Day3

=======================================

else with loop

=======================================

break, continue, pass

break:

within loop

Continue:

within loop

pass:

is use for placeholder

===================================

MultiLine statements

1.==>

result = (10 + 20) \*\

2 + 34 +\

5/4

2==>

Statements which have brackes like [], (), {} and comma, we are not required to use '\' to go to next line

appName = ['Java Prowess', 'C Prowess',

'c++ Prowess',

'prowessapps.com']

==========================================

Built-in Functions:

1. abs(x)

2. pow(base, exp, mod)

3. round(number)

4. divmod(x, y)

5. sum(iterable, start)

6. len(iterable)

7. max(iterable)

8. min(iterable)

9. sorted(iterable, reverse=False, Key=None) --> it always returns a new sorted list

def fun(v):

return v[-1]

l = ["abc", "xd", "xyb"]

# For custom sort, use key argument

# let's sort on the basis of length of string

newSortedList = sorted(l, key=fun) # c d b ===> xyb abc xd

print(type(newSortedList))

print(newSortedList)

10. all(iterable)

11. any(iterable)

12. bin(number)

13. oct(number)

14. hex(number)

================================================

'''

Use Vs Code to write the python code:

add python extensions

1. create a folder

2. open the command prompt and navigate to the folder

3. type the command to create a virtual environment:

python -m venv [environment\_name]

4. activate the virtual environment:

[environment\_name]\Scripts\activate

'''

STRING HANDLING

1. String is the collection of characters.

2. Strings are ordered block of text.

example:

name = "Mohd Daneyal"

address = """M.G. 23/23

new street

hyderabad"""

NOTE: Python supports forward(+ve) and backward(-ve) indices

NOTE\*:- Strings are immutable object.

Access the Chracters from String:

---------------------------------

We can access single chracter by using index

name[0]

name[-1]

name[-5]

String Slicing(Access the substring)

----------------------------------

[:] -- it is called slicing operator

[start : end : step] == dfault values for start=0, end=length of string, and step=1

NOTE: Step can not be zero

NOTE: slicing always return the same type of new object

NOTE: Step can not Zero(0)

Reverse of string

----------------

name[::-1]

String Operations:

------------------

1. Update/Delete (not possible)

2. Length of String -len()

3. Concatenation -- ( + )

4. Repetition of String -- (\*)

5. Iteration through the string

6. Membership Operation (in, not in)

String Methods:

---------------

1. Capitalize()

2. title()

3. lower()

4. upper()

5. find(str, start, end)

6. index(str, start, end) # ValueError: substring not found

7. rfind(str, start, end)

8. rindex(str, start, end) # ValueError: substring not found

9. count(sub\_str, start, end)

10. islower()

11. isUpper()

......

......

..... more ....

12. split(sep=" ", maxsplit=-1)

13. rsplit(sep=" ", maxsplit=-1)

14. splitlines()

15. join(iterable)

16. strip() # remove leading and trailing characters(default is space)

17. rstrip()

18. lstrip()

19. startswith()

20. endswith()

21. replace(old, new, count)

22. rjust(width, fillchar)

23. ljust(width, fillchar)

24. center(width, fillchar)

25. partition(sep)

this split the string into 3 parts and returns tuple.

The first part of the tuple is string before the specified 'sep'

The second part of tuple is "sep" it self.

The third part of the tuple is string after the specified 'sep'

26. swapcase()

------------------------