for interactive computing, it's also the basis of Jupyter notebooks (Google Colab) and Jupyter lab (DataCamp workspace).
2. What's the difference between "IDLE", "Colab", "replit" and "DataCamp"?
 - "**IDLE**" (Interactive Development Learning Environment) is an IDE that comes with your Python installation (on Windows and macOS) and has a console + a file editor + a debugger.
 - "**Colab**" (Google Colaboratory) is an online interactive notebook (or "literary") development environment based on Jupyter lab.
 - "**replit**" (replit.com), named after REPL (Read-Eval-Print-Loop) is an online platform to learn many different languages, and it provides a file editor, a console and a shell.
3. Why are there so many **technical terms** (and lingo) around learning Python? I thought it was an easy language suited for beginners?

Python is supposedly easy for beginners but it is also part of a huge, mixed commercial/open/free technological ecosystem, and as a programming language it requires infrastructure (both hardware and software), which changes continuously.
4. A **7-year old** asks you what you're doing while you're working on the **Python shell** to complete an assignment. What would you reply?

22 TODO Practice: Python scripting infrastructure

(PDF and an Org-mode version of this exercise on [GitHub](#))

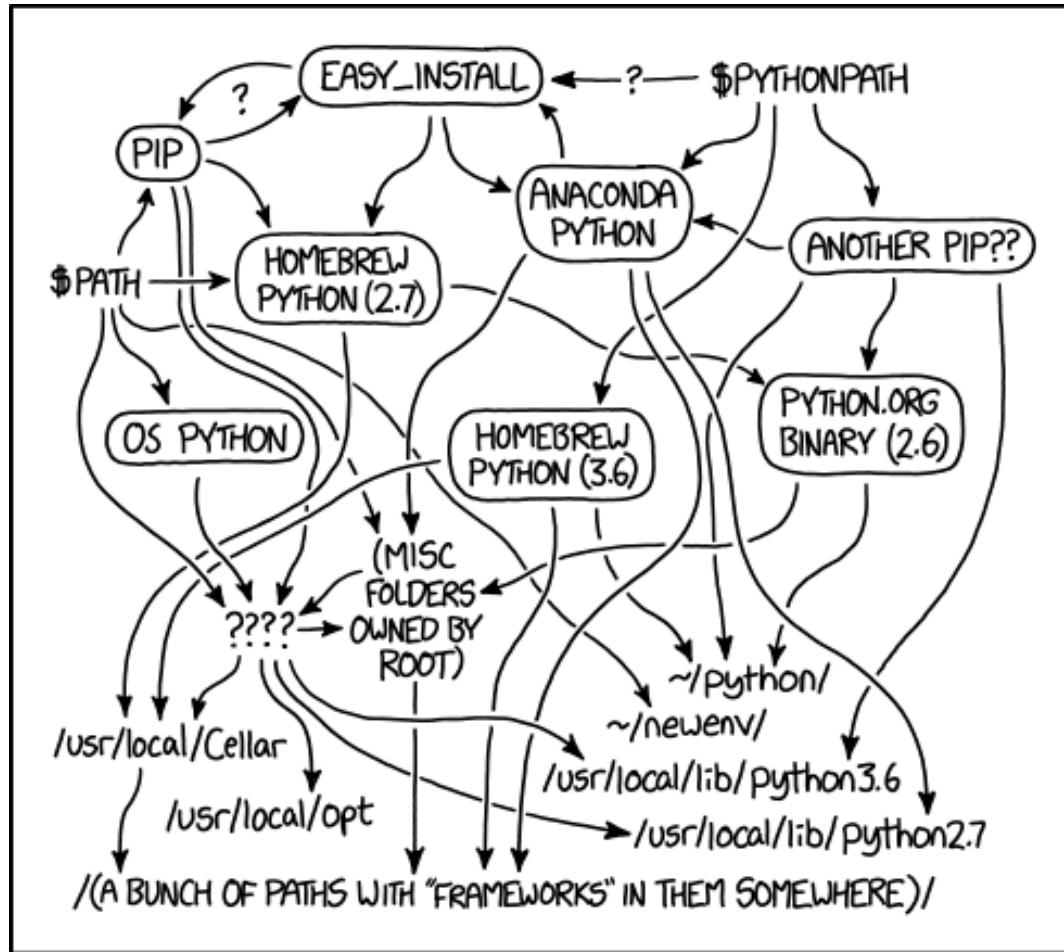


Figure 36: "Python environment" by xkcd - <https://xkcd.com/1987>



Figure 37:

Objectives:

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

23 Glossary

TERM	MEANING
Meta data	Data about data (e.g. layout instructions)
Infrastructure	Computing roadworks: hardware and software
Editor	Program to write programs in
Compiler	Program to turn source into machine code
Python	High-level, procedural, OOP language
Source code	Code for humans to read and edit (<code>.c</code>)
Machine code	Code for machines to execute (<code>.exe</code>)
Git (as in GitHub)	Software version control system (2005)
Emacs	Extensible editor written in Lisp (1985)
FOSS	Free and Open Source Software
IDLE	IDE bundled with Python
Linux	FOSS operating system (1991)
Windows, MacOS	Commercial OS (Microsoft, Apple)
Command line	Terminal, shell program to talk to the OS
Prompt	Location on your computer, e.g. <code>C:\User\</code>
Literate programming	Doc + code + output for humans and machines

24 References

- Neary M (Nov 13, 2019). Getting Started With Python IDLE. URL: realpython.com.
- Real Python (Jan 20, 2022). What is Python IDLE and How to Use the IDLE Shell. URL: [youtube.com](https://www.youtube.com).

- Images (public domain/no known copyright restrictions): Unsplash.com, Flickr.com
- PyCon (2020). Digging MUD in Python by Samuel Regandell. URL: YouTube.com.
- Python 3.11.1. (Oct 24, 2022). Python Software Foundation. URL: python.org.