| Unit 9 Strings & Special Methods [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing)    *Unit 09 of Python Programming - Unit 9 Strings & Special Methods* | |
| --- | --- |
| Learning Targets  This unit we will…  Take a closer look at some of the string methods and how we may use some of them.  I can…   * Use string methods to manipulate strings. * Use string slicing to manipulate strings. * Use the ASCII table to convert letters to numbers and back again for manipulating text. * Use string methods to evaluate strings. * Convert strings to lists for manipulation purposes and then turn them back to strings.   Vocabulary: Strings, methods, ASCII, slicing, lists, arrays, encryption, objects, immutable | |
| Learn About It!  *You can explore some, or all of these resources. If you want to see a resource again, go for it!*  [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing) *These Collab documents review the concepts of each unit with code you can run and modify.* | |
| Evidence of Learning  *Complete the following programming exercises.*  [Grading Rubric](https://docs.google.com/document/d/1shjqolaw_5tSX9T5OJ2FZuBeon7K3hDrYEJ5m1ltSEw/edit?usp=sharing) | |
| Unit Programs  Review:   1. [Colab - Strings and Special Methods](https://colab.research.google.com/drive/1I2RmdV7iPvgehWxBCoaExKQhQGYGbMwD)   Once you have reviewed the Colab document complete the problems below. There are tips, sample code, and links to sample code that you will use within the Colab documents, you also may want to refer back to early colabs. This project has special directions below about the naming of files and will give you links to some starting code with the proper file naming structure. If you work in pycharm you will need to zip your final files and turn in the one zipped file when you are done.  **Getting Started**  Once you have reviewed the Colab document, complete the problems below. There are tips, sample code, and links to sample code that you will use within the Colab documents, you also may want to refer back to early colabs. There will be two sets of problems to do, the first group can be done in a single file. There are pictures of what your output should look like below. Name the files **Unit09\_YourLastName.py**, if you do this set of problems in [repl.it](https://repl.it/) name the repl.it Unit09\_YourLastName and turn the share link into the classroom.  **Unit 09 Problems**   1. Occurrences of a string in a string - (15 points) -  Required function heading - **def stringCount(s1,s2):** Ask the user to input two strings and then pass those strings into your stringCount() method. One of the strings you ask the user for should be a longer string that will be searched. The second string will be the substring you are going to search for in the longer string. Because of how you ask for the strings you will know between s1 and s2 that you pass into your function which is the long string and which is the substring. Your function should return a number of how many times s2 appears in s1. Sample output below. Make sure your final output puts a set of “ ” around the substring and long string inputs like is shown in the sample output.      1. Longest common prefix - (15 points) -  Required function heading - **def prefix(s1,s2):** Ask the user to input two strings and then pass those strings into your prefix() method. The prefix method will compare the two words to find how much, if any prefix the words share. The method will return the shared prefix if any, if there is no shared prefix it can return an empty string (i.e. string = “”). Use a print statement to output the two input words with quotes around them as well as output the shared prefix with quotes around it.      1. Password Checker - (15 points) -  Required function heading - **def passCheck():** You will create a function that asks the user to input a potential password, the function will check the password to make sure it meets the password requirements. If the password meets the requirements your function will return true otherwise it will return false. Outside of your function print the password requirements and then have a while loop that calls the function until it receives a valid password. Look at the sample output below that shows some failed password creation attempts vs. a successful password creation. The password requirements are:    1. Password is at least 8 characters long    2. Password consists of only letters and digits    3. Password has at least two digits in it 2. Phone Keypad Convertor - (20 points) -  Required function heading - **def getNumber(s):** Required function heading - **def addDash(thenum):** Required function heading - **def phoneConvert(digits):** For this problem you will take the input of a phone number, some of which may be in letter format like 1-800-flowers. Your program has to convert any letters to numbers, based on the phone keypad, and return the converted phone number. For 1-800-flowers your program would have to return 1-800-356-9378.   The getNumber() function will take a single character as a string and if it is a number return that number, but if it is a letter return the corresponding number based on the phone keypad. The phoneConvert() function will take the entered phone number in as a parameter, it will loop over every character in the number and using getNumber() it will convert any letters in the number to numbers and then return the converted phone number. The addDash() function will add dashes “-” in the appropriate places for instance:    1. 1234567 - returns 123-4567    2. 6031234567 - returns 603-123-4567    3. 16031234567 - returns 1-603-123-4567    4. 603-1234567 - returns 603-123-4567    5. 1-603-1234567 - returns 1-603-123-4567 Your finished program should ask the user for a number and return the properly converted and formatted (i.e. with dashes) phone number. Sample output below: 3. Reverse Word - (15 points) -  Required function heading - **def reverse(s):** Create a function that takes in a string parameter and returns a reverse of that string. Write a program that asks the user to enter a word to be reversed and pass the word into your reverse function while outputting the returned string so that the first letter of the output is capitalized.      1. Word Scramble - (20 points) -  Required function heading - **def cryptor(string1, mode):** Create a function that takes a string parameter, and a trigger, either a 0/1 or boolean value. This function will use your reverse() function as one of the steps of encrypting your input word. For the other step of encryption it will use the ASCII table to swap every letter in the input string with another character from the ASCII table. Ask the user for an input and pass it into the cryptor() function, save the return of the function to a variable. Output to the user what the original value entered was and then output the encrypted version. Next, output the encrypted version saved in the variable and then pass that encrypted version into cryptor() so that the original statement is output.     Make sure you have a comment block at the top of your program with your name, the date and a list of the programs that are being run in the program. Also make sure to comment your variables, control structures, and each problem. Also use white space between the problems.  ############################################################  # Name : Date: #  # Unit 9 Problems #  # String in a String, Longest common prefix, #  # password checker, phone keypad convertor #  # reverse a word, string encrypt/decrypt #  ############################################################  When your code works and is commented, turn it into the classroom. | |



These memes are classic.