K.SANGEETHAJ

ASSIGNMENT 1

STRING FUNCTIONS /METHODS

1] CAPITALIZE

In [62]:
```
string_1 = "python is the new emerging programming language"
```
executed in 15ms, finished 10:25:42 2021-10-19

In [63]:
```
string_1.capitalize()     # It makes the first letter of the string to uppercase
```
executed in 7ms, finished 10:25:42 2021-10-19

Out[63]: 'Python is the new emerging programming language'

In [64]:
```
string_2 = "john sir class is easy for the beginners"
```
executed in 4ms, finished 10:25:43 2021-10-19

In [65]:
```
string_2.capitalize()
```
executed in 6ms, finished 10:25:44 2021-10-19

Out[65]: 'John sir class is easy for the beginners'

2] CASEFOLD

In [66]:
```
string_1 ="Python is the New emerging prograMMing laNguagE"
```
executed in 15ms, finished 10:25:44 2021-10-19

In [67]:
```
string_1.casefold()             # It is used to assemble all the improper case mi
```
executed in 9ms, finished 10:25:45 2021-10-19

Out[67]: 'python is the new emerging programming language'

In [68]:
```
string_2= "joHn sir clASs is EASy for the begiNNers"
```
executed in 12ms, finished 10:25:45 2021-10-19

In [69]:
```
string_2.casefold()
```
executed in 15ms, finished 10:25:45 2021-10-19

Out[69]: 'john sir class is easy for the beginners'

3] CENTER

In [70]:
```
string_1 = "Python is awesome"
string_1.center(44)                # It is used for the alignment(centering)
```
executed in 16ms, finished 10:25:46 2021-10-19

Out[70]:
```
'             Python is awesome              '
```

In [71]:
```
string_2= "DS is the top growing sector"
string_2.center(54,'*')              # Either spaces or star is used
```
executed in 5ms, finished 10:25:46 2021-10-19

Out[71]:
```
'*************DS is the top growing sector*************'
```

4] COUNT

In [72]:
```
string_1= "Tea is better than coffee,isn't it?"
string_1.count("is")            # It is used to count the substrings
```
executed in 10ms, finished 10:25:47 2021-10-19

Out[72]:
```
2
```

In [73]:
```
string_2= "john sir class is easy for the beginners"
string_2.count("s")
```
executed in 18ms, finished 10:25:47 2021-10-19

Out[73]:
```
6
```

5] ENDSWITH

In [74]:
```
string_1= "python is simple"
string_1.endswith("simple")     # It is used find whether it has the same ending
```
executed in 6ms, finished 10:25:47 2021-10-19

Out[74]:
```
True
```

In [75]:
```
string_2= "machine learning is part of AI"
string_2.endswith("i")          # It is false bcoz case sensitive
```
executed in 7ms, finished 10:25:48 2021-10-19

Out[75]:
```
False
```

6] EXPANDTABS

In [76]:
```
string_1="bike\tis\tbetter\tthan car "
string_1.expandtabs(tabsize= 10)      # It is used to alter the tabsize(default =
```
executed in 17ms, finished 10:25:49 2021-10-19

Out[76]:
```
'bike      is        better    than car '
```

In [77]:
```
string_2= "I\tlike\ttea"
string_2.expandtabs(5)
```
executed in 4ms, finished 10:25:49 2021-10-19

Out[77]:
```
'I    like tea'
```

7] FIND

In [78]:
```
string_1= "Kohli is the best batsman"
string_1.find("n")                    # It is used to find the substrings
```
executed in 4ms, finished 10:25:49 2021-10-19

Out[78]:
```
24
```

In [79]:
```
string_2= "dhoni is best wicket keeper"
string_2.find("keeper")
```
executed in 5ms, finished 10:25:50 2021-10-19

Out[79]:
```
21
```

8] FORMAT

In [80]:
```
name="MSD"
position = "4th"
print("{} is the {} down batsman for INDIA".format(name,position)) # It provides
```
executed in 9ms, finished 10:25:50 2021-10-19

```
MSD is the 4th down batsman for INDIA
```

In [81]:
```
name="Apple"
occupation ="doctor"
print("Daily an {} keeps {} away".format(name,occupation))
```
executed in 17ms, finished 10:25:50 2021-10-19

```
Daily an Apple keeps doctor away
```

9] INDEX

In [82]:
```
string_1="python is the new emerging programming language"
string_1.index("is")      # provides index numbers
```
executed in 5ms, finished 10:26:06 2021-10-19

