# (Part II )

Review Questions!

Here are some sample questions to review what you have learned at Python II course. You can find all the answers in the powerpoint slides. If you need more explanation please don't hesitate to email the course instructors or TAs and remember some of these questions are designed to make you think deeper about the code so it's completely normal to not get them right on the first try. We highly suggest to run the code in the questions. Happy studying!
**Suggestion:** If you have difficulty with a question write down your train of thought about how the code will run and what it will return. It is a great exercise and it will help when you encounter more complicated pieces of code in the future.

Section 1
Please choose the best answer for each multiple choice question.

1. Which choice is a list:
    a) my_list= 2
    b) my_list= “abc”
    c) my_list= [1, '1']
    d) my_list=””

2. What does this code return?

```
>>> sequence =[  'caa', 'ata', 'atg', 'aat']
>>> print (sequence[0])
```

    a) ata
    b) aat
    c) caa
    d) error

3. len() is a…
    a) variable
    b) list
    c) function
    d) string

4. What is the output of this code

```
x = 1
while x <= 5:
        if x==1:
            break
        print(x)
        x += 1
```

a)
```
1
2
3
4
```

b)
```
1
2
3
4
5
```

c) Nothing

Section 2

5. What is this function missing?

```
def timesTwoFour(two)
        two *= 2
        four = two * 4
        return two, four
```

6. Annotate this code to the best of your abilities after the #

```
import random #

def my_sequence(): #
        my_num= [1,2,3,4,5] #
        my_base=['G', 'T', 'C', 'A']
        print (random.choice(my_base)*random.choice(my_num)) #
```

my_sequence()#

7. What do you think the code above will return?

8. Write a function totalLen that takes a list of strings and, using a for loop, returns the sum of their lengths.
Example:
>>> totalLen(["AB", "CD", "EF"])
6
totalLen(["ABAC", "CD", "E"])
7
totalLen(["A", "", "E"])
2

9. Write a function ATGPos that takes a DNA sequence and returns a list of the positions of all possible start codons in the sequence.
(For example, if str = "CGATGCCACATGCCGCT", it should return [2,9].)

10. Using a while loop, write a function find_ith(dna_seq, base, i) that returns the position of the ith instance of the base. For example:
>>> str = "ACGTACGTACGTACGT"
>>> find_ith(str, 'A', 3)
8
Because s[8] is the third A in the sequence.
>>> find_ith(str, 'G', 2)
6