Chapter 13 :

Computer Science

Class XI (As per CBSE Board)

New Syllabus 2019-20

Visit : python.mykvs.in for regular updates

String Manipulation

String is a sequence of characters,which is enclosed between either single (' ') or double quotes (" "), python treats both single and double quotes same.

String Manipulation

Creating String

Creation of string in python is very easy.

e.g.

a='Computer Science'

b="Informatics Practices"

Accessing String Elements

e.g.

```
str='Computer Sciene'
print('str-',   str)
print('str[0]-',   str[0])
print('str[1:4]-',   str[1:4])
print('str[2:]-',   str[2:])
print('str *2-',   str *2 )
print("str +'yes'-",   str +'yes')
```

OUTPUT

```
('str-', 'Computer Sciene')
('str[0]-', 'C')
('str[1:4]-', 'omp')
('str[2:]-', 'mputer Sciene')
('str *2-', 'Computer ScieneComputer
Sciene')
("str +'yes'-", 'Computer Scieneyes')
```

String Manipulation

Iterating/Traversing through string

Each character of the string can be accessed sequentially using for loop.

e.g.

```
str='Computer Sciene‘
for i in str:
    print(i)
```

OUTPUT

C
o
m
p
u
t
e
r

S
c
i
e
n
e

String Manipulation

String comparison

We can use (> , < , <= , <= , == , !=) to compare two strings. Python compares string lexicographically i.e using ASCII value of the characters. Suppose you have str1 as "Maria" and str2 as "Manoj" . The first two characters from str1 and str2 (M and M) are compared. As they are equal, the second two characters are compared. Because they are also equal, the third two characters (r and n) are compared. And because 'r' has greater ASCII value than 'n' , str1 is greater than str2 .

e.g.program

```
print("Maria" == "Manoj")
print("Maria" != "Manoj")
print("Maria" > "Manoj")
print("Maria" >= "Manoj")
print("Maria" < "Manoj")
print("Maria" <= "Manoj")
print("Maria" > "")
```

OUTPUT

```
False
True
True
True
False
False
True
```

String Manipulation

Updating Strings

String value can be updated by reassigning another value in it.

e.g.

```
var1 = 'Comp Sc'
var1 = var1[:7] + ' with Python'
print ("Updated String :- ",var1 )
```

OUTPUT

```
('Updated String :- ', 'Comp Sc with Python')
```

String Manipulation

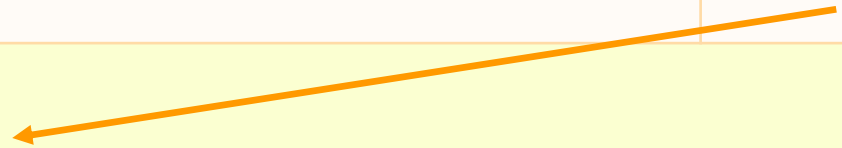
String Special Operators

e.g.

a="comp"

B="sc"

Operator	Description	Example
+	**Concatenation – to add two**	**a + b = comp sc**
*	Replicate same string multiple times	a*2 = compcomp
[]	Character of the string	a[1] will give o
[:]	Range Slice –Range string	a[1:4] will give omp
in	Membership check	p in a will give 1
not in	Membership check for non availability	M not in a will give 1
%	Format the string	



print ("My Subject is %s and class is %d" % ('Comp Sc', 11))

String Manipulation

Format Symbol

%s -string conversion via str() prior to formatting
%i -signed decimal integer
%d -signed decimal integer
%u -unsigned decimal integer
%o -octal integer
%x -hexadecimal integer (lowercase letters)
%X -hexadecimal integer (UPPERcase letters)
%e -exponential notation (with lowercase 'e')
%E -exponential notation (with UPPERcase 'E')
%f -floating point real number
%c -character
%G -the shorter of %f and %E

String Manipulation

Triple Quotes

It is used to create string with multiple lines.

e.g.

Str1 = “””This course will introduce the learner to text mining and text manipulation basics. The course begins with an understanding of how text is handled by python”””

String Manipulation

String functions and methods

Method	Result
str.capitalize()	To capitalize the string
str.find(sub)	**To find the substring position**
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
str.lstrip(char) str.rstrip(char)	Returns a copy of the string with leading/trailing characters

String Manipulation

#Python Program to calculate the number of digits and letters in a string

```
string=raw_input("Enter string:")
count1=0
count2=0
for i in string:
      if(i.isdigit()):
            count1=count1+1
      count2=count2+1
print("The number of digits is:")
print(count1)
print("The number of characters is:")
print(count2)
```

String Manipulation

Searching for Substrings

METHOD NAME	METHODS DESCRIPTION:
endswith(s1: str): bool	Returns True if strings ends with substring s1
startswith(s1: str): bool	Returns True if strings starts with substring s1
count(substring): int	Returns number of occurrences of substring the string
find(s1): int	Returns lowest index from where s1 starts in the string, if string not found returns -1
rfind(s1): int	Returns highest index from where s1 starts in the string, if string not found returns -1

E.g. program

```
s = "welcome to python"
print(s.endswith("thon"))
print(s.startswith("good"))
print(s.find("come"))
print(s.find("become"))
print(s.rfind("o"))
print(s.count("o"))
```

OUTPUT

```
True
False
3
-1
15
3
```