Q1. EXPLAIN THE KEY FEATURES OF PYTHON THAT MAKES IT A POPULAR CHOICE?

A1. FOLLOWING ARE THE FOREMOST REASON THAT MAKES PYTHON SO POPULAR

1.IT HAS A WIDELY USE IN PROGRAMMING INDUSTRY.

2.IT HAS WIDE USE IN DATA INDUSTRY.

3.IT CONTAIN A VAST AND CONTINOUS GROWING ECOSYSTEM OF LIBRARIES (137000) THIS IS THE MOST IMPORTANT REASON OF ITS POPULARITY.

4.IT CAN BE USED IN MULTIPLE DOMAINS LIKE FRONTEND, BACKEND,DATA ANALYSIS ETC.

OTHER REASONS CAN BE AS FOLLOWS

1. IT IS EASY TO LEARN.
2. IT IS EASY TO READ.
3. IT IS VERSATILE.
4. IT HAS HUGE COMMUNITY OF ACTIVE USERS WHICH MAKES IT A POPULAR LANGUAGE.

Q2.DESCRIBE THE ROLE OF PREDEFINED KEYWORDS IN PYTHON AND PROVIDE EXAMPLE OF HOW THEY ARE USED IN A PROGRAM?

A2. KEYWORDS>> PYTHON KEYWORDS ARE SOME PREDEFINED OR RESERVED WORDS THAT HAVE SPECIAL MEANING AND CAN NOT BE USED AS IDENTIFIERS

THE KEYWORD CANNOT BE USED AS IDENTIFIERS LIKE VARIABLES, FUNCTIONS ETC.

THERE ARE 35 KEYWORDS IN PYTHON'''

THE MAIN ROLE OF KEYWORD IS TO DEFINE THE SYNTAX OF THE CODE.

```
help('keywords')
```

```
Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

```
#BELOW ARE EXAMPLES OF HOW TO USE KEYWORDS

# here is an example of (KEYWORD>>if) if is use to make conditional statement

if 10>6:
  print("greater")
```

```
greater
```

```
marks=25
if marks>20:
  print("execellent")
```

```
execellent
```

```
#here is an example of (KEYWORD>>  in) IT IS USE TO CHECK WHETHER A VALUE IS PRESENT IN LIST OR RANGE

A=["SAURABH",1,4.5, "DELHI"]
if "SAURABH" in A :
  print ("true")
```

```
true
```

```
#example of if and else keyword

age=18
if age>18:
  print('ride')
else:
    print('dont ride')
```

```
dont ride
```

Q3. compare and contrast the mutable and immutable objects in python with examples?

A3. The main difference between mutable and immutable objects is that mutable objects can be altered after their creation and immutable objects can not be altered.

mutable means to edit or change

immutable means which we can not change

```
#BELOW WE CAN UNDERSTAND THE DIFFERENCE B/W BOTH THE CONCEPTS

#MUTABILITY
a=[20,30,40,3.5,4+6j,"delhi"]
a[0]
```

```
20
```

```
a[3]
```

```
3.5
```

```
#here we change the assign vale of a[3]
a[3]=4.5
a
```

```
[20, 30, 40, 4.5, (4+6j), 'delhi']
```

```
#IMMUTABILITY
b="saurabh"
b
```

```
'saurabh'
```

```
b[1]
```

```
'a'
```

```
b[1]='o'# we can not change the 'str'
b
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-9-0671c4295b16> in <cell line: 1>()
----> 1 b[1]='o'
      2 b

TypeError: 'str' object does not support item assignment
```

FROM THE ABOVE ILLUSTRATION WE CAN UNDERSTAND THAT LIST IS A TYPE OF MUTABLE OBJECT BECAUSE IT SUPPORT ITEM ASSIGNMENT WHILE STR IS IMMUTABLE BECAUSE IT DOES NOT SUPPORT ITEM ASSIGNMENT.

HOWEVER IT IS VERY IMPORTANT TO KNOW WHILE WRITING A CODE WHICH OBJECT IS MUTABLE OR NOT SO THAT WHILE WRITING THE DATA OR CODE WE CAN CHANGE THE DATA OR NOT.

Q.4DISCUSS THE DIFFERENT TYPE OF OPERATORS IN PYTHON AND PROVIDE EXAMPLES OF HOW THEY ARE USED?

A.4 PYTHON OPERATORS>> These are special keywords or symbols which are use to perform operation on values or variables.

Because we want to manage, do computation and make decision using data.

