

# Lecture 3

## PYTHON ESSENTIAL

BY

MOHAMMED ABDELFAH ALI

# Objectives

2

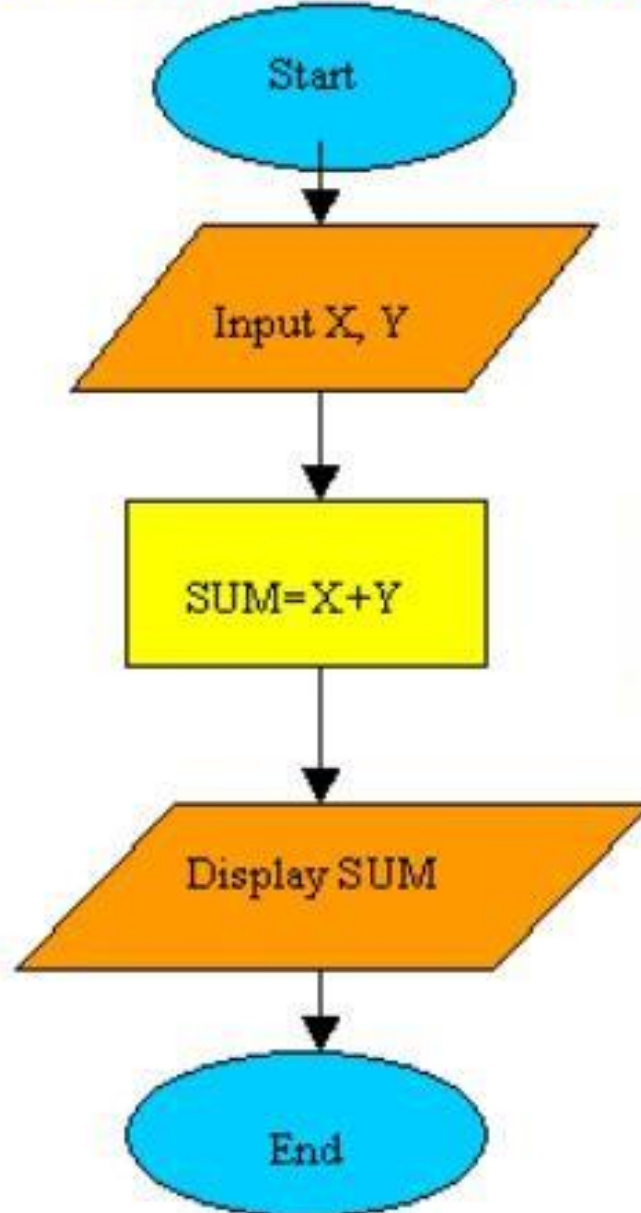
- ▶ Flowcharts
- ▶ Loop Statements
  - While Statement
  - For Statement
  - Break
  - Continue
  - Nested Loop
- ▶ Variable
  - Lists
  - Tuples
- ▶ Functions

# Flowcharts

# Flowcharts

4

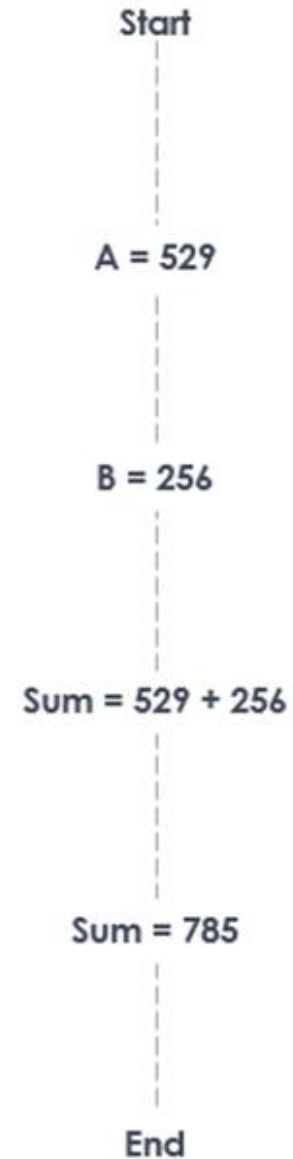
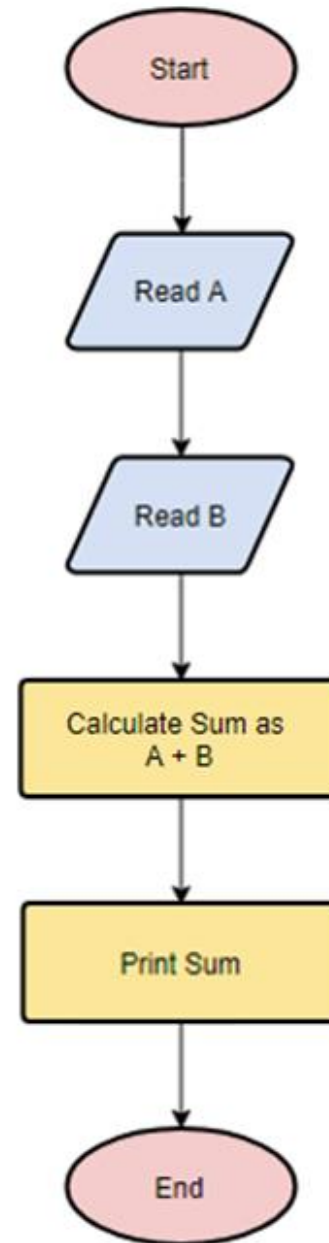
Flowchart **SUM**: To display total of two numbers



