23/2/23 NUMPY STRING FUNCTIONS

1. add()
numpy.char.add()

eg. import numpy as np
print ("concat")
print (np.char.add (['welcome', 'to'], ['Apple'

O/P
'Concat
'welcome to' 'Apple'

└ It is used to concatenate the corresponding array elements (strings)

2. multiply()
eg. numpy.char.multiply()
import numpy as np
print ("String :")
print (np.char.multiply ("hi", 3))

op String:
hihihi

└ It returns the multiple copies of the specified string.

3. center()

└ It returns the copy of the string where the original string is centered with the left & right padding filled with the specified number of fill

eg. import numpy as np
print (" Padding * ')
print (np. char. center (" Hello! ", 10, '*'))

o/p Padding *
Hello!*

4. capitalize ()

└ It returns a copy of the original string in which the first letter of the string is capitalized

eg. import numpy as np
print (" Using capital ")
print (np. char. capitalize (" hey ! "))

o/p Using capital
Hey!

5. title ()

└ It returns the title cased version of the string

eg. import numpy as np
print (" Title is : ')
print (np. char. title (" HeLLo world to you"))

o/p Title is:
Hello World To You

6. lower()

└ It returns a copy of the string where all letters are in lowercase

eg. import numpy as np
print ("lower:")
print (np.char.lower ("DONT LIE."))

o/p lower:
dont lie.

7. upper()

└ It returns a copy of the string where all letters are in uppercase

eg. import numpy as np
print ("upper:")
print (np.char.upper ("tell no lies"))

o/p upper:
TELL NO LIES

8. split()

└ returns a list of words in the string

eg. import numpy as np
print ("Doing split")
print (np.char.split ("Hello world"), sep = " ")

op Doing split
['Hello', 'world']

9. strip()

↳ returns a copy of the string with leading & trailing spaces removed

eg. import numpy as np
str = " hey world "
print ('doing strip')
print (np.char.strip(str))

O/P doing strip
hey world

10. join()

↳ returns a string which is the concat of all strings specified in the given sequence

eg. import numpy as np
print (np.char.join (':', 'AB'))

O/P A:B