Parul®
University

Python Programming

CHAPTER-3

Program Flow Control

Parul® University

DIGITAL LEARNING CONTENT

What is Control Structure?

- Control structure identifies the sequence of execution of statements or code flow in python

- It can be:
  - Sequential Execution
  - Conditional Execution
  - Iterative Execution

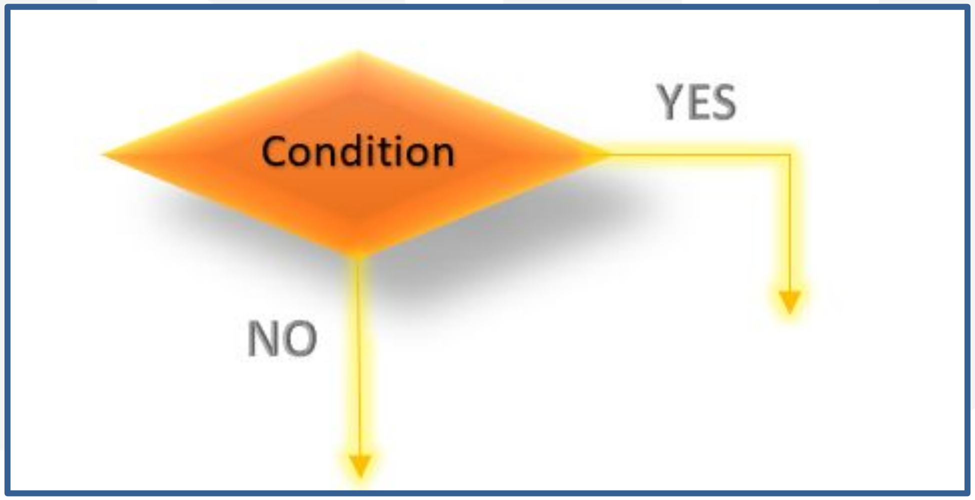
Image source : https://opensource.com/

Parul® University

DIGITAL LEARNING CONTENT

Conditional Execution

- Execute statements based on condition
- Example: Problem of making decision



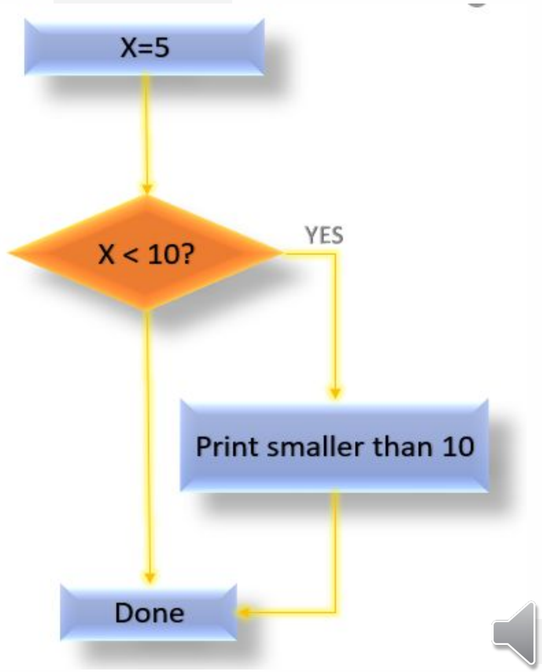
Conditional Execution : if Statement

- One – way decision statement
- Syntax:

```
if <expression> :
    #when condition is true
```

- Example:

```
x = 5

if x < 10 :      #this is if statement

    print('x is smaller than 10') # if block
```