# Flowcharts

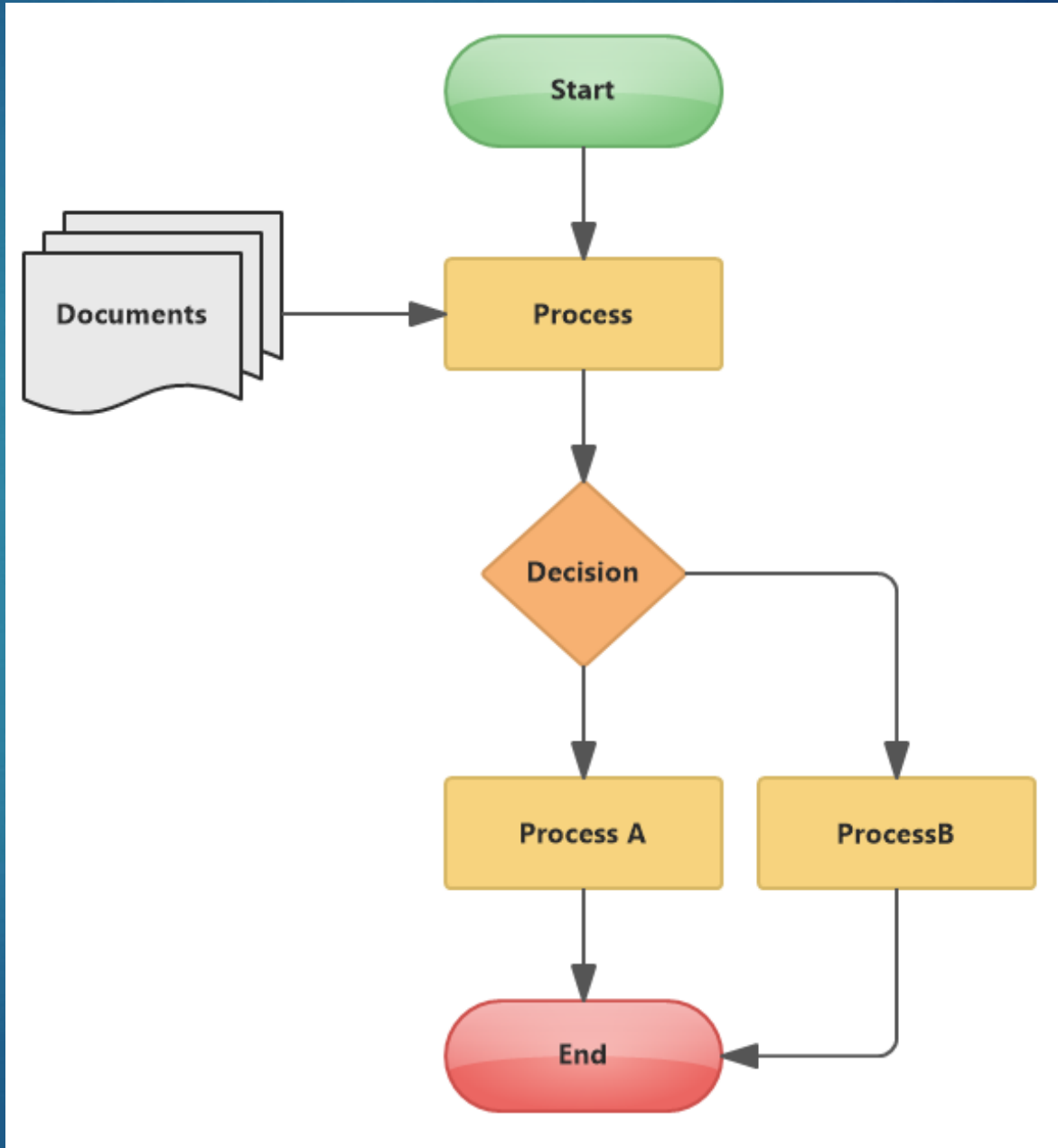
5

Find the sum of 529 and 256



# Flowcharts

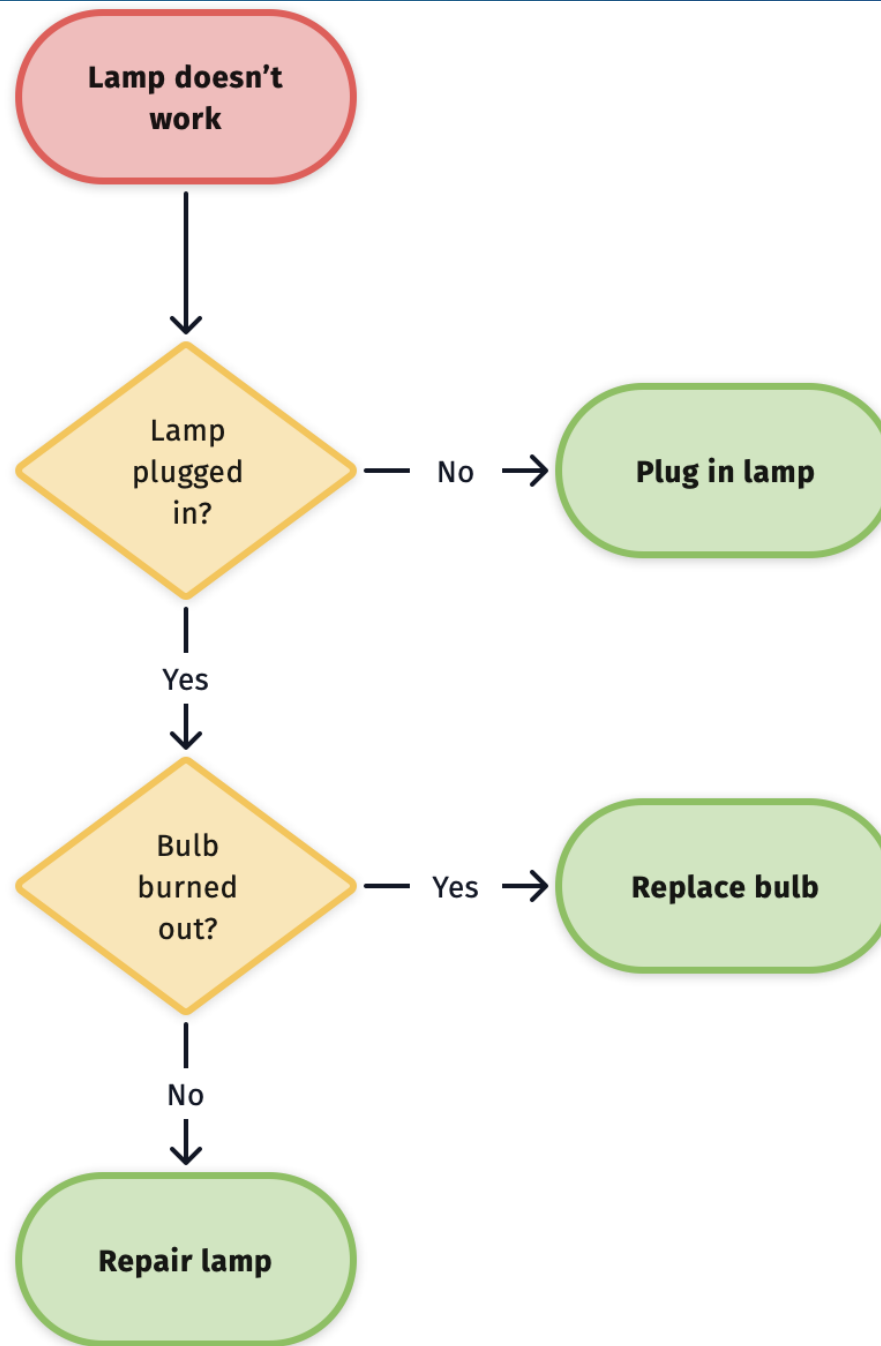
6





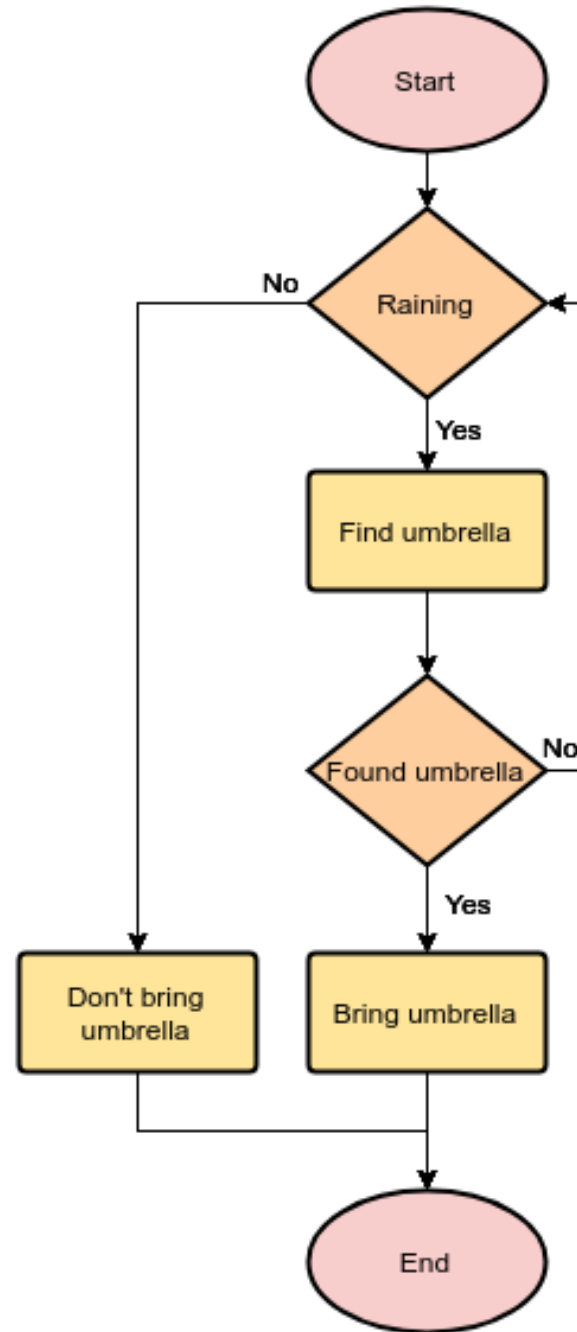
# Flowcharts

7



# Flowcharts

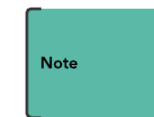
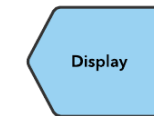
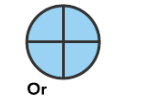
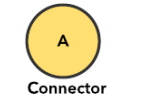
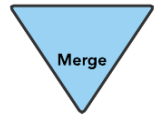
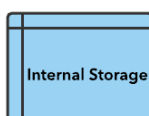
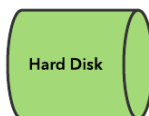
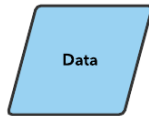
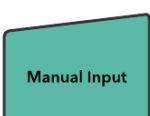
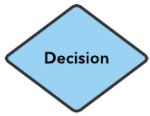
8



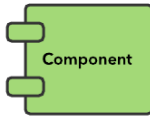
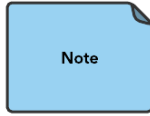
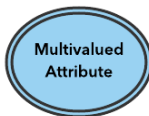


# Flowcharts

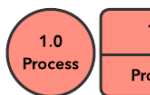
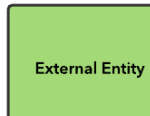
## Flowchart



