

1. TRUE / FALSE QUESTIONS

- _____ Indexing works with both strings and lists.
- _____ Once a string is created, it cannot be changed.
- _____ You can use the `for` loop to iterate over the individual characters in a string.
- _____ In slicing, if the end index specifies a position beyond the end of the string, Python will use the length of the string instead.
- _____ Indexing of a string starts at 1 so the index of the first character is 1, the index of the second character is 2, and so forth.
- _____ The index -1 identifies the last character of a string.
- _____ The following code will display 'yes + no':
- ```
mystr = 'yes'
yourstr = 'no'
mystr += yourstr
print(mystr)
```
- \_\_\_\_\_ If a whole paragraph is included in a single string, the `split()` method can be used to obtain a list of the sentences in the paragraph.
- \_\_\_\_\_ The `strip()` method returns a copy of the string with all the leading whitespace characters removed but does not remove trailing whitespace characters.
- \_\_\_\_\_ The `isupper` method converts a string to all uppercase characters.
- \_\_\_\_\_ The repetition operator (\*) works with strings as well as with lists.
- \_\_\_\_\_ When you call a string's `split` method, the method divides the string into two substrings.

**2. COMPLETION QUESTIONS:** Fill in the blanks.

- a) Each character in a string has a(n) \_\_\_\_\_ which specifies its position in the string.
- b) The `isalpha()` method returns \_\_\_\_\_ if the string contains only alphabetic characters and is at least one character in length.
- c) A(n) \_\_\_\_\_ is a span of characters that are taken from within a string.
- d) When the operand on the left side of the \* symbol is a string and the operand on the right side is an integer, the \* becomes the \_\_\_\_\_ operator.
- e) The third number in string slicing brackets represents the \_\_\_\_\_ value.
- f) The \_\_\_\_\_ operator can be used to determine whether one string is contained in another string.
- g) The \_\_\_\_\_ method returns `True` if the string contains only numeric digits.
- h) The \_\_\_\_\_ method returns a copy of the string with all the alphabetic letters converted to lower case.
- i) Method \_\_\_\_\_ returns a new string where all leading and trailing whitespace has been removed.
- j) Python represents strings as sequences of \_\_\_\_\_.

**3. ALGORITHM WORKBENCH QUESTIONS**

- a) Write a loop that counts the number of lowercase characters that appear in the string referenced by `mystring`.
- b) Assume `mystring` references a string. Write code that makes a copy of the string with all occurrences of the lowercase letter 't' converted to uppercase.
- c) Assume `mystring` references a string. Write a statement that uses a slicing expression and displays the last 3 characters in the string.
- d) Look at the following statement:  
`levels = 'Beginner, Average, Advanced, Expert'`  
Write a statement that splits this string, creating the following list from `levels`:  
`['Beginner', 'Average', 'Advanced', 'Expert']`

**MULTIPLE CHOICE QUESTIONS**

4. What will be assigned to the variable `s_string` after the following code executes?
- ```
special = '1357 Country Ln.'  
s_string = special[ :4]
```
- a) '7'
b) '1357'
c) 5
d) '7 Country Ln.'
5. What will be assigned to the variable `s_string` after the following code executes?
- ```
special = '1357 Country Ln.'
s_string = special[-3:]
```
- a) '135'  
b) '753'  
c) 'Ln.'  
d) 'y Ln'
6. This string method returns `true` if a string contains only alphabetic characters and is at least one character in length.
- a) the `isalpha` method  
b) the `alpha` method  
c) the `alphabetic` method  
d) the `isletters` method

7. What will be assigned to the variable `some_nums` after the following code executes?

```
special = '0123456789'
some_nums = special[0:10:2]
```

- a) '0123456789'
- b) '24682468'
- c) '02468'
- d) '02020202020202020202'

8. This string method returns a copy of the string with all leading and trailing whitespace characters removed.

- a) `clean`
- b) `strip`
- c) `remove_whitespace`
- d) `rstrip`

9. Which method would you use to determine whether a certain substring is present in a string?

- a) `endswith(substring)`
- b) `find(substring)`
- c) `replace(string, substring)`
- d) `startswith(substring)`

10. What will be the value of the variable `string` after the following code executes?

```
string = 'abcd'
string.upper()
```

- a) 'abcd'
- b) 'ABCD'
- c) 'Abcd'
- d) Nothing; this code is invalid

11. What list will be referenced by the variable `list_strip` after the following code executes?

```
my_string = '03/07/2018'
list_strip = my_string.split('/')
```

- a) ['3', '7', '2018']
- b) ['03', '07', '2018']
- c) ['3', '/', '7', '/', '2018']
- d) ['03', '/', '07', '/', '2018']

12. What is the return value of the string method `rstrip()`?

- a) the string with all whitespaces removed
- b) the string with all leading whitespaces removed
- c) the string with all leading tabs removed
- d) the string with all leading spaces removed

13. Which method would you use to determine whether a certain substring is the suffix of a string?

- a) `endswith(substring)`
- b) `find(substring)`
- c) `replace(string, substring)`
- d) `startswith(substring)`

### **PROGRAMS**

14. Write a program that counts the number of digits, non-digit characters, whitespace characters and words in a given string.

