



(<http://csmatters.org>) 2 - 15

0b10 - 0b1111

Strings: Traversing, Slicing, and Parsing

Unit 2. Developing Programming Tools

Revision Date: Sep 08, 2019

Duration: 2 50-minute sessions

Lesson Summary

Summary

Students will use the online book *Python for Everybody* to complete a two-session guided lab in which they will explore the use of strings in Python.

Outcomes

Students will be able to:

- Identify a string as a sequence of characters that are identified by their index value, beginning with 0 (zero)
- Use the `len` function to get the number of characters in a string
- Traverse strings using both `while` and `for` loops
- Slice strings using `[m:n]`
- Parse strings using the `find` method and slicing
- Debug simple string programs to find and correct problems

Overview

Session 1:

1. Getting Started (5 min)
2. Guided Activity (45 min)
 1. Preparation [5 min]
 2. Code Review [40 min]

Session 2:

1. Getting Started (5 min)
2. Guided Activity (35 min)
 1. Preparation [5 min]
 2. Code Review [30 min]
3. Summative Assessment (10 min)

Learning Objectives

CSP Objectives

- *EU AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*
 - LO AAP-1.C - Represent a list or string using a variable.
- *EU AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*
 - LO AAP-2.D - Evaluate expressions that manipulate strings.
 - LO AAP-2.K - For iteration: a. Write iteration statements. b. Determine the result or side-effect of iteration statements.

Math Common Core Practice:

- MP7: Look for and make use of structure.
- MP8: Look for and express regularity in repeated reasoning.

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

- Strings are made up of individual characters in a sequence. The characters are identified by their position in the string; this position is referred to as an *index value*.
- Index values in Python begin with zero (0).
- Python provides a number of built-in functions and methods that allow a programmer to manipulate strings.
- Strings can be traversed using both `for` and `while` loops. Students must decide which loop structure is appropriate, based on the nature of a program's requirements.

Essential Questions

- How can computing and the use of computational tools foster creative expression?
- How are algorithms implemented and executed on computers and computational devices?
- How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?
- How do computer programs implement algorithms?

- How do people develop and test computer programs?
- Which mathematical and logical concepts are fundamental to computer programming?

Teacher Resources

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

In the Lesson Resources folder:

- *Python for Everybody* by Charles Severance, http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf).

This book is available for download as a .pdf file. We recommend downloading the book and making it available on a shared group drive on local computers for easy access for students. Is it licensed under Creative Commons and is free to download and share. A copy of the pdf is available in the Lesson Resources folder.

Other:

- More information on the `find` method for Python 3.4 can be found at: <https://docs.python.org/3.4/library/stdtypes.html?highlight=find%20method#str.find> (<https://docs.python.org/3.4/library/stdtypes.html?highlight=find%20method#str.find>)
- Multiple choice quiz (located in the Lesson Resources folder).

Lesson Plan

Session 1

Getting Started (5 min)

Journal Question: What is a string?

- Have students share their ideas.

Guided Activity (45 min)

Preparation [5 min]

- Students should open the book *Python for Everybody* by Charles Severance, Chapter 6: Strings (page 67). (Available at http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) and in the lesson folder.)
- Students should launch PyCharm and make a new file called `stringtester.py` to test code as they go through the lesson.
- Recommend teachers project their screens with the book and PyCharm environment when appropriate.
- Have students test code samples from the book as the students and teacher go through the lesson together.

Code Review [40 min]

Section 6.1: *A string is a sequence*

- Introduce the concept of index values for each character in a string, beginning with 0 (zero).
- After reading the brief text in Section 6.1, have students type the sample code for fruit and letter with `[]` surrounding the index value of a letter and run it. Have students try to 'break' the code by typing in an index value too large for the word and see what kind of error they get.
- Code check for understanding: have students type in a different word for their fruit variable and change the index value. Does the printed letter match what was anticipated?
- Concept check for understanding: reinforce understanding by writing sample words on board, pointing to a letter at random and asking what the index value of that letter.
- Repeat checks for understanding with made-up problems as needed.

Section 6.2: *Getting the length of a string using len*

- Introduce the `len` function, which returns the number of characters in a string. Explain that empty spaces between words and punctuation also count as characters. Remind students of previous lessons concerning ASCII values for all characters, including blank spaces and punctuation.
- Using the `fruit` variable from the previous sample, have students return the number of characters using the `len` function.
- Stress the point in the book concerning `IndexErrors`: that the length of the string and the highest index value in that same string are not the same value. The length of the string `banana` is 6; the index values are numbered from 0 to 5, inclusive.
- Code check for understanding: Have students type in sample code that will return the length of a variety of strings, with and without spaces and punctuation. Did their code run without error?
- Concept check for understanding: Write a sentence on the board with spaces and punctuation. Have students count and write the "length" of the sentence on a post-it note and stick them on a designated place. Have a student or two organize the post-its based on the answers. Together with the class, count out the value of `len`, then have students type the sentence into the console in PyCharm to check their answers.
- Repeat checks for understanding with made-up problems as needed. Some famous quotes can be used to keep it interesting, such as, "The artist is nothing without the gift, but the gift is nothing without work." - Emile Zola (1840-1902).

Section 6.3: *Traversal through a string with a loop*

- Introduce the concept of printing each character of a string on a separate line by using a `while` loop and incrementing the index value of each character.
- Students should already be familiar with iteration using both `while` and `for` loops from previous lessons. Review syntax for both as needed.
- Have students traverse the string by running the sample code.

Session 2

Getting Started (5 min)

Journal Question: Explain why the length of a string is one digit higher than the highest index value of the same string.