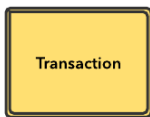
## UML



## Data Flow



## BPMN



## ERD

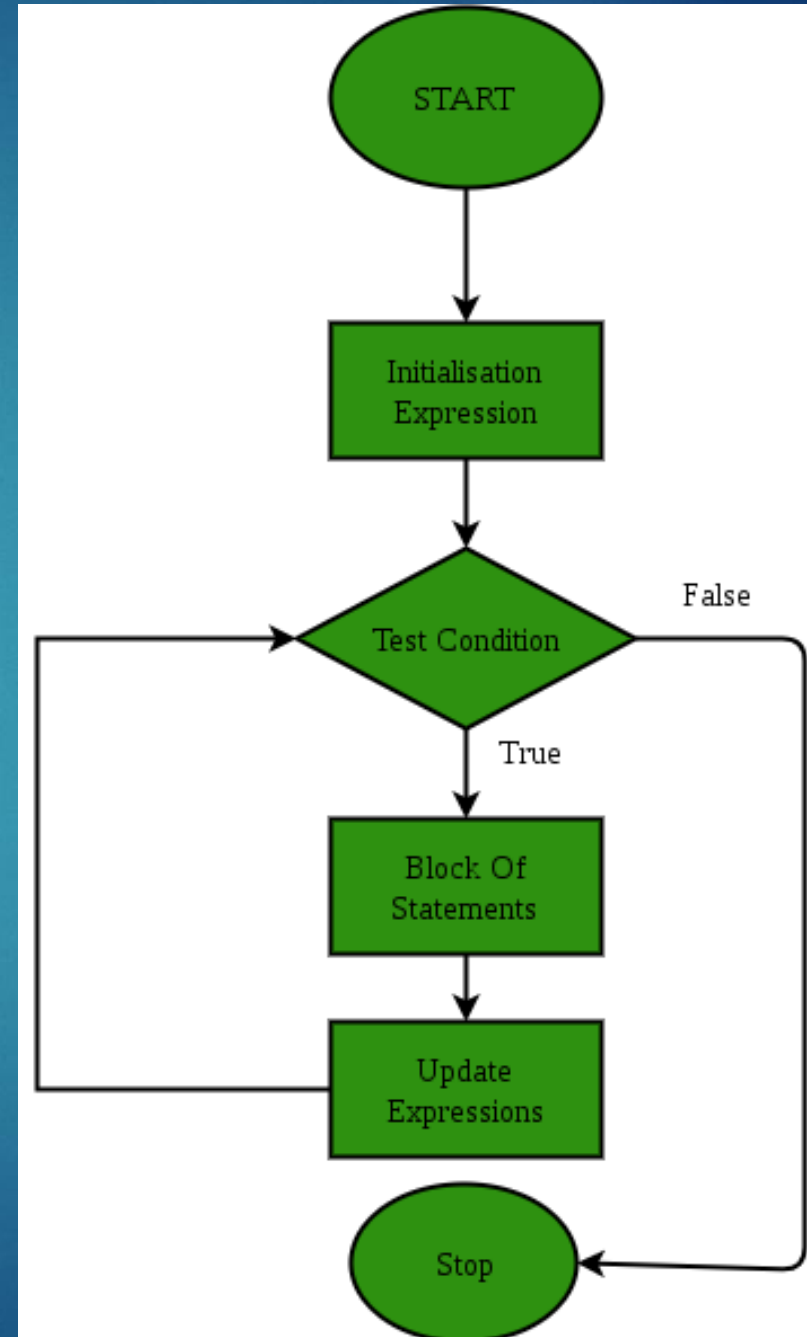
| Entity |       |      |
|--------|-------|------|
| Key    | Field | Type |
| Key    | Field | Type |
| Key    | Field | Type |



