

Python

Introduction to Iteration For

List

- For the time being, list is a collection of variables.
- Variables can be of any type like integer, float, string etc.
- For example:
 - `A=[1,2,3,4,5]` is a list of integers
 - `B=["adam", "Bella ", "Charlie", "David"]` is a list of string
- We will learn details on list on the next lectures.

Yes we can.

- The number of element can be found using *len(list)*
- See the following code:

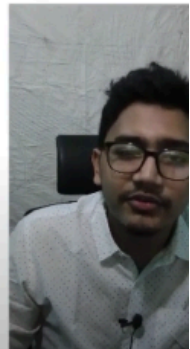
```
a=[1,2,3,4,5]
i=0
while i<len(a):
    print(a[i])
    i=i+1
```

Rewrite using *for*

- What if we can say the program to explore the list from first to last by itself?
- There comes the *for* loop.
- See the Given code.

```
a=[1,2,3,4,5]
for i in a:
    print(i)
```

- This is exactly the same program which we saw in the previous slide



Another program

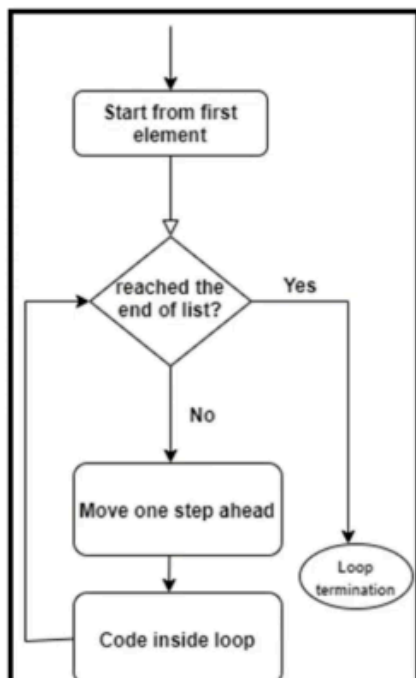
- Find the output of the program:

```
b=["Adam","Bella","Charlie"]  
  
for i in b:  
    print("Hello ",i)
```

Output

```
Hello Adam  
Hello Bella  
Hello Charlie
```

Flow chart



```
b=["Adam","Bella","Charlie"]  
  
for i in b:  
    print("Hello ",i)
```

Example

- Find the sum of the following list:

a=[22,44,33,51,64]

```
a=[22,44,33,51,64]
sum=0
for i in a:
    sum=sum+i
print(sum)
```

Largest Number in the List

```
: a = [1,2,3,4,5,6,8,9]
largest = -999
for i in a:
    if i > largest:
        largest=i
print(largest)
```

range() function

- Generates the **integer numbers between a given start integer to a stop integer**.

- Syntax:**

```
range(start, stop, step)
```

- A **start** integer is a starting number of the sequence. By default, it starts with 0 if not specified.
- A **stop** argument is an upper limit. range() function generates numbers up to this number but not including this number.
- The **step** is a difference between each number in the result. The default value of the step is 1 if not specified.
- All the arguments(start, stop, step) must be integers.

Let us see an example,

range(1,6) generates a sequence of integers from 1 to 5; not 6. The generated sequence will be [1,2,3,4,5].

range() function(Examples)

- range(0,7) will generate a sequence like [0,1,2,3,4,5,6]
- range(7) will generate a sequence like [0,1,2,3,4,5,6]
- range(0,9,1) will generate a sequence like [0,1,2,3,4,5,6,7,8]
- range(0,9,2) will generate a sequence like [0,2,4,6,8]
- range(-1,6,2) will generate a sequence like [-1,1,3,5]
- range(-1,-5,-1) will generate a sequence like [-1,-2,-3,-4]
- range(-1,-7,-2) will generate a sequence like [-1,-3,-5]

[N.B: print(range(0,9,1)) will not print the integers.]

Indexing in range()

- All the values generated by range() function are indexed.

For example,

range(0,9,2) will generate a sequence of [0,2,4,6,8]

These values are indexed like this:

Values	0	2	4	6	8
Index	0	1	2	3	4
Negative index	-5	-4	-3	-2	-1

Let us see an example,

```
seq = range(0,9,2)
print(seq[0])
print(seq[2])
print(seq[-1])
psrint(seq[-4])
```

Output

```
0
4
8
2
```

```
color= ['white','black','red']
cars = {'BMW','Audi','toyota'}
for i in color:
    for j in cars:
        print(i,j)
```

```
white Audi
white BMW
white toyota
black Audi
black BMW
black toyota
red Audi
red BMW
red toyota
```