



MLH Localhost + Basic Training by Capital One Part I

Presenter's Guide

Introduction

This is the presenter's guide for Part I of III of a workshop series created in partnership with MLH Localhost and Capital One's Division for Social Innovation. These workshops are creating with the intention of teaching learners with little to no background in coding the basics of using Python and how they can use it in an artificial intelligence use case.

Pre-Work

Before using this workshop in your classroom (or any setting), you should run through the workshop yourself from beginning to end.

Things to watch out for:

- Is Twitter accessible on your school's network?
- Do students have email addresses?
- Will students be sharing computers?

Materials:

- Computers that are connected to the internet, ideally using Chrome
- A projector

Objectives

By the end of this workshop, participants will be able to:

- Define variables, loops, functions
- Declare and assign variables and use them in loops and functions
- Use loops to reduce repetitive code
- Write functions, call them, save their output to variables, and use the variables in other functions

Summary of Module

- **Participants sign in** – this information is only used to count how many students go through the program and is not shared with anyone.
- **Short warm up** – discuss prior programming experience

- **Define AI** – a few slides that highlight places where students might already engage with AI.
- **Get excited** – try out the final product from the final workshop, then discuss how it works.
- **Review steps to complete** – there are a lot of steps to getting to the final product. It's a good idea to review these at the beginning so students know what's coming up.
- **Review definitions** – this is a preview of what students will learn during the workshop.
- **Guidelines** – we have included some best practices about how students should use what they're learning. Feel free to edit this slide or add additional ones that cover your own classroom rules and expectations.
- **Create a Glitch account** – it is not required to create a Glitch account to use Glitch. However, any "anonymous" projects are deleted after 5 days. If you plan to complete the workshop within 5 days, students can proceed without creating an account. If not, students should make accounts or they will lose their work.
- **Copy the "starter code" for practice** – the code has one file for each of the concepts students will practice during this workshop.
- **Practice basic Python skills** – students will solve a few simple challenges to become familiar with Python syntax.
- **Copy the "starter code" for the project** – students will focus on writing the code for the workshop that is most relevant to AI use cases. For that reason, we provide them with a lot of code to begin with.
- **Intro to web scraping** - what is the definition of web scraping, why are we doing this, etc.
- **Write code!**
- **Test the working project.** At the end of the workshop, participants will be able to interact with a "bot" that returns individual messages from their fave Twitter celebrity.
- **Review and quiz!**
- **Next steps** - we list a few practice problems the students can try in the future.

Pacing

More Experienced Students:

- If you are working with students who have some background in coding (any language), we estimate that it will take a total of **2 hours** to complete this workshop from beginning to end.
- Suggested pacing (30 minute blocks)
 - Session I: Intro and Set Up Glitch (Slide 1 ~ Slide 31, sections 1 and 2)

- Session II: Python Practice (Slide 34 ~ Slide 58)
- Session III: Write Code for Project (Slide 59 ~ 81)
- Session IV: Write Code for Project, close up (Slide 82 ~ end)
- Suggested pacing (45 minute blocks)
 - Session I: Intro and Set Up Glitch (Slide 1 ~ Slide 45)
 - Session II: Python Practice/Write Code (Slide 45 ~ 72)
 - Session III: Finish code, wrap up (Slide 72 ~ end)
- Suggest pacing (1 hour blocks)
 - Session I: Intro and Python Practice (Slide 1 ~ 58)
 - Session II: Write Functions (Slide 59 ~ end)

Less Experienced Students:

- If you are working with students who have little to no background in coding, we estimate that it will take a total of **3 hours** to complete this workshop from beginning to end.
- Suggested pacing (30 minute blocks)
 - Session I: Intro (Slides 1 through 20). This might be a little dry because they don't get to code during this session – feel free to turn this into a home reading assignment, etc.
 - Session II: What's Python?/ Set Up Glitch (Slides 22 - 34)
 - Session III: Practice Python Part I (Slides 35 - 46)
 - Session IV: Practice Python Part II (Slides 47 - 58)
 - Session V: Write Functions Part I (Slides 59 - 81)
 - Session VI: Write Functions Part II (Slides 82 - end)
- Suggested pacing (45 minute blocks)
 - Session I: Intro (Slides 1 - 25)
 - Session II: Practice Python (Slides 22 - 46)
 - Session III: More Python (Slides 47 - 69)
 - Session IV: Write Functions (Slides 70 - end)
- Suggest pacing (1 hour blocks)
 - Session I: Intro and What's Python (Slides 1 - 34)
 - Session II: Practice Python (Slides 35 - 58)
 - Session III: Write Functions (Slides 59 - End)

Troubleshooting

- The most common problem you will run into are typos.
- Python is depended on its indentation. Ensure students are indenting correctly
- If students miss a step, their code won't run.
- Encourage students to Google the error.
- Encourage students to read their code line by line and see if something doesn't make sense.
- Pair students together.

Resources

- [Working version](#)
- [Starter code](#)
- Codecademy [Python 2](#) Course
- [Learn Python](#)