

Game Programming with Python

Introduction

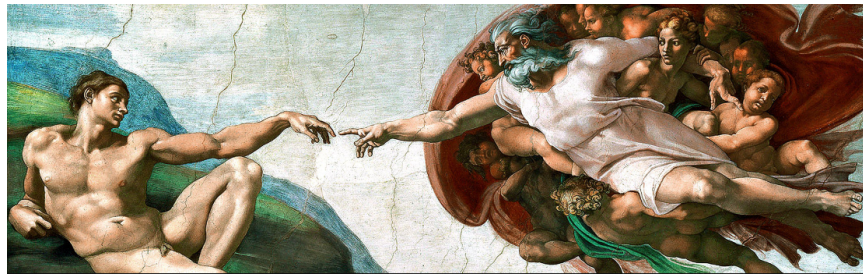
Marcus Birkenkrahe

September 24, 2024

Contents

1	Introduction	1
2	Overview	2
3	Why should you (still) learn how to program?	2
4	Will AI not do all the programming for us?	3
5	Getting started with IPython	3
6	Getting started with IDLE	5
7	Getting started with Google Colaboratory	8
8	Using AI to import and visualize data	9
9	Programming assignment: Colab notebook without AI	9

1 Introduction



- Some of this workshop material is based on Invent Your Own Computer Games with Python (4e) by Al Sweigart (free online).

2 Overview

1. Programming as an art (with or without AI).
2. Programming with IPython, IDLE, and Google Colaboratory.
3. Programming with Python: numbers, operations, variables.
4. Simple programs: "Hello World" and user input/output.
5. Quiz 1.
6. Python loops and conditional statements.
7. Random numbers.
8. Write and understand a simple game.
9. Quiz 2.
10. What do to next.

3 Why should you (still) learn how to program?

- Programming is fun.
- Programming is a useful skill to have.
- Programming trains your brain to think logically.
- Programming gets you used to making and fixing mistakes.

Simple example how to beat frustration by debugging:

```
printf("Hello, class of 2028");
```



Figure 1: Pieter Bruegel the Elder, Hunters in the Snow (1565)

4 Will AI not do all the programming for us?

"So much of what we do as software developers is just magic to people. It's really hard for people outside of software to gauge how complicated some tasks are, or what software is capable and not capable of doing. So AI just seems magical, but I have a really dim and critical view of ChatGPT and a lot of these large language models and various tools. It's not like blockchain or NFTs. There is something there, but there is also just so much hype surrounding these tools. So what I have to tell people is, yes, it absolutely is worth it to learn how to program."

Al Sweigart (Stackoverflow.com podcast, May 24, 2024)

What does this cartoon mean to you?

- CS = Computer Science
- GIS = Geographical Information System (like Google Maps).

5 Getting started with IPython

- IPython is a Python *shell* program - an environment to connect to the Python interpreter. It's not the same as

WHEN A USER TAKES A PHOTO,
THE APP SHOULD CHECK WHETHER
THEY'RE IN A NATIONAL PARK...

SURE, EASY GIS LOOKUP.
GIMME A FEW HOURS.

... AND CHECK WHETHER
THE PHOTO IS OF A BIRD.

I'LL NEED A RESEARCH
TEAM AND FIVE YEARS.



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

Figure 2: <https://xkcd.com/1425/>

- The Python interpreter takes your commands and executes them directly.
- Alternatively, you must download and install the Python interpreter on your personal computer (from python.org).
- If you have Python on your PC, you can open a command line (CMD) and open the interpreter (aka console or shell) with the command `python3`:

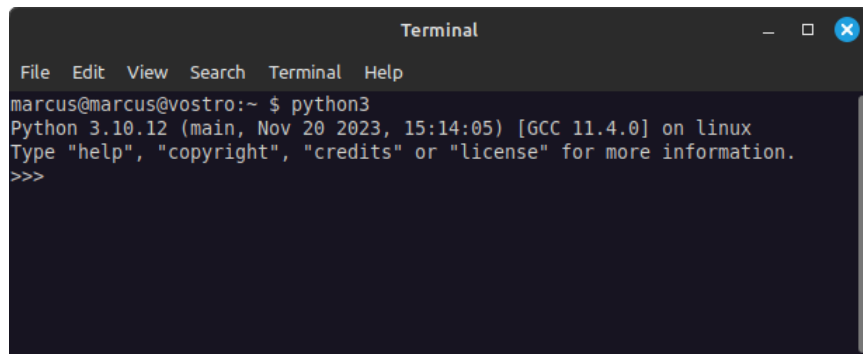
A screenshot of a Linux terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal shows the command `python3` being executed, resulting in the prompt `>>>`. The output text reads: `marcus@marcus@vostro:~ $ python3`, `Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux`, and `Type "help", "copyright", "credits" or "license" for more information.`

