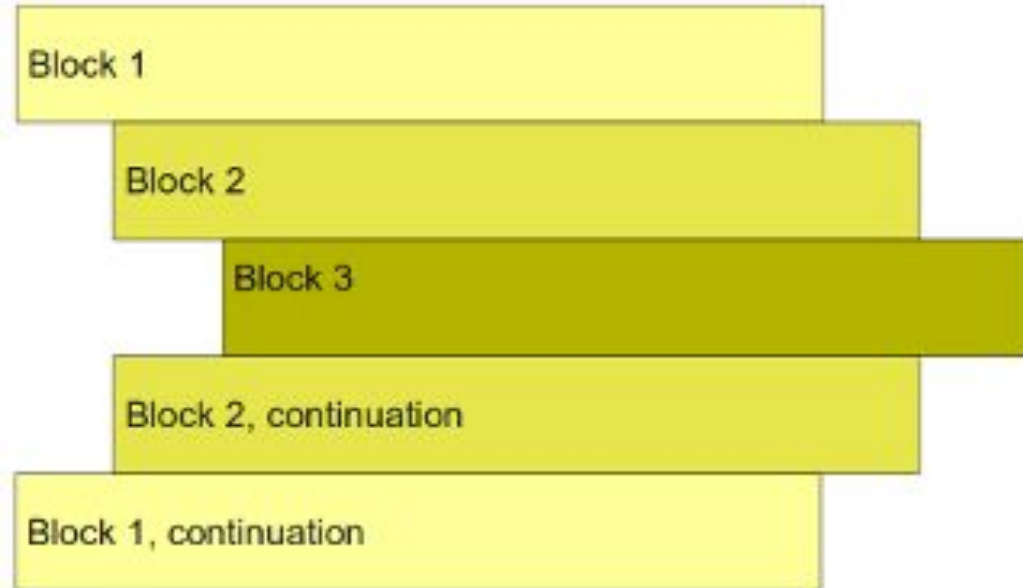# Lecture 3 : Conditions and Loops

Blocks in Python

Python programs get structured through indentation, i.e. code blocks are defined by their indentation.

In Python all the brackets '**( )**', '**{ }**', '**[ ]**' are used in data structures like lists, tuples, dicts etc. so they **can't be used in declaring any block.**

Block 1

Block 2

Block 3

Block 2, continuation

Block 1, continuation

Blocks in any other Language

```
if ( condition )


{

    do some stuff in the block ...


}
                              blocks


do other stuff out of block ...
```

Blocks in Python

colon marks the start of the block

```
if ( condition ) :

    do some stuff in the block ...


do other stuff out of block ...
```

Indent

1 tab or 4 spaces

Program Comparison

other language

```
if ( 10 % 2 == 0 )

{


print("it's even")



}
```

python

```
if 10 % 2 == 0 :


    print("it's even")
```

Conditions in Python

If else condition

Syntax :

      if condition:

           Statements..

      else:

           Statements..

```python
a=40
```

```python
if a>=30:
    print('Its True')
else:
    print('Nothing True')
```

```
Its True
```

```python
a=20
if a>=30:
    print('Its True')
else:
    print('Nothing True')
```

```
Nothing True
```

'in' keyword

```python
string='in keyword finds the occurence of an element'
```

```python
if 'the' in string:
    print(True)
```
True

```python
print('of' in string)
```
True

```python
print('off' in string)
```
False

Conditions in Python

Chaining if condition

Syntax :

```python
a=40
b=30
```

```python
if condition:
    statements...

elif condition:
    statements...

else:
    statements...
```

```python
if a is 50:
    print('a is 50')
elif b is not 10:
    print('b is not 10')
else:
    print('Nothing true')
```

```
b is not 10
```

Iteration in Python

Iterables in Python is simply an object that can be iterated upon. An object which will return data, one element at a time.

Some basic Iterators:

- string
- List
- Tuple
- Set
- Range
- Dictionary

Loops in Python

For loop

Syntax :

```python
for var in iterator:
    statement1
    statement2
    .
    .
    ...
```

```python
for i in range(10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

```python
numbers = [132,432,342,43245,43]
for num in numbers:
    print(num)
```

```
132
432
342
43245
43
```

What happened under the hood

```python
for i in "LOOP":
    print(i)
```

L
O
O
P

Loops in Python

For Else loop

Syntax :

```python
for var in iterator:
    statement1
    statement2
    .
    .
    .
else:
    statements..
```

```python
for i in range(20):
    if i==4:
        print('found 4')
        break
    print(i)
else:
    print('4 not found')
```

```
0
1
2
3
found 4
```

Loops in Python

```python
for i in range(1, 20, 2):
    if i==4:
        print('found 4')
        break
    print(i)
else:
    print('4 not found')
```

```
1
3
5
7
9
11
13
15
17
19
4 not found
```

Loops in Python

While loop

Syntax :

```python
while condition:
    statement1
    statement2
    .
    .
    .
```

```python
num=5
while num>0:
    print(num)
    num-=1
```

```
5
4
3
2
1
```

```python
while True:
    print('This loop will run forever')
```

```
This loop will run forever
This loop will run forever
This loop will run forever
This loop will run forever
This loop will run forever
```

Program to take inputs from user and display

```python
movies=[]  #empty list to store names
print("Enter Movie names :\n")
while True: #infinite loop
    movie_name=input(">>>") #take user input
    if movie_name:
        movies.append(movie_name) #store names in list
    else:
        break
print(movies)
```

```
Enter Movie names :

>>> the movie
>>> the movie 2
>>> the movie returns
>>> the movie starts
>>> the movie reboot
>>>
['the movie', 'the movie 2', 'the movie returns', 'the movie starts', 'the movie reboot']
```

1. Write a program to print all the no. divisible by 9 or 12 in range 100 - 200.
2. Write a program to take