An Introduction To Interactive Programing In Python (Part 2)

by Joe Warren, Scott Rixner, John Greiner, Stephen Wong

Quiz 5a – Mouse inputs and more lists *Skanda S Bharadwaj*

Question 1

What is the type of the parameter for a mouseclick handler?

Refer to the CodeSkulptor documentation.

Your Answer		Score	Explanation
There is no parameter.			
String			
✓ Tuple	Correct	10.00	More specifically, a tuple (pair) of non-negative integers
Boolean			

Number	
List	
Total	10.00 / 10.00

Question 2

Which of the following expressions mutate, i.e., change, list `my_list`? If you've forgotten what the operations do, you can look in the CodeSkulptor documentation.

Your Answer		Score	Explanation
✓ `my_list.reverse()`	Correct	3.00	
`another_list.extend(my_list)`	Correct	0.50	This mutates `another_list`, not `my_list`.
`my_list + [10, 20]`	Correct	0.50	

✓	`my_list.append(10)`	Correct	3.00
✓	`my_list.extend([10, 20])`	Correct	3.00
Total			10.00 / 10.00

Question 3

We want to remove the element at the front of a list. For example, we want the following code to print `"apple"` and `["pear", "blueberry"]`, respectively. What function or method call should replace the question marks?

```
fruits = ["apple", "pear", "blueberry"]

fruit = ???

print fruit, fruits
```

Your Answer	Score	Explanation
`fruits.pop()`		
`fruits.remove(0)`		

✓ `fruits.pop(0)` Correct	10.00	
`fruits[0]`		
`fruits.remove("apple")`		
`fruits[1:]`		
Total	10.00 / 10.00	

Question 4

Which of the following uses of `range()` will generate the list `[2, 5, 8, 11, 14]`?

First, think about what each of these returns, but also try each in CodeSkulptor.

Your Answer	Score	Explanation
✓ `range(2, 17, 3)` Correct	7.78	

`range(2, 14, 3)`	Correct	1.11	
`range(2, 15) * 3`	Correct	1.11	
Total		10.00 / 10.00	

Question 5

To correctly compute the product of a list `numbers` of numbers, what statement should replace the question marks?

```
numbers = …

???

for n in numbers:

    product *= n
```

Your Answer		Score	Explanation
`product = []`			
✓ `product = 1`	Correct	10.00	

`product = 0`

`product = numbers[0]`

`product = numbers[1]`

Total 10.00 / 10.00

Question 6

We can loop over strings, too!

The following incomplete function is a simple, but inefficient, way to reverse a string. What line of code needs to replace the questions marks for the code to work correctly?

```
def reverse_string(s):
    """Returns the reversal of the given string."""
    ???
    for char in s:
        result = char + result
    return result
```

```
print reverse_string("hello")
```

Your Answer		Score	Explanation
`result = 0`			
`result = []`			
✓ `result = ""`	Correct	10.00	
`result = " "`			
Total		10.00 / 10.00	

Question 7

Imagine a game on a map. At the beginning, we might want to randomly assign each player a starting point. Which of the following expressions may we use in place of the question marks to correctly implement this functionality?

```
import random
```

```
def random_point():
    """Returns a random point on a 100x100 grid."""
    return (random.randrange(100), random.randrange(100))


def starting_points(players):
    """Returns a list of random points, one for each player."""
    points = []
    for player in players:
        point = random_point()
        ???
    return points
```

Your Answer		Score	Explanation
✓ `points.append(point)`	Correct	7.50	
`points.extend(point)`	Correct	0.50	
`point.append(points)`	Correct	0.50	

`point.extend(points)`	Correct	0.50
`points + point`	Correct	0.50
`points += point`	Correct	0.50
Total		10.00 / 10.00

Question 8

The following function is supposed to check whether the given list of numbers is in ascending order. For example, we want `is_ascending([2, 6, 9, 12, 400])` to return `True`, while `is_ascending([4, 8, 2, 13])` should return `False`.

```
def is_ascending(numbers):

    """Returns whether the given list of numbers is in ascending order."""

    for i in range(len(numbers)):

        if numbers[i+1] < numbers[i]:

            return False

    return True
```

However, the function doesn't quite work. Try it on the suggested tests to verify this for yourself. The easiest fix is to make a small change to the highlighted code. What should it be replaced with?

Your Answer		Score	Explanation
`range(len(numbers - 1))`	Correct	1.00	
`range(len(numbers)) - 1`	Correct	1.00	
`range(1, len(numbers))`	Correct	1.00	
✓ `range(len(numbers) - 1)`	Correct	7.00	
Total		10.00 / 10.00	

Question Explanation

Note that here we loop not over `numbers`, but over a new list, a range of indices. This function uses that approach because it looks at **two**elements of `numbers` on each iteration.

Question 9

Turn the following English description into code:

1. Create a list with two numbers, 0 and 1, respectively.
2. For 40 times, add to the end of the list the sum of the last two numbers.

What is the last number in the list?

To test your code, if you repeat 10 times, rather than 40, your answer should be 89.

Answer for Question 9

You entered:

<table>
<tr><th>Your Answer</th><th></th><th>Score</th><th>Explanation</th></tr>
<tr><td>165580141</td><td>Correct</td><td>20.00</td><td>Correct. By the way, here's code that simplifies things a bit by not even building the list
<pre>
x = 0

y = 1

for i in range(40):

    x, y = y, x + y

print y
</pre></td></tr>
<tr><td>Total</td><td></td><td>20.00 / 20.00</td><td></td></tr>
</table>

Question Explanation

This process computes *Fibonacci numbers*. Since 0 and 1 are considered the 0th and 1st Fibonacci numbers, respectively, we effectively asked you to give the 41st Fibonacci number. While a simple and seemingly arbitrary computation, the Fibonacci sequence appears repeatedly in mathematics and nature.