Name :- mulla suhan Jameer
college Name :- DACOE, Karad
Batch :- D9
class assessment :1

Date:- 27-12-2023

Page No. □ Date □

25/30

Q.1) Explain the difference bet^n a list and a tuple in python. provide an example for each.

→

List	Tuple
i) It is a mutable	i) It is immutable.
ii) The implication of iterations is time consuming in the list	ii) The implication of the iterations are much faster in tuples.
iii) operations like insertion & deletion are better performed	iii) Elements can be a accessed better.
iv) consumes the more memory	iv) consumes the less memory
v) many built-in methods are available	v) Does not have many built-in methods.
vi) unexpected errors & changes can easily occur in lists.	vi) unexpected errors & changes can rarely occur in tuples.
vii) List syntax - num_list = [1,2,3,4,5] print (num-list)	vii) tuple syntax - num_tuple = (1,2,3,4,5) print (num_tuple)
alphabet-list = ['a','b','c','d'] print (alphabet_list)	alphabet_tuple = ('a','b','c'] print (alphabet_tuple)
mixed-list = ['a',1,'b',2,'c'] print (mixed_list)	mixed_tuple = ('a',1,'b',2) print (mixed_tuple)

7

nested-list = [1,2,3[4,5],6] print (nested-list)	nested-tuple = (1,2,3,(4,5),6) print (nested-tuple)
list-numbers = [1,2,3,4,5] print (list-numbers)	tuple-numbers = (1,2,3,4,5) print (tuple-numbers)
type (list-numbers)	type (tuple-numbers)
viii) Example of list :-	viii) Example of tuple :-
list-num = [1,2,3,4,5] print (list-num)	tuple-num = (1,2,3,4,5) print (tuple-num)

Q.2) Describe the purpose of set data type in python. provide an example to illustrate its use

→ Set :-

i) set are used to store multiple items in a single variable.

ii) set is one of 4 built-in data types in python used to store collections of data, the other 3 are list, tuple & dictionary, all with different qualities & usage

iii) A set is a collection which is unordered, unchangeable *, & unindexed.

iv) sets are written with the curly brackets e.g. create set

```
s1= {"suhan", "khushabu", "mumtaj", "jameer"}
print (s1)
```

2

- set data type in python :-

a set is an unordered collection of data type that is iterable, mutable, & has no duplicate elements.

e.g.

```
s2 = {1, 2, 3, 4, 5}
print (s2)

s3 = {True, False, False}
print (s3)
```

Q.3) what is the key difference betn a float & an integer data type in python? Give an example where using a float would be more appropriate.

→ • Integer (int) :- This data type is used to represent whole numbers without any decimal points

e.g. '1', '42', '-10' are integers.

2

• Float (float) :- This data type is used to represent numbers with decimal points or numbers in scientific notation.

e.g. '3.14', '2.0', '-0.5' are floats.

Here's an example where using a float would be more appropriate.

```
#using float for calculations involving division.
    result = 10/3
    print (result)          # output : 3.333333333335
```

In this example, if you use integer division ('//'), the result would be truncated to the nearest integer, losing the fractional part. However, by using a float, you can retain the decimal precision, which is often necessary in calculations that involve division

```
# using integer division.
    result-integer = 10//3
    print (result-integer)    # output : 3
```

if precision matters & you want to capture the fractional part of the division, using a float is more appropriate.

Q.4) How does the dictionary data type in python differ from lists & tuples ? provide an example of a dictionary & explain its a structure.

Page No.
Date

→ 1) lists :-

- lists are ordered collections of items
- elements in a list are accessed by their index.
- lists are mutable, meaning you can change, add, or remove elements after the lists is created.

```
e.g. my-list = [1, 2, 3, 'suhan', 'khushabu']
     print (my-list)
```

2) Tuples :-

- Tuples are ordered collections of items, similar to lists.
- elements in a tuple are accessed by their index.
- Tuples are immutable, meaning their values cannot be changed after creation

```
e.g. my-tuple(1, 2, 3, 'suhan', 'khushabu')
     print (my-tuple)
```

3) Dictionaries :-

- Dictionaries are unordered collections of the items.
- Instead of indexing, dictionaries use keys to access values.
- Dictionaries are mutable.

- Each item in a dictionary is key-value pair, where the key and value are separated by colon (':')

e.g. my-dict = {'name': 'suhan', 'age': '21' }
    print (my-dict} ['age'])

In the dictionary e.g. 'name', 'age' are keys & 'suhan', '21' are their corresponding values.
The structure of dictionary is
{key1: value1, key2: value2, ---}

Q.5) What is doc string & use of this string in python?

→ - The docstring is enclosed in triple-double-quotes (""" ---- """).
- It provides brief description of function's purpose.
- It includes a "parameters" section listening the input parameters with their types & descriptions.
- It includes a "Returns" section

```
def calculate-square(x):
"""
 This fun^n calculates the square of given no.

 parameters:
```

```
-x (int): The no. to be squared.

Returns:
int: The square of the input no.
"""

return x ** 2
```

- help (calculate_square)

print(calculate_square.__doc__)

Q.6) Explain purpose of // operator in python. provide an example to illustrate its use.