```
# for example we want to add two numbers
a=50
b=30
a+b # here (+) symbol is an operator which is helping us adding values
```

```
80
```

these are the categories of python operators

Python
Operators
Arithmetic Operators
Relational Operators
Assignment Operators
Logical Operators
Bitwise Operators
Membership Operators
Identity Operators

```
# airthmetic operators >> +, -, /, *, %
a=50
b=30
a+b
```

```
80
```

```
a-b
```

```
20
```

```
a*b
```

```
1500
```

```
a/b
```

```
1.6666666666666667
```

```
# comparision operator>> this compares two values and returns boolean value
5==5# here we use two times (==)
```

```
True
```

```
5!=5# this is not equals to
```

```
False
```

```
5>10
```

```
False
```

```
5<=10
```

```
True
```

logical operators

two types AND,OR

```
#AND
True and True
```

```
True
```

```
True and False
```

```
False
```

```
False and True
```

```
False
```

```
False and False
```

```
False
```

```
# or operator
True or True
```

```
True
```

```
True or False
```

```
True
```

```
False or True
```

```
True
```

```
False or False
```

```
False
```

```
# assignment operator
a=100
a
```

```
100
```

```
a+50
```

```
150
```

```
b=60
b+20
```

```
80
```

```
b
```

```
60
```

```
b=60
b+=20
```

```
b
```

```
80
```

membership operator

```
a="delhi"
'd' in a
```

```
True
```

```
'd' not in a
```

```
False
```

```
a="delhi"
'del'in a
```

```
True
```

identity operator

it compares the location of two variables in the memory

```
# example
a=5
b=6
a is b
```

```
False
```

```
a is not b
```

```
True
```

bitwise operator ; it operates at a bit level

```
#example
10&10
```

```
10
```

```
bin(10)
```

```
'0b1010'
```

```
3|5
```

```
7
```

```
bin(3)
```

```
'0b11'
```

```
bin(5)
```

```
'0b101'
```

```
bin(7)
```

```
'0b111'
```

```
#negation
~5
```

```
-6
```

```
#bitwise xor operator
```

```
5^3
```

```
6
```

```
bin(5)
```

```
'0b101'
```

```
bin(3)
```

```
'0b11'
```

```
bin(6)
```

```
'0b110'
```

shift operator>> two types left shift and right shift

It shift the bits by left or right by a specified number of position by filling the zeros.

```
#example left shift
35<<5
```

```
1120
```

```
bin(35)
```

```
'0b100011'
```

```
bin(1120)
```

```
'0b10001100000'
```

```
#example right shift
20>>2
```

```
5
```

```
bin(20)
```

```
'0b10100'
```

```
bin(5)
```

```
'0b101'
```

Q7.DESCRIBE DIFFERENT TYPES OF LOOPS IN PYTHON AND THEIR USE CASES WITH EXAMPLES?

loop statement is which allows you to execute block of code repeatedly

loops are of two types

while loop

for loop

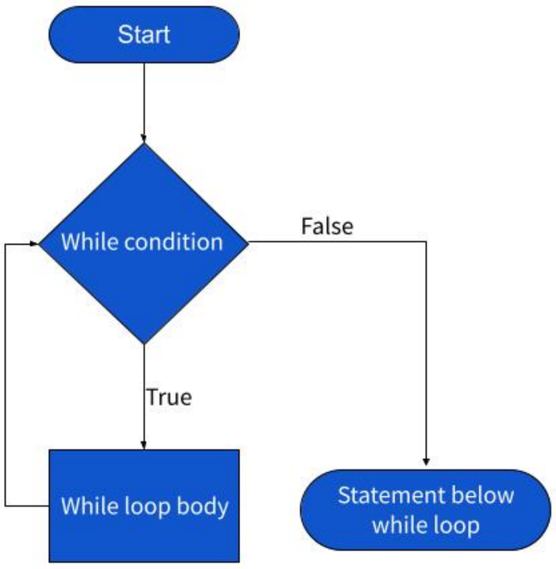
```
#example while loop
n=9
i=1
while i<n:
  print(i)
  i=i+1
```

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