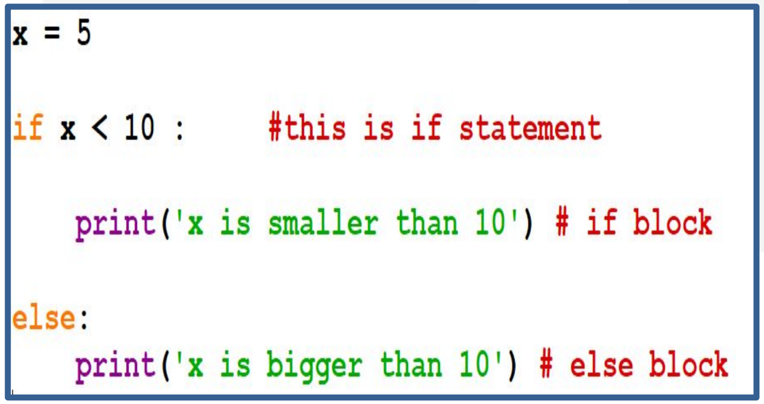


Conditional Execution : if..else Statement

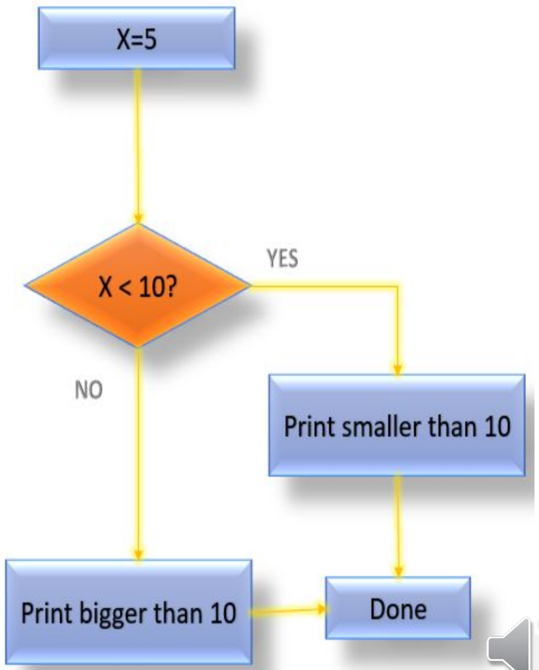
- Two – way decision statement
- Syntax:

```
if  <expression> :
        #when condition is true
else:
        # when condition is false
```

- Example:

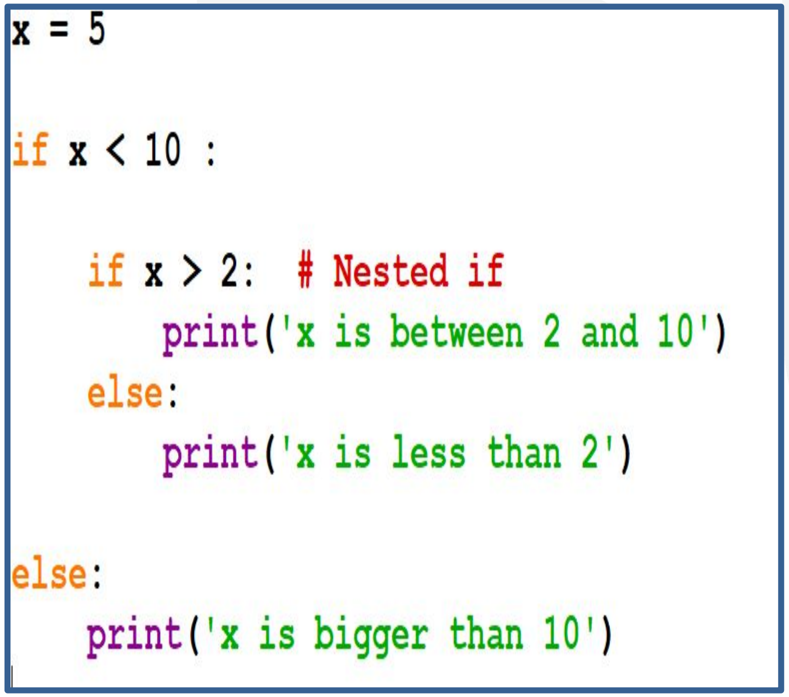


```
x = 5

if x < 10 :      #this is if statement

    print('x is smaller than 10') # if block

else:
    print('x is bigger than 10') # else block
```

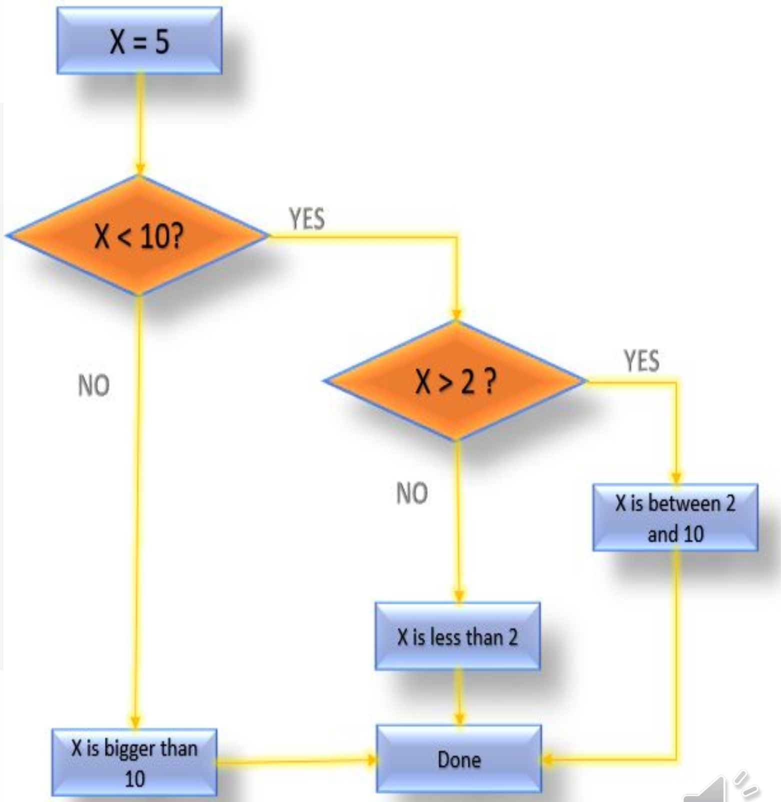


Conditional Execution : Nested Decision

- Another if statement in if statement
- Example:

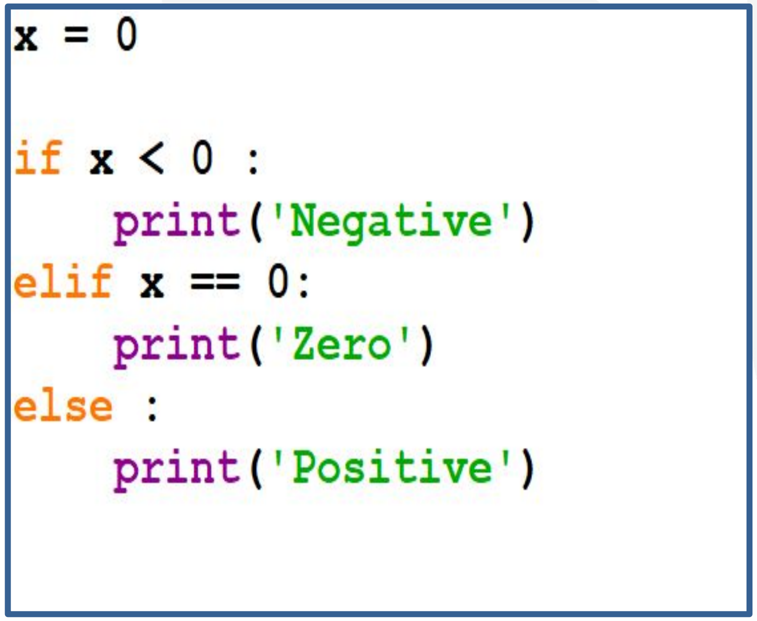


```
x = 5

if x < 10 :

    if x > 2:   # Nested if
        print('x is between 2 and 10')
    else:
        print('x is less than 2')

else:
    print('x is bigger than 10')
```

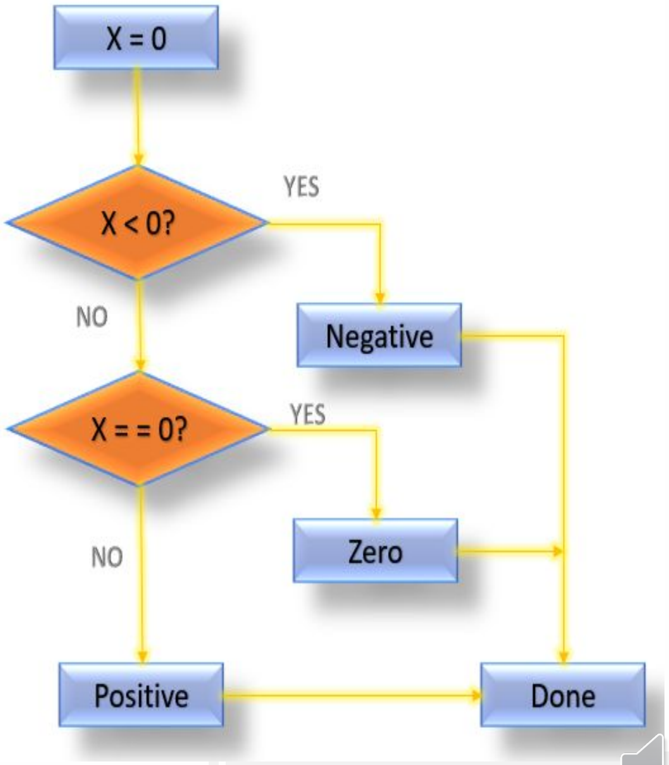


Conditional Execution : elif statement

- Multi – way decision statement
- Example:



```
x = 0

if x < 0 :
    print('Negative')
elif x == 0:
    print('Zero')
else :
    print('Positive')
```



Greet Your Friends!!

