Web Application Development using Python

Course Introduction

Prepared by George Khoury

Outline

- **About your instructor**
- **Refresh your knowledge**
- **Course structure**
  - Learning Tree
  - Roadmap
- **A little bit about Python**
  - What is Python?
  - Why Python?
- **Compiler vs. Interpreter**
- **Setup your environment**

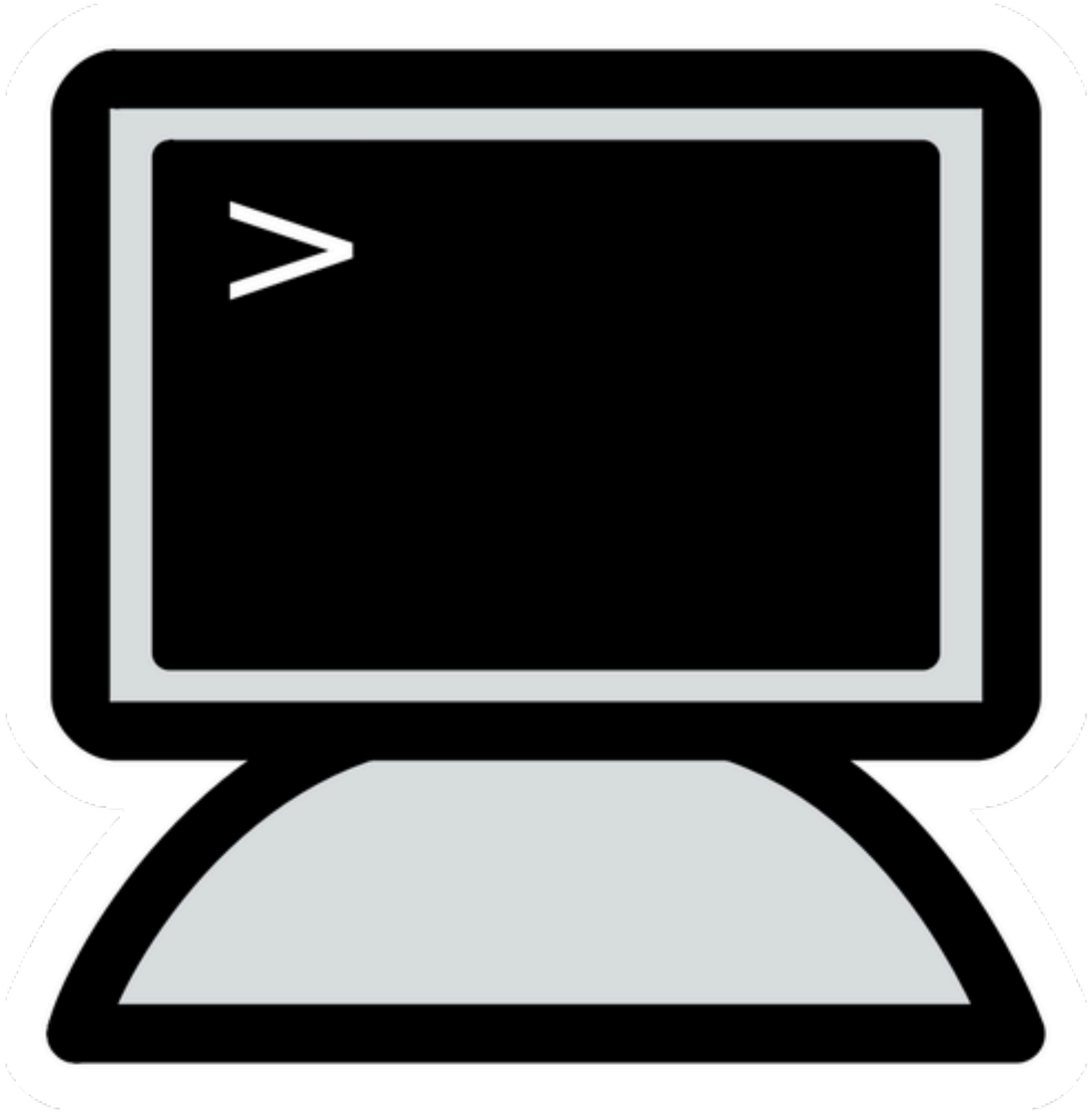
About Your Instructor

- B.Sc. in Computer Science from the German Jordanian University.
- Lead Technologist at the Jordan Open Source Association.
- Technical Manager at Quttous, eCommerce division.
- Build Engineer at Salalem Learning.
- Software Engineer at SAP.

