

PYTHON PROGRAMMING

---

# Unit 0: Pre-Work

# Agenda

## Here's what we'll be covering:

- Who is your instructor
- What is General Assembly?
- Classroom culture
- Course overview
  - Accessing the course materials
  - Homework and quizzes
  - Your final project
  - How to get a certificate of completion
- Installing, and configuring the development environment

# Agenda

## Here's what we'll be covering:

- Introduction to Programming
  - Define programming
  - Begin writing pseudocode
- Introduction to Python
  - Get acquainted with the Python programming language
  - Write “Hello, World” and create comments

## Meet Your Instructor

---

# Suresh Melvin Sigera

- Chief Technology Officer at Spokata.
- Professor of Computer Science at the College of Staten Island, The City University Of New York.
- I love Korean BBQ!!!!



## Meet Your Instructor

---

# How to contact me?

- [sureshsigera@gmail.com](mailto:sureshsigera@gmail.com)
- Slack
- Office Hrs: Wednesday 5:00 PM to 6:00 PM



# What is General Assembly?

- More 20 global campuses across 6 countries
- Thriving alumni community of 50,000+ full- and part-time graduates
- Corporate training with 350+ companies, including 39 of the Fortune 100
- 500,000+ attendees at bootcamps, workshops, and events



# Classroom culture

## Let's all agree to:

- Treat each other with respect
- Avoid bringing distractions into class



# Course overview

[Let's talk about the course.](#)





# Accessing the course materials

- Lessons for the course will can be [viewed](#) here



# Homework and quizzes

- Isn't graded, but you must submit your answers for instructor feedback. We will go over it the next class!
- I might give you surprise quizzes

# Your final project

- Each day, you'll build skills in Python and understand different ways in which you can use it to build applications
- On the day 5, you'll start brainstorming towards to your Python based project focus area
- At the end of 10th week, you'll demo your project for the class



## How to get a certificate of completion

- Complete 80% of the homework
- Don't miss more than 3 classes
- Complete the final project



# Computers Out: Prepare for Class



## Installing, and configuring the development environment

Please be prepared to access the following [Anaconda Python distribution](#) during this lesson:

1. [WINDOWS](#)
2. [MACOS](#)
3. [LINUX](#)



PYTHON PROGRAMMING



# Introduction to Programming



# Learning Objectives

**After this lesson, you'll be able to:**

- Define programming
- Begin writing pseudocode (algorithm)

# Define programming

What is programming?

Computer programming is a way of giving computers instructions about what they should do next. These instructions are known as code, and computer programmers write code to solve problems or perform a task. In other words, **programming is the process of taking an algorithm and encoding it into a notation**, a programming language, **so that it can be executed by a computer**. Although many programming languages and many different types of computers exist, the important first step is the need to have the solution. Without an algorithm there can be no program.



# What is an algorithm?

An algorithm is “a finite set of precise instructions for performing a computation or for solving a problem”.

- A program is one type of algorithm
- All programs are algorithms
- Not all algorithms are programs!

Example of an algorithm

- Directions to somebody's house is an algorithm
- A recipe for cooking a cake is an algorithm
- The steps to compute the cosine of  $90^\circ$  is an algorithm

# What is an algorithm?

In other words, an algorithm is a **list of steps to solve a problem** (or complete a task?) written in "plain English". The algorithm's steps should be written out and numbered in the order in which they are to be executed. Algorithms should be as extensive as necessary to allow one (you?) to complete the task.

Remember, your **algorithm is not only going to tell your program what to do, but how to do it.**

# Some algorithms are harder than others

Some algorithms are easy

- Finding the largest (or smallest) value in a list
- Finding a specific value in a list

Some algorithms are a bit harder

- Sorting a list

Some algorithms are very hard

- Finding the shortest path between Miami and Seattle

Some algorithms are essentially impossible

- Factoring large composite numbers

# What is Pseudocode

Pseudocode is the precursor to the actual code of an algorithm, to solve a problem. It's very high level, and includes very descriptive step-by-step instructions written in plain text.

The neat thing about pseudocode is that there is no universal standard for the industry; it's very informal, and can include anything from simple words, to phrases, to symbols of that nature. But basically it's just jargon used to describe something in **any spoken language to help us understand better and quicker.**



Solo Exercise:

# Algorithm in pseudocode

10 min



1. Write an algorithm in pseudocode that finds the average of two numbers.



Solo Exercise:

# Algorithm in pseudocode



Write an algorithm in pseudocode that finds the average of two numbers.

**Start** AverageOfTwo

**Input:** Two numbers

**Add** the two numbers

**Divide** the result by 2

**Print** the result

**End**



Here are some example of real-life algorithms.

1. **Google** - Relevant content search, and ranking.
2. **Facebook** - The Facebook Algorithm is a process that ranks all available posts that can display on a user's News Feed based on how likely that user will have a positive reaction.
3. **Instagram** - When a post receives a ton of likes and comments, this signals to the Instagram algorithm that your post is quality, engaging content that more people will want to see, so the Instagram algorithm will show it to more users.
4. When you go home from work.



Group Exercise:

# Algorithms in real-life

20 min



Objective : How to get home from work?

## Requirements:

1. Get someone else to test it.
2. How can you be sure that your algorithm will "work ok"?

Once you are done, let's share the solutions.



PYTHON PROGRAMMING



# Introduction to Python



# Learning Objectives

**After this lesson, you'll be able to:**

- Get acquainted with the Python programming language
- Write “Hello, World” and create comments



# Key features of Python

- It's simple
- It's versatile
- It's always improving
- It's popular!



# When to use Python

- Putting up websites
- Analyzing data
- Building robots
- Most use cases!

# When NOT to use Python

Other programming languages exist - Python isn't great for everything!

- Mobile apps
- Huge programs
  - Python is interpreted - the computer reads it as it goes
  - Other programming languages are read in advance!
- Sometimes too easy
  - Easy to expect things to work that don't!

# Comments in Python

Comments are very important in your programs. They are used to tell you what something does in English, and they are used to disable parts of your program if you need to remove them temporarily. Here's how you use comments in Python:

# Comments in Python

Comments are very important in your programs. They are used to tell you what something does in English, and they are used to disable parts of your program if you need to remove them temporarily. Here's how you use comments in Python:

ex2.py

```
1  # A comment, this is so you can read your program later.
2  # Anything after the # is ignored by python.
3
4  print("I could have code like this.") # and the comment after is ignored
5
6  # You can also use a comment to "disable" or comment out code:
7  # print("This won't run.")
8
9  print("This will run.")
```



## Computers Out: Sample Code-a-long Tutorial



Let's do some hands on coding! Go Ahead and fire up your IDE.



# Finish That Sentence

What are your biggest takeaways from today?



“Something that really got me thinking is...”

“The best thing I got out of today is...”

“I discovered...”

“I still want to learn about...”

“I was surprised that...”

# Ask Me Anything!



---

## **Before Next Class:**

**Check Slack  
Ask Questions  
Office Hours**

“

**Don't Forget Your  
Exit Tickets!**

<https://goo.gl/forms/QyV919BvtH3Wh5G2>

See you next time!



**Thank you!**