

Data Bootcamp: Code Practice #2

Revised: January 25, 2016

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

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

```
x = [1, 2, 3]
y = 4
z = x + y
```

2. *Review.* For the same `x` and `y` as the previous question: What function tells us what type they are? What function tells us how many elements they contain?
3. What type is each expression? Why?

```
2
'2'
2.0
"2.0"
2>1
'Itamar' > 'Chase'
[1, 2]
(1, 2)
{1: one, 2: two}
```

4. What value does each of these comparisons have?

```
1>=0
1 >= 1
1 == 1
1 == 1.0
1 > 1
'Spencer' == "Spencer"
```

5. 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')
```

6. What is the result of this code? Why?

```
if True:
    print('on the one hand')
else:
    print('but on the other')
```

What happens if we replace `True` with `False` in the first line?

7. What is the value of `x` after the following block of code?

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

8. Suppose we have two lists, `x = [1, 2, 3, 4]` and `y = ['x', 'y', 'z']`. Adapt the code below to determine which has more elements:

```
if <insert expression>:
    print('x has more')
else:
    print('y has more')
```

9. 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. ??
10. Explain in words what slicing does.
11. How would you extract (“slice”) the first element (the integer 1) from the list `x` below? The last element? All but the first element?

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

12. Use slicing to extract each word from

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

Suggestion: Number every character in `sentence`.

13. Consider the list

```
x = [1, 2, "a", 'b', "fast", 'slow', 3, "Raghu", 'Liuren', 10]
```

- (a) How would you slice out the first item? The last item?
- (b) How would you slice out the items from `'b'` to 3 inclusive?

14. Using the same list `x`, write a loop that prints every element on a new line.
15. *Challenging.* Using the same list `x`, write a loop that prints every element of type `str`.
16. 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)
```

17. 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"]
```

18. Bond example ??
19. 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
```

20. Define a function that takes your birth year and the current year as inputs and prints every year in between (and including) those years.
21. 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
```

22. *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]
```

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

24. Dictionaries...