Refresh your knowledge

Prerequisites for this course

1. Feel comfortable with the **terminal**!
2. Familiarise yourself with the basics of **object-oriented programming**.
3. Refresh your knowledge in **software engineering principles**.
4. Read about **web application development**.



Refresh your knowledge

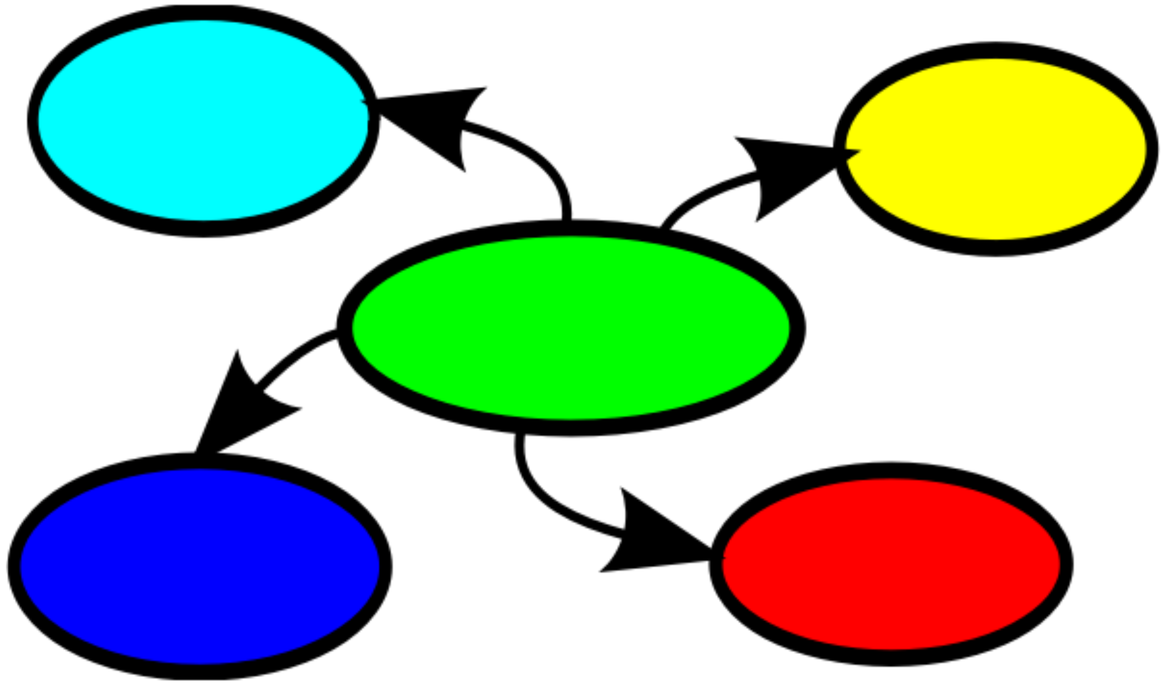
Learning Resources

1. https://github.com/geokhoury/htu-devops-labs
2. https://www.edx.org/course/introduction-to-linux
3. https://www.tutorialspoint.com/What-is-object-oriented-programming-OOP
4. https://12factor.net/
5. https://www.tutorialspoint.com/software_engineering/index.htm
6. https://www.imad.tech/

Course Structure

- Lecture Notes
- Class Assignments (**CA**)
- Practical Assignments (**PA**)
- Capstone Project

