2023

GETTING TO KNOW PYTHON

MODUL 1

AI

Python is a recommended programming language for learning artificial intelligence (AI).

TABLE OF CONTENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Introduction

Python is a high-level, versatile, and beginner-friendly programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python has gained immense popularity in various fields, including web development, data analysis, artificial intelligence, scientific computing, and more, making it one of the most widely used programming languages in the world.

Here are some key features and characteristics of Python:

Readability: Python's clean and easy-to-understand syntax allows programmers to express their ideas concisely and clearly. This readability makes it an excellent choice for beginners.

Versatility: Python is a multipurpose language, suitable for a wide range of applications. You can build web applications, desktop software, scientific simulations, data analysis tools, and even AI and machine learning models using Python.

Large Standard Library: Python comes with a comprehensive standard library that provides modules and packages for various tasks, reducing the need to write code from scratch. This makes development faster and more efficient.

Community and Ecosystem: Python has a vast and active community of developers and enthusiasts. You can find extensive documentation, tutorials, and libraries, making it easier to learn and solve problems.

Cross-Platform: Python is available on multiple platforms, including Windows, macOS, and various Unix-based systems, ensuring that your code can run on different operating systems without major modifications.

Interpreted Language: Python is an interpreted language, which means you can write and run code without the need for a separate compilation step. This makes it an excellent choice for rapid development and prototyping.

Open Source: Python is open-source software, meaning it is freely available, and anyone can contribute to its development. This fosters innovation and collaboration within the Python community.

Python's versatility and ease of use make it an ideal choice for both beginners looking to learn programming and experienced developers working on a wide range of projects. Whether you're interested in web development, data analysis, machine learning, or any other field, Python is a valuable tool to have in your programming arsenal.

Install Python

Step 1 — Downloading the Python Installer

1. Go to the official <u>Python download page for Windows</u>.
2. Find a stable Python 3 release. We will install Python version 3.10.10.
3. Click the appropriate link for your system to download the executable file: Windows installer (64-bit) or Windows installer (32-bit).
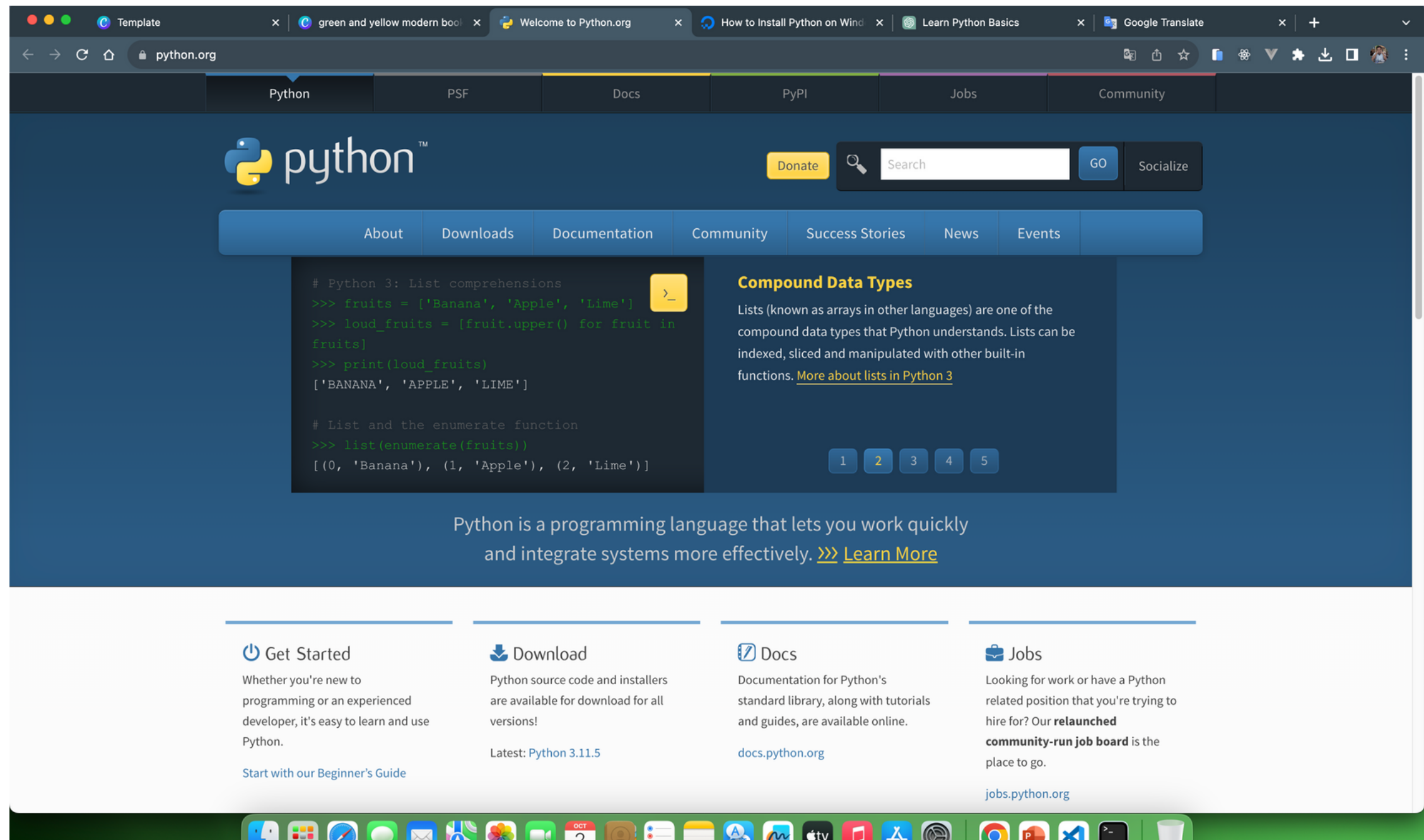
Step 2 — Running the Executable Installer

