Lab 4: Python – Strings

A string is a sequence of characters.

1. String Concatenation

Example (a):

```
>>> str1 = 'operations'
>>> str2 = 'research'
>>> dept = str1+str2   [The + operator, concatenates or joins strings together ]
>>> print(dept)
```

Example (b):

```
>>> fname = input('Enter your first name: ')
>>> sname = input('Enter your surname: ')
>>> fullname=fname+sname
>>> print(fullname)
```

What happens? Is it printing correctly? Is there any space in between first name and surname? Try

```
>>> fullname=fname+'  '+sname
>>> print(fullname)
>>> print( sname*3)                    → Prints sname 3 times
```

How to modify it if you want to include middle name also?

2. Built-in String Methods

```
Execute the following and observe.
>>> dept='ieor'
>>> dept.capitalize()                  → Capitalize the first character
>>> dept.upper()                       → Capitalize the entire string
>>> college='IIT Bombay'
>>> college.upper()
>>> college.lower()                    → Convert the string to lowercase
```

Update the previous program to print the full name with first name, middle name, surname and all first letters should be in upper case.

Execute the following and observe.

```
>>> dept='industrial engineering and operations research'
>>> dept.title()                       → Convert the string to title case
>>> print('@' *5)                      → Prints '@' 5 times
>>> print('Length=',len('mystr'))      → Prints the length of the string
```

3. Indexing

You can access the characters one at a time with the bracket operator '[]'. The expression in brackets is called an *index*. The index indicates which character in the sequence you want.

"The quick brown fox jumps over the lazy dog" is an English-language pangram—a phrase that contains all of the letters of the English alphabet. Google and find out what is a pangram?

```
>>> mystr='The quick brown fox jumps over the lazy dog'
>>> print(mystr+' is a pangram')
>>> print(mystr[1])
```

Does it print the first character 'T' of the string?

For computer scientists, the index is an offset from the beginning of the string, and the offset of the first letter is zero. So 'T' is the 0th letter ("zero-eth") of mystr, 'h' is the 1th letter ("one-eth"), and 'e' is the 2th ("two-eth") letter.

Character	T	h	e	(space)	q	u	i	c	k
Index	0	1	2	3	4	5	6	7	8

You can use any expression, including variables and operators, as an index, but the value of the index has to be an integer. Otherwise you will get an error. Try

```
>>> k=2
>>> print(mystr[k])
>>> print(mystr[k**3])
>>> print(mystr[1.3])
>>> print(mystr[-2])
>>> print(mystr[1:5])
```

Match the following

Command	**Description**
`mystr[5:]`	Prints the last 5 characters
`mystr[:5]`	Prints all characters from the starting but not the last 5 chacters
`mystr[:-5]`	Prints the first 5 characters
`mystr[-5:]`	Prints all characters from the 5th position to end

Understand the following string operations

```
(a) mystr[::2]
(b) mystr[::-1]
(c) mystr[::-2]
```

Index can be an integer or a range of integers. The general syntax is

`string[start_pos: end_pos: increment]`

For example execute

```
>>> mystr[2:11:3]
```

4. Traversal with a loop

You can iterate over a string using 'while' or for loop

```
index = 0
while index < len(mystr):
    letter = mystr[index]
    print('Position=', index, ' Character=', letter)
    index = index + 1
```

This loop traverses the string and displays each letter on a line by itself. The loop condition is **index < len(mystr)**, so when index is equal to the length of the string, the condition is false, and the body of the loop is not executed. The last character accessed is the one with the index **len(mystr)-1**, which is the last character in the string.

Another way to write a traversal is with a for loop:

```
for char in mystr:
    print(char)
```

5. Some more built-in functions

in	: Membership - Returns true if a character exists in the given string
not in	: Membership - Returns true if a character does not exist in the given string
find	: Determine if a string occurs. It returns index if found and -1 otherwise

Try

```
>>> 'dog' in mystr
>>> 'dog' not in mystr
>>> 'god' in mystr
>>> 'god' not in mystr
>>> mystr.find('dog')
>>> mystr.find('he')
```

In my string 'he' appears two times. Suppose you wanted to check whether 'he' appears between 5th and 25th position, you can provide additional parameters in 'find'. Try

```
>>>mystr.find('he',5,25)
>>> mystr.find('he',5,35)
```

Tasks

In a new notebook, do the following exercises. Use comments / print messages to neatly label the code. For each question write and solve using a single cell, if possible. After you complete the questions, submit your file in Moodle.

Q1. Write a function that takes a string as an argument and displays the letters backwards, using a while/for loop.

Q2. Write a program that takes a string and a character and displays the number of occurrence of the characters in the string.

Q3. Write a program to accept two strings as input and check whether the second string is in the first string or not.

Q4. Write a program to accept three strings, namely string1, string2, and string3 and print the substring of string1from the position of string 2 to the position of string 3.

For example:
string1 = “The quick brown fox jumps over the lazy dog”
string2 = “brown”
string3 = “lazy”
Then the output should be “brown fox jumps over the”

Q5. A palindrome is a word which reads in the same way in either forward or reverse direction. For example: level, refer, hannah, malayalam, madam, “Was it a car or a cat I saw”

Write a program to check whether a string is a palindrome or not.

Q6. Write a program that takes a string and displays the string in the following styles.
All odd characters are in upper case. Sample output: MyNaMeIsKhAn
All even characters are in upper case. Sample output: mYnAmEiSkHaN

Q7. Write a program that takes a string/paragraph and returns the following.
Number of words
Number of characters with space
Number of characters without space

There are more built-in functions available for the string operations. Please read http://docs.python.org/2/library/stdtypes.html#string-methods for more details.