Learning Tree



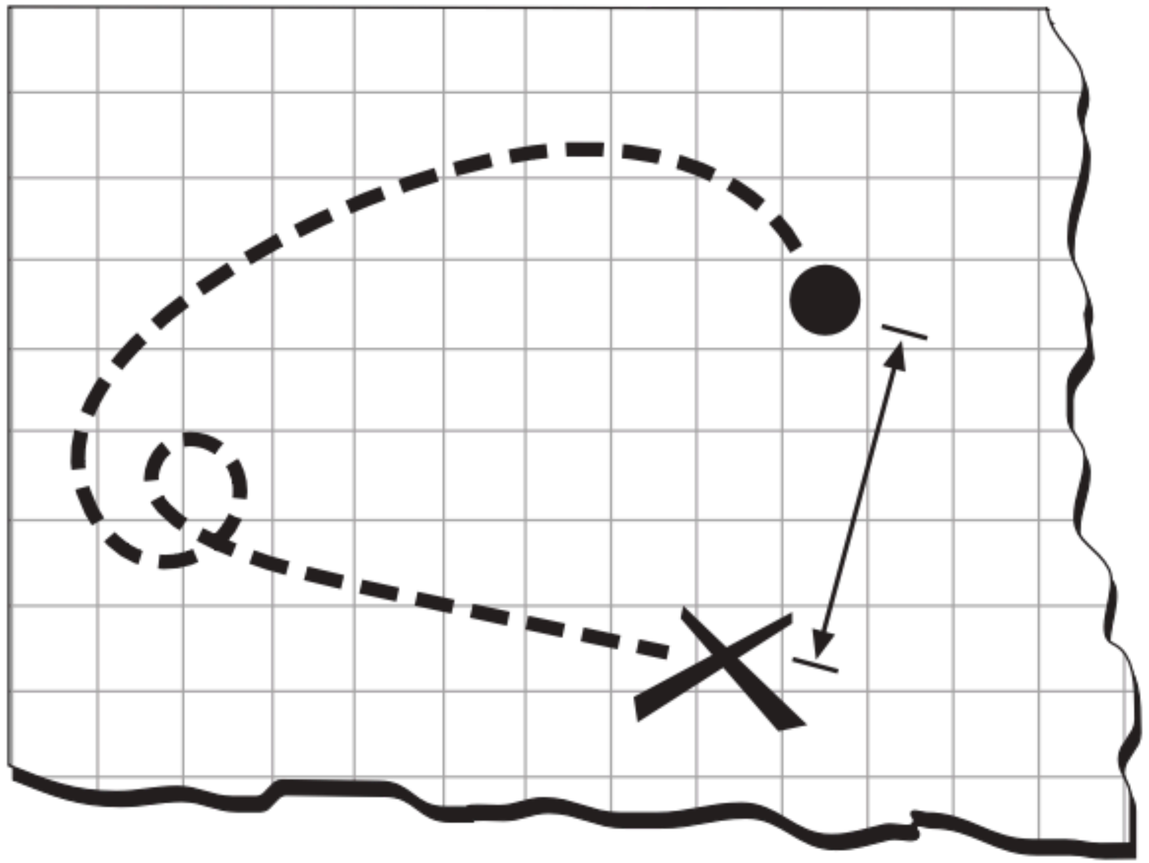
Course Roadmap

What will you learn?

- **Introduction to Python - Part 1**
  - Basic data types and structures (int, float, string, list,)
  - Basic input output (print(), raw_input(), etc)
  - Basic flow control
  - Functions

