

# Learning Quiz 12: Formatted Output

**Due** Oct 16 at 1pm

**Points** 5

**Questions** 5

**Available** until Dec 4 at 11:59pm

**Time Limit** None

**Allowed Attempts** Unlimited

## 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	<a href="#">Attempt 1</a>	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---COW



Other Incorrect Match Options:

- cat-cat--cat---cat
- pig-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!**

'{-&gt;7.3f}'.format(11/3)

'--3.667'

**Correct!**

'{:&gt;9.3f}'.format(11/3)

'----3.667'

**Correct!**

'{: ^&lt;9.6f}'.format(11/3)

'3.666667^'

**Correct!**

'{:&lt;9.3f}'.format(11/3)

'3.667----'

**Correct!**

'{:e}'.format(301/3)

'1.003333e+02'

**Correct!**

'{:e}'.format(31/3)

'1.033333e+01'

**Question 3****0.83 / 1 pts**

A programmer would like the following string to be stored into a:

'queen-----king-jack--'

Fill in the blanks so that the programmer can get the desired string stored into a. All answers should be single digit.

a = '{ 2 :< 12 }--{ 3 :> 6 }{ 1 :< 6 }'.format('ace','jack','queen','king')

	<b>Answer 1:</b>
Correct!	2
	<b>Answer 2:</b>
ou Answered	12
orrect Answer	8
orrect Answer	6
orrect Answer	7
orrect Answer	5
orrect Answer	9
	<b>Answer 3:</b>
Correct!	3
	<b>Answer 4:</b>
Correct!	6
orrect Answer	7
orrect Answer	8
orrect Answer	9
orrect Answer	5
	<b>Answer 5:</b>
Correct!	1
	<b>Answer 6:</b>
Correct!	6

**Question 4****1 / 1 pts**

Consider the list of directives on page 107. Match the directive and a possible output.

**Correct!****%a**

Thu

**Correct!****%A**

Tuesday

**Correct!****%b**

Feb

**Correct!****%B**

March

**Correct!****%l**

01

**Correct!****%M**

01

**Correct!****%p**

AM

**Correct!****%S**

01

**Correct!****%y**

01

**Correct!****%Y**

2019

**Correct!****%Z**

Pacific Standard Time ▼

Correct!

%d

01 ▼

Correct!

%H

01 ▼

## Question 5

1 / 1 pts

By default, integers are presented in decimal notation. Use format to convert decimal number 50 to other formats.

Correct!

'{:b}'.format(50)

'110010' ▼

Correct!

'{:c}'.format(50)

'2' ▼

Correct!

'{:d}'.format(50)

'50' ▼

Correct!

'{:o}'.format(50)

'62' ▼

Correct!

'{:x}'.format(50)

'32' ▼

Quiz Score: **4.83** out of 5