There are several string methods that will return Boolean values:

Method	True if
str.isalnum()	String consists of only alphanumeric characters (no symbols)
str.isalpha()	String consists of only alphabetic characters (no symbols)
str.islower()	String's alphabetic characters are all lower case
str.isnumeric()	String consists of only numeric characters
str.isspace()	String consists of only whitespace characters
str.istitle()	String is in title case
str.isupper()	String's alphabetic characters are all upper case

Use:

```
number = "5"
letters = "abcdef"

print(number.isnumeric())
print(letters.isnumeric())
```

```
Output:
True
False
```

String Methods(..)

Method	Description
str.capitalize()	Returns the copy of the string with its first character capitalized and the rest of the letters are in lowercased.
string.casefold()	Returns a lowered case string. It is similar to the lower() method, but the casefold() method converts more characters into lower case.
string.count()	Searches (case-sensitive) the specified substring in the given string and returns an integer indicating occurrences of the substring. Syntex: str.count(substring, start, end), str.count(substring)
string.endswith()	Returns True if a string ends with the specified suffix (case-sensitive), otherwise returns False. Syntex: str. endswith (suffix, start, end), str.endswith (suffix)
string.find()	Returns the index of the first occurence of a substring in the given string (case-sensitive). If the substring is not found it returns -1. Syntex: str.find(substr, start, end), str.find(substr)
string.index()	Returns the index of the first occurence of a substring in the given string. Syntex: str.index(substr, start, end), str.index(substr)
string.join()	Returns a string, which is the concatenation of the string (on which it is called) with the string elements of the specified iterable as an argument. i.e sep = '-->' mystr = 'Hello' print(sep.join(mystr)) Output: `'H-->e-->l-->l-->o'`
string.ljust()	Returns the left justified string with the specified width. If the specified width is more than the string length, then the string's remaining part is filled with the specified fillchar.

Method	Description
	mystr = 'Hi' print(mystr.ljust(4)) Output: 'Hi ' Print(mystr.ljust(4, '-')) Output: 'Hi--' Print(mystr.ljust(2, '-')) Output: 'Hi'
string.lower()	Returns the copy of the original string wherein all the characters are converted to lowercase.
string.lstrip()	Returns a copy of the string by removing leading characters specified as an argument. mystr = ' Hello World ' mystr.lstrip() # removes leading spaces Output: 'Hello World '
string.partition()	Splits the string at the first occurrence of the specified string separator sep argument and returns a tuple containing three elements, the part before the separator, the separator itself, and the part after the separator. mystr = 'Hello a World' print(mystr.partition(' ')) Output: ('hello', 'a ', 'world')
string.replace()	Returns a copy of the string where all occurrences of a substring are replaced with another substring. Syntax: str.replace(old, new, count) mystr = 'apples, bananas, apples, apples, cherries' print(mystr.replace('apples','lemons')) Output: lemons, bananas, lemons, lemons, cherries
string.rfind()	Returns the highest index of the specified substring (the last occurrence of the substring) in the given string. Syntax: str.replace(old, new, count) greet = 'Hello World!' print('Index of l: ', greet.rfind('l')) Output: Index of l: 9

Method	Description
string.rindex()	Returns the index of the last occurence of a substring in the given string.
string.rsplit()	Splits a string from the specified separator and returns a list object with string elements. langs = 'C,Python,R,Java,SQL,Hadoop' print(langs.rsplit(',')) Output: ['C', 'Python', 'R', 'Java', 'SQL', 'Hadoop']
string.rstrip()	Returns a copy of the string by removing the trailing characters specified as argument.
string.split()	Splits the string from the specified separator and returns a list object with string elements.
string.splitlines()	Splits the string at line boundaries and returns a list of lines in the string.
string.startswith()	Returns True if a string starts with the specified prefix. If not, it returns False.
string.strip()	Returns a copy of the string by removing both the leading and the trailing characters.
string.swapcase()	Returns a copy of the string with uppercase characters converted to lowercase and vice versa. Symbols and letters are ignored.
string.title()	Returns a string where each word starts with an uppercase character, and the remaining characters are lowercase.
string.upper()	Returns a string in the upper case. Symbols and numbers remain unaffected.