- **Introduction to Python - Part 2**

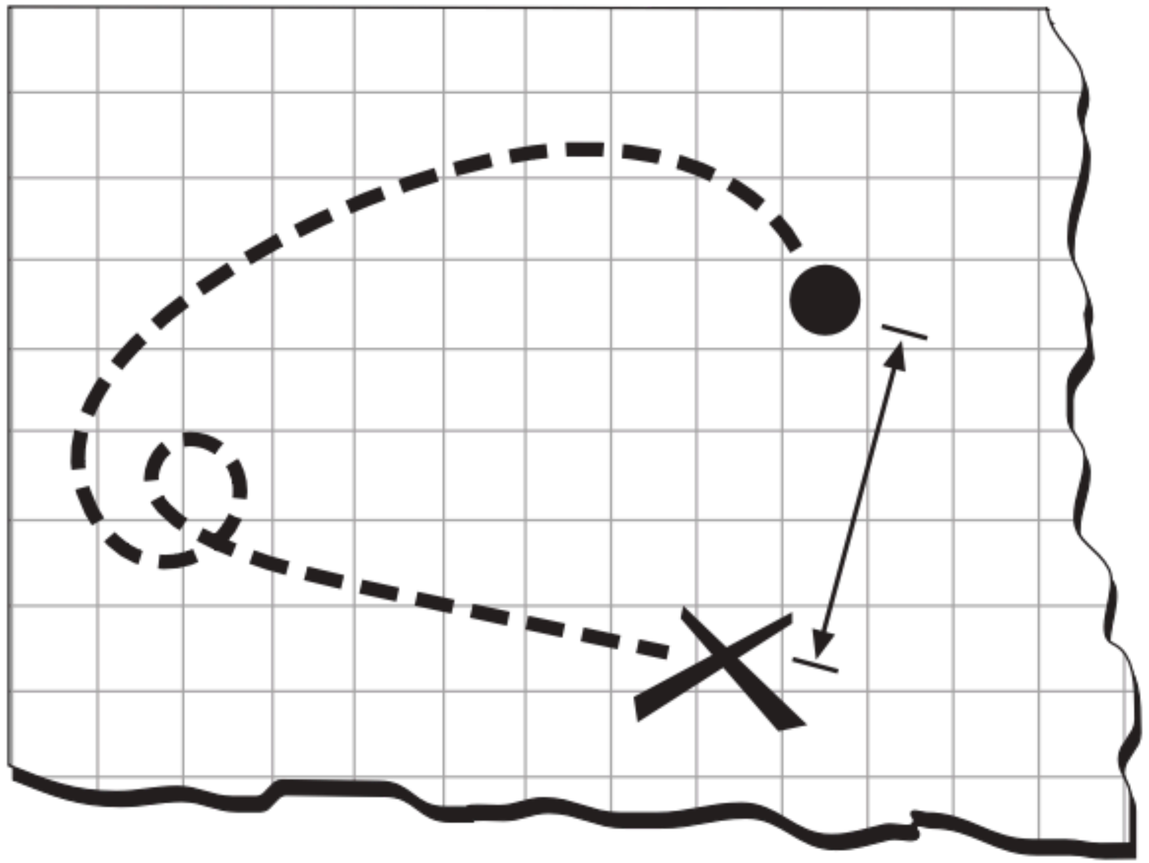
- **Version Control Systems**



Course Roadmap

What will you learn?

- Web application development using Flask
- Web application development patterns
- REST APIs using Flask



What is Python?

- **Python** is an *interpreted*, *object-oriented*, *high-level* programming language with dynamic semantics.
- Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it:
  - Very attractive for Rapid Application Development.
  - Good choice for scripting or glue language to connect existing components together.

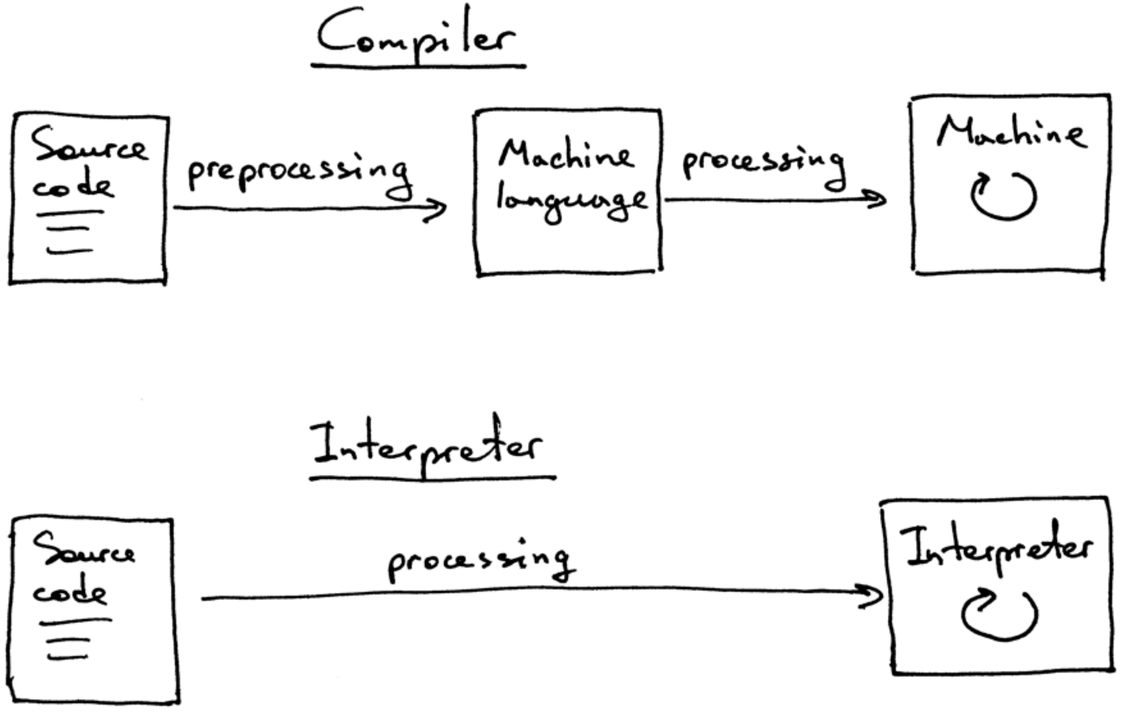
Why Python?

