GATE

CRASH COURSE

Data Science & AI

Subject

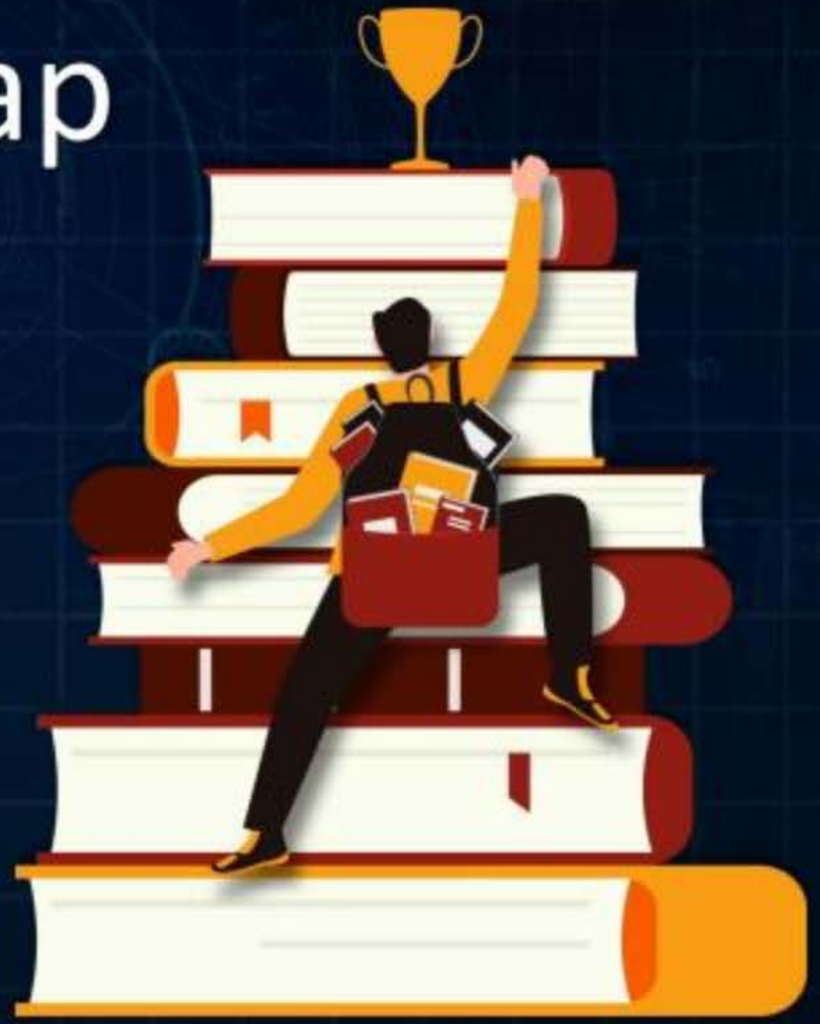
Python - For Data Science Tuple,
Sets and Dictionaries
Lec No. 05

By – Satya Sir

Last Class

Quick Recap

1. Homework Question Solution
2. Iterative Control Statements
3. Unconditional Statements
4. Strings and Lists
5. Examples



Topics to be covered

1. Homework Questions Solution
2. Nested Lists
3. String Methods
4. Tuples
5. Sets and Examples

Homework Question - 1

```
for i in range(1,7,2):
    for j in range(i):
        print(i+j)
        if j%2==1:
            break
```

The number of times print statement is executed is 5

i=1 → j=0 Print 1 0%2==1 False

i=3 → j=0 Print 3 | j=1 Print 4 j%2==1 True | j=2 X | j=3 X

i=5 → j=0 Print 5 | j=1 Print 6 1%2==1 True | j=2 X | j=3 X | j=4 X

o/p: 1 3 4 5 6

Homework Question - 2

```
for i in range(1,7,2):
    for j in range(i):
        print(i+j)
        if j%2==1:
            continue
```