Out[82]:
```
7
```

In [83]:
```
string_2="DS is the top growing sector"
string_2.index("t")
```
executed in 18ms, finished 10:26:06 2021-10-19

Out[83]:
```
6
```

10] ISALNUM

In [84]:
```
string_1= "sap1234"
string_1.isalnum()  # it checks whether it is alphanum
```
executed in 10ms, finished 10:26:19 2021-10-19

Out[84]: True

In [85]:
```
string_2="TOM 1234"
string_2.isalnum()      # There is presence of space
```
executed in 10ms, finished 10:26:20 2021-10-19

Out[85]: False

11] ISALPHA

In [86]:
```
string_1="pyhton"
string_1.isalpha()  # it checks whwther it is alpha
```
executed in 17ms, finished 10:26:35 2021-10-19

Out[86]: True

In [87]:
```
string_2 ="java123"
string_2.isalpha()
```
executed in 8ms, finished 10:26:35 2021-10-19

Out[87]: False

12] ISASCII

In [88]:
```
string_1 ="orange"
string_1.isascii()  # it checks whether it is ascii
```
executed in 7ms, finished 10:26:46 2021-10-19

Out[88]: True

In [89]:
```
string_2 ="12345"
string_2.isascii()
```
executed in 7ms, finished 10:26:47 2021-10-19

Out[89]: True

13] ISDECIMAL

In [90]:
```
string_1="123.56"
string_1.isdecimal()      # it checks whether it is decimal
```
executed in 13ms, finished 10:26:57 2021-10-19

Out[90]: False

In [91]:
```
string_2="90678"
string_2.isdecimal()
```
executed in 5ms, finished 10:26:57 2021-10-19

Out[91]: True

14] ISDIGIT

In [92]:
```
string_1="as123"
string_1.isdigit()   # it checks whether it is a number
```
executed in 9ms, finished 10:27:06 2021-10-19

Out[92]: False

In [93]:
```
string_2="9087567"
string_2.isdigit()
```
executed in 15ms, finished 10:27:07 2021-10-19

Out[93]: True

15] ISLOWER

In [94]:
```
string_1="assam"
string_1.islower()   # it checks whether all the strings are in lower case
```
executed in 16ms, finished 10:27:15 2021-10-19

Out[94]: True

In [95]:
```
string_2="America"
string_2.islower()
```
executed in 5ms, finished 10:27:15 2021-10-19

Out[95]: False

16] ISNUMERIC

In [96]:
```
string_1 ="12345"
string_1.isnumeric() # it checks whether all strings are numeric
```
executed in 9ms, finished 10:27:26 2021-10-19

Out[96]: True

In [9]:
```
string_2="root12"
string_2.isnumeric()
```
executed in 17ms, finished 09:02:03 2021-10-19

Out[9]: False

17] ISIDENTIFIER

In [97]:
```
string_1="pyhton"
string_1.isidentifier()    # it checks whether it is a identifier in python
```
executed in 17ms, finished 10:27:46 2021-10-19

Out[97]: True

In [98]:
```
string_2="97alpha"
string_2.isidentifier()
```
executed in 20ms, finished 10:27:46 2021-10-19

Out[98]: False

18] ISPRINTABLE

In [99]:
```
string_1="@ 123"
string_1.isprintable()      # it checks whether all characters in the string are
```
executed in 17ms, finished 10:28:00 2021-10-19

Out[99]: True

In [100]:
```
string_2="\nch(24)"
string_2.isprintable()
```
executed in 4ms, finished 10:28:00 2021-10-19

Out[100]: False

19] ISSPACE

In [101]:
```
string_1 ="    "
string_1.isspace()        #  it checks if there are only whitespace characters ir
```
executed in 17ms, finished 10:28:18 2021-10-19

Out[101]: True

In [102]:
```
string_2="12 +14"
string_2.isspace()
```
executed in 18ms, finished 10:28:19 2021-10-19

Out[102]: False

20] ISTITLE

In [103]:
```
string_1="Python Is Good"
string_1.istitle()              # it checks if the string is a titlecased string
```
executed in 17ms, finished 10:30:03 2021-10-19

Out[103]: True

In [104]:
```
string_2 ="java is differnt from C"
string_2.istitle()
```
executed in 16ms, finished 10:30:03 2021-10-19