After installer Downloaded, click file downloader.exe to install python, select add python.exe and follow next instruction, you can click next afther this.
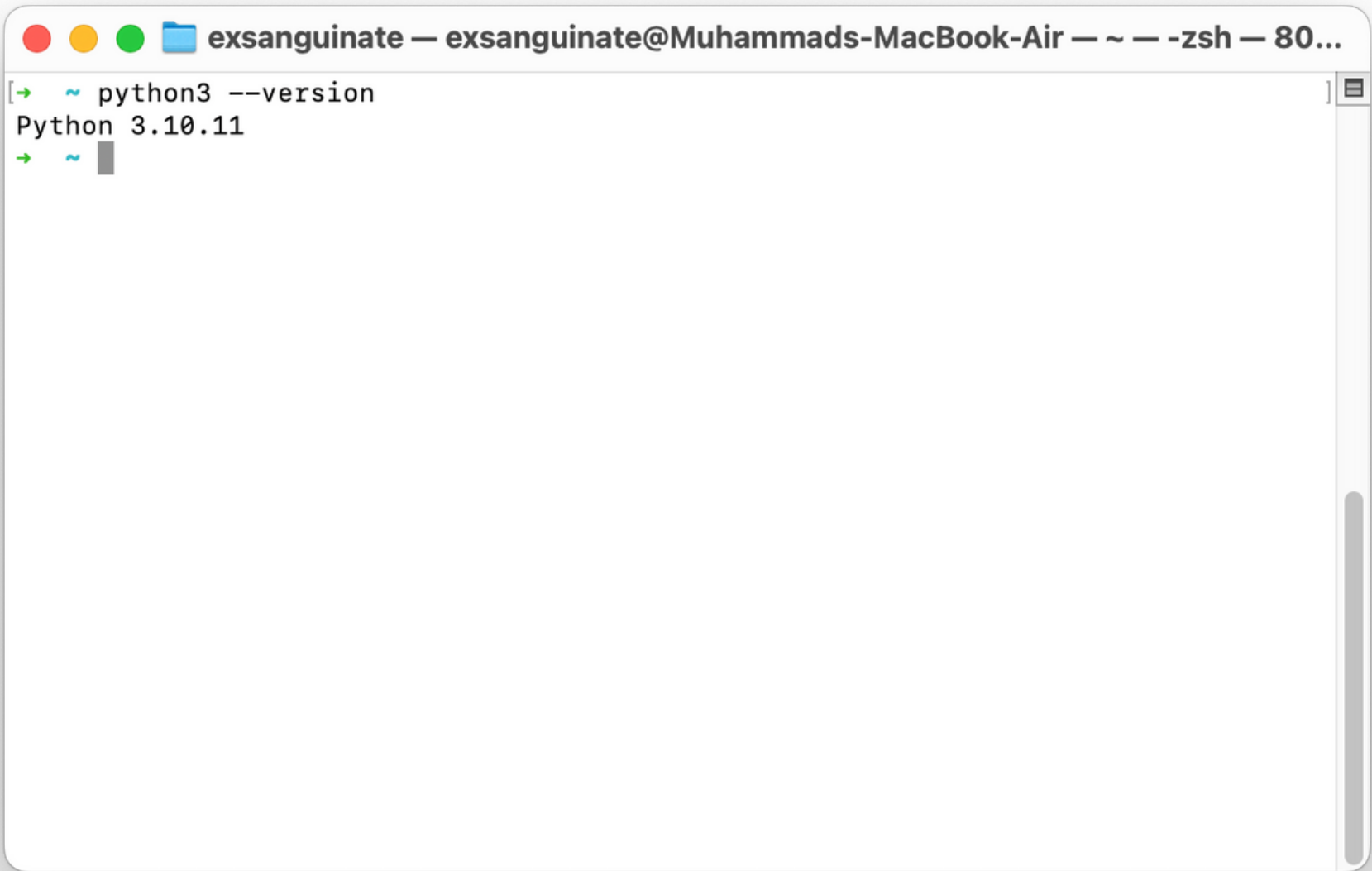
Step 3 — Verify the Python Installation