Problems:

1. Consider the following string:
   `hannah = "Did Hannah see bees? Hannah did."`
   a. What is the value displayed by the expression len(hannah)?
      Ans: 32

b. What is the value returned by the method call hannah[12::]?
Ans: ee bees? Hannah did.
c. Write an expression that refers to the letter b in the string referred to by hannah.
Ans: hannah[15:16:]

2. How long is the string returned by the following expression? What is the string?
"Was it a car or a cat I saw?".substring(9, 12)
Ans: 3, car
3. Suppose your python lab instructor has given to you some sentence. You have to reverse the words of sentence. How will you do in python?

```
def reverse(sentence):
    words = sentence.split(' ')
    reverse_words = ' '.join(reversed(words))
    return reverse_words


input = input("Enter Words of string...")
print (reverse(input))
```

4. Ravi gives some sentence to Ankita. Ankita have to recognise even length of word from given sentence. How will Ankita do it in python?

```
def print_Even_Length_Words(str):
    str = str.split(' ')
    for word in str:
        if len(word)%2==0:
            print(word)

string = input("Enter string")
print_Even_Length_Words(string)
```

5. Suppose your lab instructor has given some strings. You have to count total number of vowels are present in this string. How will you do it in python?

```
def vowel_count(string):
    v_count = 0
    vowel = set("aeiouAEIOU")
    for alphabet in string:
        if alphabet in vowel:
            v_count = v_count + 1
    print("Total number of vowels :", v_count)
```

```
string = input("Enter String...")
vowel_count(string)
```

6. Rahul gives two strings to Vivek. Vivek wants to find out uncommon words from both strings. Write a python program to find out uncommon words from both strings.

```
def uncommon(str1,str2):
    str1=str1.split()
    str2=str2.split()
    xyz = set(str1).symmetric_difference(set(str2))
    return xyz

str1 = input("Enter first string: ")
str2 = input("Enter second string: ")
print(list(uncommon(str1,str2)))
```

7. Suppose your instructor has given one string and one substring. You have to find out, given substring is present in given string or not. How will you do it in python?

```
def check_substring(str, str1):
    if (str.find(str1) == -1):
        print("NO")
    else:
        print("YES")

str = input("Enter String:")
str1 = input("Enter sub string:")
check_substring(str, str1)
```

8. Write a program that asks the user for a sentence. Display the sentence backwards, letter by letter. Example:
   Enter a sentence:
   Mary had a little lamb.
   Your sentence backwards:
   .bmal elttil a dah yraM

```
def reverseSentence(st):
    result=st[::-1]
    return result

print(reverseSentence("Hello World."))
```

9. A list contains 5 strings. Write a program that creates the list (you can hard-code the list or fill it with user inputs), then displays the index numbers of elements that contain the letter "e" in upper-case or lower-case.

```
def getEIndex(sts):
    i=0
    while i<len(sts):
        if "e" in sts[i] or "E" in sts[i]:
            print(i)
            i+=1

sts=["Elephent","apple","banana"]
getEIndex(sts)
```

10. Write a function that takes a string as input and reverse only the vowels of a string.
    Example 1:
    Given s = “hello”, return “holle”.
    Note:
    The vowels does not include the letter “y”.

```
def reverseVowels(st):
    if st=="":
        return st
    a=st
    # char[] a =str.toCharArray();
    s=0
    e=len(a)-1
    flag=True
    while flag:
        while not (a[s] in "aeiou" or a[s] in "AEIOU"):
            s+=1
            if s>=e:
                flag=False
                break
        while not (a[e] in "aeiou" or a[e] in "AEIOU"):
            e-=1
            if s>=e:
                flag=False
                break
        if flag:
```

```
            a=swap(a,s,e)
            # t=a[s]
            # a[s]=a[e]
            # a[e]=t
            s+=1
            e-=1
            if s>=e:
                flag=False
                break

        return str(a)

def swap(string, i, j):
    string = list(string)
    string[i], string[j] = string[j], string[i]
    return ''.join(string)

print(reverseVowels("hello"))
```

11. Write a program that prints all ASCII codes of characters of a string . If code is 70, print DEL.

```
def asciiChar(st):
    # sResult=""
    for s in st:
        if ord(s)==70:
            print("DEL")
        else:
            print(ord(s))

asciiChar("Fellow friend")
```

12. Raghav is active on social media, his identifier is #Raghav or @Raghav. Now he wants to know how popular he is. Given an aggregated text with messages; how often does #Raghav or @Raghav occur there?

```
def countPopularity(posts):
    count=0
    for post in posts:
        if "#Raghav" in post or "@Raghav" in post:
```

```
            count+=1
    return count

posts={"You are a champ #Raghav",
"#ShareYourEars was a huge hit! Thanks to your support, @DisneyParks
donated $2M to @MakeAWish & @MakeAWishIntl!",
"Congrats @Raghav"}

print("Count: ",countPopularity(posts))
```