# Loop Statements

# Loop Statements

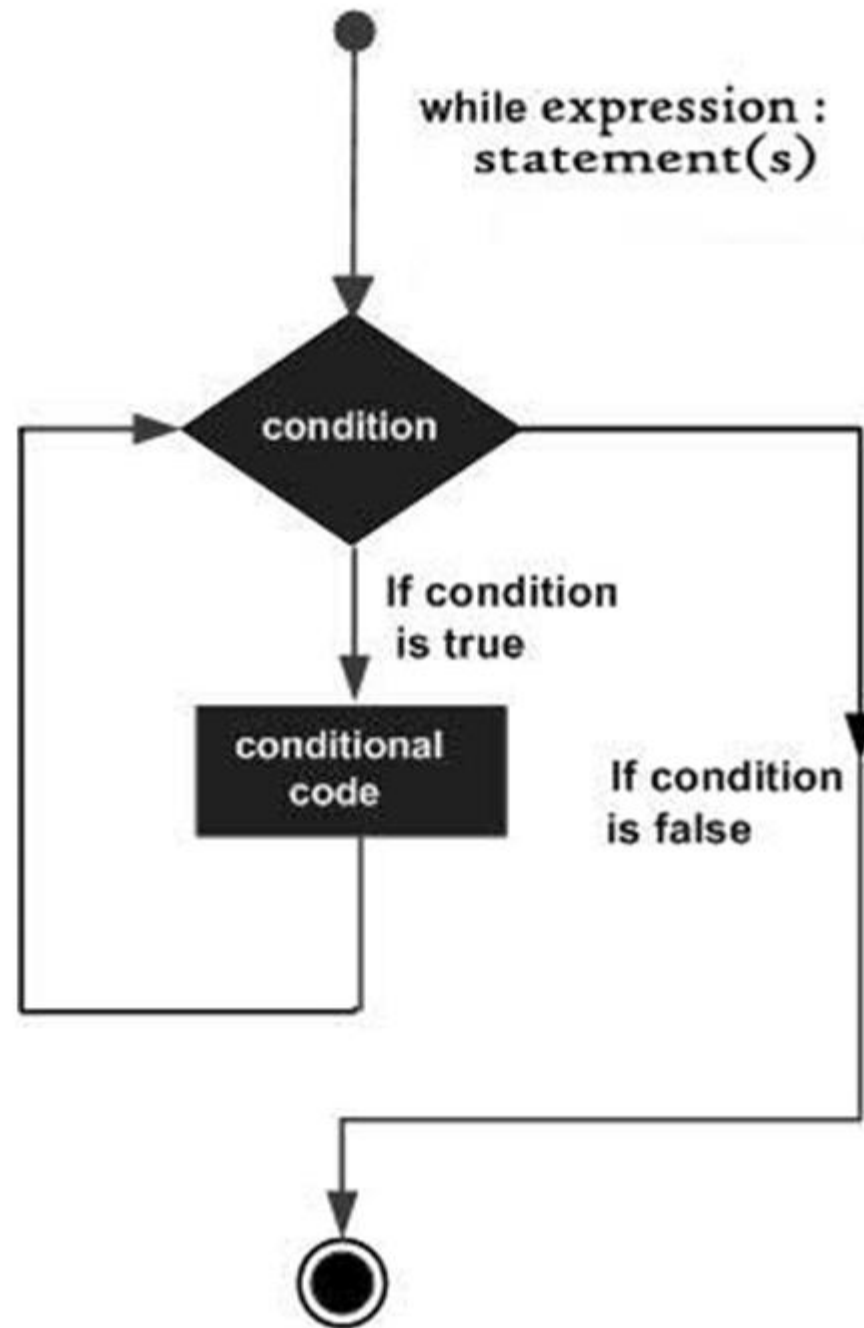
- ▶ A loop statement allows us to execute a statement or group of statements multiple times.
- ▶ The following diagram illustrates a loop statement .
- ▶ Example: **Print numbers from 0 to 10.**



