01 my frist program

In [14]:
```
print(2+3)
print("hello world")
```

```
5
hello world
```

02 operators

In [15]:
```
print(2+1)
print(3-1)
print(6/2)#without floating values kaleya
print(2*3)
print(13%2)
print(6//2)#without floating values ka leya
print(2**4) #power kaleya out are 16
print(3**2/2*3/3+6-4)
#(PEMDAS) it is all about operation rules it can be solve one by one and the sequenc
#parenthesis Exponents Mutiply Divide Addition Subtraction
```

```
3
2
3.0
6
1
3
16
6.5
```

03_strings

In [16]:
```
print('test for single quotes')
print("test for quotes")
print('''test for tripple quotes''')
print("what's")
```

```
test for single quotes
test for quotes
test for tripple quotes
what's
```

04_comment

In [17]:
```
print("how are you")
print("we are learninng python with hamad ur rehman")
print(2+6)
#ctrl+/ are used for commenting the lines before commenting we will select the line
print(2+8)#print operatord funcations with numbers
```

```
how are you
we are learninng python with hamad ur rehman
8
10
```

05_input_variables

In [18]:
```
fruit_basket=input("what is your favourite fruite")
print(fruit_basket)
x=9
print(x)
```

```
what is your favourite fruite4
4
9
```

input_varable_02

In [19]:
```
#variable: object containing specific values
x = 5
print(x) #numeric or integer variable

y="we are learing python with hammad"  #string variable
print(y)

x=15
print(x)

x=x+10   #x=15+10
print(x)

#types/class of vaiable
type(x)
print(type(x))   #output: int class

print(type(y))  #output: str class

#print_types_class
#Rules to assign a variable:
# 1: the vaiable should contain letter number underscore
# 2: do not start with numberd
#3: space are not allowed
# 4: do not use keyword used in funcation (break,mean,media e.t.c)
# 5: short and descriptive
# 6: case senstivity(lowercase, uppercase letter lowercase letter should be used)

#input functions
fruit_basket=input ("what is your favourite fruite ")
print(fruit_basket)
#these two lines are called cin labrabry

#input second funcation for example just
name=input("enter your name: ")
greeting=("hello")
print(greeting,name)

# another way of second funcation
name=input("enter your name: ")
print("hello",name)

#thrid stage of input funcation
name=input("what is your name")
age=input("how old are")
greeting="hello"
print(greeting,name," , you are still young")
```

```
5
we are learing python with hammad
15
25
<class 'int'>
<class 'str'>
what is your favourite fruite 6
6
enter your name: hamad
hello hamad
enter your name: ali
hello ali
what is your nameahmad
how old are23
hello ahmad  , you are still young
```

06_conditions

In [20]:

```
#logical operators are either "true or false" or "yes or no" or "0 or 1"
# equal to                  ==
# not equal to              !=
# less than                 <
# greater than              >
# less than and equal to    <=
# GREATER than and equal to >=

# is 4 equal to 4
print(4==4)   #output are ture
print(4!=4)   #output are false
print(4>3)   #output are ture
print(3>6)   #output are false
print(3<=4)   #output are ture
print(5>=4)   #output are ture
#application of logical operators
hamad_age=4
age_at_school=5
print(hamad_age==age_at_school)

#inout funcaton and logical operator
age_at_school=5
hamad_age=input("how old is hamad") #input funcation
hamad_age=int(hamad_age)          # it is used to convert string to int
print(type(hamad_age))
print(hamad_age==age_at_school)
```

```
True
False
True
False
True
True
False
how old is hamad22
<class 'int'>
False
```

07_conversions

In [21]:

```
x=10       #integer
```

```
y=10.2     #float
z="hello"  #string

#implicite type conversion
x=x*y
print(x,"type of x is " ,type(x))   # output are float because it conver integer to

 #explicit type conversion
 age=input("what is your age")
age=int(age)
print(age,type(age))
```

```
  File "C:\Users\hamad\AppData\Local\Temp/ipykernel_7992/618709986.py", line 10
    age=input("what is your age")
    ^
IndentationError: unexpected indent
```

08_ifel_else

In [22]:

```
hamad_age=4
required_age_at_school=5

#question can hammad go to school?

if hamad_age==required_age_at_school:
    print("hamad can join the school")
elif hamad_age > required_age_at_school:
    print("hamad shold join school")
elif hamad_age==2:
    print("you should take care hamad he is still a baby")
else:
   print("hamad can not join to school")
```

```
hamad can not join to school
```

09_funcations

In [23]:

```
def print_codanic():
    print("we are learning with hamad")
    print("we are learning with hamad")
    print("we are learning with hamad")
print_codanic()

#2
def print_codenices():
      text="we ware learning with hammad g"
      print(text)
      print(text)
print_codenices()

#3
def print_code(text):
    print(text)
    print(text)
print_code("we are learnini just")

#4
def school_calcilator(age, text):
    if age==5:
```

```
        print("hamad can join school")
    elif age>5:
        print("hamad should go to higher school")
    else:
         print("hamad still a baby")

school_calcilator(15,"hamad")

#5
def future_age(age):
    new_age=age+20
    return new_age
    print(new_age)
future1=future_age(18)
print(future1)
```

```
we are learning with hamad
we are learning with hamad
we are learning with hamad
we ware learning with hammad g
we ware learning with hammad g
we are learnini just
we are learnini just
hamad should go to higher school
38
```

10_important librarayes

In [24]:

```
#if you want to print the values of pi
import math
print("the values of pi is",math.pi)

import statistics
x=[150,250,350,450]
print(statistics.mean(x))

#same important libraries are numpy,pandas
```

```
the values of pi is 3.141592653589793
300
```

11_loops

In [25]:

```
 #while and for loop
 # while loops
x=0
while(x<5):
    print(x)
    x=x+1


    #for loop   is ma hum range da tai hai
    for x in range(5,10):
        print(x)


 #array
days = ["Mon","Tue","Wed","Thu","Fri"]
```

```
for d in days:
    if(d=="wed"):break #loop stops
    if(d=="wed"):continue #skips d it mean before  the wed and wed are skips are the
    print(d)
```

```
0
5
6
7
8
9
Mon
Tue
Wed
Thu
Fri
```

array with in numpy labraries

In [1]:
```
import numpy as np
a= np.array({1,2,3,4})
a
```

Out[1]:
```
array({1, 2, 3, 4}, dtype=object)
```

one-d array in numpy

In [2]:
```
import numpy as np
a = np.array([5,5,5])
a
```

Out[2]:
```
array([5, 5, 5])
```

In [8]:
```
# zero array in one-D-array
b= np.zeros(2)
b
```

Out[8]:
```
array([0., 0.])
```

In [9]:
```
#array
c=np.ones(3)
c
```

Out[9]:
```
array([1., 1., 1.])
```

In [10]:
```
# create an empty array with e elements
d= np.empty(3)
d
```

Out[10]:
```
array([1., 1., 1.])
```

In [11]:
```
# with range of element
```

In [ ]:
```
e = np.arange(6)
e
```

Out[11]:
```
array([0, 1, 2, 3, 4, 5])
```

In [12]:
```
# with specific range of elements
f=np.arange(2,20)
f
```

Out[12]:
```
array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19])
```

In [13]:
```
# continue
g= np.arange(2,20,2)
g
```

Out[13]:
```
array([ 2,  4,  6,  8, 10, 12, 14, 16, 18])
```

In [14]:
```
#linerly spaced arrays
h= np.linspace(0,10,num=5) #give use 5 nums
h
```

Out[14]:
```
array([ 0. ,  2.5,  5. ,  7.5, 10. ])
```

In [15]:
```
# spacefis data type in array
i = np.ones(5,dtype=np.int8)
i
```

Out[15]:
```
array([1, 1, 1, 1, 1], dtype=int8)
```

In [16]:
```
j =np.ones(3,dtype=np.float64)
j
```

Out[16]:
```
array([1., 1., 1.])
```

two D-array

In [20]:
```
l= np.zeros((3,4))
l
```

Out[20]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [22]:
```
l=np.ones((5,6))
l
```

Out[22]:
```
array([[1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1., 1.]])
```

In [23]:
```

```

```
l=np.empty((3,4))
l
```

Out[23]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [32]:
```
# making and reshpaing a 3d array
c=np.arange(24).reshape(2,3,4)
c
```

Out[32]:
```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],

       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```

In [ ]:
```

```