Welcome to QuickPass!

Table of Contents

1. Course Overview

The intent of QuickPass is to provide a streamlined training and enrollment process to prepare you for the challenges and technical assessmements required for admission into Code Platoon's Full-stack Immersive program.
In this course, students will learn core competencies that are crucial in developing the foundational knowledge base of any software engineer:

1. Programmatic Thinking and Problem Solving
2. Foundational Data and Data Collection Types
3. Functions and Data Manipulation

The course has no hardware requirements to attend; all of the material and practical exercises are web- or cloud-based so you can focus on the content. It is recommended that you establish a GitHub and repl.it account as they offer great online environments to enter and run code.

Course Modules

1. Problem Solving
2. Python Data Types and Methods (Strings, Numbers, Booleans)
3. Python Conditional Statements, Truthy/Falsy Logic & Functions
4. Python Data Ordered Data Structures (Lists, Tuples)
5. Python Unordered Data Structures (Sets & Dictionaries)
6. Python Loops and Data Manipulation
7. Python Advanced Data Collections (Nested Data Collections)
8. Course Mini Project / Review
9. Course Mini Project / Review

2. Instructor

Aaron Wood

Aaron is a retired Air Force Veteran with over 21 years of service as a meteorologist. He is a graduate of Code Platoon's Full-stack Software Engineering Bootcamp: Lima Platoon in 2020. He has held several software engineer positions as a consultant and also as an instructor teaching introductory Python data science courses to military and government personnel.

aaron@codeplatoon.org

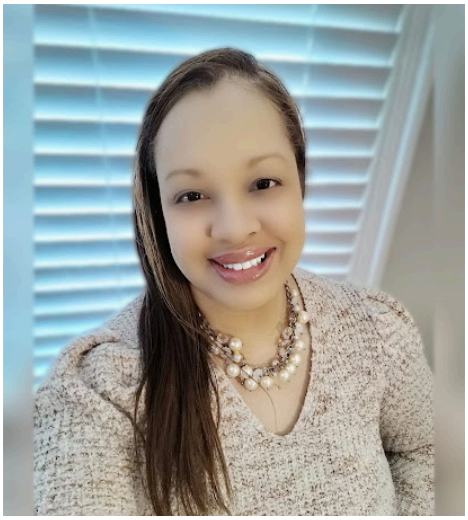
3. Code Platoon Support Team

Greg Drobny

Greg Drobny is a former Airborne Infantryman, PSYOP Team Chief, political consultant, professional mil blogger, and is Code Platoon's Student Outreach and Recruitment Manager.

greg@codeplatoon.org

Tish Johnson



Tish Johnson is the Student Support and Compliance Manager at Code Platoon. In this role, she harnesses her college administration experience to manage the enrollment process for admitted students, including processing VA educational benefits.

tish@codeplatoon.org

Guillermo Aguilar

As Code Platoon's Pre Work Coordinator, Guillermo Aguilar develops the pre work curriculum and live technical assessments for incoming students. He served in Afghanistan and as an Army National Guardsman for eleven years.

guillermo@codeplatoon.org

4. QuickPass Tools

Slack

- Slack

Slack is the primary communication tool of Code Platoon and this class will have its own channel for communication outside of the normal class times. You can use Slack on any platform and, like your favorite messaging service, it supports most types of media and, more importantly, code examples that you can share and discuss with your classmates. You should have already received an invitation to join Code Platoon's workspace, but if you did not, please reach out to any staff member.

GitHub & GitHub Codespaces

- GitHub
- GitHub Codespaces

If you have not already done so, we recommend establishing a GitHub account and getting familiar with basic navigation.

GitHub Codespaces offer an easy way to start up a container with Visual Studio Code (A powerful and Free interactive Development Environment). You will be able to install the various tools & extensions to customize your workspace.

Jupyter Notebooks

- Jupyter

The class repository will contain jupyter notebooks that you can use on your own machine. We also include HTML and PDF versions with the same information. Jupyter notebooks allow you to run Python code in "cells" instead of a single file which provides a more detailed look into how code executes. The cell you are reading now is a markdown cell, which is mainly for reading and presenting links. The cell below is a code cell. It is Python in this example, but you can run other languages with Jupyter Notebooks.

- Note: The first time you run a cell, you will need to select a Python kernel to run the code. If running in VScode, it will ususally prompt for a Python interpreter at the top. There should also be a "Select Kernel" button at the top-right portion of the VSCode window.

- Code Cell Example

In [ ]:
```
# If you want, you can run this cell with the play button found in upper left or at the top of the window.

import time

st = time.time()

target_num = 5000000
count = 0

while count < target_num:
    count += 1

et = time.time()

total_time = et - st
print(f"The time it took for Python to count to {target_num} took {round(total_time, 2)} seconds")
```

```
The time it took for Python to count to 5000000 took 0.23 seconds
```

Online REPLs

REPL (Read, Evaluate, Print, Loop) is a computer environment where user inputs are read and evaluated, and then the results are returned to the user. Once you install Python on your computer, you can enter a local REPL where you can enter Python commands and see the results. It is useful for testing out simple commands to see what they do. For learning purposes, using an online REPL can aid in learning and they have different additional features to visualize code. Also, because they are based in the browser, you can execute code from anywhere and don't have to worry about installing anything on your local machine.