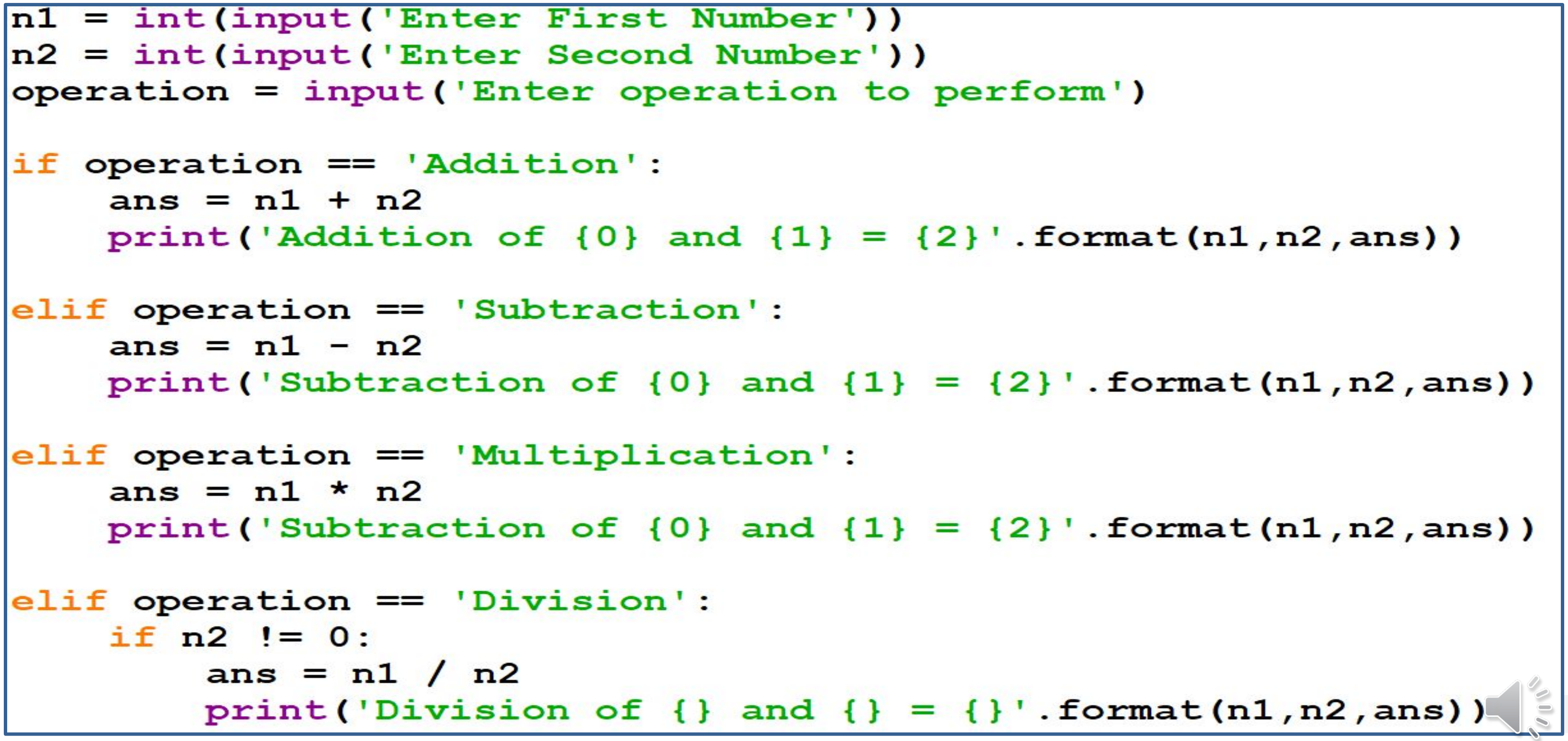
```
lang = input('Enter your Language')

if lang == 'English':
    print('Hello! Good Morning!!')

elif lang == 'Hindi':
    print('Namstey! Suprabhat!!')

else:
    print("I don't know the language") #string enclosed with ""
```

```
Enter your LanguageHindi
Namstey! Suprabhat!!
```

Parul® University

DIGITAL LEARNING CONTENT

Making Your Simple Calculator!!

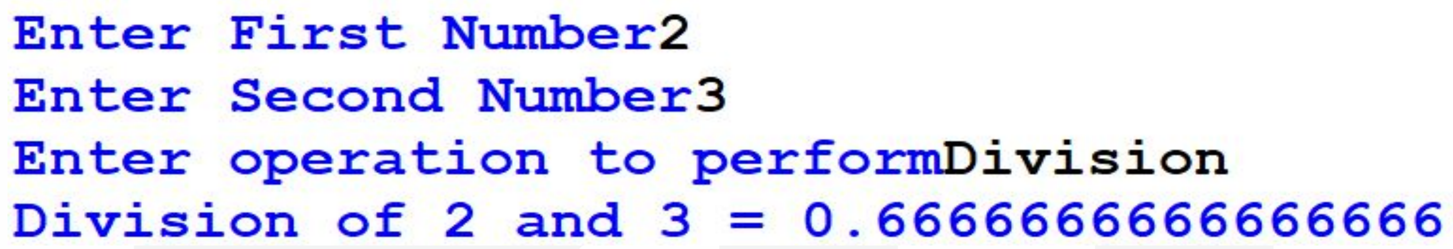


```
n1  = int(input('Enter First Number'))
n2  = int(input('Enter Second Number'))
operation = input('Enter operation to perform')

if operation == 'Addition':
    ans = n1 + n2
    print('Addition of {0} and {1} = {2}'.format(n1,n2,ans))

elif operation == 'Subtraction':
    ans = n1 - n2
    print('Subtraction of {0} and {1} = {2}'.format(n1,n2,ans))

elif operation == 'Multiplication':
    ans = n1 * n2
    print('Subtraction of {0} and {1} = {2}'.format(n1,n2,ans))

elif operation == 'Division':
    if n2 != 0:
        ans = n1 / n2
        print('Division of {} and {} = {}'.format(n1,n2,ans))
```

