**Items, Split, .upper, .lower, .append. Array**

Note: split() and strip() are string methods further explained elsewhere. split() separates a string into tokens using any whitespace as the default separator. The tokens are returned in a list (i.e., 'a b c'.split() returns ['a', 'b', 'c']). strip() returns a copy of a string with leading and trailing whitespace removed.

where = 'Sn = Smith'

a, v = (word.strip() for word in where.lower().split('='))



b = where[:]

print(a,b,v)

print(a[0],b[1],v[2])



* sn Sn = Smith smith



* s n i



**slice** Use slicing To extract substring from the whole string then we use the syntax like

string\_name[beginning: end : step]

x = ducky

print(x[0:2])

.split( )

.items( )

command = input().lower().strip()

.upper( )

.append( )

isupper(), islower(), lower(), upper()

isupper() is a built-in method used for string handling. This method returns True if all characters in the string are uppercase, otherwise, returns “False”. islower() does not take any parameters

# checking for uppercase characters

string **=** 'GEEKSFORGEEKS'

print(string.isupper())

item.lower( ) *It converts the given string in into lowercase and returns the string.*

*same for .upper but for upper case*

The split() method splits a string into a list.

You can specify the separator, default separator is any whitespace.

string.split(separator, maxsplit)

separator Optional. Specifies the separator to use when splitting the string. By default any

whitespace is a separator

maxsplit Optional. Specifies how many splits to do. Default value is -1, which is "all occurrences"

txt = "hello, my name is Peter, I am 26 years old"

x = txt.split(", ")

print(x)

* ['hello', 'my name is Peter', 'I am 26 years old']

You could have used (“,”, 1) it would have only separated the first instance of “,”

**,strip( )**

The strip() method removes any leading (spaces at the beginning) and trailing (spaces at the end) characters (space is the default leading character to remove)

txt = ",,rrttgg11.....banana....rr"

x = txt.strip(",.grt1")

print(x)

* banana

**.append( )** Python’s append() function inserts a single element into an existing list. The element will be added to the end of the old list rather than being returned to a new list. Adds its argument as a single element to the end of a list. The length of the list increases by one.

|  |
| --- |
| my\_list **=** ['geeks', 'for']  my\_list.append('geeks')  print my\_list |

* ['geeks', 'for', 'geeks']

my\_list = ['geeks', 'for', 'geeks']

another\_list = [6, 0, 4, 1]

my\_list.append(another\_list)

print my\_list

['geeks', 'for', 'geeks', [6, 0, 4, 1]]

**extend( )** Iterates over its argument and adding each element to the list and extending the list. The length of the list increases by a number of elements in its argument.

|  |
| --- |
| my\_list **=** ['geeks', 'for']  another\_list **=** [6, 0, 4, 1]  my\_list.extend(another\_list)  print my\_list |

* ['geeks', 'for', 6, 0, 4, 1]

**insert( )** This method can be used to insert a value at any desired position. It takes two arguments-element and the index at which the element has to be inserted.

list\_name(index,element)

test = ['geeks', 'geeks']

test.insert(1, 'for')

print(test)

[‘geeks’, ‘for’, ‘geeks’]

**Using ‘+’ operator to add an element in the list in Python**

sample\_list =[ ]

n = 10

for i in range(n):

**# i refers to new element**

sample\_list = sample\_list+[i]

#if you remove +[ ] => [ ]

print(sample\_list)

Output:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

**Using append**

sample\_list =[ ]

n = 10

for i in range(n):

# i refers to new element

sample\_list.append(i)

print(sample\_list)

Output:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Comprehension List

fruits = ["apple", "banana", "cherry", "kiwi", "mango"]  
newlist = []  
  
for x in fruits:  
  if "a" in x:  
    newlist.append(x)  
  
print(newlist)

**Array**

# importing "array" for array creations

**import** array as arr

# array with int type

Chart, table, treemap chart

Description automatically generateda = arr.array('i', [1, 2, 3])

print ("Array before insertion : ", end =" ")

for i in range (0, 3):

print (a[i], end =" ")

print()

* **Array before insertion : 1 2 3**

# inserting array using

# insert() function

a.insert(1, 4)

print ("Array after insertion : ", end =" ")

for i in (a):

print (i, end =" ")

print()

* **Array after insertion : 1 4 2 3**

# array with float type

b = arr.array('d', [2.5, 3.2, 3.3])

print ("Array before insertion : ", end =" ")

for i in range (0, 3):

print (b[i], end =" ")

print()

* **Array before insertion : 2.5 3.2 3.3**

# adding an element using append()

b.append(4.4)

print ("Array after insertion : ", end =" ")

for i in (b):

print (i, end =" ")

print()

Array after insertion : 2.5 3.2 3.3 4.4