MODULE 0.1

Welcome and Course Introduction

TECHNION
Azrieli School of Continuing Studies of the Technion

Program

- Getting to know each other
- What is Python?
  - History
  - Compiled languages versus interpreted languages
  - PIP
- Jupyter notebook
- Course expectations
- Creating your preferred setup

Learning Objectives

- You will be able **to describe** the goals and expectations of the course
- You will be able **to describe** the main characteristics of Python, including its history, its interpreter and its package installer (*pip*).
- You will be able **to write** your first Python program to the interactive interpreter, the zsh shell and to Jupyter Notebook
- You will be able **to choose** your own preferred set of tools to write your Python scripts

Welcome and Introduction

- Give a short introduction of yourself?
- Any prior experience with Python?
- What do you expect / hope to get out of this course?

Course Expectations

Course Roadmap

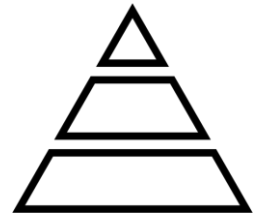
- Building blocks of coding
- Variables and data types
- Branching structures
- Loops
- Lists and tuples
- Dictionaries and data structures
- Web scraping
- Web automation
- Encryption
- Sockets
- Scapy

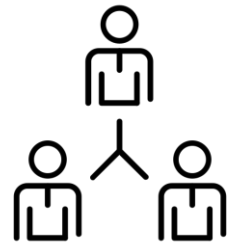
Expectations

- Short introductions
- A lot of teamwork (!!)
- Projects
  - Small coding challenges
  - Large projects
  - Scrums
- Code reviews and presentations / show case
- Own initiative