Parul® University

DIGITAL LEARNING CONTENT

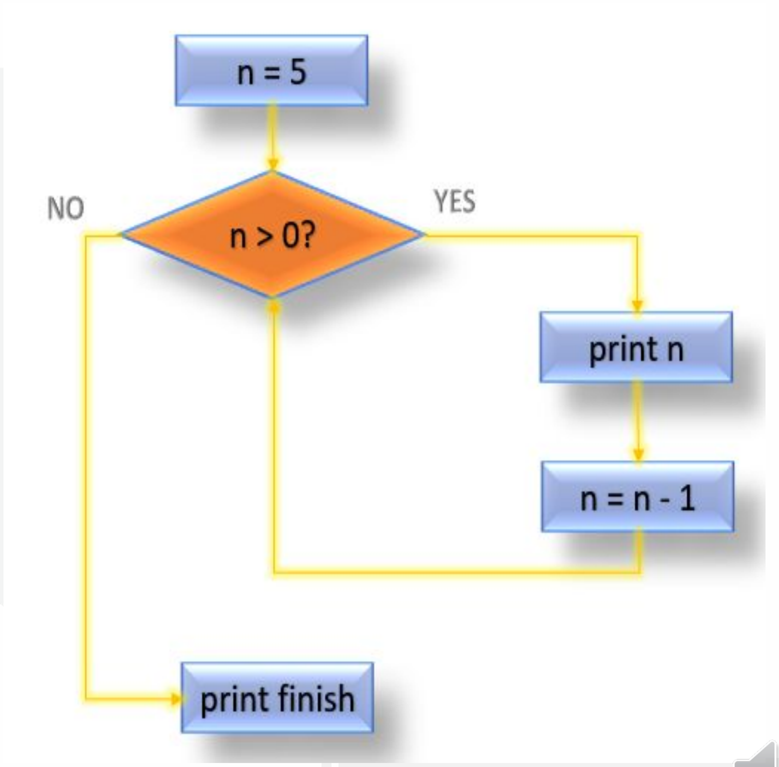
Making Your Simple Calculator!!



```
Enter First Number2
Enter Second Number3
Enter operation to performDivision
Division of 2 and 3 = 0.6666666666666666
```

Iterative Execution

- Execute statements repeatedly for several times
- Example: counter problem



Parul® University

DIGITAL LEARNING CONTENT

Iterative Execution: while statement

- Indefinite loop : keep going until a logical condition becomes False
- Syntax:

```
while  <condition> :
        #when condition is true repeat
```

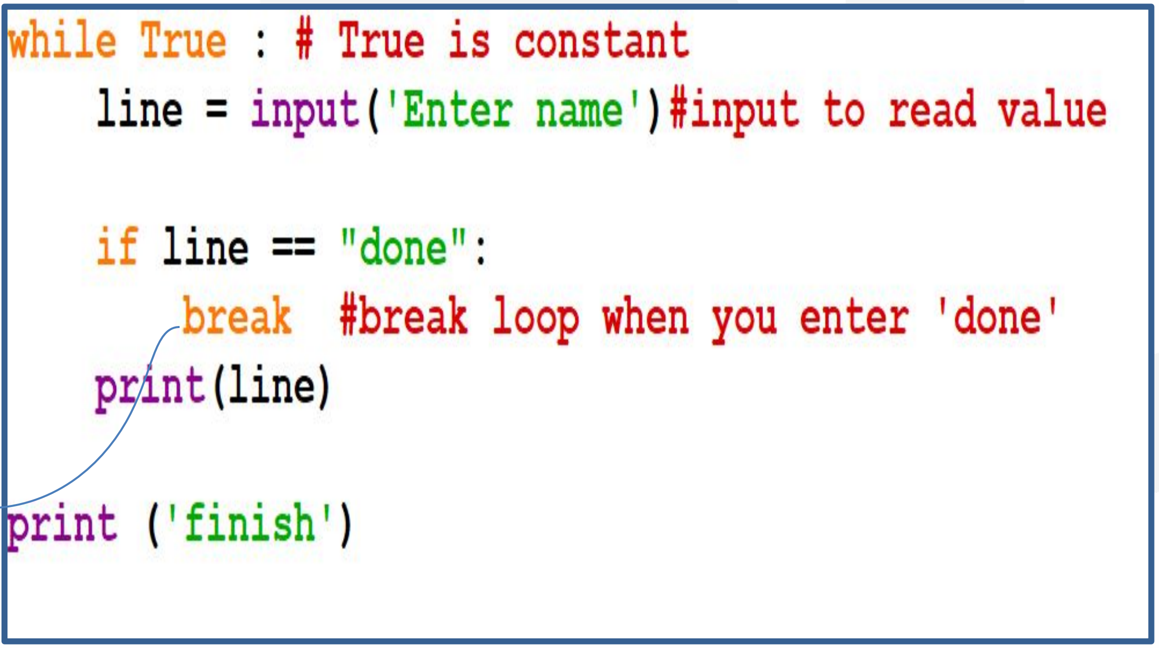
- Example:

```
n = 5

while n > 0 : # while statement

    print(n)
    n = n - 1

print ('finish')
```