# while Loop

while expression:  
statement(s)

12



# while Loop

13

```
count = 0
```

```
while (count < 9):
```

```
    print ('The count is:', count)
```

```
    count = count + 1
```

```
print ("Good bye!")
```

# Else Statement with While Loop

14

```
count = 0
```

```
while count < 5:
```

```
    print (count, " is less than 5")
```

```
    count = count + 1
```

```
else:
```

```
    print (count, " is greater than 5")
```



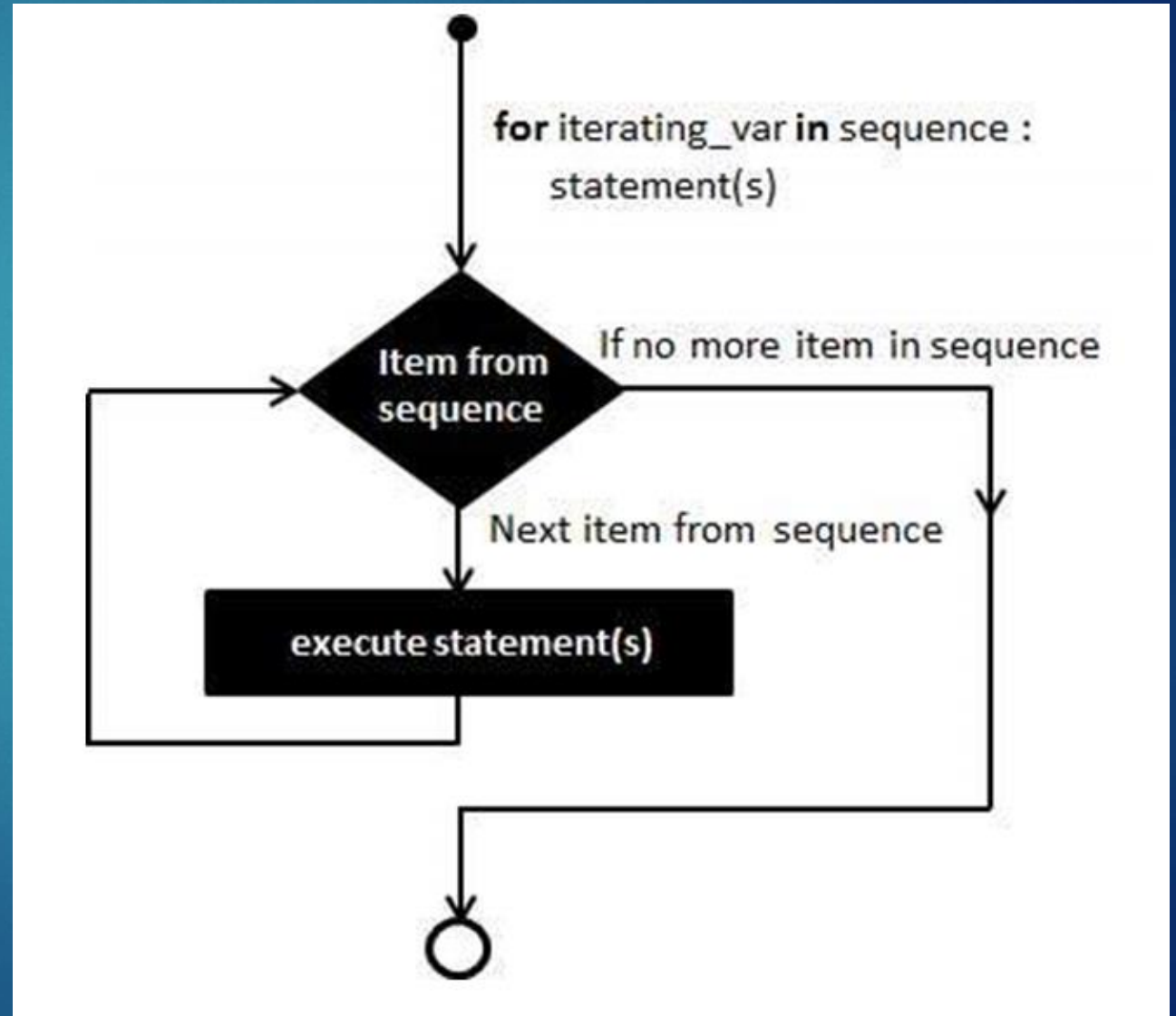
# for Loop

15

**for** iterating\_var **in** sequence:  
    statements(s)

Example:

- print letters of the word
- print items of the list



# for Loop

16

```
for letter in 'Python':    # First Example
```

```
    print ('Current Letter :', letter)
```

```
print ("Good bye!")
```

```
fruits = ['banana', 'apple', 'mango']
```

```
for fruit_Item in fruits:    # Second Example
```

```
    print ('Current fruit :', fruit_Item)
```

