String Methods.

1. upper() → convert the string to uppercase.

2. lower() → convert the string to lowercase.

3. title() → Capitalize each word in a string.

4. capitalize() → 1st letter of the 1st word is capitalized.

5. swapcase() → convert upper to lower & lower to upper.

ex:

s1 = "Tom Vs Jerry"

s1.upper() → TOM VS JERRY.

s1.lower() → tom vs jerry.

s1.title() → Tom Vs Jerry.

s1.capitalize() → Tom vs Jerry.

s1.swapcase() → tOM vS jERRY.

(6) replace() → Replace the old value with the new value.

- If the 1st argument doesn't match any substring of the original string, then it will return the original string.

(7) startswith() → True → if the string starts with spec. prefix else False.

(8) Endswith() → True → if the string Ends with spec. prefix else false.

(9) lstrip() → leading whitespace removed.

(10) rstrip() → Trailing whitespace removed.

(11) strip() → leading + trailing w.s rem.

ex: s1 = " He is a boy "

```
print(s1.replace('boy','cricketer')) // He is a cricketer.
print(s1.startswith('he')) // True
print(s1.Endswith('the')) // False
print(s1.lstrip()) // He is a boy
print(s1.rstrip()) // He is a boy
print(s1.strip()) // He is a boy.
```

12) isupper() → True if string are in upper else False

13) islower() → True if string are in lower else False.

Note: Both isupper() & islower() will return True for whitespaces, digits, spl char & symbols. but the string must contain atleast one upper/lower case.

ex: S1 = "hello", S2 = "IRONMAN"
S3 = "5I7G9", S4 = "%L*S"
S5 = "KIM 19", S6 = "12^c"

isupper()	islower()
S1 = False.	S1 = True.
S2 = True.	S2 = False
S3 = True.	S3 = True.
S4 = True	S4 = True.
S5 = True.	S5 = True.
S6 = True.	S6 = True.

14) isalnum() → True → combination of nos and/or alph. (no space, spl char)

15) isalpha() → True → if all char are alphabets (no space, digits, spl char)

16) isspace() → True → if all char are space.

Ex:
```
s1 = "Harry potter"
s2 = "Boy23"
s3 = "catherine"
s4 = "Agi-8:10"
s5 = "Agk2721il"
```

Spl. char:
[@, _, !, #, $, %, ^, &, *, (,), <, >, ?, /, \, |, {, }, ~, :]

isalpha()
```
s1 = False
s2 = False
s3 = True
s4 = False
s5 = False
```

isalnum()
```
s1 = False
s2 = True
s3 = True
s4 = False
s5 = True
```

Ex:
```
s1 = "Harry potter"
s2 = "  "
s3 = " "
s4 = "  9"
```

isspace()
```
s1 → False
s2 → True
s3 → False
s4 → False
```

17) isdigit() → True → only nos.
(no spaces, alpha, spl char, symbols)

18) isnumeric() → True → numeric
ie., Int, subscript, superscript,
fractions, numerals, (no space)

19) isdecimal() → True → decimal nos.
(no space, subscript, superscript,
currency numerators, Fractions,
roman numerals).

ex: S1 = "123 456" S2 = "5943"
S3 = "kim-19" S4 = "wc 2020"
S5 = "20% ^@23", S6 = "Hello".

isdigit()	isnumeric()	isdecimal()
S1 → False	S1 = False.	S1 = False.
S2 = True.	S2 = True.	S2 = True.
S3 = False	S3 = False.	S3 = False.
S4 = False	S4 = False	S4 = False.
S5 = False	S5 = False	S5 = False.
S6 = False.	S6 = False.	S6 = False.

20) Find) → Return the position of string: 3 Arguments.

Find (sub, start, end)

→ start, end are optional. If absent, then search starts from Index 0.

→ If substring is not found, it returns -1.

21) Index → Return the pos. of str.

3 Arguments. Index (sub, start, end).

→ start, end are optional. If absent, then search starts from Index 0.

→ If substring is not found, it raises value Error exception.

ex: s = " w e l c o m e t o

p y t h o n "

9 10 11 12

is find()	is index().
s.find('yth') → 12.	s.index('yth') → 12.
s.find('to') → 8.	s.index('to') → 8.
s.find('o') → 4.	s.index('o') → 4.
s.find('me', 3, 10) → 5.	s.index('me', 3, 10) → 5.
s.find('neon') → -1.	s.index('et', 5, 15) → 6.
s.find('et', 5, 15) → 6.	s.index('neon') → value Error.

22) count() → Returns the no. of occu. of the string.

3. Arguments

count(sub, start, end).

- Start, End are optional. If absent, the count starts from Index 0.

- If sub string not found, it returns 0.

ex:

```
s = "Messi is Light years ahead of Ronaldo".
print( s. count(' i'))   // 3
       s. count('ea', 20)   // 1
       s. count( 's', 7, 20)   // 2
       s. count(' ')   // 6
       s. count('goal')   // 0
```

23. split() : Returns the list of words in sting . using sep as delimiter string.

2. Arguments.

split (sep, maxsplit).

- sep is optional. If absent, splitting is done based on engle empty space.

- max split is optional. specifies the no. of times to split.

ex :. s = " python is better than Java"

s. split() → ['python', 'is', 'better', 'than', 'Java']

s. split ('a') → ['python', 'is', 'better', 'th', 'n', ' J', 'v', '].

s. split ('t', 2) → ['py', 'hon . 'is', 'be', 'ter, 'than', 'Java'].

X ——— X