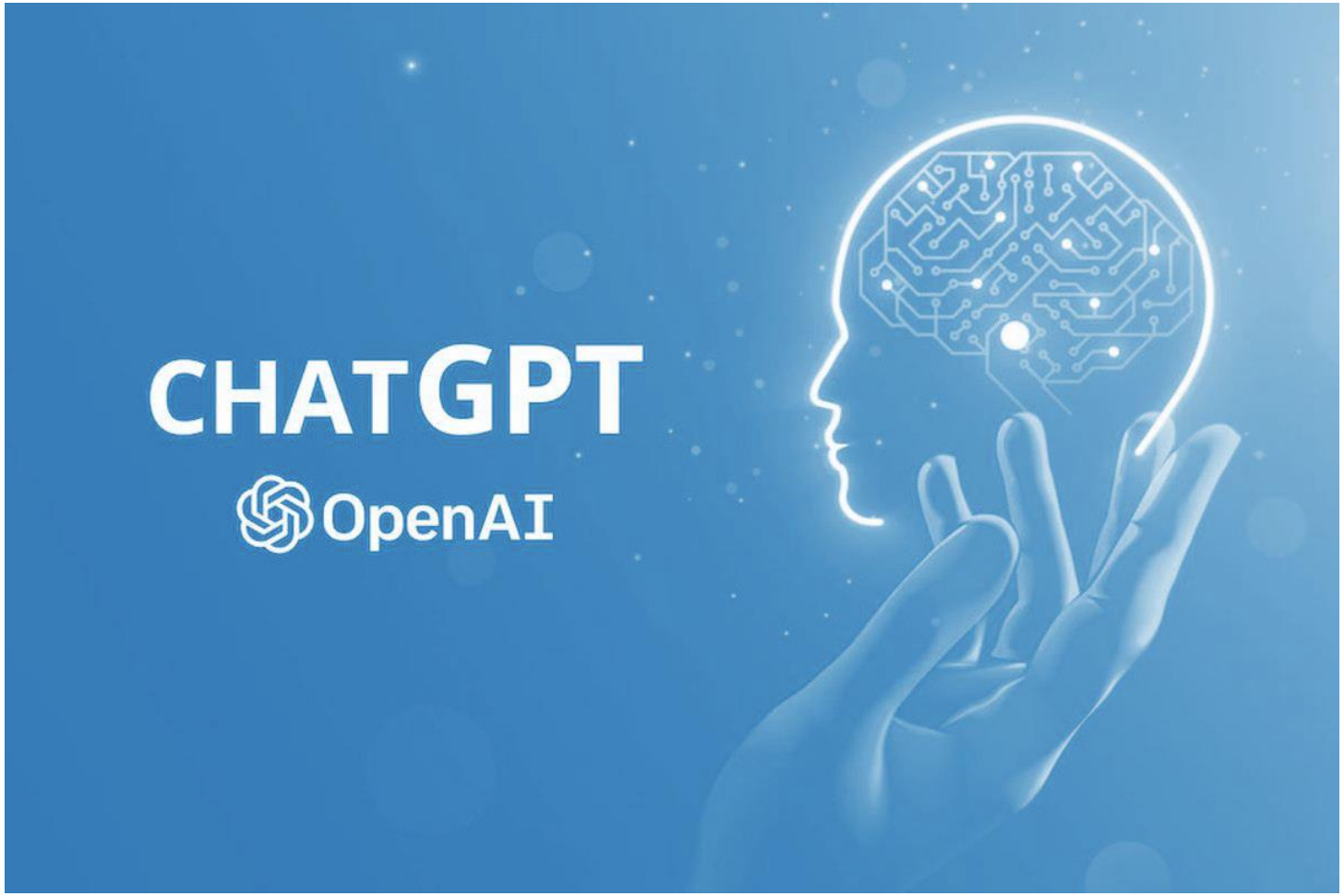
Course Expectations

- The lessons will build upon each other. Make sure you **don’t miss important parts** and do all the labs in time.
- **Ask questions** if you are not sure about something, during or after class while practicing with the materials
- We work as a team, and I want to see **leadership and cooperation.** If you are finished with a lab or assignment, help your fellow students to make sure no one falls behind.





ChatGPT



What is Python?

What is Python?

- Python is a high-level programming language (HLPL), that is widely used in various applications such as web development, data science, artificial intelligence, and many others. Its characteristics:
  - High-level, interpreted language
  - Portable (can be run on multiple platforms, including Windows, Linux, and macOS) o Easy to learn and read
  - Large standard library and a staggering amount of external libraries
  - Used a lot in the field of cyber security - A lot of tools, exploits and POCs are written in python

High-Level Programming Language

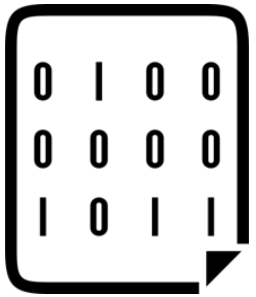
- High-level programming languages are a lot closer to the logic of human communication.
  - Easy to use
  - More flexible

High-Level:

```
x = 17 + 5
>>> 22
```

Assembly:

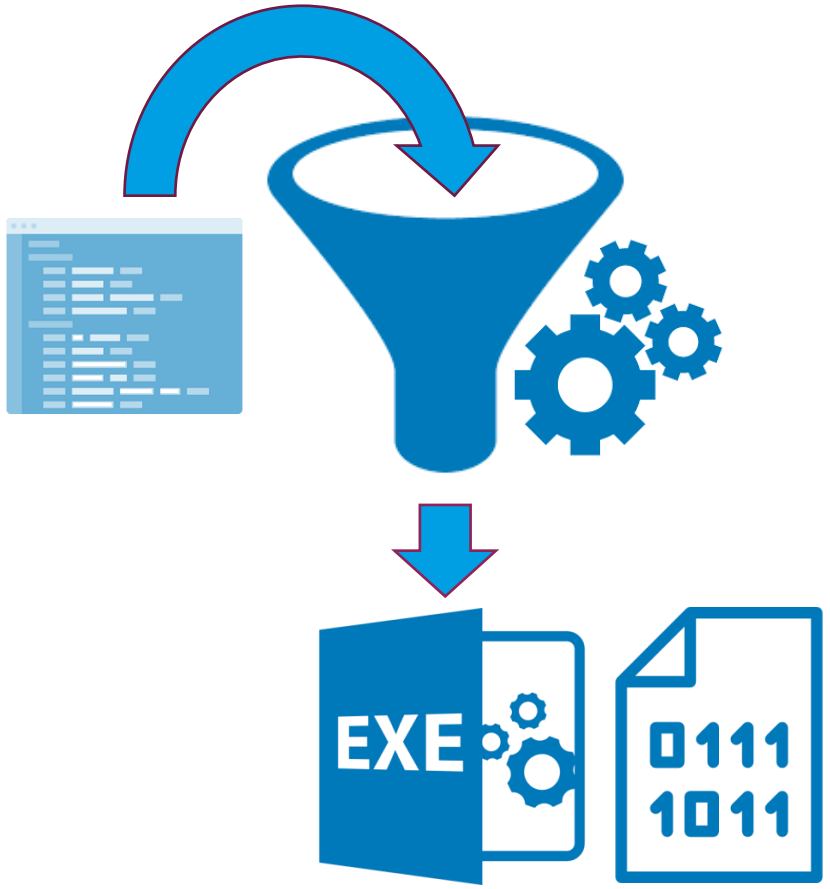
```
MOV ax, 5
MOV bx, 17
ADD ax, bx
```

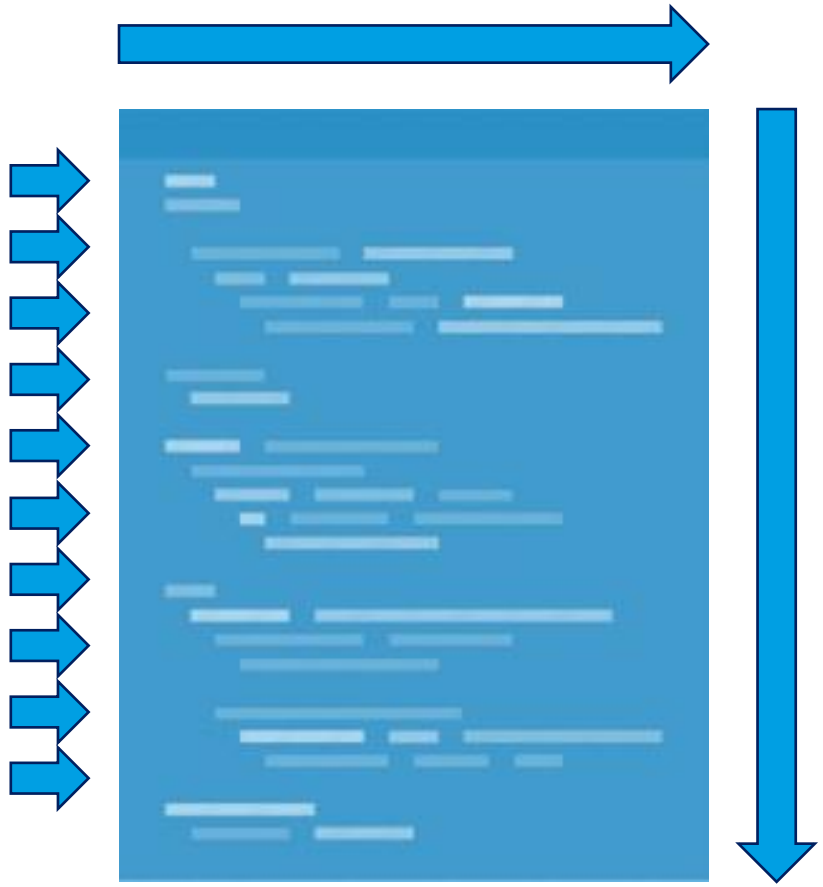


Compiled and Interpreted Languages

- In compiled languages, the whole script (code file) is translated from readable language to machine code.
  - This machine code outputs a compiled file that is ready for execution. (for example: .exe)
  - You can run this compiled file repeatedly
- An interpreted language does not need to be compiled before execution.
  - The source code is processed and executed line-by-line by an interpreter, which reads the code and translates it into machine code on the fly.
  - Easier to learn and use, but may result in slower execution times since each line of code must be interpreted at runtime.

Compiler versus Interpreter





A Short History Of Python

- Python was created in the late 1980s by Guido van Rossum, a Dutch programmer. He named he language after his favorite comedy group, Monty
- Python and released the first version of Python (version 0.9.0) in February 1991.
- The language became quickly popular amongst developers and the academic community, due to its ease of use and open-source license.
- Python 2.0 was first released in 2000. Its latest version, 2.7, was released in 2010.
- Python 3.0 was released in 2008. Its latest version, 3.10, was released in 2021.
- Since January 1, 2020, Python 2.7 has "retired" and no longer be maintained.

The Python Interpreter

- The Python interpreter for python 3 is simply called python3
- The Python interactive interpreter is called the Python shell or simply the Python REPL (Read-Eval-Print Loop).
  - It allows users to enter and execute Python code interactively, line by line, and receive immediate feedback on the results.
  - A powerful tool for testing code snippets, learning and experimenting with new features, and debugging code

PIP Installs Packages

- Pip, or “Pip Installs Packages”, is the package management system for Python.
- It is used to install, upgrade, and manage Python packages and their dependencies.
- It is a command-line tool that allows you to easily download and install Python packages from the Python Package Index (PyPI) and other repositories.
- Usage: pip install <package-name>

Jupyter Notebook

Jupyter Notebook

- Allows users to create and share documents that contain live code, output, visualizations, and narrative text.
- It supports a variety of programming languages, including Python, R, Julia, and many others.
- You can write and execute code in small chunks called cells, which can be run individually or as part of a larger sequence.
- You can document your code and your output in a single document that can be easily shared and replicated by others.
- Useful for educational purposes

Install Jupyter Notebook

- Install jupyter notebooks with pip:

```
pip install notebook
```

- Add jupyter to the PATH variable for zsh:

```
echo 'export PATH=$PATH:/home/kali/.local/bin' >> .zshrc
```

- Run Jupyter notebooks:

```
Jupyter-notebook
```

Building Your Working Environment

Building Your Working Environment

- Class:
  - Operating system: Kali-Linux 2023.2
  - Jupyter notebooks
- Team:
  - Choose your text editor
  - Discuss other useful tools
  - Arrange knowledge sharing (cloud, Git, etc…)
  - Discuss roles and responsibilities

Text Editors For Linux

1. **Visual Studio Code**: has syntax highlighting, code completion, debugging, and Git integration.
2. **PyCharm**: Python IDE with a lot of features, including intelligent code completion, debugging, refactoring, and Git integration.
3. **Sublime Text**: A popular code editor with support for many programming languages, including Python. It has a powerful plugin system and a clean, customizable interface.
4. **Atom**: Another free and open-source code editor with a large community of users and a wide range of plugins and themes. It has built-in support for Git and GitHub, as well as syntax highlighting and code completion.
5. **Emacs and Vim** –both haveasteep learning curve, but they are highly customizable andextensible.

LAB 0.1 – Hello, World!

- Download `Lab 0.1 –Hello world` into to your coursefolder
- Open Jupyter notebooks from your course folder and display the lab.
- Read the lab carefully
- Prepare a team presentation of your working environment

Presentations

- Present your working environment and your team to the class
  - Introduce your team…
  - What tools do you use?
  - How does the “Hello, world!” program look in your IDE?
  - Go over some of the features of your IDE…
  - How do you organize knowledge sharing as a team?
- (Share your screen via Teams)

Learning Objectives

- You will be able **to describe** the goals and expectations of the course
- You will be able **to describe** the main characteristics of Python, including its history, its interpreter and its package installer (*pip*).
- You will be able **to write** your first Python program to the interactive interpreter, the zsh shell and to Jupyter Notebook
- You will be able **to choose** your own preferred set of tools to write your Python scripts

Thank you

TECHNION
Azrieli School of Continuing Studies of the Technion