

## Python Mega Assignment # 1

- Which of the following terms are related to dictionaries?
  - value
  - item
  - index
  - key
- Just like lists, + operator is used to extend dictionaries?
  - True
  - False
- To access items from a dictionary, we specify the index of that item within [] like myDict[0]?
  - True
  - False
- When we use [] to access the value from a dictionary which does not exist in that dictionary....?
  - Value within [] is added to the dictionary
  - Value None is returned
  - New dictionary is created
  - None of above
- What does return the pop method of a dictionary?
  - list
  - tuple containing the pair of last item of the dictionary
  - dictionary
  - value of the key, if it exists in the dictionary
- What does return popitem method return?
  - dictionary
  - tuple containing the pair of last item of the dictionary
  - list
  - value of key, if it exists in the dictionary
- Which of the following 2 methods can be used to iterate through the items of a dictionary?
  - items()
  - values()
  - indexes()
  - keys()
- Which one of the following is used to enclose a dictionary?
  - () parenthesis
  - { } curly brackets
  - [] square brackets
  - "" quotation marks
- Write Python Program add key-value pair in dictionary and check if a Given Key or Value or Both Exists in a Dictionary or Not.
- Write a Python Program to Count the Frequency of Words Appearing in a String Using a Dictionary and print only the words having Even (divisible by 2) frequency.
- X = ["Feb", Apr, Mar, May, Jun, Jul, Aug, Jan]. What will be output of following?  
X[0:3]  
X[2:8]

```
X[4:9]
X[1:7:2]
X[-1:-7]
X[-7:7]
X[-1:-8:-2]
X[:4]
```

12. Remove the correct number from the list X

```
X = [ 9,2,8,4,5]
```

```
X__?__
```

```
print (X)
```

Output: [2,8,4,5]

1) .delete(9)

2) .rm(9)

3) .remove(9)

13. p = 3

```
q = 'hello! '
```

```
print( q __?__ p)
```

hello! hello! hello!

1) \*

2) \*\*

3) +

14. y = "this is a random sentence"

```
print (y__?__)
```

Output: THIS IS A RANDOM SENTENCE

1) .upper()

2) .upcase()

3) .capitalize()

15. p = True

```
q = 'True'
```

`r = 2`

`r = 2.0`

`print(type(p))`

`print(type(q))`

`print(type(r))`

`print(type(s))`

16. What are the optional arguments to the function?

`function_1(R1, q, p=None, R2= None)`

1) q and R2

2) p and R2

3) p and R1

4) R1 and q

17. Which command invokes method X() of the object p?

1) X(p)

2) p\$x()

3) X().p

4) p.x()

18. X=4 , Y= 2

`print(X % Y)`

`print(X / Y)`

`print(X // Y)`

`print(Y % X)`

19. x = [[4, 1, 1], [5, 9, 0]]

for i in \_\_?\_\_:

```
for j in __?__:
```

```
    ?
```

```
Output:      4
```

```
            1
```

```
            1
```

```
            5
```

```
            9
```

```
            0
```

```
x = [[4, 1, 1], [5, 9, 0]]
```

```
for i in __?__:
```

```
    for j in __?__:
```

```
        ?
```

```
Output:      4      5
```

```
            1      9
```

```
            1      0
```

```
x = [[4, 1, 1], [5, 9, 0]]
```

```
for i in __?__:
```

```
    for j in __?__:
```

```
        ?
```

```
Output:      4      1      1      5      9      0
```

```
x = [[4, 1, 1], [5, 9, 0]]
```

```
for i in __?__:
```

```
    for j in __?__:
```

```
        ?
```

```
Output:      4      1      1
```

```
            5      9      0
```

```
20. q = [10.62, 16.14, 6.45, 17.11]
```

```
for __?__, z in enumerate(q) :
```

```
print( 'Item ' + str( j ) + ' - ', str ( z ))
```

Output:           Item 0 – 10.62  
                  Item 1 – 16.14  
                  Item 2 – 6.45  
                  Item 3 – 17.11

- 1) z
- 2) i
- 3) j
- 4) x
- 5) k
- 6) y

21. Which of these about a dictionary is false?

- a) The values of a dictionary can be accessed using keys
- b) The keys of a dictionary can be accessed using values
- c) Dictionaries aren't ordered
- d) Dictionaries are mutable

22. What is the output of the following:

```
D = dict()
```

```
for i in range (3):
```

```
    for j in range(2):
```

```
        D[i] = j
```

- a. {0: 1, 1: 1, 2: 1}
- b. {1: 0, 1: 1, 1: 2}
- c. {0: 1, 1: 2, 2: 3}
- d. {1: 2, 1: 1, 1: 0}

23. You are writing a function that increments player score in a soccer game

If no value is specified for points, then point must start with 1

If no value is specified for bonus, then bonus should be True

**01 def increment\_score ( bonus , score , points ):**

To meet the first requirement line 01 must be change to

- ☐ def increment\_score ( bonus , score , points = 1 ): (True or False)