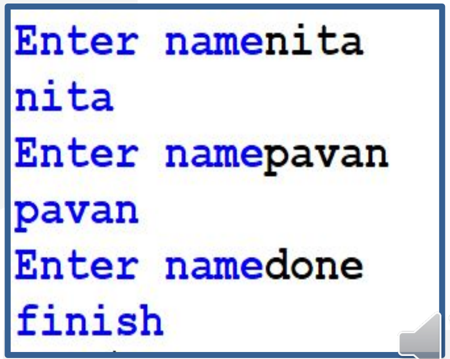
```
5
4
3
2
1
finish
```

Breaking the loop

- To end the current loop in between, use ‘ break ’ statement
- Example:



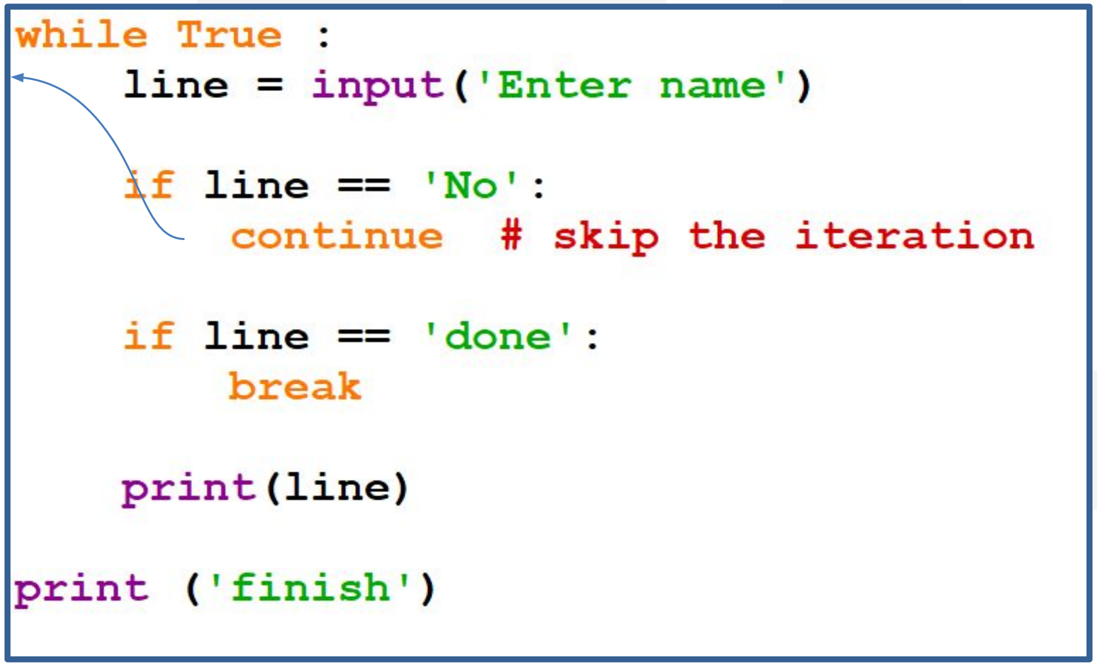
```
while True : # True is constant
    line = input('Enter name')#input to read value

    if line == "done":
        break  #break loop when you enter 'done'
    print(line)

print ('finish')
```



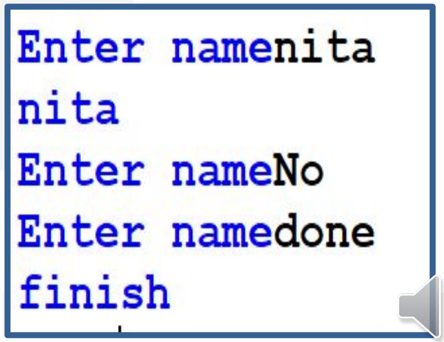
```
Enter namenita
nita
Enter namepavan
pavan
Enter namedone
finish
```

Skipping the iteration (Continue)

- To end the current iteration and start the next iteration of loop, use continue statement
- Example:



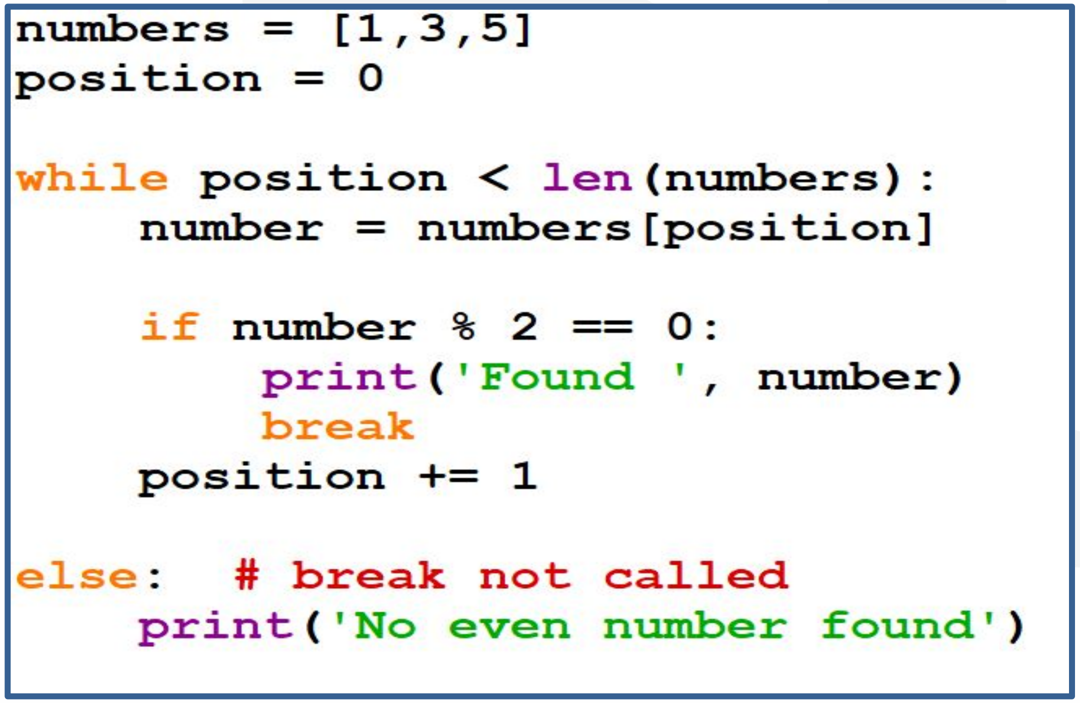
```
while True :
    line = input('Enter name')

    if line == 'No':
        continue   # skip the iteration

    if line == 'done':
        break

    print(line)

print ('finish')
```



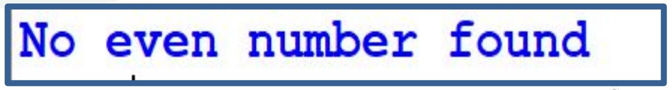
```
Enter namenita
nita
Enter nameNo
Enter namedone
finish
```

while with else statement

- If while loop ended normally without break call, control passes to an optional else



```
numbers = [1,3,5]
position = 0

while position < len(numbers):
    number = numbers[position]

    if number % 2 == 0:
        print('Found ', number)
        break
    position += 1

else:  # break not called
    print('No even number found')
```



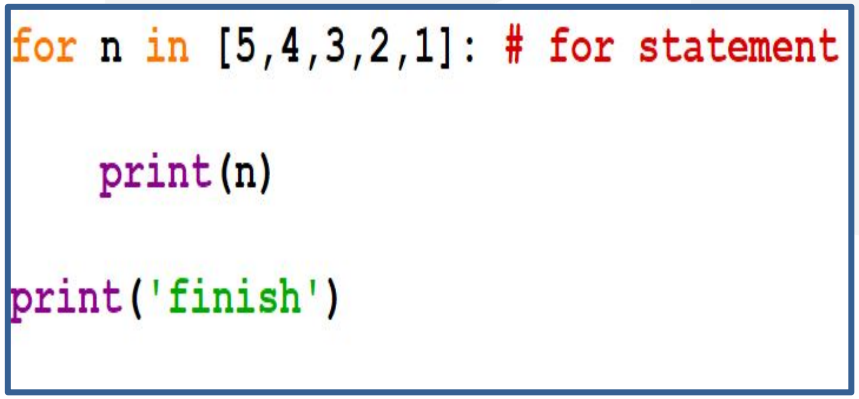
```
No even number found
```

Iterative Execution : for statement

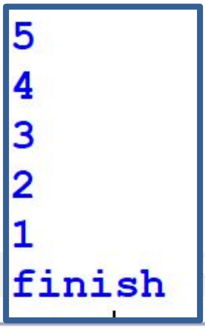
- Definite loop : Repeat for exact number of times
- Syntax:

```
for  <iterator_variable> in <range> :
        #repeat upto range
```

- Example:



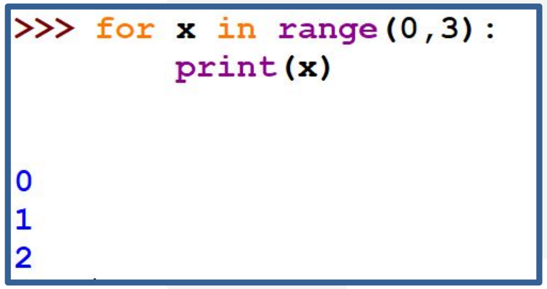
```
for n in [5,4,3,2,1]: # for statement

    print(n)

print('finish')
```



