(http://csmatters.org/pd-new) P - 03

0bP - 0b11

Python: Conditional Statements and Functions

Unit Programming

Revision Date: Sep 07, 2019
Duration: 75 50-minute sessions

Lesson Summary

Summary: In this lesson, teachers will learn about conditional statements and functions, actively working through examples of both. They will be introduced to the coding practice of putting the primary portion of a program into a main function. They will also be given time to work through practice multiple choice problems from the *AP Computer Science Principles Course and Exam Description.*

Outcomes:

- Teachers will understand how Python uses if-statements and while loops to control the flow of a program.
- Teachers will practice reading code in Python.
- Teachers will gain an understanding of how control structures will be represented on the AP exam.
- Teachers will have a basic understanding of functions.

Overview (75 min):

- Overview of if statements (15 min)
- Overview of functions (15 min)
- Scope (5 min)
- Practice with Functions (20)
- AP Exam Practice (20)

Learning Objectives

Key Concepts

Teachers should understand:

- how boolean logic works.
- know how to utilize if statements and loops.
- have a basic understanding of what functions are and why they are used.

Teacher Resources

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

PROG03_Python Conditional Statements and Functions Folder (https://drive.google.com/open?id=0B9aVxjdD4rTDcWItYVVhbXRMbVU)

Flowchart for squirrelParty program - example 1: https://drive.google.com/open?id=0B7-BtBqRDjG0S2lPYlF6TWJ6MEk (https://drive.google.com/open?id=0B7-BtBqRDjG0S2lPYlF6TWJ6MEk)

Flowchart for squirrelParty program - example 2: https://drive.google.com/open?id=0B7-BtBqRDjG0QkQxNHVwM2dNTkE (https://drive.google.com/open?id=0B7-BtBqRDjG0QkQxNHVwM2dNTkE)

Completed example of squirrelParty program: https://drive.google.com/open?id=0B7-BtBqRDjG0U2tpOHBvSkdnUGc (https://drive.google.com/open?id=0B7-BtBqRDjG0U2tpOHBvSkdnUGc)

Code skeleton for squirrelParty program: https://drive.google.com/open?id=0B7-BtBqRDjG0dzVHWG51RE9FaFE (https://drive.google.com/open?id=0B7-BtBqRDjG0dzVHWG51RE9FaFE)

Link to the Runestone book "CS Principles: Big Ideas in Programming": http://interactivepython.org/runestone/static/StudentCSP/index.html (http://interactivepython.org/runestone/static/StudentCSP/index.html)

Link to "Purple Book (https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf)" (AP CSP Course and Exam Description)

Lesson Plan

TOTAL: 75 min

Materials: The instructor should follow the Conditional Statements and Functions presentation (https://drive.google.com/open?id=1YVeQjBg_nK9Rv-ighyaj-M97wSy0udY8MhJcZhMmWMs). The teachers should have access to the presentation and the *Runestone Interactive* website.

Python Console: Pycharm, repl.it, or any other Python 3 interpreter

Note: Pair teachers based on programming experience with Python.

Overview of if Statements (15 min)

Slides 2-11: Conditional statements are used to make decisions within a program. The instructor introduces the various forms that conditional (if) statements and blocks can take. As they go through the slides, the instructor should ask teachers to consider how they make decisions in their every-day lives and how they might translate a given decision-making process into one of the forms for the *if* statement (*if, if-else, if-elif, if-elif-else*, etc.)

Example: How does one decide...

- whether or not to take an umbrella when going outside?
- what to eat on a menu when dining out?
- which route to take on the way to work?

Aspects of *if* statements/blocks to emphasize to teachers:

- When using the *if-elif-else* sequence, one can leave out the second or third part (you are not required to provide an *else*, nor are you required to provide an *elif*. There MUST be an *if* at the start)
- If you are using *if-elif*, *if-elif-else*,or *if-else*, these blocks of code must occur in that order sequentially from top to bottom. The code will error if you have an if statement, then some extra code, then the else.
- You can chain together as many *elif*'s as you want with one *if* statement! There can be only one *if* or *else* per logic path (block).

For each form of the conditional statement, the instructor should point out the way in which it is represented within the language-agnostic pseudocode and block language syntaxes specified in the AP CSP Course and Exam Description (https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf).

As the instructor goes through the different examples, they should type them out and run the code so that the teachers can see how the code will execute. It is helpful to discuss and make predictions about what the code will do before running it. It is also a helpful teaching practice to pose modifications to the decision-making scenario and have teachers contribute suggestions as to how to change the code. (*What if we were in a state where the minimum age for driving is different? How would we change the code to reflect that?*)

Overview of functions (15 min)

Slides 12-14: A **function** is a piece of code that accomplishes a specific task. Other code can "call" the function to do that task, using the funcName() syntax. Teachers will come up with a list of functions they've already used (turtle functions, range function, etc.).

