

Python Basics I

Saif Ali

Department of Electrical Engineering
Jamia Millia Islamia

WELCOME TO PYTHON!



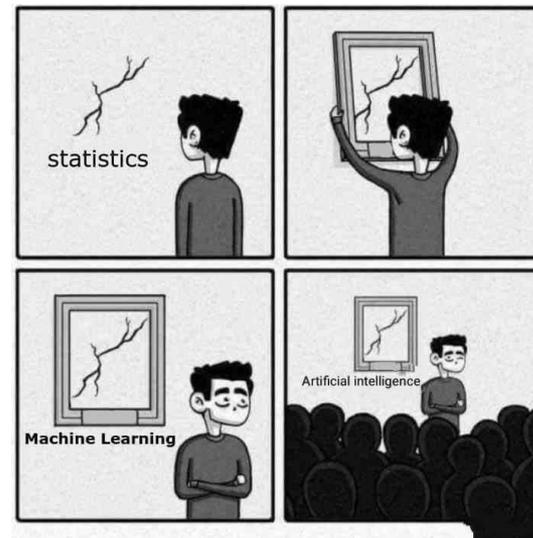
About me

Computer scientist, applied economist

Areas of specialization

Geo-spatial data science, econometrics and statistics,

VR and interactive media.



Source:

<https://towardsdatascience.com/no-machine-learning-is-not-just-glorified-statistics-2643952234e3>



A Fun AI Quiz

1. AI is a sub-discipline of ML. TRUE or FALSE?
2. AI is the science of human behaviour. TRUE or FALSE?
3. Models are representations of reality. TRUE or FALSE?
4. There is fact-based (scientific) knowledge and _____ (ethical) knowledge ?



A Fun AI Quiz

1. AI is a sub-discipline of ML. **FALSE, it's the other way round.**
2. AI is the science of human behaviour. **Debatable. AI <> Neuroscience.**
3. Models are representations of reality. **TRUE. Abstractions of real-world processes.**
4. There is fact-based (scientific) knowledge and **value-based** (ethical) knowledge
?



About the course

- Short-term (~ 50 hours over 18 days)
- Training program on Artificial Intelligence (AI) and Machine Learning (ML)
- Attended by a **diverse** audience (levels, background, interests)
- A combination of offline and online, theory and hands-on learning
- A basic **scientific introduction** to AI and ML with some Python skills
- Goal: use it for your own work/research.
- Technologies: **WebEx** (meetings), **Google Colab** (for Python), **WhatsApp** (general communication)



House Rules

For online attendees:

1. Chatbox will remain open to all at all times for relevant conversation.
2. You can ask a question at any time by typing into the Q/A box.
3. We may take questions in batches or as they arrive.
4. Unmuting may be available depending on what type of session it is.
5. If you need the mic, raise your hand.

For offline attendees:

1. Please ask your questions directly to the instructors.
2. Instructors will repeat the question for the online audience (most of the time).

For all attendees:

1. We may also communicate with you through WhatsApp. Please join the official group.



About You - in 5 Words - Live Poll

Join using link (online):

<https://app.sli.do/event/29FNDvDZ4PKCpzHzS5mByw/embed/polls/38e39668-1d5e-4929-bfe0-5d7319132941>



Join using QR code (offline):



Thinking like a Computer Scientist/Programmer

The single most important skill for a computer scientist is **problem solving**.

Problem solving means

- the ability to clearly articulate the problem
- think creatively about solutions
- the ability to clearly articulate the solution

As it turns out, the process of **learning to program** is an excellent opportunity to practice **problem-solving skills**.



Thinking like a Computer Scientist/Programmer

There are many kinds of problems and many ways of problem-solving

1. Through policy making, governance
2. Through behavioural change (teaching, preaching)
3. Through storytelling (art, cinema, documentary films, journalism)
4. Through qualitative research (surveys, ethnography)
5. Through business and finance
6. Through quantitative research (statistics)
7. Through science and technology (invention, r & d)
8. **Through computers and programming**



Thinking like a Computer Scientist/Programmer

This way of thinking combines some of the best features of **mathematics**, **engineering**, and **natural science**.

Like **engineers**, they **design** software programs, assemble components into systems

Like **mathematicians**, computer scientists use formal languages to **code** their designs into programs computations Hint: think of talking to a friend vs a DL application

Like **scientists**, they **observe** the workings of complex programs, form hypotheses, and **test** predictions.

Quick Chatbox Quiz

Try to recall the last time you “designed” something.
What complex system did you observe recently?



Thinking like a Computer Scientist/Programmer

Natural language: There are two people, one is older than the other by 7 years. If the older person is 14 years old, how old is the younger person?