Open CMD and write command "python --version"

The command written will display the version of Python that is already installed. Since we have successfully installed Python, let's proceed to explore some of the syntax rules of Python. We will now try to run our Python file in Visual Studio Code. If your computer doesn't have Visual Studio Code installed, you will need to install it first. This tutorial will not cover how to install Visual Studio Code. Please find instructions on how to install Visual Studio Code on your own.
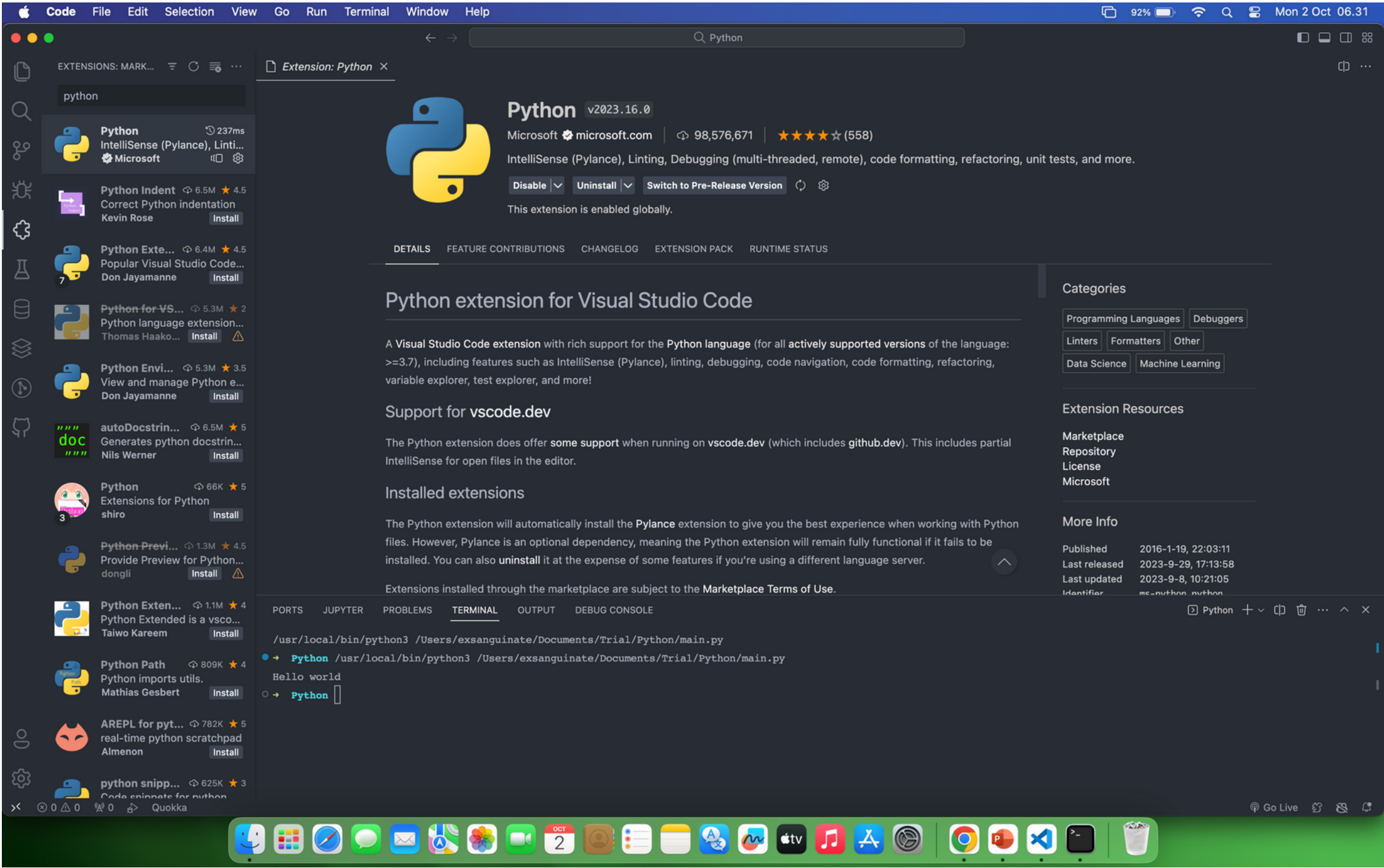
Install the Python extension in Visual Studio Code. The purpose is to enable IntelliSense (Pylance), linting, multi-threaded debugging, remote debugging, code formatting, refactoring, unit tests, and more.