→ The '//' operator in python is the floor division operator. It performs division & returns the largest integer that is less than or equal to the result.

e.g.
```
result = 10 // 3
print (result)
```

Q.7) Differentiate betn == & is operators in python provide examples to demonstrate their usage.

→ 1) '==' operator :-

- The "==" operator is used for value equality comparison.
- 1 It checks if the values of the two operands are equal.

e.g.

```
a = [1,2,3]
b = [1,2,3]

result = a ==b
print (result)
```

2) "is" operator :-

- The 'is' operator used to identity comparison.
- It checks if the two operands refer to the same object in memory.

e.g.

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

result1 = x is y
print ( result1)

result2 = x is z
print (result 2)
```

Q.8) what is the use of += operator in python? provide an e.g. to demonstrate its functionality

→ The "+=" operator in python is a shorthand assignment operator that performs addition & assignment in a single step.

e.g.

```
1) x=5
   x += 3          # equivalent x=x+3
   print (x)       # output : 8
```

2

```
2) word = "Hello"
   word += "world"      # equivalent word=word+world
   print (word)         # output : Hello world.
```

Q.9) Discuss the role of the 'in' operator in a python. provide an e.g. how it can be used.

→ The 'in' ~~tun~~ operator in python is used to check it a specified value is present in a sequence, such as string, list, tuple or set.
It returns a Boolean value ('True' or 'False')

e.g. check it an element present in a list.

```
1) fruits = ['apple', 'banana', 'grape', 'orange']
   print ('banana' in fruits)     # output: True
   print ('mango' in fruits)      # output : False
```

2

2) check it character is present in string

```
word = 'python'
print ('y' in python)     # output true
```

3) check if key is present in dictionary.

```
person = {'name': 'suhan', 'age': 21 }
print ('gender' in person) # output : False
```

Q.10) explain the concept of ternary operator (x if condition else y) in python. provide an example scenario where it can be employed

→ syntax of ternary operator
x if condition else y

e.g.

```
# determine ~~fw~~ if a number is even or odd.
num = 10

# using traditional if-else statement
if num % 2 == 0:
    result = " ven"
else:
    result = "odd"
print (result)          # output : even

# using ternary operator :-
result = "even" if num % 2 ==0 else "odd"
print (result)     # output : even
```

Q.11) what is the purpose of the if statement in python? provide an example demonstrating the use of an if statement.

→ The 'if' statement in python is used for conditional execution of code. It allows you to specify the block of code that will be executed only if a certain condition is true if the condition is false the block of code is skipped

e.g 1

```
number = -5
if number > 0:
    print ("The number is positive")
else:
    print ("The number is negative")
```

Q.12) Difference betn while & for loops in python Give an example of each loop

→ In python, both 'while' & 'for' loop structures that allow use to repeatedly execute the block of code, but they have different use cases & structures.

1) 'while' loop :-

- The 'while' loop is used to repeatedly execute a block of code as long as specified condition is true

2

In this given example, the 'while' loop prints numbers from 1 to 5. The loop continues executing as long as 'count' is less than / equal to 5 e.g # print numbers from 1 to 5 using while loop.

```
count = 1
while count <= 5:
    print ( count)
    count += 1
```

2) 'for' loop :-

- The for loop is used to iterate over a sequence (such as list, tuple, string, / range) or other iterable objects

e.g. # print elements in a list using for loop.

```
fruits = ['apple', 'banana', 'orange', 'grape']
for fruit in fruits:
    print (fruit)
```

Q.13) Explain the significance of the break statement in python provide a scenario where using break is appropriate.

→ The 'break' statement in python is used to exit a loop prematurely, regardless of whether the loop's condition is still true.

e.g. Find the 1st occurrence of number in a list

```
numbers = [1, 3, 5, 7, 9, 11, 13]
```

```
search_number = 11
for num in numbers:
    if num == search_number:
        print (f"Number {search_number} found!")
        break
else:
    print (f"Number {search_number} not found in
                                        the list.")
```

9

Q.14 Discuss the role of the continue statement in python. provide a code snippet demonstrating its use.

→ The 'continue' statement in python is used to skip the rest of the code inside a loop for the current iteration & proceed to the next iteration.

e.g.

```
# skip printing even numbers in a list
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for num in numbers:
    if num % 2 = = 0:
        continue
    print(num)
```

- The 'for' loop iterates through the elements in the 'numbers' list.

Q.18) How does the else clause in a loop contribute to the control flow in python? provide an e.g. illustrating the use of the else clause in a loop.

→ In python, the 'else' clause in a loop provides a block of code that is executed when the loop condition becomes false.

for else general syntax

```
for variable in iterable:
else:
```

e.g.

```
# check if number is a prime
num = 11

for i in range(2, num):
    if num % i == 0:
        print(f"{num} is not prime number")
        break
else:
    print(f"{num} is a prime number")
```

In this example, the 'for' loop iterates through the range of numbers from 2 to 'num -1'