(http://csmatters.org) 2 - 12

0b10 - 0b1100

Functions: Parameters and Return Values

Unit 2. Developing Programming Tools

Revision Date: Feb 06, 2020
Duration: 1 50-minute session

Lesson Summary

Summary

In Python, a function is a named sequence of statements that belong together. In this lesson, students learn why functions are used, how they are used, and how they are defined.

Outcomes

- Students learn how functions affect program flow, how functions use local variables, and how they use global variables.
- Students learn preferred means of communication with functions and how to incorporate them into their programs. Attention is also given to debugging programs that contain functions.
- Students will be introduced to procedures (functions) on the exam reference sheet.

Overview

1. Getting Started (5 min)
2. Guided Activities (35 min)
    1. Arguments and return values [5 min]
    2. Compound Functions [5 min]
    3. Function Definitions [5 min]
    4. Return Values [10 min]
    5. Line Numbers [10 min]
3. Wrap Up (10 min)

Source

Students will work extensively with the online version of How to Think Like a Computer Scientist (HTLACS) hosted by Runestone Interactive.

Learning Objectives

CSP Objectives

- *EU AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.*
    - LO AAP-3.A - For procedure calls: a. Write statements to call procedures. b. Determine the result or effect of a procedure call.
    - LO AAP-3.B - Explain how the use of procedural abstraction manages complexity in a program.

Math Common Core Practice:

- MP1: Make sense of problems and persevere in solving them.
- MP5: Use appropriate tools strategically.

Common Core Math:

- F-IF.1-3: Understand the concept of a function and use function notation
- F-IF.4-6: Interpret functions that arise in applications in terms of the context
- F-BF.3-5: Build new functions from existing functions

Common Core ELA:

- RST 12.4 - Determine the meaning of symbols, key terms, and other domain-specific words and phrases
- RST 12.10 - Read and comprehend science/technical texts

NGSS Practices:

- 5. Using mathematics and computational thinking

Key Concepts

Students will understand the purpose of functions and how they allow a program to be built and maintained in a modular way.

Predictable Misunderstandings:

- students often think that functions (or any code) written will run. They don't connect that it must be called in order to be run.

- students often think that functions have access to variables that they don't have access.

Essential Questions

- How can computing and the use of computational tools foster creative expression?
- How does abstraction help us in writing programs, creating computational artifacts and solving problems?
- How are algorithms implemented and executed on computers and computational devices?
- How are programs developed to help people, organizations or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?
- How do computer programs implement algorithms?
- How does abstraction make the development of computer programs possible?
- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?
- How does computing enable innovation?

Teacher Resources

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

Other:

- Function Video
    - http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html (http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html)
- Python for Everybody
    - http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)
- Runestone Interactive
    - http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html (http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html)

Lesson Plan

Getting Started (5 min)

1. Show first 3:30 of Function Video: http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html (http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html)

2. Students respond in their journals to the following prompts.
3. Open Getting Started Unit 2-12.ppt in Teacher Resources when reviewing the questions below with the students (allow students to share their responses before displaying the answers on the ppt).

- What is a function?
- Why are they used?
- What is a parameter?
- What are two types of parameters?

Guided Activities (35 min)

Arguments and return values [5 min]

Point out: Data values can be stored in simple variables, lists of items, or standalone constants and can be passed as input to, or output from, procedures.

- Students read Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) Chapter 4 Functions sections 4.1 to 4.3.
- Students copy the sentences below. Replace the words *argument* and *result* with synonyms.
    - “It is common to say that a function "takes" an argument and "returns" a result. The result is called the **return value**.”
- Visit this web page (https://docs.python.org/2/library/functions.html (https://docs.python.org/2/library/functions.html)) that lists the built in functions in Python.
- Verify that the Python Version is 2.7.
- Ask, "How many built in functions are in Python 2.7?"
- Change the Python version to 3.4.1. How many built-in functions are in Python 3.4.1?
- Why do you think there are a different number of functions in Python 3.4.1 than in 2.7?