The number of times print statement is executed is 9

i=1 : j=0 → Print 1

i=3 : j=0 → Print 3 ; 1 → Print 4 ; 2 → Print 5

i=5 : j=0 → Print 5 ; 1 → Print 6 ; 2 → Print 7 ; 3 → Print 8 ; 4 → Print 9

o/p:

1
3
4
5
5
6
7
8
9

String Methods

Method	Description
capitalize()	Converts the first character to upper case
casefold()	Converts string into lower case
center()	Returns a centered string
count()	Returns the number of times a specified value occurs in a string
encode()	Returns an encoded version of the string
endswith()	Returns true if the string ends with the specified value
expandtabs()	Sets the tab size of the string
find()	Searches the string for a specified value and returns the position of where it was found
format()	Formats specified values in a string
format_map()	Formats specified values in a string
index()	Searches the string for a specified value and returns the position of where it was found
isalnum()	Returns True if all characters in the string are alphanumeric
isalpha()	Returns True if all characters in the string are in the alphabet
isascii()	Returns True if all characters in the string are ascii characters
isdecimal()	Returns True if all characters in the string are decimals
isdigit()	Returns True if all characters in the string are digits

String Methods

Method	Description
isidentifier()	Returns True if the string is an identifier
islower()	Returns True if all characters in the string are lower case
isnumeric()	Returns True if all characters in the string are numeric
isprintable()	Returns True if all characters in the string are printable
isspace()	Returns True if all characters in the string are whitespaces
istitle()	Returns True if the string follows the rules of a title
isupper()	Returns True if all characters in the string are upper case
join()	Converts the elements of an iterable into a string
ljust()	Returns a left justified version of the string
lower()	Converts a string into lower case

String Methods

Method	Description
lstrip()	Returns a left trim version of the string
maketrans()	Returns a translation table to be used in translations
partition()	Returns a tuple where the string is parted into three parts
replace()	Returns a string where a specified value is replaced with a specified value
rfind()	Searches the string for a specified value and returns the last position of where it was found
rindex()	Searches the string for a specified value and returns the last position of where it was found
rjust()	Returns a right justified version of the string
rpartition()	Returns a tuple where the string is parted into three parts
rsplit()	Splits the string at the specified separator, and returns a list
rstrip()	Returns a right trim version of the string

String Methods

Method	Description
split()	Splits the string at the specified separator, and returns a list
splitlines()	Splits the string at line breaks and returns a list
startswith()	Returns true if the string starts with the specified value
strip()	Returns a trimmed version of the string
swapcase()	Swaps cases, lower case becomes upper case and vice versa
title()	Converts the first character of each word to upper case
translate()	Returns a translated string
upper()	Converts a string into upper case
zfill()	Fills the string with a specified number of 0 values at the beginning
split()	Splits the string at the specified separator, and returns a list

Strings

```
S = 'Gate'

Print(S.lower())      # gate
Print(S.upper())      # GATE
Print(S.isalpha())    # True
Print(S.isNumeric())  # False
Print(S.isascii())    # True
Print(S.islower())    # False
Print(S.isupper())    # False
```

```
Print(S.center(10))

  Gate
--------

Print(S.center(10,'#'))

###Gate###
```

Nested Lists

Add Elements to the list

```
l = [11, 22]
```

11	22

```
l.append(33)
```

11	22	33

```
l.append(55)
```

11	22	33	55

```
l.append(44)
```

11	22	33	55	44

```
print(l)      # [11, 22, 33, 55, 44]
```

append() method add elements by default at the end of the list

```
l = [10, 20, 30, 40]
```

Index value

```
l.insert(3, 5)
l.insert(-2, 4)
l.insert(-1, -3)
l.insert(2, 3)
print(l)
```

