Due Oct 16 at 1pm Poil
Allowed Attempts Unlimited

Points 5 Questions 5

Available until Dec 4 at 11:59pm

Time Limit None

Instructions

Prior to completing this quiz, you should have read:

• Section 4.2, Formatted Output (p. 98-107)

Please also go over Practice Problems 4.3, 4.4, 4.5, 4.6, 4.7 in the textbook (solutions at the end of the chapter) before attempting this quiz.

This quiz was created for learning purposes. You may attempt this quiz as many times as you would like. The highest score *prior to the deadline* will count towards the final course grade. No late submissions will be accepted.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	27 minutes	4.83 out of 5

Score for this attempt: **4.83** out of 5 Submitted Oct 15 at 5:13pm This attempt took 27 minutes.

Question 1 1 / 1 pts

The format() string method takes in an infinite number of arguments and formats the arguments based on given place holders. The first argument will be index 0, second will be index 1, and so forth.

```
>>> newtext = '{0} -- {1} -- {2}'.format('zero', 'one', 'two')
>>> print(newtext)
zero -- one -- two
```

Notice in the above example, the curly brackets get replaced by corresponded words passed into the format() function while everything outside of curly brackets prints as-is.

We can change the order of the curly brackets:

```
>>> newtext = '{2} -- {0} -- {1}'.format('zero', 'one', 'two')
>>> print(newtext)
two -- zero -- one
```

Notice that {0} always corresponds to the first string (in this case 'zero'), {1} always corresponds to the next one, and so forth.

If no numbers are specified, the strings will print in the order that they appear:

```
>>> newtext = '{} -- {} -- {}'.format('zero', 'one', 'two')
>>> print(newtext)
zero -- one -- two
```

Match the string to their printout.

Correct!

'{}-{}--{}'.format('cat', 'dog', 'cow', 'pig')

cat-dog--cow---pig 🔻

Correct!

'{1}-{2}--{3}---{0}'.format('cat', 'dog', 'cow', 'pig')

dog-cow--pig---cat

Correct!

'{1}-{1}--{1}---{1}'.format('cat', 'dog', 'cow', 'pig')

dog-dog--dog---dog

Correct!

'{2}-{2}--{2}---{2}'.format('cat', 'dog', 'cow', 'pig')

cow-cow--cow ~

Other Incorrect Match Options:

- · cat-cat--cat
- pig-pig---pig

Question 2 1 / 1 pts

We can enter different specifications inside the curly brackets after a colon as well. For example, the following format requires 5 character spaces and 7 character spaces for each respective placeholder, not counting the characters in between curly brackets:

```
>>> '{0:5}---{1:7}'.format('Joe', 'Bruin')
'Joe ---Bruin '
```

Numbers will be right aligned while strings will be left aligned.

```
>>> '{0:5}---{1:7}'.format(12, 13)
' 12--- 13'
```

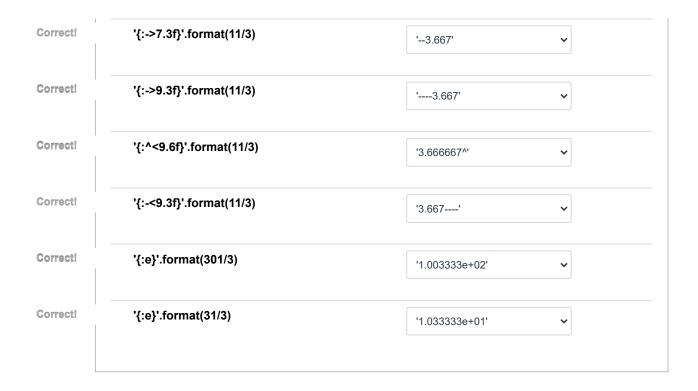
You can use < for left-alignment, > for right-alignment, and ^ for center alignment.

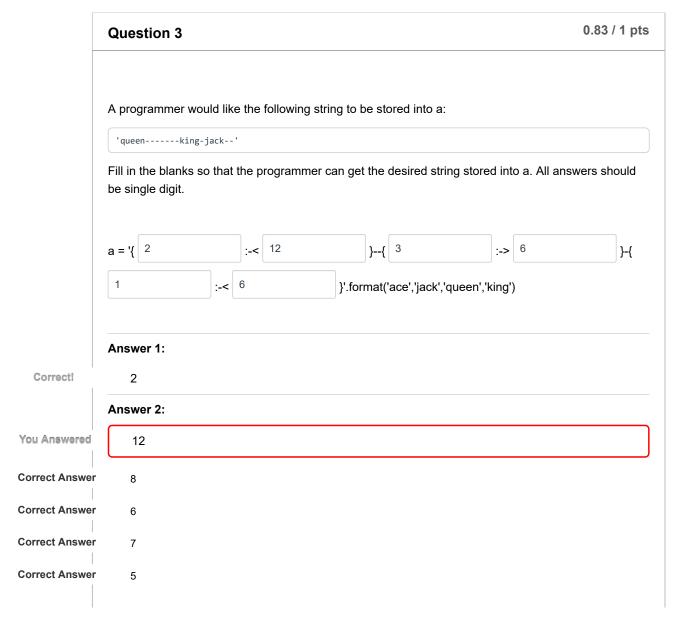
```
>>> '{1:<15}{0:^10}{2:>15}'.format('center', 'left', 'right')
'left center right'
```

Special characters in front of <, >, or ^ will replace blank spaces with that character

```
>>> '{:-<15}'.format('example')
'example------'
>>> '{:<>10}'.format('test')
'<<<<<test'
```

Test out other special formats for numbers, and match them to their corresponding value.





Correct Answer	9			
	Answer 3:			
Correct!	3			
	Answer 4:			
Correct!	6			
Correct Answer	7			
Correct Answer	8			
Correct Answer	9			
Correct Answer	5			
	Answer 5:			
Correct!	1			
	Answer 6:			
Correct!	6			