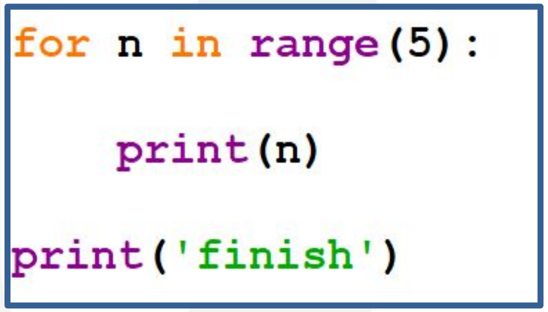
```
5
4
3
2
1
finish
```

Iterative Execution : for statement

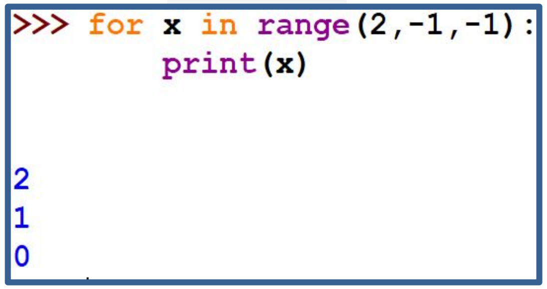
- If you do need to iterate over a sequence of numbers, the built-in function range() comes in handy
  - **range (start, stop, step)**



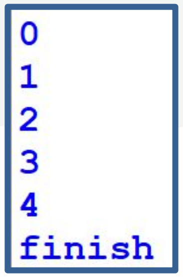
```
>>> for x in range(0,3):
        print(x)

0
1
2
```



```
for n in range(5):

    print(n)

print('finish')
```

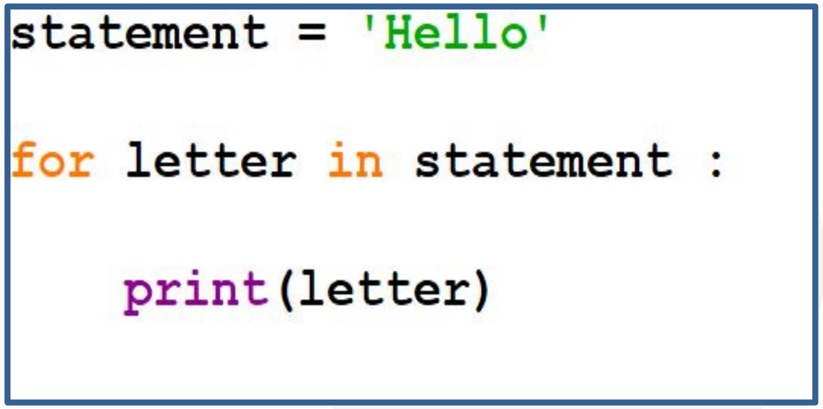


```
>>> for x in range(2,-1,-1):
        print(x)

2
1
0
```

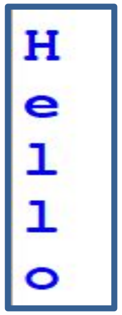


```
0
1
2
3
4
finish
```

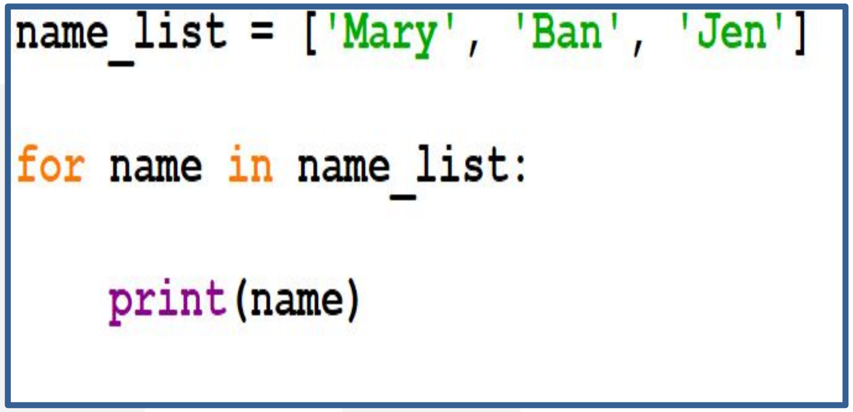
for loop with string



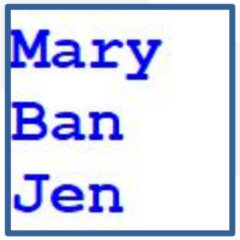
```python
statement = 'Hello'

for letter in statement :

    print(letter)
```



```
H
e
l
l
o
```



```python
name_list = ['Mary', 'Ban', 'Jen']

for name in name_list:

    print(name)
```



```
Mary
Ban
Jen
```

Parul® University

DIGITAL LEARNING CONTENT

Smallest number from list!!

```
num_list = [23,45,34,12,30]
smallest = None #None is constant having value as '0'

for number in num_list :

    if smallest is None :
        smallest = number

    elif smallest > number :
        smallest =number

print('The smallest number is',smallest)
```

```
The smallest number is 12
```

Fibonacci Sequence!!

```python
nterms = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0

if nterms <= 0:
   print("Please enter a positive integer")
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1, end = '   ')
       nth = n1 + n2
       n1 = n2
       n2 = nth
       count += 1
```

```
How many terms? 5
Fibonacci sequence:
0   1   1   2   3
```

Assignment

1. Explain the types control structures

2. Write a program to check number is prime or not

3. Write a program to check the entered number is Armstrong number or not

4. Differentiate between break and continue

5. Write a PYTHON program that prints 1 2 4 8 16 32 … 2^n

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in