Out[104]: False

21] ISUPPER

In [18]:
```
string_1="sun rises in the east"
string_1.isupper()                    # it checks if all the strings are in the up
```
executed in 18ms, finished 09:02:05 2021-10-19

Out[18]: False

In [19]:
```
string_2= "HAPPY MORNING"
string_2.isupper()
```
executed in 17ms, finished 09:02:05 2021-10-19

Out[19]: True

22] STRIP

In [22]:
```
string_1= "     THANK YOU"
string_1.strip()        # Removes unnessary white spaces
```
executed in 18ms, finished 09:10:21 2021-10-19

Out[22]: 'THANK YOU'

In [24]:
```
string_2="android takes over the ios"
string_2.strip('an')          # even removes the string that is unwanted
```
executed in 16ms, finished 09:20:59 2021-10-19

Out[24]: 'droid takes over the ios'

23] REPLACE

In [29]:
```
string_1="GOAD MORNING"
string_1.replace("A","O",1) # It replaces the required string with the new ones
```
executed in 7ms, finished 09:26:49 2021-10-19

Out[29]: 'GOOD MORNING'

In [30]:
```
string_2="FOLLOW AN TRAFFIC RULES"
string_2.replace("AN","THE")
```
executed in 11ms, finished 09:27:57 2021-10-19

Out[30]: 'FOLLOW THE TRAFFIC RULES'

24] SWAPCASE

In [105]:
```
string_1="sun rises in the east"
string_1.swapcase()                # converts the string into the opposite case
```
executed in 19ms, finished 10:31:23 2021-10-19

Out[105]: 'SUN RISES IN THE EAST'

In [106]:
```
string_2="KEEP QUIET"
string_2.swapcase()
```
executed in 13ms, finished 10:31:24 2021-10-19

Out[106]: 'keep quiet'

25] SPLIT

In [107]:
```
string_1="python is the progrmamming language"
string_1.split()      # it breaks up a string at the specified separator
```
executed in 12ms, finished 10:31:34 2021-10-19

Out[107]: ['python', 'is', 'the', 'progrmamming', 'language']

In [108]:
```
string_2="milk is good for health"
string_2.split()
```
executed in 11ms, finished 10:31:35 2021-10-19

Out[108]: ['milk', 'is', 'good', 'for', 'health']

26] STARTSWITH

In [42]:
```
string_1="c is the basic programming language"
string_1.startswith( "c")          # It checks whether the string startswith the
```
executed in 19ms, finished 09:44:53 2021-10-19

Out[42]: True

In [43]:
```
string_2="Be carefull while driving"
string_2.startswith("be")
```
executed in 16ms, finished 09:44:53 2021-10-19

Out[43]: False

27] TITLE

In [44]:
```
string_1="photography is my hobie"
string_1.title()                # It returns a string with first letter of each w
```
executed in 6ms, finished 09:47:16 2021-10-19

Out[44]: 'Photography Is My Hobie'

In [45]:
```
string_2="please be silent"
string_2.title()
```
executed in 8ms, finished 09:49:10 2021-10-19

Out[45]: 'Please Be Silent'

28] ZFILL

In [47]:
```
string_1="program is easy"
string_1.zfill(20)   #It returns a copy of the string with '0' characters padded
```
executed in 11ms, finished 09:53:08 2021-10-19

Out[47]: '00000program is easy'

In [48]:
```
string_2="come fast"
string_2.zfill(10)
```
executed in 4ms, finished 09:54:00 2021-10-19

Out[48]: '0come fast'

29] SPLITLINES

In [55]:
```
string_1="Milk\nChicken\r\nBread\rButter"
string_1.splitlines()   # It splits the string at line breaks and returns a list
```
executed in 8ms, finished 10:04:24 2021-10-19

Out[55]: ['Milk', 'Chicken', 'Bread', 'Butter']

In [56]:
```
string_2="Milk\nChicken\r\nBread\rButter"
string_2.splitlines(True)
```
executed in 5ms, finished 10:05:28 2021-10-19

Out[56]: ['Milk\n', 'Chicken\r\n', 'Bread\r', 'Butter']

30] RINDEX

In [59]:
```
string_1="Let it be, let it be, let it be"
string_1.rindex("it")           # it returns the highest index number
```
executed in 13ms, finished 10:10:22 2021-10-19

Out[59]: 26

In [61]:
```
string_2="come fast,fast and more fast"
string_2.rindex("fast")
```
executed in 6ms, finished 10:14:05 2021-10-19

Out[61]: 24

==========END==========

In [ ]: