```
# -*- coding: utf-8 -*-
"""Untitled0.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1iCbej1ijyXHPIsqf5FWu5jKTX_zRG5e_

1. Following are the Key Features of Python:
*  widely used in the industry.
*  simple and easy to learn thats why it makes easy for the beginners to
learn this language.
* Python has the large standard library with the structured codes and
modules to perform the taskes.
* Its flexibility allows it to be used across various domains, from web
development to data analysis and automation.

2.  Predefined keywords are the backbone of the python programmig which
control the flow,logic, structure and behaviour of the programme.
for eg:-
"""

if 19>10:
   print("19 is greater than 10") #here "if" is the predefined key word.

if 20<55:
  print("20 is lessar than 55")

"""3. Mutable Objects:
* mutable objects can be changed, let's say we have a list having some
items, those items can be replaced by other item intstructed by the code.
* Memory location remains the same.

for eg:-
"""

list__cont = [11,12,13,17,25,30,29]
list__cont[0]

list__cont[-6]

list__cont[-6]= "kalhan"

list__cont[0: ]

#Hence '12' in the list has been replaced or changed by 'Kalhan' only the
items in the list can be changed or replaced.

"""   Immutable Objects:
   * these are the objects which cannot be changed or replaced.
   let's we have a list having an item only, the letters of the item
cannot be changed.
   * new memory location is created for that.

   for eg:-
"""
```

```
list__cont = ["kalhanjotshi"]# "kalhanjotshi"is a string of which letter
cannot be changed.
list__cont[3] = r
list__cont

"""see. an error has occured. #strings are immutable objects

4. TYPE OF OPERATORS

i. Aerthamatic Operatrs: +,-,*,/
For eg:-
"""

a=9
b=12
add = a+b
add

subtract = b- a
subtract

multiply = a*b
multiply

divide = b/a
divide

"""ii. Modolous operator>>> "%" for the remainder."""

a = 25
b =7
a%b

"""iii. Exponention operator>>> "**"  means, power eg: 3^2= 9"""

3**2

2**100

"""iv. Floor Operator>>> "//" means removing the decimal. for eg. if
19/5= 3.8 it will be 3.

"""

8//9

9//8

5//19

19//5

"""v. Comparison Operators>>> "==";"!=";">=";"<=". """

a=7
b=2
a==b
```

```
a!=b

a>=b

a<=b


"""vi. Logical Operators>>> "and" & "or"

"""

True and True

True and False

False and True

False and False

True or True

False or False

True or False

False or True

"""vii.Assignment Operators:

"""

a=17
b=3
a+b

a+=3
a

a+b

b-=3
b

a+b

a+=10
a

a+b

b+=10
b

a+b

"""Viii. Membership operator:- "in" &"not in"."""
```

```
a ="universe"

"i" in a

"i" not in a

"m" in a

"""ix. Identity Operator:-

"""

a=9
b=2
a!=b

a is b

b != a

"""x. Bitwise Operator:- "&" and "|":

result is obtained on the basis of their binary codes.
"""

10 & 10

11 & 15

bin(11)

bin(15)

8|11

bin(8)

bin(11)

"""xi. Bitwise Xor operator:- "^"
"""

7^10

bin(7)

bin(10)

bin(13)

"""xii. Shift operators:

- Right shift operator>>> remove the number of elements from the right by
the number which is at "lesser than" place.

"""

13 >> 3
```

```
bin(13)  # end codes i,e 101 of 0b1101 has been removed so, 1.

"""- Left shift operator>>>number which is in place of 'greater than' is
the times the binary 0's is added to its code."""

15 << 2

bin(15)

# 1111 is the code and 00 is added because 2 is the number of times so
111100, so this is the binary code of 60.

"""5. Type casting>>> when the value of the the variable is missmatched
by\n mistake and throws it error, in order to correct it Type casting is
used.'

There Are two types of type casting: "implicit typecasting" and "explicit
type casting"

for eg:-
"""

#EXPLICIT TYPE CASTING
a = "7"
b = 5
a+b

#see it throws an error

type(a)

#type of a if is string by mistake now it is corrected by type casting...

a = "7"
b = 5
int(a)+b

#see it converts string into an integer then the code is executed
correctly.

b = 9.8
type(b)
float_value=int(b)
type(float_value)

#integer to float

b=57
type(b)
int_value=float(b)
type(int_value)

#IMPLICIT TYPE CASTING: Python automatically understands the data type:

a=7
b=6
a+b
```

```
b = 5.5
a+b

#it automatically understands.

#conecatination:

a="kalhan jotshi"
b= 8
a+b #it will throw an error as one is string and other is integer

a= "Kalhan Jotshi"
b= "Rahul Handoo"
a+b

a = "kalhan"
b = "Jotshi"
a + b

"""6. Conditional statements>>> When a condition is given it helps to
code decision.:  "if";"if else" ; "if elif else"; "nested if else"
"""

# If statements
a=7
if a>3:
  print("a is greater")

if a<11:
  print("a is lessar")

lecture_5 = "easily understood"
if lecture_5:
  print(" because of good teaching")

weather = "sunny"
if weather:
  print("i will go for swimming")

number = 34
if number % 2 == 0:
  print(" number is even")

number = 27
if number == 27:
  print("number is multiple of 3")

marks = 80
co_act = True
if ((marks>35) and (co_act == True)):
  print(" student is all rounder")

#if else statement:

age = 21
if (age >= 21):
  print("he will be allowed to drink")
```

```
else:
  print("he will be kicked out of the bar")

gun = "licence"
if (gun == "licence"):
  print("he will be allowed to have it")
else:
  print("he will be behind the bars")

licence_age = 15
if (licence_age>=18):
  print("he will get the licence")
else:
  print("he will not get licence")

#if elif else statement:
a= 34.5
if (a==35):
  print("you are passed")
elif(a<35):
  print("you are failed")
elif (a>=35):
  print("passed")
else:
  print("negative")

score = 90
if((score>90) and (score<=100)):
  print("1st division")
elif((score>=80) and (score<=90)):
  print("2nd division")
elif((score>=70) and (score<=80)):
  print("3rd division")
elif((score>=60) and (score<=70)):
  print("4th division")
elif(score>=50) and (score<=60):
  print("5th division")
else:
  print("failed")

a=int(input("enter a num."))
if a>100:
  print("the num. is greater than hundred")
elif 0<=a<=100  :
  print("num is positive and greater than zero")
else:
  print("the num is negative")

#creating a login page
name = input("please enter a name")
email = input("please enter an email")
password = input("please enter a password")

if name=="":
    print("please enter a name. it can't be empty")
else:
    if "@" not in email:
        print("enter a valid email address")
```

```
    else:
        if len(password)>6:
            print("invalid password")
        else:
            print("login successful")

"""7. Types of Loop>>> while Loop and For loop.
- while loops are used until the condition of the code is true.
- for loop are used
"""

#while loop
a=10
b=2
while b<10:
  print(b)
  b=b+1

count_num = 4
while count_num>-1:
  print(count_num)
  count_num = count_num - 1
```