15. Write a program that asks the user to enter a sentence and checks whether the sentence contains more than one space between words. If so, the program should remove the extra spaces. For example, "Hello    World" should be "Hello World". (Hint: Use `split` and `join`.)

16. (*Check SSN*) Write a program that prompts the user to enter a Social Security number in the format ddd-dd-dddd, where d is a digit. The program displays Valid SSN for a correct Social Security number or Invalid SSN otherwise.

17. (*Check substrings*) You can check whether a string is a substring of another string by using the `find` method. Write your own function to implement `find`. Write a program that prompts the user to enter two strings and then checks whether the first string is a substring of the second string.

18. (*Occurrences of a specified character*) Write a function that finds the number of occurrences of a specified character in a string using the following header:

```
def count(s, ch):
```

The `str` class has the `count` method. Implement your method without using the `count` method. For example, `count("Welcome", 'e')` returns 2. Write a test program that prompts the user to enter a string followed by a character and displays the number of occurrences of the character in the string.

19. (*Occurrences of a specified string*) Write a function that counts the occurrences of a specified non-overlapping string `s2` in another string `s1` using the following header:

```
def count(s1, s2):
```

For example, `count("system error, syntax error", "error")` returns 2. Write a test program that prompts the user to enter two strings and displays the number of occurrences of the second string in the first string.

20. (*Reverse a string*) Write a function that reverses a string. The header of the function is:

```
def reverse(s):
```

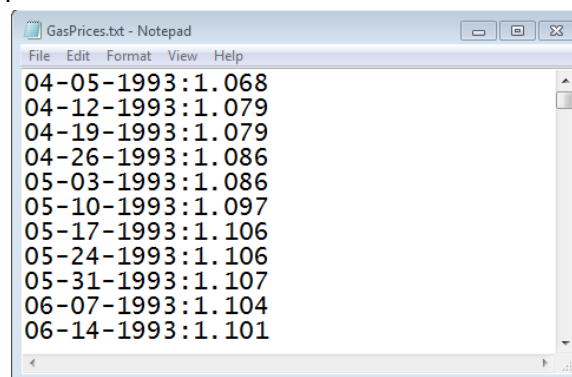
Write a test program that prompts the user to enter a string, invokes the `reverse` function, and displays the reversed string.

21. (Bioinformatics: find genes) Biologists use a sequence of letters **A**, **C**, **T**, and **G** to model a *genome*. A *gene* is a substring of a genome that starts after a triplet **ATG** and ends before a triplet **TAG**, **TAA**, or **TGA**. Furthermore, the length of a gene string is a multiple of **3** and the gene does not contain any of the triplets **ATG**, **TAG**, **TAA**, and **TGA**. Write a program that prompts the user to enter a genome and displays all genes in the genome. If no gene is found in the input sequence, the program displays **no gene is found**. Here are the sample runs:

```
Enter a genome string: TTATGTTTAAAGGATGGGCGTTAGTT Enter
TTT
GGGCGT
```

```
Enter a genome string: TGTGTGTATAT Enter
no gene is found
```

22. Write a program that gets strings containing a person's first and last name as separate values, and then displays their "initials", "name in address book", and "username". For example, if the user enters a first name of "John" and a last name of "Smith", the program should display "J.S.", "John SMITH", and "jsmith".
23. In the student sample program files for this chapter, you will find a text file named `GasPrices.txt`. The file contains the weekly average prices for a gallon of gas in the United States, beginning on April 5th, 1993, and ending on August 26th, 2013. Below figure shows an example of the first few lines of the file's contents:



Each line in the file contains the average price for a gallon of gas on a specific date. Each line is formatted in the following way:

MM-DD-YYYY:Price

MM is the two-digit month, DD is the two-digit day, and YYYY is the four-digit year. Price is the average price per gallon of gas on the specified date.

For this assignment, you are to write one program that read the contents of the file and prints the dates when the gas prices are highest and lowest along with the prices.

(You can access the `GasPrices.txt` from Google Classroom)

24. The file named `text.txt` stores some text as one sentence per line. Write a program that reads the file's contents and calculates the average number of words per sentence. (You can access the `text.txt` from Google Classroom)