

Learning Quiz 17: while loops

Due Oct 30 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, be sure to read:

- Sections 5.4-5.5: while Loop and More Loop Patterns (p. 143-148)

Please also go over Practice Problem 5.10 and 5.11 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	11 minutes	5 out of 5

Score for this attempt: 5 out of 5

Submitted Oct 28 at 4:24pm

This attempt took 11 minutes.

Question 1	1 / 1 pts

while-loops are similar to for-loops in that they will repeat a set of indented lines multiple times. However, unlike for-loops, while-loops do not necessarily have a set number of times that it will repeat the indented lines. The while loop will stop only if a given condition is broken.

The following code will loop through only one time because the second time around, the condition is no longer satisfied:

```
a = True
while a:
    print("a is now ", a)
    a = False
print("Loop ended at a =", a)
```

This next code will loop through multiple times and stop after a becomes greater than 70:

```
a = 1
while a <= 70:
    print("a is now ", a)
    a *= 2
print("Loop ended after a became", a)
```

It is possible to get "stuck" in an infinite while loop. To stop the following lines of code, press "ESC" at the top-left corner of your keyboard or the red square at the top left corner of PyCharm:

```
a = True
i = 0
while a:
    print(i)
    i += 1
```

In general, be sure that somewhere in the body of your while loop, you update the variable so that you're one step closer to stopping the while loop's condition.

Consider the following lines of code:

```
stopper = 100
while stopper > 10:
    print("You're still in the loop")
    stopper /= 2
```

How often will the program go through the while loop?

Correct!☒ 4☐ 3☐ 0☐ 2☐ 5☐ 1**Question 2****1 / 1 pts**

Which while-loop will produce the same output as the following for-loop?

```
for i in range(10):  
    print(i)
```

☐

```
i = 1  
while i < 10:  
    print(i)  
    i += 1
```

☐

```
i = 0  
while i <= 10:  
    print(i)  
    i += 1
```

☐

```
i = 1  
while i < 10:  
    print(i)  
    i += 1
```

Correct!☒

```
i = 0
while i < 10:
    print(i)
    i += 1
```

☐

```
i = 0
while i < 10:
    print(i)
    i += 1
```

☐

```
i = 0
while i <= 10:
    print(i)
    i += 1
```

Question 3**1 / 1 pts**

In a while loop, the condition is the statement that Python will check before executing a set of indented lines. The condition needs to equate to True if you want Python to run the lines:

The following code will skip the indented block completely because the condition(`num < 10`) is never met:

```
num = 100
while num < 10:
    print(num)
    num *= 0.05
print("Moved on from while loop")
```

What is the condition of the following while loop?

```
x = 1000
while x > 100:
    print(x)
    x /= 2
print("Moved on from while loop")
```

Correct!☒**x > 100**

☐ `x /= 2`☐ `print("Moved on from while loop")`☐ `x = 1000`

Question 4

1 / 1 pts

The following line of code will produce an infinite loop.

```
x = 100
while x > 20:
    print(x)
    ## insert code here ##
```

Which of the following, if added to the end of the while loop, will prevent an infinite loop?

Correct!☒ `x -= 1`☐ `x *= 1`☐ `x += 1`☐ `x /= 1`

Question 5

1 / 1 pts

It is possible to stop a loop in the middle of the indented block. For loops inside functions, one way to stop the loop is to return a value. In functions,

Python will stop a function right away when it sees a "return" and it will continue with the rest of a given program. Consider the following program:

```
def myloop():
    '''returns a list of names input by the user'''
    looptime = True
    roster = []
    while looptime:
        name = input('Please enter name to add in roster: ')

        if name == '':
            return roster

        roster.append(name)

print('Time to get names')
myloop()
print('function ended')
```

Consider the following function:

```
def myfun(word):
    while True:
        return 'done'
        print(word)

userin = input('Enter name: ')
myfun(userin)
```

True/False: The program will never print the user's input.

Correct!

☒ True

☐ False

Quiz Score: **5** out of 5