Formal language:

Let the age of the older person be y and younger person be x .

$$y = 14$$

$$y - x = 7$$

What is the value of x ?

Note

x and y are “variables”. The idea of using variables for problem solving was first introduced by Muḥammad ibn Mūsā al-Khwārizmī



What is a program?

A program is a sequence of instructions that specifies how to perform a computation.

- Mathematical (with numbers)
- Symbolic (with text)

A few basic instructions appear in just about every language:

input

Get data from the keyboard, a file, or some other device.

output

Display data on the screen or send data to a file or other device.

math

Perform basic mathematical operations like addition and multiplication.

conditional execution

Check for certain conditions and execute the appropriate sequence of statements.

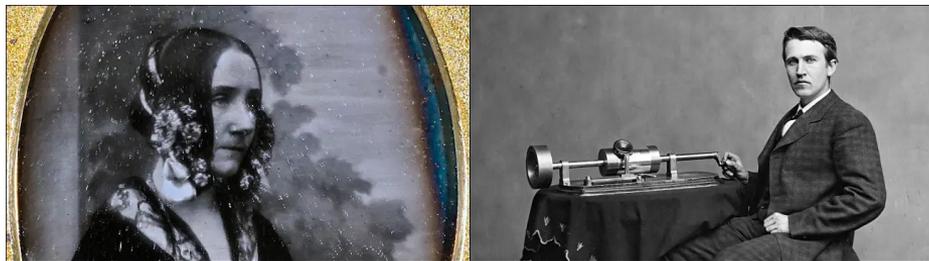
repetition

Perform some action repeatedly, usually with some variation.



What is debugging?

“The key to success is *failure*.”



Programming **errors** are called **bugs** and the process of tracking them down and correcting them is called debugging.

Quick Chatbox Quiz

Share a short incident in the chatbox where you learned from failure and used the learning to achieve success.

Ada Lovelace

1815-1852 British mathematician, considered the first computer programmer

Thomas Edison

American inventor and businessman



The Python Programming Language

Is it a **formal** language or natural language?

It is a **high-level** language, not a low-level language.

It is an **interpreter-based** language, not a compilation-based language.

- Immediate mode
- Scripted mode

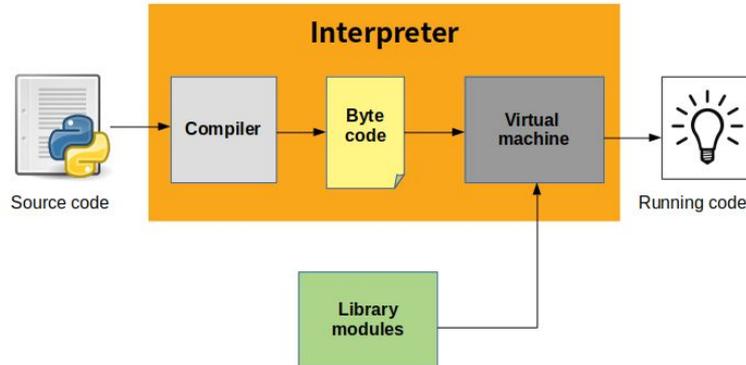
It is a **portable** language.



The Python Programming Language

In order to run Python programs, you need a **programming environment**

- a text editor to **write** your code
- a Python interpreter to **interpret and run** your code
- a command line interface to invoke the Python interpreter



Source: <https://i.imgur.com/PJME67T.png>



The Python Programming Language

Steps to run a Python program

- install a Python interpreter (one time only)
- use a text editor to **write** your code (save it as **.py** file)
- from a command line, invoke the Python interpreter to interpret and run your code

The image illustrates the process of running a Python program through three sequential steps:

- Installation:** The first screenshot shows the "Python 3.7.2 (64-bit) Setup" window. It offers two options: "Install Now" (which includes IDLE, pip, and documentation) and "Customize installation" (which allows choosing location and features). The "Add Python 3.7 to PATH" checkbox is checked.
- Code Writing:** The second screenshot shows a text editor window titled "test.py" containing the following Python code:


```

1 my_file = open("C:/Documents/Python/test.txt", mode="w")
2 print("What is the file name?", my_file.name)
3 print("What is the mode of the file?", my_file.mode)
4 print("What is the encoding format?", my_file.encoding)
5
6 text = ["Hello Python!", "Good Morning!", "Good Bye"]
7 my_file.writelines(text)
8 print("Size of the file is:", my_file.tell())
9 print("Current position is at byte:", my_file.tell())
10 my_file.seek(0)
11 print("Content of the file is:", my_file.read())
12 my_file.close()
13
14 file = open("C:/Documents/Python/test.txt", mode="r")
15 line_number = 2
16 current_line = 1
17 data = ""
18
19 for line in file:
20     data = line
21     print("Data present at current line is:", data)
22     break
23     current_line = current_line + 1
24
25 bin_file = open("C:/Documents/Python/bin_file.exe", mode="wb+")
26 message_content = data.encode("utf-32")
27 bin_file.write(message_content)
28 bin_file.seek(0)
29 bdata = bin_file.read()
30 print("Binary Data is:", bdata)
31 ndata = bdata.decode("utf-32")
32 print("Normal Data is:", ndata)
33 file.close()
34 bin_file.close()
      
```
- Execution:** The third screenshot shows a command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The user has entered the command `python test.py` at the prompt, which has been executed successfully.



