أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

# Python Basics Workshop

Day 1: February 9

# Agenda

- Morning Session (9:00 – 12:00)
  - Python Basics
  - Variables & Data Types
  - Control Structures
  - Functions & Modules
  - Organizing Your Code & Project Structure
  - Import System
  - Package Management with pip
- Lunch Break (12:00 – 13:30)

# Agenda

- Afternoon Session (13:30 – 15:30)
  - Programming Best Practices
  - Code Style (PEP 8)
  - Documentation
  - Error Handling
  - Project Documentation & Writing README
  - Documentation Tools
  - Code Comments

# Python Basics

- What is Python?
  - A high-level, interpreted language
  - Emphasizes readability and simplicity
- Why Python?
  - Versatile with a large community and libraries
- Setting Up
  - Installing Python and setting up your environment
  - https://www.python.org/downloads/
  - https://www.jetbrains.com/pycharm/
    - Download the free version
  - If you are too lazy:
    - https://colab.research.google.com/

# Variables and Data Types

- Variables
  - Containers for storing data values
  - Dynamically typed: no need to declare a type
- Basic Data Types
  - Numbers (integers, floats)
  - Strings
  - Booleans
- Type Conversion
  - Converting between types (e.g., int(), str())

# Control Structures

- Conditional Statements
  - if, elif, else
- Loops
  - for loops for iterating over sequences
  - while loops for repeated execution
- Loop Controls
  - break and continue

# Functions and Modules

- Functions
  - Defining functions using def
  - Parameters, arguments, and return values
- Modules
  - Importing built-in and custom modules
  - How modules help organize code

# Organizing Your Code & Project Structure

- Code Organization
  - Modularizing your code for readability and reuse
- Project Structure
  - Folder and file organization
  - Naming conventions and best practices

# Package Management with pip

- What is pip?
  - Python's package installer
- Using pip
  - Installing packages (e.g., pip install package_name)
  - pip install numpy
  - pip install pygame
  - All pip packages can be found here:
    - https://pypi.org/
    - You can create your own!
  - You will see better options later(Spoiler conda and docker)
- Virtual Environments
  - Isolating project dependencies
  - pip freeze

# Break

- You can find the dinner in building 13
- Come back by 13:30(1:30PM ☺)

# Programming Best Practices

- Writing Clean Code
  - Maintainability, readability, and simplicity
- DRY Principle
  - Don’t Repeat Yourself
- Code Reviews & Refactoring
  - Continuous improvement of code quality

# Code Style (PEP 8)

- PEP 8 Guidelines
  - Conventions for naming, spacing, and line length
  - Link
- Benefits
  - Consistent and readable code
- Tools
  - Style checkers (e.g., flake8, black)
    - We don’t use either, but we should 🥱

# Documentation

- Why Document?
  - Makes your code accessible and maintainable
- Types of Documentation
  - Inline comments, docstrings, and external docs
- Best Practices
  - Clear, concise, and up-to-date documentation

# Error Handling

- Understanding Errors
  - Syntax errors vs. runtime errors
- Try/Except Blocks
  - Catching and handling exceptions gracefully
- Custom Exceptions
  - Creating exceptions tailored to your application

# Project Documentation & Writing README

- README Essentials
  - Project overview, installation instructions, and usage examples
- Best Practices
  - Clarity, structure, and regular updates
  - https://github.com/othneildrew/Best-README-Template
- Additional Info
  - Licensing, contributing guidelines, and contact info

# Code Comments

- Best Practices
  - When and where to comment your code
- Types of Comments
  - Inline comments vs. block comments
- Balance
  - Write enough to explain “why” without over-commenting