```
n=5
i=2
while i<n:
  print(i)
  i=i+1
```

```
2
3
4
```

BELOW IS THE TEST CONDITION OF WHILE LOOP



```
# WHILE LOOP WITH ELSE
n=9
i=1
while i<n:
  print(i)
  i=i+1
else:
  print("loop executed successfully")
```

```
1
2
3
4
5
6
7
8
loop executed successfully
```

```
##break statement
n=9
i=1
while i<n:
  print(i)
  i=i+1
  if i==5:# break statement exits the loop
    break
else:
  print("loop executed successfully")
```

```
1
2
3
4
```

```
## continue statement
n=9
i=1
while i<n:
  i=i+1
  if i==4:
    continue # continue >> skips the iteration
  print(i)
else:
  print("loop executed successfully")
```

```
2
3
5
6
7
8
9
loop executed successfully
```

FOR LOOP== IT IS USED TO ITERATE OVER A SEQUENCE OF ELEMENTS

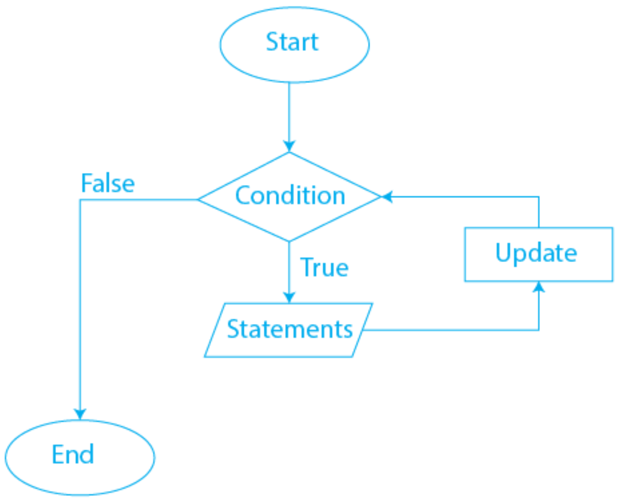
```
# FOR LOOP
for i in 'saurabh':
  print(i)
```

```
s
a
u
r
a
b
h
```

```
a='saurabh'
for i in a:
  print(i)
```

```
s
a
u
r
a
b
h
```

below is the test condition of FOR LOOP



```
#ELSE block
l=[2,4,5,'saurabh','data analytics']
for i in l:
  print(i)
else:
  print('this will execute when for loop ends without break')
```

```
2
4
5
saurabh
```

```
data analytics
this will execute when for loop ends without break
```

```
#break in for loop
l=[2,4,5,'saurabh','data analytics']
for i in l:
  if i==5:
    break
  print(i)
else:
  print('this will execute when for loop ends without break')
```

```
2
4
```

Q5.EXPLAIN THE CONCEPT OF TYPECASTING IN PYTHON WITH EXAMPLES ?

A5. TYPE CASTING - THE PROCESS OF CHANGING A DATATYPE OF A VARIABLE OR OBJECT IN PYTHON IS CALLED AS TYPECASTING OR TYPECONVERSION.

IT IS IMPORTANT BECAUSE SOMETIMES WHILE EXECUTING OR COMPUTATION WITH OPERATORS,THERE CAN BE A MISMATCH BETWEEN DATATYPES