0	1	2	3
10	20	30	40
-4	-3	-2	-1

0	1	2	3	4
10	20	30	5	40
-5	-4	-3	-2	-1

0	1	2	3	4	5
10	20	30	4	5	40
-6	-5	-4	-3	-2	-1

0	1	2	3	4	5	6
10	20	30	4	5	-3	40

10	20	3	30	4	5	-3	40

Question

Remove | Delete Elements from the lists

remove(), Pop(), clear(), del keyword

Ex:

```
l = [1, 5, 7, 9, 5, 3, 7, 1, 10]

l.remove(5)  # First Occurance of given Element gets removed.   l = [1,7,9,5,3,7,1,10]

l.Pop()  # It removes last Element if index Not Specified.      l = [1,7,9,5,3,7,1]

l.Pop(4) # removes value of specified index                     l = [1,7,9,5,7,1]

del l[2] # l = [1,7,5,7,1]

l.clear() # It clears all content but list remains. # l = []

del l # It removes complete list from Memory
```

Question

- Arrays are implemented using lists in Python.
- So, multi-dimensional array is implemented in the form of Nested lists.

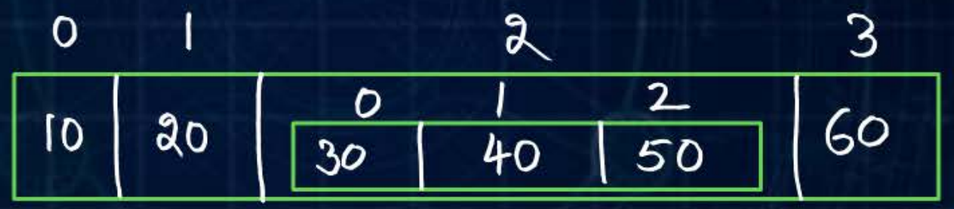
Examples :

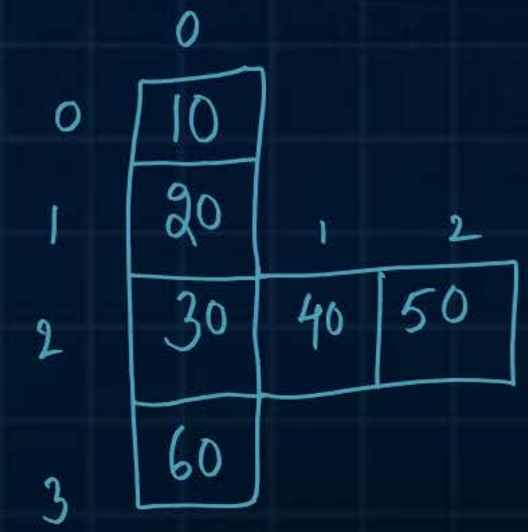
```
x = [10, 20, [30, 40, 50], 60]

Print( len(x)) # 4

Print( len(x[2])) # 3

Print (x[2][1]) #4
```





