Name : Jadhav Romesha Adhikrao.
College Name : DACOE, Karad.
Batch : D9

24/30

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

List	Tuple
1) It is mutable.	1) It is immutable.
2) The implication of iterat^n is time consuming in list.	2) The implication of iterat^n are much transfer in tuples.
3) Applications like insertion & delecation are better performed.	3) Elements can be a accessed better.
4) It consumes more memory.	4) It consumes less memory.
5) Many built-in methods are available.	5) Does not have many built-in methods.
6) Unexpected errors & changes are can be easily occurs in list.	6) Un-expected errors & changes can rarely occur in tuples.
7) List syntax -	7) tuple syntax.
num_list = [1,2,3,4,5]	num_tuple = (1,2,3,4,5)
print (num_list).	print (tuple num_tuple)
alphabet_list = ['a','b','c','d']	alphabet_tuple ('a','b','c','d')
print (alphabet_list)	print (alphabet_tuple)
mixed_list = ['a',1,'b',2,'c']	mixed_tuple = ('a',1,'b',2)
print (mixed_list)	print (mixed_tuple)
nested_list = [1,2,3,[4,5],6]	nested_tuple = (1,2,3,(4,5),6)
print (nested_list)	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 (list_numbers)
8) Example of list	8) Example of tuple
list_num = [1,2,3,4,5] print (list_num)	tuple_num=(1,2,3,4,5) print (tuple_num)

Q2. Describe the purpose of set data type in python. provide an example to illustrate its use.

→

set :-

1) sets are used to store multiple items in single variable.

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

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

4) sets are written with the curly brackets.

eg. Create, set.

S1 = {"Romesha", "Aniket", "Aryansh", "Riya"}

• Set data type in python :-
A set is a unordered collection of data types that is iterable, mutable, & has no duplicate elements

eg $S_2 = \{1, 2, 3, 4, 5\}$
print (S_2)
$S_3 = \{$ True, false, false$\}$
print (S_3)

Q.3 What is the key difference bet^n a Float & an integers data type in python ? Give an example where using a float would be more appropriate.

→ • Integer (int) :- It is used to represent whole numbers without any decimal points.
eg. '1', '42', '-10', are integers.

• float (float) :- It is used to represent no. with decimal points or no. in scientific notation.
eg. '3.14', '2.0', '-0.5', '0.6' are float.

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

```
# using float for calculations involving division
result = 10/3
print (result)    # output : 3.3333333
```

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) #o/p = 3.
```

if precision matters & you wan 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 structure.

→ 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.

eg. `my_list = [1, 2, 3, "Ramesha", "Aniket"]`
`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
not be
eg. m

- Tuples are immutable, meaning their values cannot be changed after creation.
eg my-tuple (1,2,3, 'Romesha', 'Aniket')
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 each dictionary is key-value pair, where the key & value are separated by colon (:)
eg. my-duct = {'name' : 'Romesha', 'age' : '21'}
print (my-duct ['age'])

In the dictionary eg. 'name', 'age' are keys & ~~sultan~~ 'Romesha', '21' are there corresponding values
The structure of dictionary is
{key 1 : value1, key 2 : value 2, ...}

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

→ • The docstring is enclosed in triple - double quotes (""" ... """).
• It provides brief funct^n descript^n of functions purpose.
• It includes a "parameters" section listening the input parameters with their types & descriptions. It includes a 'return' section.

```
def calculate_square(x):
    """
    This f^n calculates the square of given n.
    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 ~~values~~ use.

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

eg. result = 10//3.
print (result)

Q.7 Differentiate bet^n == & is operators in python provide examples to demonstrate their usages.

1) '==' operator :-

- The operator is used for value equality comparision.
- It checks if the values of 2 operands are equal

eg. a = [1, 2, 3]
b = [1, 2, 3].
result = a == b
print (result)

2) "is" operator :-
- The 'is' operator used to identify comparison
- It checks if the 2 operands refer to the same object in memory

eg.
x = [1, 2, 3]
y = [1, 2, 3]
z = x.

result 1 = x is y
print (result 1)
result 2 = x is z
print (result2)

Q.8. What is the use of t= operator in python? provide an eg. to ~~determ~~ demonstrate its functionality

→ The "t=" operator in python is the short-hand assignment operator that performs addition & assignment in a single step

eg. 1) x = 5
x t = 3 # equivalent x = x+3
print (x) # output : 8
2) word = "Hello"

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

Q.9 Discuss the rate of the 'in' operator in python, provide an eg. how it can be used.

→ The in operator in python is used to check if a specified value is present in a sequence such as string, list, tuple, or set.
It returns a Boolean value ('True' or 'False')
eg.
Check if an element person1 in a list

1) fruits = ('apple', 'banana', 'grape', 'orange')
print ('banana' in fruits) # output : True
print ('mango' in fruits)   # output : false.

2) check if character is present in string.
word = 'python'
print ('y' in python) # output : true

3) check if key is present in dictionary
person = {'name': 'Romesha', 'age': 21}
print ('gender' in person) # o/p : 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.

eg # determine 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) # o/p : even.

using ternary operator.

result = "even" if num % 2 == 0. else "odd".

print (result) # o/p : 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 exicution 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.

eg. number = -5.

if number > 0:

print ("The number is positive")

```
else:
    print ("The number is negative")
```

Q.12 Differentiate betⁿ while & for loop in python Give an example for each loop.

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

1) 'While' loop :-

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

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.

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.

eg. find the 1st occurance 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.")
```

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.

eg. # 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.15 How does the else clause in a loop contribute to the control flow in python? provide an eg. 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 interable:
else
```

eg.

```
# check if number is 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 eg. the 'For' loop iterates through the range of numbers from 2 to 'num-1'.