Strings

```
"""Strings"""
# Strign declaration
str1="Hello"
print("String: ",str1)
#acceesing the string
print("str1[0]: ",str1[0])
#accessing last item
print("str1[-1]: ",str1[-1])
#accessing second element
print("str1[1]: ",str1[1])
```

Result:

```
String: Hello
str1[0]:  H
str1[-1]:  o
str1[1]:  e
```

Strings

- A string is a collection of characters

String Slicing

```
"""String Slicing"""
str1="Hello World!"
# 5 is excluded and 0-4 indices are used.
print("str1[0:5]:",str1[0:5])
```

Result:

```
str1[0:5]: Hello
```

Immutability

CHANGE

```
"""Immutability"""
str1="Hello"
str1[0]='a'
print("changing the first character:",str1[0])
```

Result: It's gives Error(Immutability)

```
    str1[0]='a'

TypeError: 'str' object does not support item assignment
```

DELETION

```
"""Immutability"""
str1="Hello"
del str1[0]
print("deleting the first character:",str1[0])
```

Result: It's gives Error(Immutability)

```
    del str1[0]

TypeError: 'str' object doesn't support item deletion
```

String Function

FUNCTIONS

```
greet = "hello world!"
g=["a","b","c"] #sequence of string to join
j="*"
greet1="PYTHON PROGRAMMING"
print("To Upper:",greet.upper()) #convert to upppercase
print("To Lower:",greet1.lower()) #convert to lowercase
print("Is AlphNumeric?",greet.isalnum())    #only alphanumeric characters (no symbols)
print("Is Alpha:",greet1.isalpha()) #only alphabetic characters (no symbols)
print("Is Lower:",greet.islower())  #alphabetic characters are all lower case
print("Is Numeric:",greet.isnumeric())  #only numeric characters
print("Is whitespace:",greet.isspace()) #only whitespace characters
print("Is Title Case:",greet.istitle()) #In title case
print("Is Upper:",greet.isupper())  #alphabetic characters are all upper case
print("Length of greet:",len(greet)) #find length of string
print("Join demo:", j.join(g)) #join the list of string with the help of delimiter
print("Split Demo:",greet.split(" ")) # split the string into list of strings with the delimiter
print("Replace Demo:",greet.replace("hello","Hi")) # replace value in the string
```

RESULT

```
To Upper: HELLO WORLD!
To Lower: python programming
Is AlphNumeric? False
Is Alpha: False
Is Lower: True
Is Numeric: False
Is whitespace: False
Is Title Case: False
Is Upper: False
Length of greet: 12
Join demo: a*b*c
Split Demo: ['hello', 'world!']
Replace Demo: Hi world!
```

String Module

- It’s a built-in module.

```
import string
print("All the letters: ",string.ascii_letters)
print("All lowercase letters: ",string.ascii_lowercase)
print("All uppercase letters: ",string.ascii_uppercase)
print("All digits: ",string.digits)
print("All hexadecimal digits: ",string.hexdigits)
print("All punctuations: ",string.punctuation)
```

Result:

```
All the letters:  abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
All lowercase letters:  abcdefghijklmnopqrstuvwxyz
All uppercase letters:  ABCDEFGHIJKLMNOPQRSTUVWXYZ
All digits:  0123456789
All hexadecimal digits:  0123456789abcdefABCDEF
All punctuations:  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

```
"""Returns - space, tab, linefeed, return, formfeed, and vertical tab characters
' \t\n\r\x0b\x0c'    """
# import string library function
import string
print("Hello")
s = string.whitespace
print(s)
print("Hello..This message is after whitespace")
```

Result:

```
♂
Hello..This message is after whitespace

In [35]:
```

String Programs

String Char-Case Change

- A String is Palindrome or Not
- A String Is an Anagram or Not
- Find the length of a string.
- Copy one string to another string.
- Concatenate two strings.
- Compare two strings.
- Convert lowercase string to uppercase.
- Convert uppercase string to lowercase.
- Toggle case of each character of a string.
- Find a total number of alphabets, digits or special character in a string.
- Count the total number of vowels and consonants in a string.
- Count the total number of words in a string.
- Find the reverse of a string.
- Check whether a string is a palindrome or not.
- Reverse order of words in a given string.

8. String Questions: Level Up

- Find the first occurrence of a character in a given string.

1. Find the last occurrence of a character in a given string.
2. Search all occurrences of a character in a given string.
3. Count occurrences of a character in a given string.
4. Find the highest frequency character in a string.
5. Find the lowest frequency character in a string.
6. Count the frequency of each character in a string.
7. Remove the first occurrence of a character from a string.
8. Remove the last occurrence of a character from a string.
9. Delete all occurrences of a character from a string.
10. Remove all repeated characters from a given string.
11. Replace the first occurrence of a character with another in a string.
12. Replace the last occurrence of a character with another in a string.

1. Put all occurrences of a character with another in a string.
2. Find the first occurrence of a word in a given string.
3. Find the last occurrence of a word in a given string.
4. Search all occurrences of a word in a given string.
5. Count occurrences of a word in a given string.
6. Remove the first occurrence of a word from the string.
7. Remove the last occurrence of a word in a given string.
8. Delete all occurrence of a word in a given string.
9. A Trim leading white space characters from a given string.
10. Trim trailing white space characters from a given string.
11. Trim both leading and trailing white space characters from a given string.
12. Remove all extra blank spaces from the given string.

Lists as Array

```
"""Lists"""
#list is a sequesnce
l=[10,20,30,40,50]
print("l=",l)
l1=[1,2,[3,4]]
print("l1=",l1)
l2=["abc","efg","hij"]
print("l2=",l2)
print("l2[0]:",l2[0])
print("abc in l2:","abc" in l2)
print("abcd in l2:","abcd" in l2)
```

Result:

```
l= [10, 20, 30, 40, 50]
l1= [1, 2, [3, 4]]
l2= ['abc', 'efg', 'hij']
l2[0]: abc
abc in l2: True
abcd in l2: False
```

Membership using 'in':

Traversing a list

```
print("Membership test:")
for i in l2:
    print(i)
```

Result:

```
Membership function test:
abc
efg
hij
```

List Operations & Slicing

```
"""List operations"""
a1=[1,2,3]
a2=[4,5,6]
print("Concatenation:",a1+a2)
print("Repeat a1 three times:",a1*3)
```

Result:

```
Concatenation: [1, 2, 3, 4, 5, 6]
Repeat a1 three times: [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

```
"""List Slicing"""
a=[1,2,3,4,5,6]
print("a[0:3]=",a[0:3]) #prints index 0 to 2
print("a[:4]=",a[:4]) #prints index 0 to 3
print("a[4:]=",a[4:]) #prints index 4 to 5
print("a[:]",a[:]) #print all the items in list
```

Result:

```
a[0:3]= [1, 2, 3]
a[:4]= [1, 2, 3, 4]
a[4:]= [5, 6]
a[:] [1, 2, 3, 4, 5, 6]
```

List Methods

```
"""List as an Array"""
x1 = ['a', 'b', 'c']

x1.append('d')
print("x1 after append()=",x1)   #Adds an element at the end of the list"""

x2=['w','x']
x1.clear() #Removes all the elements from the list
print("x1 after clear()=",x1)

x3=[]
x3=x2.copy() #Returns a copy of the list
print("x3 after copy()=",x3)

print("count of w in x3 =",x3.count('w')) #Returns the number of elements with the specified value

x4=('g','h')   #declaring tuple
x3.extend(x4) #Add the elements of a list (or any iterable), to the end of the current list
print("x3 after extend() with a tuple x4 =",x3)

r=x3.index('x')
print("Index of x in x3:",r) #Returns the index of the first element with the specified value

q='e'
x3.insert(1,q) #Adds an element at the specified position
print("x3 after insert() at position 1=",x3)

x3.pop(2)
print("x3 after pop() at index 2:",x3) #Removes the element at the specified position

x3.remove('w')
print("x3 after remove() for value w:",x3) #Removes the first item with the specified value

x3.reverse()
print("Reverse of list x3:",x3) #Reverses the order of the list

x3.sort()
print("Sort the list x3:",x3) #Sorts the list
```

Result:

```
x1 after append()= ['a', 'b', 'c', 'd']
x1 after clear()= []
x3 after copy()= ['w', 'x']
count of w in x3 = 1
x3 after extend() with a tuple x4 = ['w', 'x', 'g', 'h']
index of x in x3: 1
x3 after insert() at position 1= ['w', 'e', 'x', 'g', 'h']
x3 after pop() at index 2: ['w', 'e', 'g', 'h']
x3 after remove() for value w: ['e', 'g', 'h']
Reverse of list x3: ['h', 'g', 'e']
Sort the list x3: ['e', 'g', 'h']
```