To meet the second requirement line 01 must be change to

- ☐ def increment\_score ( bonus = True , score , points = 1 ): (True or False)
- ☐ Once a parameter is defined with default value, any parameter to the right must also be defined with default values (True or False)

24. What will be output?

```
def avg ( x , y , z = 50 ):  
    adding = x + y + z  
    avg_value = adding / 3  
    return avg_value  
  
y = avg ( x = 5 , y = 9 , z = 20 )  
  
print(y)
```

25. What will be output? Describe it with reason and logic behind. Do multiple experiments with arguments / parameters to remove error, if occurs.

```
def avg ( *opt_values , name ):  
  
    avg_value = sum (opt_values) / len(opt_values)  
    print('name is: ' + name + 'Marks: ' + str(avg_value))  
  
avg ( 5 , 9 , 20, 34, 87, 112 , 'Ali' )
```

26. Final output is not required. Just take copy pencil, think and write the output of each line, write down the link between parameters and arguments. Remove one or two \*\* from other\_info and observe the output.

```
def display_result(winner, score, **other_info):  
    print("The winner was " + winner)  
    print("The score was " + score)  
  
display_result(winner="Manchester", score="1-0", overtime ="yes", injuries="none")
```

27. The position of parameters and arguments is re-arranged. Just think and find the logic behind output or error.

```
def display_result(winner, **other_info, score):
    print("The winner was " + winner)
    print("The score was " + score)

display_result(winner="Manchester", overtime="yes", injuries="none", score="1-0")
```

28. What will be the output of the following Python expression if X=123?

```
print("%06d"%X)
```

a) 123000  
b) 000123  
c) 000000123  
d) 123000000

29. What will be the output of the following Python expression if x=22.19?

```
print("%5.2f"%x)
```

a) 22.1900  
b) 22.00000  
c) 22.19  
d) 22.20

30. What will be the output of the following Python code?

```
'{0:f}, {1:2f}, {2:05.2f}'.format(1.23456, 1.23456, 1.23456)
```

a) Error  
b) '1.234560, 1.22345, 1.23'  
c) No output  
d) '1.234560, 1.234560, 01.23'

31. Write down the output of each line after each iterations. Do multiple experiments to change values

i = 1

while False:

if i%2 == 0:

break

print(i)

i += 2

32. Write down the output of each line after each iterations. Do multiple experiments to change values

```
x = "abcdef"
```

```
i = "a"
```

```
while i in x:
```

```
    x = x[:-1]
```

```
    print(i, end = " ")
```

33. Write down the output of each line after each iterations. Do multiple experiments to change values

```
for i in ".join(reversed(list('abcd'))):
```

```
    print (i)
```

34. Flow of the program. Write the output of each line after every iteration of 'i'

```
for i in range(10):
```

```
    if i == 5:
```

```
        break
```

```
    else:
```

```
        print(i)
```

```
else:
```

```
    print("Here")
```

35. What is the output? And understand the functionality of lambda function

```
y = 6
```

```
z = lambda x: x * y
```

```
print z(8)
```

36. Write output and give proper logic of whatever the output comes.

```
i=0
```

```
def change(i):
```

```
    i=i+1
```

```
    return i
```

```
change(1)
```

```
print(i)
```



40. What will be output? Define this output clearly

```
def change(one, *two):
```

```
    print(type(two))
```

```
    print(two)
```

```
change(1,2,3,4)
```

41. What will be output? Define this output clearly

```
def find(a, **b):
```

```
    print(type(b))
```

```
find('letters',A='1',B='2')
```

42. Write output and define each line's output for each iteration of 'i'

```
def foo(i, x=[]):
```

```
    x.append(i)
```

```
    return x
```

```
for i in range(3):
```

```
    print(foo(i))
```

43. Evaluate the following Python arithmetic expression: and write which segment will execute first?  
(Brackets, Exponents, Multiplication, Addition / Subtraction, Left to right rule)

$$(3*(1+2)**2 - (2**2)*3)$$

44. You are creating a function that manipulates a number. The function has the following requirements:

- A float is passed into the function
- The function must take the absolute value of the float
- Any decimal points after the integer must be removed

A. `math.fmod(x)`

B. `math.frexp(x)`

C. `math.floor(x)`

D. `math.ceil(x)`

E. `math.fabs(x)`

45. You are writing code that generates a random integer with a minimum value of 5 and a maximum value of 11.

Which two functions should you use? Each correct answer presents a complete solution. (Choose two.)

- A. `random.randint(5, 12)`
- B. `random.randint(5, 11)`
- C. `random.randrange(5, 12, 1)`
- D. `random.randrange(5, 11, 1)`

**46. Write a program that receives marks from user and check the grade.**

Marks greater than equal to 90 then A grade

Marks between 80 to 90, B grade

Marks between 70 to 80, C grade

Marks between 60 to 70, D grade

Marks less than equal to 60 then E grade