04

List, Tupple, Set & Dictionary

List:

A list is a collection of mutable elements defined with square brackets [], and its elements are separated by commas.

List Examples:

```python
my_list = [1, 2, 3, 4, 5]
fruits = ['apple', 'banana', 'cherry']
mixed_data = [1, 'hello', 3.14, True]
```

Tuple:

A tuple is similar to a list but is immutable. It is defined using regular parentheses ().

```python
my_tuple = (1, 2, 3, 4, 5)
coordinates = (3.14, 2.71)
```

Set:

A set is a collection of unique elements, and it is unordered. Sets can be defined using curly braces {} or the set() function.o')

```python
my_set = {1, 2, 3, 4, 5}
unique_chars = set('hello')
```

Dictionary:

A set is a collection of unique elements, and it is unordered. Sets can be defined using curly braces {} or the set() function.o')

```python
my_set = {1, 2, 3, 4, 5}
unique_chars = set('hello')
```

Latihan:

1. **Soal Set (Himpunan)**:
   - Buatlah sebuah set yang berisi angka 1 hingga 5.
   - Bagaimana cara menambahkan elemen baru ke dalam set?
   - Apakah set memungkinkan duplikasi elemen? Mengapa atau mengapa tidak?
2. **Soal Tuple**:
   - Buatlah sebuah tuple yang berisi beberapa nama buah.
   - Apa yang membedakan tuple dari list dalam hal sifat mutability (kemampuan untuk diubah)?
3. **Soal Dictionary**:
   - Buatlah sebuah dictionary yang berisi informasi sederhana tentang sebuah buku, seperti judul, penulis, dan tahun terbit.
   - Bagaimana cara mengakses nilai dari dictionary berdasarkan kunci (key)?
   - Apakah dictionary memungkinkan kunci yang sama? Mengapa atau mengapa tidak?
4. **Soal Campuran**:
   - Buatlah sebuah tuple yang berisi beberapa nama kota.
   - Kemudian, buatlah sebuah dictionary yang mengaitkan masing-masing kota dengan jumlah penduduknya.
   - Bagaimana Anda akan menambahkan kota dan jumlah penduduk baru ke dalam dictionary ini?
5. **Soal Pemahaman Konsep**:
   - Jelaskan perbedaan utama antara set, tuple, dan dictionary dalam Python.
   - Kapan Anda akan memilih salah satu dari ketiganya daripada yang lain dalam situasi pengembangan nyata?

OOP

Class

Classes provide a means of bundling data and functionality together.

Example:

```python
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year
        self.speed = 0

    def accelerate(self, mph):
        self.speed += mph

    def brake(self, mph):
        self.speed -= mph

    def honk(self):
        print(f"{self.make} {self.model} honks!")

    def get_speed(self):
        return self.speed
```

This Car class has a constructor __init__ that initializes the car's make, model, and year. It also initializes the car's speed to 0. The class includes three methods:
1. accelerate(self, mph): Increases the car's speed by the given number of miles per hour (mph).
2. brake(self, mph): Decreases the car's speed by the given number of miles per hour (mph).
3. honk(self): Prints a message indicating that the car is honking, including its make and model.
4. get_speed(self): Returns the current speed of the car.