- -

List Loop

Code:

```
"""list loops"""
l=["Social","Science","Maths","English"]
for i in range(0,len(l)):
    print(l[i])
```

Result:

```
Social
Science
Maths
English
```

Lists are mutable

- Mutability means add or remove or change
- List allow duplicated

```
"""Lists allow duplicates, add, remove and change"""
l=["a","b","c","d","a"]
print("*Dulicate values in the list*")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
# changing a value in list
l[2]="z"
print("*A value location 2 changed in list*")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
print("*Remove a value from the list*")
l.remove("b")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
print("*Adding to the list*")
l.append("f")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
```

Result:

```
*Dulicate values in the list*
l0:=a
l1:=b
l2:=c
l3:=d
l4:=a
*A value location 2 changed in list*
l0:=a
l1:=b
l2:=z
l3:=d
l4:=a
*Remove a value from the list*
l0:=a
l1:=z
l2:=d
l3:=a
*Adding to the list*
l0:=a
l1:=z
l2:=d
l3:=a
l4:=f
```

Aliasing

```
"""Aliasing"""
l=["a","b","c"]
m=l  #aliasing
print("List l is:")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
print("List m is:")
for i in range(0,len(m)):
    print("m{}:={}".format(i,m[i]))

#Changing a value in the aliased list m"
m[2]="z"
print("List l after changing value in m:")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
```

Result:

```
List l is:
l0:=a
l1:=b
l2:=c
List m is:
m0:=a
m1:=b
m2:=c
List l after changing value in m:
l0:=a
l1:=b
l2:=z
```

Cloning

```
"""Cloning"""
l=["a","b","c"]
m=l[:] #cloning with slicing the list
print("List l is:")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
print("List m is:")
for i in range(0,len(m)):
    print("m{}:={}".format(i,m[i]))

print("Changing a value in the cloned list m:")
m[2]="z"
print("List l after changing value in m:")
for i in range(0,len(l)):
    print("l{}:={}".format(i,l[i]))
```

Result:

```
List l is:
l0:=a
l1:=b
l2:=c
List m is:
m0:=a
m1:=b
m2:=c
Changing a value in the cloned list m:
List l after changing value in m:
l0:=a
l1:=b
l2:=c
```

List parameters

```
"""List as parameters"""
def list_para(l):
  for i in l:
      print("{}".format(i))

l = ["a", "b", "c"]
print("Display List")
list_para(l)
```

Result:

```
Display List
a
b
c
```

List Comprehension

Syntax:

```
n_list = [any_expr for i in
iterable(can be list or tuple or set)
if condition]
```

Example:

```
"""List Comprehensions"""

l=["abc","bca","cb"]
n=[i for i in l if "a" in i ]
print("New list n is:{}".format(n))
```

Result:

```
New list n is:['abc', 'bca']
```

List comprehension programs

- List with values greater than 5
- List with squares of values
- Built-in functions(lower(),upper()) as expressions
- Maximum value in each row of matrix
- List of values ends with “a” and length > n
- Matrix to a list

List or Arrays Programs

1. Insert An Element Desired or Specific Position In An Array
2. Remove Duplicates Items In An Array
3. Delete Element From Array At Desired Or Specific Position
4. Print “I AM GENIUS” Instead Of Your Name Using Array
5. Check String Is Palindrome Or Not Using For Loop
6. Convert All Input String Simultaneously Into Asterisk (*)
7. Read and print elements of the array. – using recursion.
8. Print all negative elements in an array.
9. Sum of all array elements. – using recursion.
10. Find a maximum and minimum element in an array. – using recursion.
11. Get the second largest element in an array.
12. Count the total number of even and odd elements in an array.
13. Count the total number of negative elements in an array.
14. Copy all elements from an array to another array.

1. Insert an element in an array.
2. Delete an element from an array at the specified position.
3. Count frequency of each element in an array.
4. Print all unique elements in the array.
5. Count the total number of duplicate elements in an array.
6. Delete all duplicate elements from an array.
7. Merge two arrays to the third array.
8. Find the reverse of an array.
9. Put even and odd elements of an array in two separate arrays.
10. Search an element in an array.
11. Sort array elements in ascending or descending order.
12. Sort even and odd elements of the array separately.
13. Left rotate an array.
14. Right rotate an array.