```
# HERE IS THE EXAMPLE OF TYPECASTING
4+5
```

```
9
```

```
a='saurabh'
type(a)
```

```
str
```

```
'2'+5 # here type of '2' is str thats why it is not executing
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-353d714f42c2> in <cell line: 1>()
----> 1 '2'+5

TypeError: can only concatenate str (not "int") to str
```

```
# now we can change the type of '2' by using typecasting
a='2'
b=5
a+b
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-6-9dd437232c8e> in <cell line: 4>()
      2 a='2'
      3 b=5
----> 4 a+b

TypeError: can only concatenate str (not "int") to str
```

```
type(a)
```

```
str
```

```
int(a)+b# here we change the str '2' to integer by using the concept of type casting
```

```
7
```

```
#string to integer
a='2'
print(type(a))
print(type(int(a)))
```

```
<class 'str'>
<class 'int'>
```

AS SHOWN ABOVE THE EXAMPLE OF TYPECASTING, SIMILARLY IT CAN BE USE FOR FLOAT TO INT OR INTEGER TO FLOAT VARIABLES.

Q6 HOW DO CONDITIONAL STATEMENT WORK IN PYTHON? ILLUSTRATE WITH EXAMPLES.

A6. CONDITINAL STATEMENTS - CONDITIONAL STATEMENT IS A PART OF FLOW CONTROL MECHANISM.

IT HELPS US TO CODE DECISIONS BASED ON SOME PRECONDITION.

IF

IF ELSE

IF ELIF ELSE

NESTED IF ELSE

THESE ARE THE EXAMPLES OF CONDITIONAL STATEMENTS

```
# LET SEE THE USE OF THEM
# if statements
#if condition is true:
  #print this block of code
```

```
marks= 90
if marks>80:
  print('you are execellent')
```

```
you are execellent
```

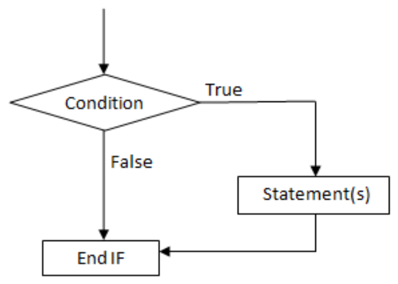


fig: Flowchart for if statement

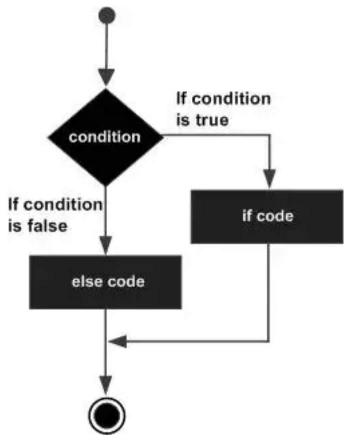
```
# in practical world, multiple conditions exis
# use of if else
marks= 90
co_curr= True
```

```
if(marks>80 and co_curr==True):
```

```
you are execellent
```

```
marks= 75
co_curr= True
if(marks>80 and co_curr==True):
  print('you are execellent')
else:
  print('you are average')
```

```
you are average
```



```
num=8
if num%2==0:
  print('even')
else:
  print('odd')
```

```
even
```

```
#multiple conditiond
#elif
a=1
if a>100:
  print('greater')
elif a<100:
  print('lesser')
else:
  print(a)
```

```
lesser
```

```
a=1000
if a>500:
  print('huge number')
elif a>100:
  print('greater')
elif a<100:
  print('lesser')
else:
  print(a)
```

```
huge number
```

```
#nested if else
#example
name=input('please enter name')
email=input('enter valid email')
password=input('enter password')

if name=='':
  print('enter valid name')
else:
  if "@" not in email:
    print('enter valid mail')
  else:
    if len(password)<4:
      print('enter valid password')
    else:
        print ('successfully registered')
```

```
please enter nameshubham
enter valid emailsau@ghji
enter passworddghyu
successfully registered
```

assignment concluded

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```