Figure 3: Python interpreter on my Linux box at home

- The '3' suggests that there was a 'python2': on many computers, entering `python` will start Python 2 instead of Python 3.
- Confusingly, Python 2 and Python 3 are not compatible.

6 Getting started with IDLE

- When you have Python, you also have another program called IDLE, which can be started from the command line with `idle`.
- IDLE (Interactive DeveLopment Environment) is like a word processor for writing Python programs.
- It's a separate window where you can enter commands and execute them.
- IDLE is a so-called "Integrated Development Environment" (IDE) that allows you to perform different operations during program development on one platform: writing, debugging and executing code.

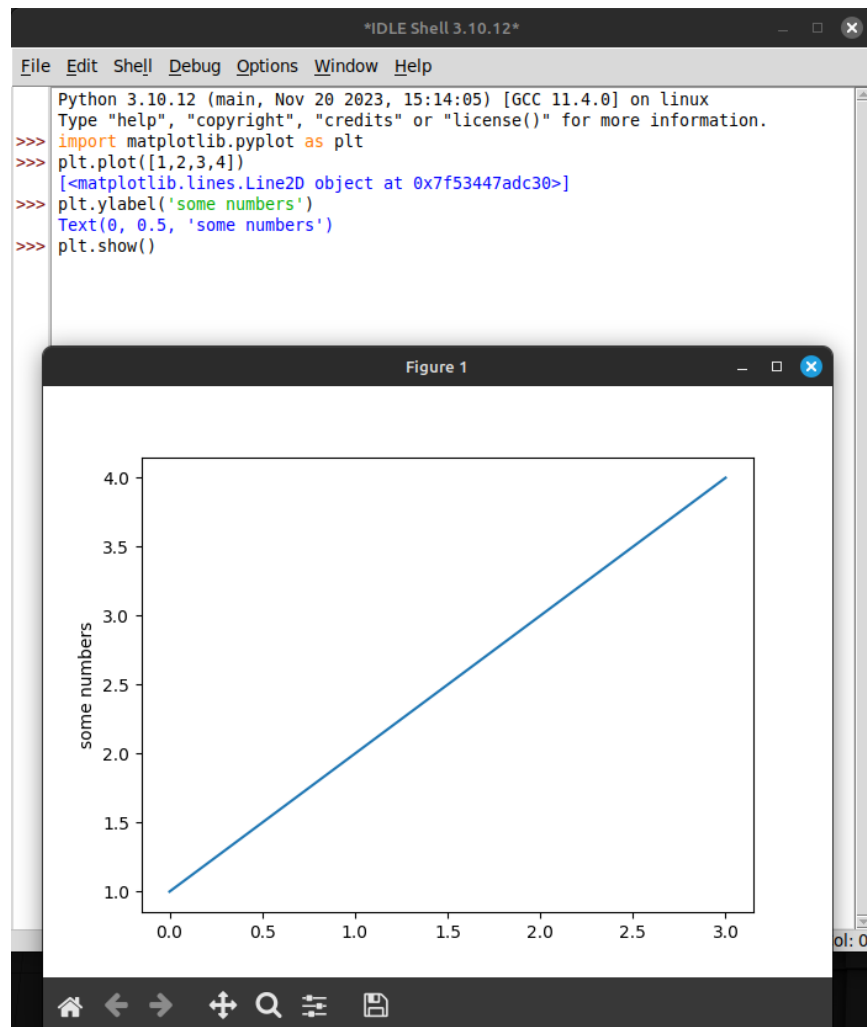


Figure 4: Executing Python code in the IDLE shell

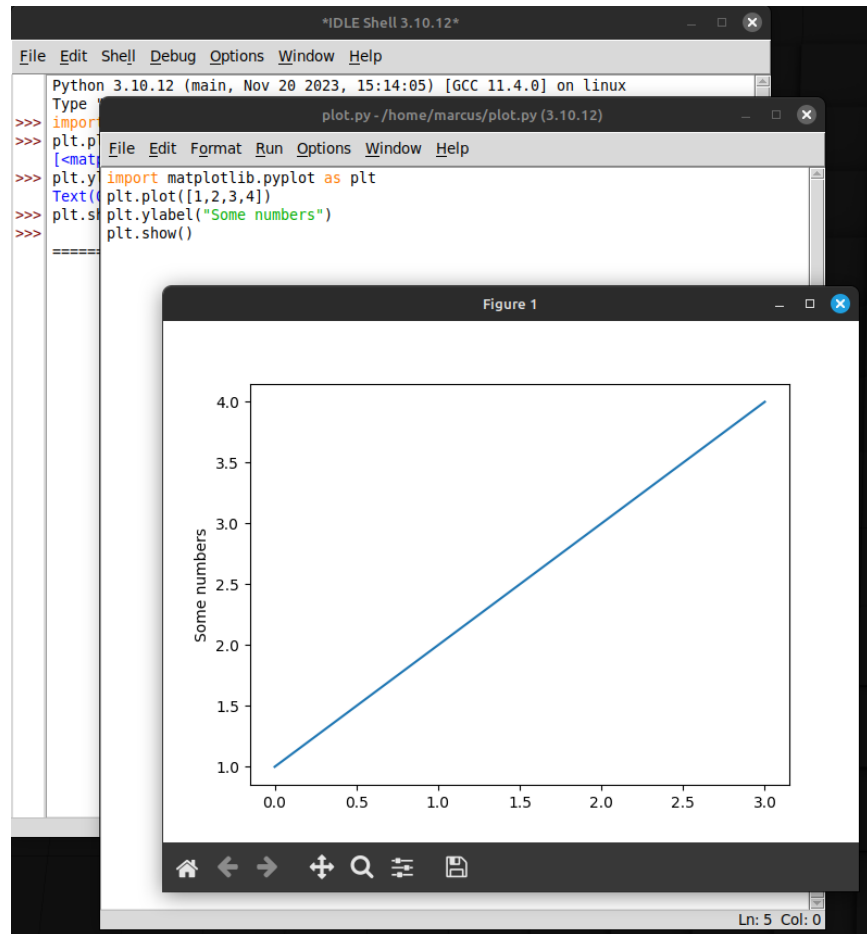


Figure 5: Executing Python code from a Python file inside IDLE

7 Getting started with Google Colaboratory

- Fortunately, Google offers a free IPython application called Colaboratory, which we will use to do all of our coding.
- To open, you must have a Google account. In a browser (any will do but Google Chrome works best), open `colab.research.google.com`.
- In this notebook, you can add text and code, and you can run the code.
- To experiment with that, open a new notebook from the **File** menu and code alongside me.
- The text can be formatted using so-called Markdown language:
 - `#` will create a headline and a section
 - `##` will create a headline and a subsection
 - `' '` will format text as code
 - `![img](URL)` will load an image from the address URL
 - etc.
- `CTRL + ENTER` will execute a code cell, and `SHIFT + ENTER` will execute it and create a new code cell.
- Examples:
 1. Create a text cell with the headline "My first text cell"
 2. Create another text cell with the sub-headline "My first code cell"
 3. Create a code cell
 4. In the code cell, type this code & run it with `SHIFT + ENTER`

```
import matplotlib.pyplot as plt
```
 5. In the next code cell, type this code & run it with `CTRL + ENTER`:

```
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```
- You now have all the ingredients of an interactive data science notebook: text, code, and output.
- Give the notebook a title, e.g. "Colabdemo.ipynb" and save it. This file will now automatically be saved to your Google Drive account.

8 Using AI to import and visualize data

- We'll import the `mtcars` data set (a CSV file on the Internet).
- We'll print the first few lines of the dataset.
- We'll plot the miles-per-gallon vs. weight data.

9 Programming assignment: Colab notebook without AI

- Create a new notebook
- Create text cells
- Create and run code cells
- Add the `matplotlib` graphics library
- Add the `random` random number library
- Generate a "noise" plot of random numbers
- Customize the plot (title, labels)
- The notebook is automatically saved to GDrive