These are just a couple of examples of online REPLs that you can use to execute python code in a browser environment:

- REPL.it
- Python Tutor

REPL.it will be used for some of your exercises, so it is recommended you create an account (not required). Python Tutor has a great feature to see how code runs step-by-step.

Programming Concepts

When learning the basics of your first programming language, it is helpful to have a basic understanding of what allows a program, whether written in Python or any other language, to run. These concepts will be mentioned again in Quick Pass, and they will also be illustrated in the lessons and practice challenges.

Introduction to Scripts

Scripts are sets of instructions written in a programming language that are executed by a computer. Unlike compiled programs, scripts are often interpreted, meaning they are executed line by line by a scripting engine or interpreter, such as Python's interpreter or the JavaScript engine in a web browser. When running a script, it is typically read top-to-bottom.

- **Key Points**:
  - Scripts can automate tasks, manipulate data, or generate content.
  - They do not require a separate compilation step.
  - Common scripting languages include Python, JavaScript, and Bash.

Basic Input and Output

- **Input**: Receiving data (e.g., from a user, file, or sensor) for processing.
- **Output**: Displaying or sending out data (e.g., to a screen, file, or network).
- **Key Points**:
  - Input and output operations are fundamental for interactive scripts and applications.
  - Python's `input()` and `print()` are examples of basic input and output functions.

Computer Memory

Computer memory is a crucial concept in computer science, involving the storage of data and instructions for processing. The two primary types of computer memory are:

- **RAM (Random Access Memory)**: Temporary storage that is fast but volatile. Python programs primarily use RAM to store data while running. Data stored in RAM is lost when the computer is turned off.
- **Storage (Hard Drive or SSD)**: Permanent storage used to keep data and programs. Slower than RAM but non-volatile.

Memory is managed by the operating system. When writing Python code, you will need to think about how the memory is being utilized and the appropriate methodologies to effectively use and preserve memory.

Programming Languages

Programming languages are avenues for us, as humans, to tell a computer what to do while still maintaining a human-readable format. Behind the scenes, when a program written in a language is run, the language instructions are processed into a non-human-readable format that computers can understand. There are hundreds of different programming languages that you can use, but there are a few that are most popular. At Code Platoon, we focus on Python and Javascript since they are the most popular languages behind full-stack applications.

In QuickPass, we are going to only focus on Python because the formatting is largely human readable and allows for a quick immersion into the key foundational concepts of programming without the overhead of learning the syntax.

Python

- Wikipedia: Python (programming language))
- The Zen of Python
- PEP8 Style Guide for Python

Python is a widely used programming language for general-purpose programming and what we will be exclusively focusing on in QuickPass. Python was created by Guido van Rossum and first released in 1991. Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.

The core philosophy of the language is summarized by the document The Zen of Python, which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

JavaScript

- Wikipedia: JavaScript

Known as the core language of the Web, 99% of websites use JavaScript on the client side for webpage operation. Do not confuse JavaScript with Java, a popular general-purpose, multi-platform language. While they share the same letters in their name, you can equate their similarities to comparing a motorcycle to a hot-air balloon. They are both modes of transportation, but operate in wildly different ways. Like Python, JavaScript is a high-level language which allows it to be more human-readable (but at the expense of increased run time). While we will not be focused on Javascript in QuickPass, it is used extensively in Code Platoon's immersive program.

Variables and Data Types

Variables are named locations in memory used to store data that can change during the execution of a script. Each variable has a data type, which dictates the kind of data it can store, such as integers, floating-point numbers, or strings.

- **Key Points**:
  - Data types include `int` (integer), `float` (decimal number), `str` (string), and more.
  - Choosing the correct data type is important for efficient memory use and program execution.

Resources

One of the most wonderful things about the software engineering field is the availability of online resources to discover a solution to your problem. It is a fact that the vast majority of your time when "developing" is actually reading & searching for solutions, tools, methods, etc. to successfully complete your program/app/webpage/game. In addition to what has already been shared above, below is a categorized listing of resources that you can use throughout this course and beyond (more will be introduced as well):

Interactive Learning Platforms

- **Codecademy** - https://www.codecademy.com/
- **freeCodeCamp** - https://www.freecodecamp.org/
- **Khan Academy** - https://www.khanacademy.org/
- **LeetCode** - https://leetcode.com/

Video Tutorials

- **Coursera** - https://www.coursera.org/
- **edX** - https://www.edx.org/
- **Udacity** - https://www.udacity.com/

Documentation and Reference

- **Mozilla Developer Network (MDN)** - https://developer.mozilla.org/
- **W3Schools** - https://www.w3schools.com/
- **Microsoft Learn** - https://learn.microsoft.com/

Books and Guides

- **Eloquent JavaScript** - https://eloquentjavascript.net/
- **Automate the Boring Stuff with Python** - https://automatetheboringstuff.com/

Forums and Communities

- **Stack Overflow** - https://stackoverflow.com/
- **GitHub** - https://github.com/
- **Reddit** - https://www.reddit.com/ (Look for specific subreddits like r/learnprogramming and r/webdev)

Practice and Challenges

- **HackerRank** - https://www.hackerrank.com/
- **CodeSignal** - https://codesignal.com/
- **Project Euler** - https://projecteuler.net/

Tools and Editors

- **Visual Studio Code** - https://code.visualstudio.com/
- **Git** - https://git-scm.com/