```
print ("Good bye!")
```

# Else Statement with For Loop

17

```
fruits = ['banana', 'apple', 'mango']  
  
for fruit_Item in fruits:      # Second Example  
    print ('Current fruit :', fruit_Item)  
  
else:  
    print("The list is finished")
```

# Nested loops

18

**for** iterating\_var **in** sequence:

**for** iterating\_var **in** sequence:

        statements(s)

    statements(s)

**while** expression:

**while** expression:

        statement(s)

    statement(s)

# Nested loops

19

```
i = 1
while(i < 3):

    j=0
    while(j <= 10):
        print ("j : ",j)
        j = j + 1

    print ("Iteration number : ", i)
    i = i + 1

print ("Good bye!")
```



# break statement

20

Terminates the loop statement and transfers execution to the statement immediately following the loop.

**for** letter **in** 'Python':   **# First Example**

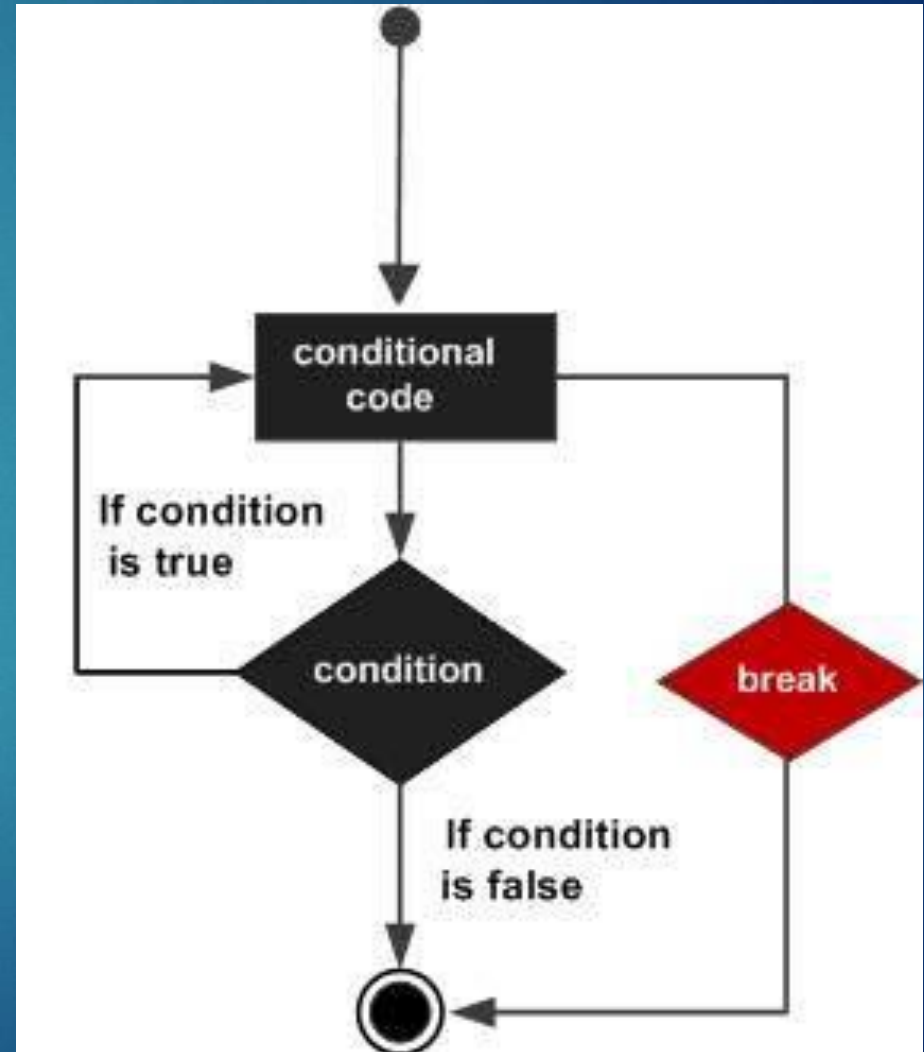
**if** letter == 'h':