- Have students share their answers. Guide the discussion back to the previous lesson, in which students traversed a string with a loop.
- Share a sample of code similar to the traversal with a while loop from the last lesson and ask students to determine the output based on an evaluation of the code. Example:

```
word = "alphabet"
index = 0
while index < len(word):
    letter = word[index]
    print (letter)
    index = index + 1
```

- Remind students of the shortcut to increment index, using `index += 1` in place of `index = index + 1`.

Guided Activity (35 min)

Preparation [5 min]

- Students should open the book *Python for Everybody* by Charles Severance, Chapter 6: Strings (page 67). (Available at http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) and in Lesson Resources folder.)
- Students should launch PyCharm and open the `stringtester.py` file used to test the code that they created during the last lesson.

Code Review [30 min]

Section 6.3: Traversal through a string with a loop continued

- Take students through a discussion of traversing a loop backwards by beginning at the end of the length of the string minus 1, and decrementing the loop control variable (index) instead of incrementing.
- Code check for understanding: Book Exercise 6.1: Write a while loop that starts at the last character in the string and works its way backwards to the first character in the string, printing each letter on a separate line, except backwards.

Sample solution:

```
fruit = "watermelon"
length = len(fruit)
index = length - 1

while index >= 0:
    letter = fruit[index]
    print(letter)
    index -= 1
```

- Concept check for understanding: Project your code with errors and have students write the errors they find on post-it notes and stick them on a designated space. Have a student, or small group of students, organize the post-its in a way which makes sense to them. Debug the program together as a class based on the post-it note results.
- Introduce the concept of using a `for` loop instead of a `while` loop to traverse a string.
- Code check for understanding: Have students type sample code found at the top of page 69 in the book, using a `for` loop to traverse a string.
- Concept check for understanding: Ask students to think about when it is more helpful to use a `while` loop than a `for` loop to traverse a string and share their ideas. Expect answers such as “it is easier to make a `for` loop” and “it is harder to move backwards through a `for` loop than a `while` loop.”
- Explain that the word `char` in the example is used as a variable and is *not* a keyword in Python. It can be replaced with another word or letter. Have students use `x` instead of `char` in their sample `for` loop code to test this concept. This is an opportunity to have a discussion about using meaningful variable names so they have self-documenting code.

Section 6.4: *String slices*

- Students should return to page 69 in the book. Introduce the concepts of substrings and slicing strings using the `[n:m]` syntax to indicate returning a string from the n^{th} character to the m^{th} character, not including the m^{th} character.
- Code checks for understanding: have students type in sample code which will slice strings in all the various ways covered in the section. For example:

```
print(fruit[0:3]) # returns app because a is in the 0th position and the
                 # second p is in the 2nd position.
                 # The slice goes to 'p' not including the 'p'.

print(fruit[:3]) # Still returns app because a is in the 0th position.
                 # Leaving the first index blank begins the slice at the
beginning of the string,
                 # and the second p is in the 2nd position.
                 # The slice goes to 'p' not including the 'p'.
print(fruit[3:]) # Returns le because l is in the 3rd position.
                 # Leaving the second index blank ends the slice at the
                 end of the string.
print(fruit[:]) # Returns apple because n begins at zero and m goes to t
                 he end of the string.
```

- Concept check for understanding: answer the question for Exercise 6.2: Given that `fruit` is a string, what does `fruit[:]` mean?
- Note: If you have time, you may want to go through Sections 6.5 - 6.9. Section 6.9 explains the difference between functions and methods, which will be useful when students use the `find` method while parsing strings.

Section 6.10: *Parsing strings*

- Introduce the `find` method and apply slicing in this section.
- As you go through this section, pay special attention to the `find` method. Explain that the `find` method returns the position at which the substring we are searching for begins.
- Code check for understanding: Have students type in practice code other than what is in the book and use the `find` method without error.

- Sample code:

```
message = 'Meet me at the clock tower @ 7:00 a.m.'  
atSign = message.find('@')  
print (atSign)
```

This code sample returns 27 , indicating that the @ sign is at index value 27 in the string.

Journal: How are string functions used in everyday life? (search engines to find information quickly, school databases to look up student information, many more). How much more efficient is a computer at looking through millions of strings of data compared to a human? Extension: what if you were looking for a particular face in a crowd in a video, how could computational tools enhance that process?

Summative Assessment (10 min)

- Coding assessment and/or quiz. Alternatively, the assessment may be assigned as homework.

Options for Differentiated Instruction

Students can work in pairs while new concepts are introduced and practiced.

One advanced student could be assigned to be the "checker" for each row and have them raise a flag or something similar when they have checked off everybody in their row as having one small group of programming exercises complete and correct. Possibly offer a token prize to the winning row.

For example, after Section 6.3, the following exercise could be assigned:

1. Everybody choose a different fruit.
2. Write code to display the first vowel in the name of the fruit and the first consonant.
3. Display the length of the fruit.
4. Print each character of their fruit name on a separate line by using a while loop.

Evidence of Learning

Formative Assessment

Code checks for understanding and concept checks for understanding are provided with each new function, method, or concept introduced.

Summative Assessment

Summative coding assessment:

- Assign your first and last name to a variable.
- Use the find method to locate the space between your first and last names.
- Use slicing to isolate and print only your first name.

Sample answer code:

```
name = 'Pat Miller'  
space = name.find(' ')  
firstName = name[0:space]  
print (firstName)
```

It is recommend that a 10-question multiple choice quiz that requires students to evaluate code samples from these lessons, determining the output or possible outcomes when the code is run be developed.



(<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>).