* For Loop :-

Syntax : x = ['navin', 29, 5.8]

```
for i in x:
    print(i)
```

Indentation is important here

* Break, continue, pass :-

Break → Come out of the loop

Continue → skip that iteration

pass → If there is no code then pass it / ignore it

* Printing patterns -

* For - else

* Arrays in python

- All the values should be in same type
- Arrays in python don't have specific size you can increase & decrease the size according to your need.

```
import array as arr
from array import *
```

Type Code	C Type	Python Type	Min size in bytes
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	PY_UNICODE	unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int		
'I'	unsigned int		
'l'	signed long		
'L'	unsigned long		
'f'	float	float	4
'd'	double	float	8

Single Dimensional Array → one row multiple columns
Multi Dimensional array → multiple row multiple columns
Array does not support multidimensional
Then we use third package which is 'Numpy'
PIP → Install packages

* Ways of creating Arrays in Numpy -

array()	
linspace()	shallow copy
logspace()	Deep copy
arange()	
zeros()	
ones()	

Shallow copy -
Its copy the elements but both the arrays are dependent on each other.

Deep copy -
Its copy all the elements but both the arrays are not linked to each other.

* Functions :-
You can use same set of lines in multiple times
This code is reusable. Function is called only when
E.g. You are explicitly calling it

```
def greet():
    print('Hello!')
    print("Good morning")
greet()
```