

(<http://csmatters.org/pd-new>) P - 06

0bP - 0b110

Python Practice Coding Session



Unit Programming

Revision Date: Jan 09, 2018

Duration: 150 50-minute sessions

Lesson Summary

Summary: In this lesson, teachers will get a brief overview of lists from the previous lesson, and complete up to three Python programming projects.

Objectives: Teachers will:

- Practice indexing with Lists, and get questions answered about the previous session.
- Develop programs using the software development process to identify the problem, plan a solution, code and test.
- Create programs according to standards using lists and strings.

Overview: (Time: 150 min)

1. Practice answering AP CS-P questions. (15 min)
2. Indexing Practice (10 min)
3. Develop solutions with pseudocode (60 min)
4. Coding Practice (65 min)

Learning Objectives

CSP Objective

Big Idea - Professional development includes learning experiences and resources to ensure that teachers understand how the subject(s) they teach addresses the Maryland content standards and the relationships between the subjects they teach and other subjects in the curriculum.

- - LO 1a - Professional development includes learning experiences and resources to ensure that teachers understand how the subject(s) they teach addresses the

Maryland content standards and the relationships between the subjects they teach and other subjects in the curriculum.

Teacher Resources

Student computer usage for this lesson is: **required**

PROG06_Python Practice Coding Session Folder (<https://drive.google.com/open?id=0B5vAY-fhOT-iSTh2WEhoQjFNZUK>)

Lesson Plan

TOTAL: 150 min

Materials: Teachers should have access to a Python IDE (preferably PyCharm). They should also have access to the "Practice Coding Session" PowerPoint.

Optional: Pair teachers based on programming experience and have them work on the programs together.

Part 1: AP Multiple Choice Practice Questions (15 min)

Slide 2: Point out that students will need to answer 74 multiple choice questions in 120 minutes. Take 6 minutes to answer the 3 questions. Work together to review and justify answers.

- **Note:** If a teacher finds the questions easy, have them write similar questions to be used as practice questions.

Part 2: Indexing Practice (10 min)

Slides 3 - 7: Review the different ways to access elements in lists. As with previous sessions, give the teachers about 5 minutes to try these questions on their own. Try to not let this section last too long. We need time to code!

Part 3: Plan pseudocode solutions (60 min)

Slide 8: Remind teachers of the importance of planning out an algorithm before trying to code it. Also remind teachers about adhering to coding conventions.

Slides 9 - 13: Teachers are given 4 programming exercises.

- Program 1: nest if statements
- Program 2: validate an email address
- Program 3: find the largest number in a list
- Program 4: extends program 3 to include a loop to get user input and choose the largest number

Notes on Program 3:

- Take 10-20 minutes to model the algorithm planning process for Program 3 (Slide 11). This program is deceptively difficult for first-time programmers because finding the largest number in a short list is something that our brains can handle without much intentional thought. It can seem as if we are processing the numbers all at once, but a computer must be told to go through the list one element at a time in order to identify the largest value.
- Slide 12 includes an algorithm for Program 3 written in natural language and in the AP CSP language-agnostic pseudocode. Modeling the algorithm planning process for the teachers will support them in their learning of programming if they are new to it and give all teachers a sense of how best to support their own students, many of whom will be new to programming.

Teachers will spend the rest of Part 3 planning the algorithms they will use in their programs.

Part 4: Coding Practice (65 min)

Teachers spend this time coding the algorithms they planned during Part 3. Monitor teacher progress. Be available as they have questions, but do your best to answer their questions in ways that guide them to the solution and expand their understanding, as opposed to just telling them how to write their code.

Sample solutions to the four programs are available under Teacher Resources in this lesson.

Options for Differentiated Instruction

Teachers can work individually, or use "pair programming". Each teacher will select the level of challenge to complete a program from concept to working code to model the process and how to deal with challenges along the way with students.



(<http://www.umbc.edu/>)



(<http://www.umd.edu/>)



(<http://www.nsf.gov/>)

Authored by: CS Matters in Maryland

Website: csmatters.org (<http://csmatters.org>)

Email: csmattersinmaryland@gmail.com (<mailto:csmattersinmaryland@gmail.com>)

This work is licensed under a
Creative Commons Attribution-ShareAlike 3.0 United States License
(<http://creativecommons.org/licenses/by-sa/3.0/us/>)

by University of Maryland, Baltimore County (<http://umbc.edu>) and University of Maryland, College Park
(<http://umd.edu>).