Question

$A = [1, 3, [5, 7, 9, [2, [6, 8, -1, [0, 3, 4], 5], 7, 9], 1, 8], 2, 6, 3]$

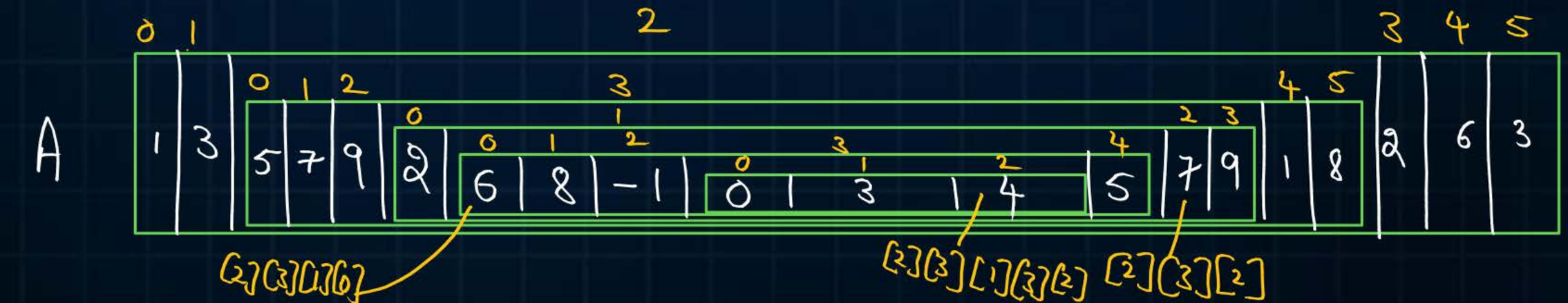
$i = A[2][3][2]$ # $i = 7$

$j = A[2][3][1][3][2]$ # $j = 4$

$k = A[2][3][1][0]$ # $k = 6$

```
Print( i+j+k)
```

o/p: $7 + 4 + 6 = 17$



Modify List Elements

Ex:

```
l = [1, 3, 5, 4, 6, 2, 8]

l[3] = 7        # l = [1, 3, 5, 7, 6, 2, 8]

l[2:5] = [-3, 10, 23]   # l = [1, 3, -3, 10, 23, 2, 8]

l[1:2] = [10, 15, 30]   # l = [1, 10, 15, 30, -3, 10, 23, 2, 8]

l[3:7] = [25, 50]       # l = [1, 10, 15, 25, 50, 10, 23, 2, 8]

Print(l)    # [1, 10, 15, 25, 50, 10, 23, 2, 8]
```

Tuples

- Ordered Collection
- Supports both +ve/-ve index
- Supports duplicate values
- Supports different type of data
- Immutable Collection

- Tuples can be Nested as like lists
- Tuples can be Modified as below:
  1) Convert tuple to list type
  2) Modify
  3) Convert list back to tuple type
- But, clear() and del keyword are valid to apply on tuples.

- Tuples can be Created as: object = (values) (or) object = tuple()

Ex: t = (10, 10, 20, 'GATE', 4.79)

t = tuple()

Unpacking Tuples

Ex: 1

```
t = (10, 20, 30)
(a, b, c) = t   # a=10  b=20  c=30
```

Ex: 2

```
t = (10, 20, 30, 40, 50, 60)
a, b, *c = t   # a=10  b=20  c=[30, 40, 50, 60]
```

Ex: 3

```
t = (10, 20, 30, 40, 50, 60)
a, *b, c = t   # a=10  b=[20, 30, 40, 50]  c=60
```

Ex: 4

```
t = (10, 20, 30, 40, 50, 60)
*a, b, c = t
# a=[10, 20, 30, 40]  b=50  c=60
```

Question

Methods of tuples

- index( ) → returns first occurance of given Element in tuple
- count ( ) → returns frequency of Element in the tuple.

Ex:

```
L = (10, 20, 30, 10, 40, 20, 10, 50)

i = L.index(20)    # i = 1

j = L.count(10)    # j = 3

Print(j - i)       # 3-1 = 2
```

Sets in Python

- It is unordered collection (Index is Not supported)
- It does not support duplicates
- It is Immutable* [add items, remove items]
- It can be created as:

  Object = { values } (or) Object = set( )

- Empty set can be created through Constructor only.

Ex:

→ S = {10,20,10,20,10}

print(S) # {10,20} (or) {20,10}

Print(type(S)) # <class, 'set'>

Ex: S = { }

Print(type(S)) # <class, 'dict'>

Ex: S = set( )

Print(type(S)) # <class, 'set'>

Sets in Python

methods	Equivalent Operators
. union( )	|
. intersection( )	&
. difference ( )	-
. Symmetric_difference( )	^
union_update( )	|=
intersection_update( )	&=
difference_update( )	-=
:	

To be Contd ...

Question

o/p = _____

Homework

count = 1

x = [1, 3, 5, [7, 9, [2, 4], 6], 8]

y = (5, 7, (9, 3, 1, (6, 2), 4), 8)

i = x[3][2][1]

j = y[2][3][0]

```
for a in i:
    for b in j:
        count = count + a+b
Print(count)
```

Summary

- String methods
- Nested lists
- Add, Remove, Change operations on lists
- Tuples, Unpacking, methods
- Sets

To be Contd ...

Thank

THANK

Keep Hustling!