- The **Python syntax** is easy to learn and emphasizes readability which reduces the cost of program maintenance.
- **Python** supports ***modules*** and ***packages***, which encourages program modularity and code reuse.
- The **Python interpreter** and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Compiler vs. Interpreter

Compiler characteristics

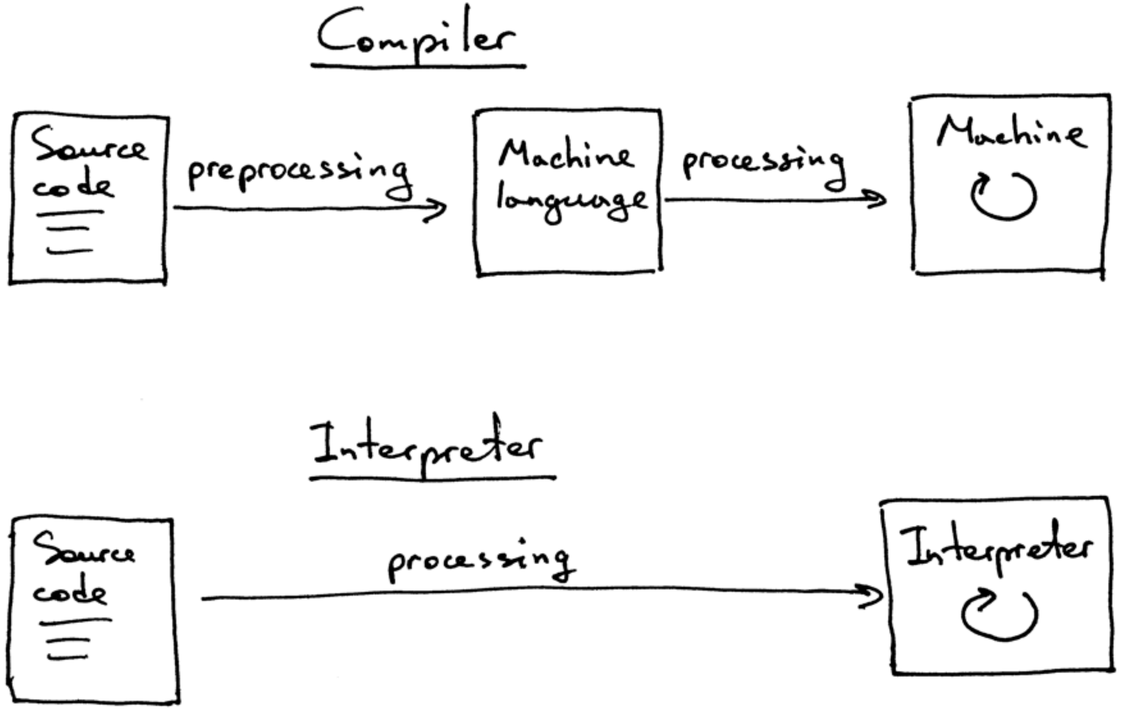
- spends a lot of time analyzing and processing the program
- the resulting executable is some form of machine-specific binary code
- the computer hardware interprets (executes) the resulting code
- program execution is fast



Compiler vs. Interpreter

Interpreter characteristics

- relatively little time is spent analyzing and processing the program
- the resulting code is some sort of intermediate code
- the resulting code is interpreted by another program
- program execution is relatively slow



Setup your environment

- A Linux based operating system, preferably Fedora.
  - https://getfedora.org/
- An integrated development environment.
  - Atom / VScode / Sublime
- The Python interrupter
  - https://www.python.org/downloads/