```
list1=["2020-10-12","2023-10-13","2023-10-08"]

for i in list1:
    print(i[-2:])

for i in list1:
    print(i.split("-")[-1])

b=[]
for i in list1:
    b.append(i.split("-")[-1])
b

l1=[1,2,3]
l2=["a","b","c"]
list(zip(l1,l2))

dict(list(zip(l1,l2)))

a=int(input("enter the number:"))
b=100
c=b/a
print(c)
```

#exceptional handling

1. try block
2. except
3. else
4. finally block

```
try:
    a=int(input("enter the number:"))
    b=100
    c=b/a
    print(c)
except:
    print("enter proper value greater zero")

try:
    a=int(input("enter the number:"))
    b=100
    c=b/a
    print(c)
except ValueError:
    print("pls give int value")
except ZeroDivisionError:
    print("enter proper value greater zero")
print("done")
```

```
try:
    a=int(input("enter the number:"))
    b=100
    c=b/a
    print(c)
except Exception as e:
    print(e)
print("done")

try:
    a=int(input("enter the number: "))
    b=100
    c=b/a
    print(c)
except Exception as e:
    print(e)
else:
    print("no error")
print("done")

try:
    a=int(input("enter the number: "))
    b=100
    c=b/a
    print(c)
except Exception as e:
    print(e)
else:
    print("no error")
finally:
    print("successfully finished")#--------------the boss

try:
    a=int(input("enter the number: "))
    b=100
    c=b/a
    print(c)
except Exception as e:
    print(e)
finally:
    print("successfully finished")
```

int, bool, float, str, list, tuple, range, dict, complex, set, frozenset

```
a=20

mutable and immutable
list        tuple, int, str, bool, float, complex, frozenset
set
dict
```

```
#set unordered, unchangeable, unindex, dot not allow duplicates values
#set items or elements are any data types

1.add
2.clear
3. copy
4. difference

#creating empty use below step
set1={()}

type(set1)

set2={1,2,3,4,5,6,6,6}

set2

set3={1,2,3,4,5,6,6,7,True}

set3

set4={1,2,3,4,5,"t",1.0,6,True,1+2j}

set4.

set4.add("h")

set4

set4.clear()

set4.

set5=set4.copy()

set5

id(set4)

id(set5)

set3

set4

z=set3.difference(set4)

z

y=set4.difference(set3)

y.

set3
```

```
set4

set3.difference_update(set4)

set3

set4.

?set4.discard

set4.discard(1) #it must be element otherwise no error

set4

set3

set6

set6= set3.intersection(set4)#it creates new set fetching matching
items

set4.update

set4.intersection_update(set3)#it remove the itmes is not present in
both sets it is updated original set

set7={"1","5"}

?set4.isdisjoint

#isdisjoint
set3.isdisjoint(set7)

set3

set4.

z=set4.issubset(set3)#bollean output true or false

z

set3.issuperset(set4)#bollean output true or false

set4.pop()#remove item from left to right

set4.

set4.remove(7)#remove specific element must be elemnt otherwise error

set4

#union

set3
```

```
set4

t=set3.union(set4)

?t.symmetric_difference

u=set3.symmetric_difference(set4)#here it will create new set which
remove both set common item

u

set3.symmetric_difference_update(set4)#here it will update original
set which remove both set common item

set3

#updation
set3.update(set4)

set3={1,2,3,5,8,9}

set3

#index
set3

set3[1]

set3[3]="7"

#frozenset ----------->immuatable
#Syntax: frozenset([Iterable])

set3

q=frozenset(set3)

type(q)

q.

q.add["v"]

dict1={"1":1,"2":2}

#copy, difference, intersection, symetric_difference, union,
issuperset, issubset, isdisjoint

frozenset(dict1)

t1=(1,3,2,1,4,5)

sorted(t1, reverse=True)
```

```
#sort is a method (list, string) it will chnage origial datatype
#sorted is function() it is cerated new data type

#built in function
#1. all

?all

all(t1)

all([ ])

all([True,False])

divmod(4,2)#otput is (floor division, quetint)

any([True,False])

any([False,False])

all([1,0])

round(1.2)

round(1.6)

#operators
1. arithemetic
+,-,*,/,%,//,**

1+2

"q"+"2"

"w"+1

int+float=float
complex+complex=complex
comple+float=complex
int+complex=complex
int+boolean=int
boolean+boolean=int
str+str=str
int+int=int
list+list=list
tple+tple=tuple
set+set (we cant add)

{1,2}+{1,2}

True+False

2+True
```

```
["s"]+["e"]

["s"]+("e")

(1,)+(2,)

{1:1}+{2:2}

2+2j+1+1j

1+1j+"w"

a=30
b=20

a-b

list-list

[1,2]-[1,2]

True-False

#* multiplication

a*b

int*int
str*str(not)
float*float
boolean*boolean
str*int
str*float

(2+2j)*(2+2j)# j^2=-1

"ganesh"*6

True*True

"ganesh"*6.2

list*list

[1,2,3]*[1,2,3]

list*int

list*complex

[1,2,3]*(2+2j)

/,%,//

int**int
```

```
complex**complex

(2+2j)**(2+2j)

(2+2j)/(2+2j)

(2+2j)%(2+2j)

#relational conditional
==,>,<,!=,>=,<=

a

b

a==b

a>b

a<b

a!=b

a<=b

a>=b

a,b=30,30

a!=b

#Assignment operotor

a

c=a

c

a=a+1#a+=1

a

a+=1

a

a=a-1#a-=1

a*=2

a/=30

a
```

```
a//=1

a

a%=1

a

a**=2
```

Logicla opearotes are and, or, not

```
a=12
b=2
if a==12 and b==3:
    print("hi")
else:
    print("not matching with seven")

a=12
b=2
if a not b:
    print("hi")
else:
    print("not matching with seven")

a = 10

if not a:
     print("Boolean value of a is True")
else:
     print("10 is divisible by either 3 or 5")
```