Slides 15,16: Functions are one of the more advanced topics in programming. Teachers need to have a strong understanding of how and why functions are created. Have teachers journal for a few minutes and then share out as to why being able to make ones own functions would be useful for programming.

Why would we want to be able to make our own functions?

- Useful for code in which tasks are repeated (Do not have to write the code for the task more than once.)
- Make it easier to organize code into a sequence of tasks. (The code is easier to create and to read.)
- Allow one to build an algorithm in incremental parts that can be separately coded and debugged.

Slides 17-20: The instructor should also emphasize that, unlike mathematical functions, a function in a program (sometimes called a **procedure**) does not have to have an input or a return. Tip: Use the term "return" instead "output". In a program, the output refers specifically to what a program outputs to the user interface (screen, speakers, etc..

Slide 21: The instructor shows the teachers an example of a program that calls a function called drawSquare(num), which draws a square whose sides are a specified number of pixels in length. This example also illustrates the practice of putting the main portion of the program into its own function called *main()*. The function definition for the main function goes below all of the other function definitions in Python. After the main function definition, there needs to be a function call for main() in order for the program to execute once it is run.

Scope (5 min)

Slides 22,23: A function is like a "black box." It only knows what is passed to it (through inputs) and what is defined within the function itself. The only communication between the function and the outside world is the input variables and returns!

Variables that are defined within a function are **local** to that function, meaning that no other part of the program has access to that variable unless it is passed as a return value of the function or defined outside of the function. For this reason, it is a good practice to define all variables in the main function if they will be needed there.

Note: Unlike with some languages like C, Python does not require that variables be defined at the top of the code. This is a double-edged sword; although it makes it easier to create a running code, it can be harder to keep track of what variables were initialized where if some sort of standard is not followed when writing the code.

Practice with Functions (20 min)

Slide 24: The instructor presents the following exercise prompt to the teachers.

When squirrels get together for a party, they like to have cigars. A squirrel party is successful when the number of cigars is between 40 and 60, inclusive, unless it is the weekend, in which case there is no upper bound on the number of cigars. Write a program that determines whether or not a given squirrel party was a success.

Students tend to want to skip the planning step, but it is crucial for their success in programming to learn to plan out an algorithm before attempting to type out code for the program. Reiterate often to teachers that they will need to repeatedly model and enforce the planning expectation. Spend 5-10 minutes planning out the algorithm for the program. Do this on a white board/paper - not on the computer. Below are two examples of flow charts that instructors developed with teachers during this session.

- Flow Chart Example 1 (https://drive.google.com/open?id=0B7-BtBqRDjG0S2lPYlF6TWJ6MEk) - Determine if it is the weekend or if it is a weekday first.

Then use the number of cigars to determine if the party is a success.

- Flow Chart Exampe 2 (https://drive.google.com/open?id=0B7-BtBqRDjG0QkQxNHVwM2dNTkE) - Determine if the number of cigars is at least 40. Then determine whether or not it is a weekend or a weekday. Then determine if the party was a success.

Make sure to get all teachers involved in this process and to not let one or two teachers monopolize the algorithm-planning process.

Once the instructor and teachers have planned an algorithm for the program, the instructor can begin live coding with the teachers. The instructor should model good teaching practice by starting with a blank file, completing the header information (name, date, brief summary of program). The instructor should create a function called squirrelParty that decides if the party was a success, using a main function in the code that calls the squirrelParty function.

An example of the end product for the program can be found here (https://drive.google.com/open?id=0B7-BtBqRDjG0U2tpOHBvSkdnUGc). (matches algorithm in flow chart example 1)

Slide 25: The instructor can also show teachers this example of skeleton code (https://drive.google.com/open?id=0B7-BtBqRDjG0dzVHWG51RE9FaFE). (matches flow chart example 1) Skeleton code can be given to students to scaffold the process of coding after they have planned out their algorithm. The instructor may also find it helpful to begin the live coding with the skeleton code to save time or support teachers who are new to programming.

AP Exam Multiple Choice Practice (20 min)

Slide 26: The instructor should use the remainder of the session to allow teachers to work on the listed practice multiple choice questions from the Fall 2017 version of the AP CSP Course and Exam Description (https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf). The instructor should circulate and be available to answer questions that teachers have. At the same time, when a teacher asks for the instructor's help, they should do their best to help the teacher without directly giving the answer. This will model how teachers can help students get to the point at which they can work through such questions on their own.

Options for Differentiated Instruction

If there are teachers who are already very familiar with programming, run the optional session PROG-3 on Programming Challenges in the PD lessons folder. There is a wide variety of challenges for any level of skill that teachers can use to practice their coding and to select example problems to use with their classes to develop algorithms for differentiated instruction.

Evidence of Learning

Formative Assessment

Teachers will get feedback on their understanding throughout the lesson through the various labeled "Tasks," where they practice new concepts.

Summative Assessment

Teachers can self-assess by completing the homework assignment.


(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)