You can create instances of this Car class and use its methods like this:

```python
my_car = Car("Toyota", "Camry", 2022)
print(f"My car is a {my_car.year} {my_car.make} {my_car.model}.")
print(f"Current speed: {my_car.get_speed()} mph")

my_car.accelerate(30)
print(f"Accelerating... Current speed: {my_car.get_speed()} mph")

my_car.brake(10)
print(f"Braking... Current speed: {my_car.get_speed()} mph")

my_car.honk()
```
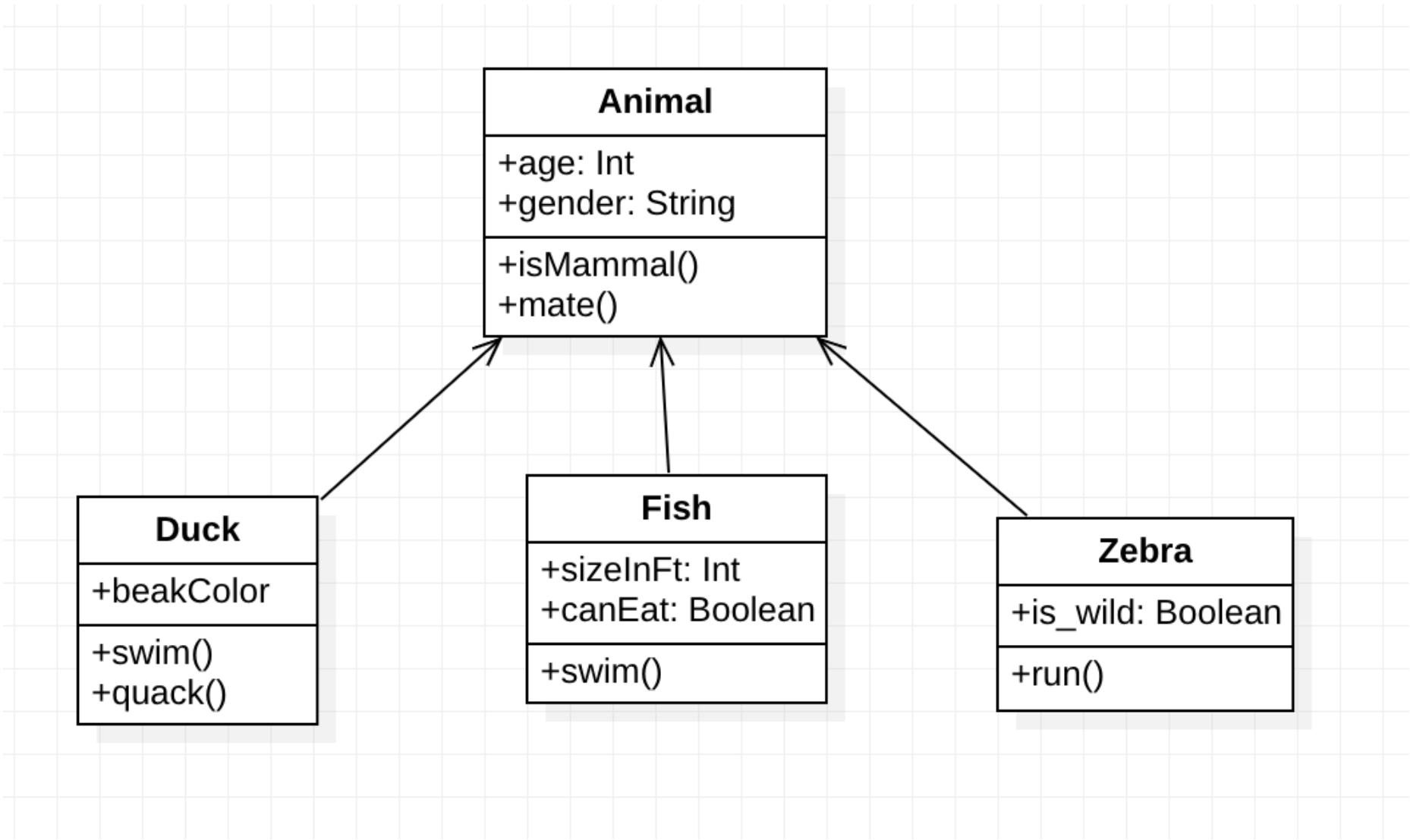
Output:

```
My car is a 2022 Toyota Camry.
Current speed: 0 mph
Accelerating... Current speed: 30 mph
Braking... Current speed: 20 mph
Toyota Camry honks!
```

Exercise

Create instances of the child classes of animals, each with 3 objects. Then, store all the objects into a tuple.
1. Find animals that belong to mammals and count how many are mammals and how many are not.
2. The duck swims at a speed of 2 meters in 10 seconds. Every 5 seconds, the duck quacks. The fish swims at a speed twice as fast as the duck. When the duck quacks, the fish's speed becomes half of the duck's speed.