**Feedback — Week 5 Quiz**[Help](https://class.coursera.org/pythonlearn-003/help/quizzes?url=https%3A%2F%2Fclass.coursera.org%2Fpythonlearn-003%2Fquiz%2Ffeedback%3Fsubmission_id%3D85011)

You submitted this quiz on **Sat 1 Nov 2014 7:35 AM PDT**. You got a score of **10.00** out of **10.00**.

Top of Form

**Question 1**

What is wrong with this Python loop:

n = 5

while n > 0 :

print n

print 'All done'

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| This loop will run forever | Correct | 1.00 |  |
| There should be no colon on the **while** statement |  |  |  |
| The **print 'All done'** statement should be indented four spaces |  |  |  |
| **while** is not a Python reserved word |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 2**

What does the **break** statement do?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Exits the program |  |  |  |
| Exits the currently executing loop | Correct | 1.00 |  |
| Resets the iteration variable to its initial value |  |  |  |
| Jumps to the "top" of the loop and starts the next iteration |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 3**

What does the **continue** statement do?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Jumps to the "top" of the loop and starts the next iteration | Correct | 1.00 |  |
| Exits the currently executing loop |  |  |  |
| Resets the iteration variable to its initial value |  |  |  |
| Exits the program |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 4**

What does the following Python program print out?

tot = 0

for i in [5, 4, 3, 2, 1] :

tot = tot + 1

print tot

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| 0 |  |  |  |
| 10 |  |  |  |
| 15 |  |  |  |
| 5 | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |

**Question 5**

What is the *iteration* variable in the following Python code:

friends = ['Joseph', 'Glenn', 'Sally']

for friend in friends :

print 'Happy New Year:', friend

print 'Done!'

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| friend | Correct | 1.00 |  |
| Sally |  |  |  |
| Glenn |  |  |  |
| Joseph |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 6**

What is a good description of the following bit of Python code?

zork = 0

for thing in [9, 41, 12, 3, 74, 15] :

zork = zork + thing

print 'After', zork

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| Compute the average of the elements in a list |  |  |  |
| Find the largest item in a list |  |  |  |
| Find the smallest item in a list |  |  |  |
| Sum all the elements of a list | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |

**Question 7**

What will the following code print out?

smallest\_so\_far = -1

for the\_num in [9, 41, 12, 3, 74, 15] :

if the\_num < smallest\_so\_far :

smallest\_so\_far = the\_num

print smallest\_so\_far

Hint: This is a trick question and most would say this code has a bug - so read carefully

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| -1 | Correct | 1.00 |  |
| 3 |  |  |  |
| 74 |  |  |  |
| 42 |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 8**

What is a good statement to describe the **is** operator as used in the following if statement:

if smallest is None :

smallest = value

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| The if statement is a syntax error |  |  |  |
| Is true if the **smallest** variable is not defined |  |  |  |
| matches both type and value | Correct | 1.00 |  |
| Looks up 'None' in the **smallest** variable if it is a string |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question Explanation**The **is** operator is stronger than the equality operator (==) as it insists on matching the two values exactly including type. This simple example shows the difference:

>>> 1.0 == 1

True

>>> 1.0 is 1

False

While 1.0 is the same *value* after the integer 1 is converted to floating point, the **is** operator does no conversion and so the two values do not match. The**is** operator is best used on small constant values like small integers, True, False, and None. The **is** operator should not be used with large numeric values or strings - these values should be compared with the == operator.

**Question 9**

Which reserved word indicates the start of an "indefinite" loop in Python?

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| for |  |  |  |
| break |  |  |  |
| while | Correct | 1.00 |  |
| indef |  |  |  |
| def |  |  |  |
| Total |  | 1.00 / 1.00 |  |

**Question 10**

How many times will the body of the following loop be executed?

n = 0

while n > 0 :

print 'Lather'

print 'Rinse'

print 'Dry off!'

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| 0 | Correct | 1.00 |  |
| 5 |  |  |  |
| This in an infinite loop |  |  |  |
| 1 |  |  |  |
| Total |  | 1.00 / 1.00 |  |

Bottom of Form