



# **SNS COLLEGE OF ENGINEERING**

**Kurumbapalayam(Po), Coimbatore – 641 107**

**Accredited by NAAC-UGC with 'A' Grade**

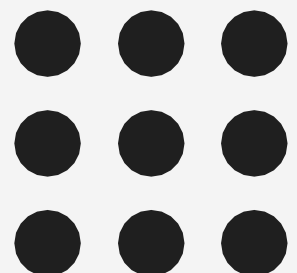
**Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai**

**Department of Artificial Intelligence and  
Data Science**

**Course Name – Computational Thinking and  
Python Programming**

**I Year / I Semester**

**Unit 3-CONTROL FLOW, FUNCTIONS**





## List as array:

### Array:

Array is a collection of similar elements. Elements in the array can be accessed by index. Index starts with 0. Array can be handled in python by module named array. To create array have to import array module in the program.

### **Syntax :**

```
import array
```

### **Syntax to create array:**

```
Array_name = module_name.function_name('datatype',[elements])
```

### **example:**

```
a=array.array('i',[1,2,3,4])
```

a- array name

array- module name

i- integer datatype

### **Example**

#### **Program to find sum of array elements**

```
import array
```

```
sum=0
```

```
a=array.array('i',[1,2,3,4])
```

```
for i in a:
```

```
sum=sum+i
```

```
print(sum)
```

### **Output**

```
10
```

1. single quotes (' ')
2. double quotes (" ")
3. triple quotes('''' ''')

### Operations on string:

1. Indexing
2. Slicing
3. Concatenation
4. Repetitions
5. Member ship

String A	H	E	L	L	O
Positive Index	0	1	2	3	4
Negative Index	-5	-4	-3	-2	-1

<b>indexing</b>	<pre>&gt;&gt;&gt;a="HELLO" &gt;&gt;&gt;print(a[0]) &gt;&gt;&gt;H &gt;&gt;&gt;print(a[-1]) &gt;&gt;&gt;O</pre>	<ul style="list-style-type: none"> <li>❖ Positive indexing helps in accessing the string from the beginning</li> <li>❖ Negative subscript helps in accessing the string from the end.</li> </ul>
<b>Slicing:</b>	<pre>Print[0:4] – HELLO Print[ :3] – HEL Print[0: ]- HELLO</pre>	<p>The Slice[start : stop] operator extracts sub string from the strings. A segment of a string is called a slice.</p>
<b>Concatenation</b>	<pre>a="save" b="earth" &gt;&gt;&gt;print(a+b) saveearth</pre>	<p>The + operator joins the text on both sides of the operator.</p>
<b>Repetitions:</b>	<pre>a="panimalar " &gt;&gt;&gt;print(3*a) panimalarpanimalar panimalar</pre>	<p>The * operator repeats the string on the left hand side times the value on right hand side.</p>
<b>Membership:</b>	<pre>&gt;&gt;&gt; s="good morning" &gt;&gt;&gt;"m" in s True &gt;&gt;&gt; "a" not in s True</pre>	<p>Using membership operators to check a particular character is in string or not. Returns true if present</p>



## Convert list into array:

fromlist() function is used to append list to array. Here the list is act like a array.

### Syntax:

arrayname.fromlist(list\_name)

### Example

#### program to convert list into array

```
import array
sum=0
l=[6,7,8,9,5]
a=array.array('i',[])
a.fromlist(l)
for i in a:
    sum=sum+i
print(sum)
```

### Output

35

#### Methods in array

```
a=[2,3,4,5]
```

	Syntax	example	Description
1	array(data type, value list)	array('i',[2,3,4,5])	This function is used to create an array with data type and value list specified in its arguments.
2	append()	>>>a.append(6) [2,3,4,5,6]	This method is used to add the at the end of the array.
3	insert(index,element )	>>>a.insert(2,10) [2,3,10,5,6]	This method is used to add the value at the position specified in its argument.
4	pop(index)	>>>a.pop(1) [2,10,5,6]	This function removes the element at the position mentioned in its argument, and returns it.
5	index(element)	>>>a.index(2) 0	This function returns the index of value
6	reverse()	>>>a.reverse() [6,5,10,2]	This function reverses the array.
7	count()	a.count()  4	This is used to count number of elements in an array

## Function Composition :

Function Composition is the ability to call one function from within another function

It is a way of combining functions such that the result of each function is passed as the argument of the next function.

In other words the output of one function is given as the input of another function is known as function composition.

find sum and average using function composition	output
<pre>def sum(a,b):     sum=a+b     return sum def avg(sum):     avg=sum/2     return avg a=eval(input("enter a:")) b=eval(input("enter b:")) sum=sum(a,b) avg=avg(sum) print("the avg is",avg)</pre>	<pre>enter a:4 enter b:8 the avg is 6.0</pre>

## Recursion

A function calling itself till it reaches the base value - stop point of function call. Example: factorial of a given number using recursion

Factorial of n	Output
<pre>def fact(n):     if(n==1):         return 1     else:         return n*fact(n-1) n=eval(input("enter no. to find fact:")) fact=fact(n) print("Fact is",fact)</pre>	<pre>enter no. to find fact:5 Fact is 120</pre>