The Python Programming Language

Installing Python

- <https://www.python.org/> is the official Python website
- you can download the latest version of Python (3.11.4) from there
- make sure you note:
 - the Python version
 - the platform on which you are programming (Windows, Linux, macOS)
 - the place where you install your Python interpreter (home directory)



Summary

1. Computer scientists and programmers solve complex problems by designing solutions and then coding those solutions in a programming language that has a formal structure.
2. Python is a high-level, interpreted language that is very popular for AI, ML and DL.
3. Naturally, Python also has formal rules
4. We need a Python programming environment before we can start coding
5. You can make your own environment on your system
6. You can use an integrated system like Jupyter Notebook or Google Colab
7. We will use Google Colab in this course.



Google Colaboratory

The screenshot shows the Google Colaboratory interface for a notebook titled 'Untitled2.ipynb'. The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with '+ Code' and '+ Text' buttons, a code editor with a code cell containing `print("Hello World!")`, and an output area showing 'Hello World!'. Numbered callouts (1-11) point to the following elements:

- 1: File name 'Untitled2.ipynb'
- 2: '+ Code' button
- 3: '+ Text' button
- 4: Menu bar (File, Edit, View, Insert, Runtime, Tools, Help)
- 5: 'All changes saved' status
- 6: Code editor area
- 7: Run button (play icon)
- 8: Clear output button (trash icon)
- 9: Output area showing 'Hello World!'
- 10: Output area showing 'Hello World!'
- 11: Clear output button (trash icon)

Try it yourself
 Open the
[“SA1-PR-Practice-Notebook”](#)
 Python notebook in Google
 Colab.

1. **Files:** Here you will be able to upload datasets and other files from both your computer and Google Drive
2. **Code Snippets:** Here you will be able to find prewritten snippets of code for different functionalities like adding new libraries or referencing one cell from another.
3. **Run Cell:** This is the run button. Clicking this will run any code that is inserted in the cell beside it. You can use the shortcut shift+enter to run the current cell and exit to a new one.
4. **Table of Contents:** Here you will be able to create and traverse different sections inside of your notebook. Sections allow you to organize your code and improve readability.
5. **Menu Bar:** Like in any other application, this menu bar can be used to manipulate the entire file or add new files. Look over the different tabs and familiarize yourself with the different options. In particular, make sure you know how to upload or open a notebook and download the notebook (all of these options are under “File”).
6. **File Name:** This is the name of your file. You can click on it to change the name. Do not edit the extension (.ipynb) while editing the file name as this might make your file unopenable.
7. **Insert Code Cell:** This button will add a code cell below the cell you currently have selected.
8. **Insert Text Cell:** This button will add a text cell below the cell you currently have selected.
9. **Cell:** This is the cell. This is where you can write your code or add text depending on the type of cell it is.
10. **Output:** This is the output of your code, including any errors, will be shown.
11. **Clear Output:** This button will remove the output.



Python Textbook

How to Think Like a Computer Scientist



Learning with Python 3 (RLE)

Version date: October 2012

by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers

(based on 2nd edition by Jeffrey Elkner, Allen B. Downey, and Chris Meyers)

Corresponding author: p.wentworth@ru.ac.za

Open Source Textbook

Book Link

<http://openbookproject.net/thinkcs/python/english3e/index.html>



Self-study and reading for tomorrow

For total beginners (non-CSE)

1. “Code: The Hidden Language of Computer Hardware and Software” by Charles Petzold”, Read [Chapter 10](#).
2. Python from the Very Beginning by John Whitington. Read [Chapter 1](#).

For intermediate and advanced programmers

1. Exception handling in Python: <https://docs.python.org/3/tutorial/errors.html>

For everyone

1. [Must watch] [Introduction to Colab](#)
2. [Should read] [Data Science vs. Artificial Intelligence & Machine Learning: What's the Difference?](#)



<https://shorturl.at/Kq64I>