Compound Functions [5 min]

- Open Runestone Interactive Functions page (http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html (http://interactivepython.org/runestone/static/thinkcspy/Functions/functions.html)) and read about user defined functions in Python.
- Read through lines 1 and 2 that define the header and body of compound statements. Note the format of the compound statement used to define a function.
- Ask:
    - What color is used to highlight the keyword def?
    - What punctuation mark ends the first(header) line?
    - How far is the body indented?
    - What color is the name of the function?
    - What word is used for the value in parenthesis?

Function Definitions [5 min]

- Run ActiveCode 1.
- Change the number in line 15 and run the code to make a smaller square.
- Change the number in line 15 and run the code to make a larger square.
- Run ActiveCode 2.
- Ask: What are the two line numbers used to call or invoke the function drawSquare?
- Run ActiveCode 3.
- Change the definition of drawMulticolorSquare to different colors.
- Ask: What line would you change to draw smaller multi-color squares?

Return Values [10 min]

- Some functions find and return values. Complete ActiveCode 5, 6 and 7.
- Change ActiveCode 7 so it prints the minimum values.

Line Numbers [10 min]

- Run CodeLens 1.
- Ask:
    - What line is executed next after line 6?
    - What line is executed next after line 3?

    - How do you think Python knew what line to go to after line 3?
    - What keyword is used to tell Python to return a value from the function named square?
    - What do you think would happen if you run the program after removing the return statement from line 3?

Exam Reference Sheet

Say: Procedures are referred to by different names, such as method or function, depending on the programming language. The exam reference sheet uses the name PROCEDURE instead of the Python word function. It provides:

1. procName (arg1, arg2, …) as a way to *call* PROCEDURE procName(parameter1, parameter 2, …)
2. Procedure DISPLAY(expression) to display the value of expression, followed by a space.
3. Procedure INPUT(), which accepts a value from the user and returns the input value.
4. result ← procName(arg1, arg2, …) to assign to result the value of the procedure “procName”.
5. PROCEDURE procName(parameter1, parameter2, …) { } which is used to *define a procedure* that takes zero or more arguments and does not return a value.
6. RETURN(expression) statement, to return control to the point where the procedure was called and return the value of expression.
7. PROCEDURE procName(parameter1, parameter2, …) { RETURN(expression) } which is used to *define a procedure* that takes zero or more arguments to return control to the point where the procedure was called and return the value of expression.

Say: The entire exam reference sheet will be given to students when they take the end of course exam.

.

Wrap Up (10 min)

- Return to Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) Chapter 4 Functions Copy and paste the code from section 4.4 to Runestone ActiveCode 4.
- Run the code multiple times.
- Ask:
    - What do you note about the results?
    - What is the name of the function used?
    - Is the function on the list of the built-in Python functions or is it defined in the program?
    - Where do you think the function is defined?
- Add the number 1 as a parameter in the function call and run the program.
- Explain the error message.
- Write one thing to remember about functions on an exit note.

Homework: Students should read Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) Chapter 4: Sections 4.7, 4.8 and 4.9. Complete Exercises 2 and 3.

Answer the following questions:

- What error message occured in Exercise 3?
- Why did the error occur?
- What happened when you change the order of print_lyrics and repeat_lyrics function definitions in Exercise 3?
- How is a function call like a road detour?
- The parameter used to define the function print_twice is named bruce. How would the function behavior change if the name was changed to wayne all three times it appears in the function definition?

Options for Differentiated Instruction

Students should be paired through this exercise since paired discussion is used for formative assessment. Question can be provided to students through a variety of formats including production of a Google form or using student response systems.

Three suggested strategies are:

- provide guide questions for reading
- provide teacher provided annotated text excerpts
- Students annotate text while reading

Evidence of Learning

Formative Assessment

After Activity D, have students compare results with their elbow partners. Discuss any unresolved issues. This strategy can be used after any check for understanding.

After Activity F, ask students to suggest a rule for creating functions that would help avoid this error.

Students work alternately between the web site, partners, and the whole group. Teachers are to monitor student responses to the questions following each activity to be sure that students are addressing the key content within each activity.

Summative Assessment

- Explain the advantages of functions/procedures in programming.
- Explain the purpose of parameters with the use of functions.
- Define a Python function with and without parameters.
- Call a Python function with and without parameters.
- Trace a function call from the main program to and from a user defined function.
- Modify a Python program to use a function.
- Create and use functions that return a string or a number.

(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)