

## Data Bootcamp: Code Practice #2

Revised: January 22, 2016

*Answer each of the questions below. Answers should include any code you wrote to find your answer. They can be handwritten, but they must be readable and look professional.*

1. (Review) Will this code run without errors? Why? If it does run, what is the value of `z`?

```
x = [1, 2, 3]
y = [42, 43]
z = x + y
```

2. (Review) Will this code run without errors? Why? If it does run, what is the value of `x`?

```
x = (1, 2, 3)
y = (42, 43)
z = x + y
```

3. (Review) What is the value of `z` in the code below? Is there another operation which we can use to create this same list?

```
x = [1, 2, 3]
z = x + x
```

4. (Review) What is the value of `y` after running the code below?

```
x = [1]
y = x
x.append(2)
y = y.pop()
```

5. *Review.* Does this code run without error? If so, what does it produce? If not, explain why.

```
x = [1, 2, 3]
y = [42]
z = x + y
print(z)
```

6. *Review.* Does this code run without error? If so, what does it produce? If not, explain why.

```
x = "I don't think that word means"
y = 'what you think it means'
z = x + y
print(z)
```

7. Describe and explain the type of these expressions:

```
12
12.0
'12.0'
"""12.0"""
[12]
1 >= 2
1<=2
'12.0' == """12.0"""
```

8. Imagine you have two lists, how could you determine which list was longer?

```
x = [1, 2, 3, 4]
y = ["x", "y", "z"]
if (insert code here):
    print("x is longer")
else:
    print("y is longer")
```

9. What do you expect the output of each of the following comparisons to be?

```
print(1 > 0)
print(1 >= 0)
print(1 == 0)
print(1 > 1)
print(1 >= 1)
print(1 <= 1)
print(1 == 1)
print(1.0 + 1e-18 == 1.0)
```

10. What do you expect the output of each of the following comparisons to be?

```
print("sarah" == 'sarah')
print([1, 2, 3] == [1, 2, 3])
print([0, 1, 2] == [1, 2, 3])
```

11. What do you expect the value of `x` to be after the following block of code?

```
cond = True
if cond:
    x = "Chase"
else:
    x = "Dave"
```

12. Write a short piece of code (using an if-else statement) that assigns a numerical value to a variable `x` and then checks whether the type of `x` is a float or integer. Print out which it is.
13. Does this code run without error? If so, what does it produce? If not, how would you fix it?

```
if 2>1
    print('Yes, 2 is still greater than 1')
```

14. Explain in words what slicing does?
15. What is the "index" of the last element of the array `x`?

```
x = [1, 2, 3, 4, 5]
```

16. Which index is each of the following elements in the array `x`?

```
x = [1, 2, "a", "b", "fast", "slow", "Ethan", "Hunt", 10]
```

- (a) "a"
- (b) 1
- (c) 10
- (d) "Ethan"
- (e) "Hunt"

17. What is the correct way to slice and get the first three elements of an array?
18. Pull out each word using slicing from the following sentence.

```
sentence = "This is a sentence; please slice it."
```

19. Finish writing the for loop below so that it adds all of the numbers less than 100.

```
x = 0
for i in :
    x += i
print(x)
```

20. For every element in the following list, `x`, print the element if it is a string and skip it otherwise.

```
x = [0, 1, "G", "o", 3, "o", "d", 1, "w", 10, "o", "r", "k"]
```

21. Bond example

22. Imagine that you have information that says the price of corn is going to change by 10 cents per day (only trades on week days) until it reaches \$2.50, after which it will begin to decrease by 10 cents a day. Finish the code below to print the price of corn every day until it reaches \$2.50.

```
days = ["M", "T", "W", "Th", "F"]
starting_price = 1.20
current_price = starting_price
while True:
    for day in days:
        current_price = current_price + 0.10
        print("The price on ", day, "is ", current_price)

    if current_price == 2.50:
        break
```

23. Define a function that takes your birth year and the current year and prints out every year inbetween (and including) those years.
24. A function is defined below. Determine what it does and write an example of you using it.

```
def score_to_percentage(pointsawarded, pointstotal):
    """
    This function changes a number of points awarded
    and a number of total points into a percentage
    """
    percentage = pointsawarded / pointstotal

    return percentage
```

25. *Challenging.* Create a function to convert a date tuple to a regular date: to convert (say) (2015, 8, 22) to August 22, 2015 expressed as a string. That is: The input is a tuple and the return is a string. *Suggestion:* You might want to use the list

```
months = ['January', 'February', etc]
```

26. Take list of change [quarters, dimes, nickels, pennies] and return value. *Bonus points:* Use a tuple.