**break**

**print** ('Current Letter :', letter)

## Execution output

Current Letter : P  
Current Letter : y  
Current Letter : t





# continue statement

21

Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

```
for letter in 'Python':    # First Example
```

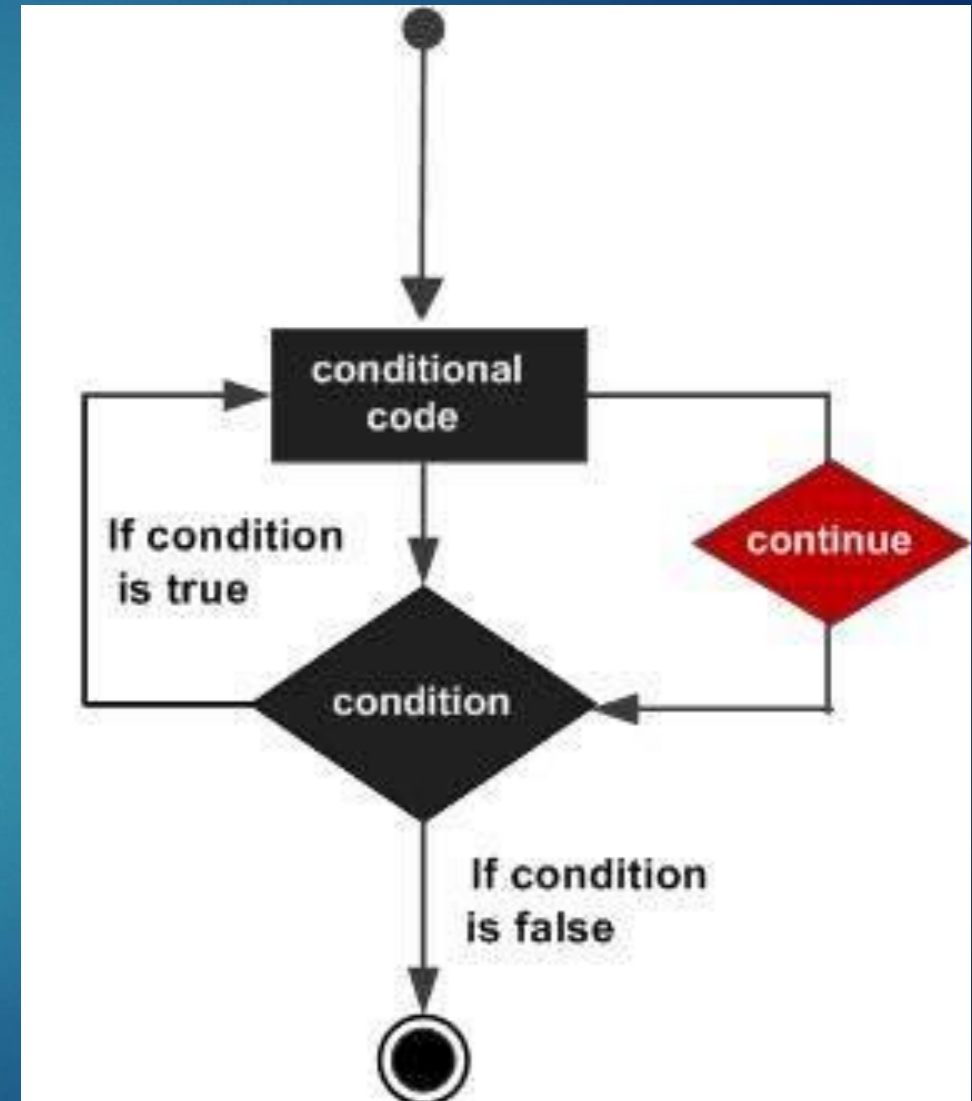
```
    if letter == 'h':
```

```
        continue
```

```
    print ('Current Letter :', letter)
```

## Execution output

```
Current Letter : P  
Current Letter : y  
Current Letter : t  
Current Letter : o  
Current Letter : n
```



# Thanks