MLH Localhost + Basic Training by Capital One Part II

Presenter's Guide

Introduction

This is the presenter's guide for Part II 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:

- Explain the difference between web scraping and data cleaning
- Explain why you might want to clean data
- Use a regular expression to remove unwanted characters from data
- Import a library into a file

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 what you did during the previous workshop
- **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.

- **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.
- **Review the previous workshop** – review the definitions and what they did last time
- **Preview the workshop** – discuss what they will learn today
- **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 data cleaning** – what is the definition of data cleaning, 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 1.5 hours to complete this workshop from beginning to end.
- Suggested pacing (30 minute blocks)
    - Session I: Intro and Set Up Glitch (Slides 1 - 25)
    - Session II: Write Code (Slides 26 - 37)
    - Session III: Write Code and Wrap Up (Slides 38 - End)
- Suggested pacing (45 minute blocks)
    - Session I: Intro and Write Code (Slides 1- 35)
    - Session II: Write Code and Wrap up (Slides 36 - 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 2 hours to complete this workshop from beginning to end.
- Suggested pacing (30 minute blocks)
    - Session I: Intro (Slides 1 - 17)
    - Session II: Explore the Code (Slides 18 - 31)
    - Session III: Write Code (Slides 32 - 48)
    - Session IV: Write Code and Wrap Up (Slides 49 - End)
- Suggested pacing (45 minute blocks)

    - Session I: Intro and Set Up Glitch (Slides 1 - 25)
    - Session II: Write Code (Slides 26 - 37)
    - Session III: Write Code and Wrap Up (Slides 38 - End)
- Suggested pacing (1 hour blocks)
    - Session I: Intro and Write Code (Slides 1- 35)
    - Session II: Write Code and Wrap up (Slides 36 - End)

Troubleshooting

- The most common problem you will run into are typos.
- 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