Conditional Statements

Used to perform either x-operations when conditional if True or perform y-operations when conditional is False

1.if 2.if...else 3.if...elif 4.match case

In [ ]:
```
tkt=input('Do you have ticket(yes/no)')
if(tkt.lower()=='yes'):
    print('wantch movie')
    print('enjoy')
if(tkt.lower()=='no'):
    print('sorry check tickets')
```

In [ ]:
```
tkt=input('Do you have ticket(yes/no)')
if(tkt.lower()=='yes'):
    print('wantch movie')
    print('enjoy')
if(tkt.lower()=='no'):
    print('sorry check tickets')
```

In [ ]:
```
#positive,negative,zero
no=int(input('enter a number'))
if(no<0):
    print('Given Number is Negative')
if(no>0):
    print('Given Number is Positive')
if(no==0):
    print('Given Number is Zero')
```

In [ ]:
```
#positive,negative,zero
no=int(input('enter a number'))
if(no<0):
    print('Given Number is Negative')
if(no>0):
    print('Given Number is Positive')
if(no==0):
    print('Given Number is Zero')
```

In [ ]:
```
#positive,negative,zero
no=int(input('enter a number'))
if(no<0):
    print('Given Number is Negative')
if(no>0):
    print('Given Number is Positive')
if(no==0):
    print('Given Number is Zero')
```

In [ ]:
```
num=int(input('Enter any number'))
if num<0:
    print('Nrgative')
else:
   if num>0:
        print (' number Positive')
   else:
       print('Zero')
```

In [ ]:
```
#vowels
word=input('enter a word').lower()
if(word.isalpha()):
    if('a' in word or 'e' in word or 'i' in word or 'o' in word or 'u' in word):
        print('word is vowel')
    else:
        print('word is consonant')
else:
    print('not a word')
```

In [ ]:
```
#vowels
word=input('enter a word').lower()
if(word.isalpha()):
    if('a' in word or 'e' in word or 'i' in word or 'o' in word or 'u' in word):
        print('word is vowel')
    else:
        print('word is consonant')
else:
    print('not a word')
```

In [ ]:
```
#vowels
word=input('enter a word').lower()
```

```
if(word.isalpha()):
    if('a' in word or 'e' in word or 'i' in word or 'o' in word or 'u' in word):
        print('word is vowel')
    else:
        print('word is consonant')
else:
    print('not a word')
```

In [ ]:
```
dd={1:'ONE',2:'TWO',3:'THREE',4:'FOUR',5:'FIVE',6:'SIX',7:'SEVEN',8:'EIGHT',9:'NINE'}
d=int(input('enter any digit'))
res=dd.get(d) if dd.get(d)!=None else 'Number'
print('{} is {} '.format(d,res))
```

In [ ]:
```
dd={1:'ONE',2:'TWO',3:'THREE',4:'FOUR',5:'FIVE',6:'SIX',7:'SEVEN',8:'EIGHT',9:'NINE'}
d=int(input('enter any digit'))
res=dd.get(d) if dd.get(d)!=None else 'Number'
print('{} is {} '.format(d,res))
```

In [ ]:
```
d=int(input('enter any digit'))
if(d in[0,1,2,3,4,5,6,7,8,9]):
    print('number is Positive and less than 10')
elif(d in[-1,-2,-3,-4,-5,-6,-7,-8,-9]):
        print('number is Negative and less than -10')
else:
        print('is digit {}'.format(d))
```

In [ ]:
```
d=int(input('enter any digit'))
if(d in[0,1,2,3,4,5,6,7,8,9]):
    print('number is Positive and less than 10')
elif(d in[-1,-2,-3,-4,-5,-6,-7,-8,-9]):
        print('number is Negative and less than -10')
else:
        print('{} is digit ,im from else block'.format(d))
```

In [ ]:
```
d=int(input('enter any digit'))
if(d in[0,1,2,3,4,5,6,7,8,9]):
    print('number is Positive and less than 10')
elif(d in[-1,-2,-3,-4,-5,-6,-7,-8,-9]):
        print('number is Negative and less than -10')
else:
        print('{} is digit ,i am from else block'.format(d))
```

PaySlip using if..elif

In [ ]:
```
import sys
eno=int(input('Enter a Employee Number:'))
ename=input('Enter a Employee Name:')
bsal=float(input('Enter Employee Basic Salary:'))
if(bsal<0):
          print('Invalid Salary No need to work')
          sys.exit()
elif(0<bsal<=10,000):
   da=bsal*(10/100)
   ta=bsal*(8/100)
   hra=bsal*(6/100)
   ma=bsal*(1/100)
   lic=bsal*(1/100)
   gpf=bsal*(1.5/100)
netsal=(da+ta+hra+ma)-(lic+gpf)
print('='*50)
print('PAYSLIP EMPLOYEE SALARY DATA')
print('='*50)
print('Employee Number {}:'.format(eno))
print('Employee Name {}:'.format(ename))
print('Employee Basic Salary {}|:'.format(bsal))
print('Empoyee DA {}:'.format(da))
print('Employee TA {}:'.format(ta))
print('Employee HRA {}:'.format(hra))
print('Employee MA {}:'.format(ma))
print('Deduction')
print('Employee LIC {}:'.format(lic))
print('Employee GPF {}:'.format(gpf))
print('='*50)
print('Employee NETSALARY {}:'.format(netsal))
```

In [ ]:
```
import sys
eno=int(input('Enter a Employee Number:'))
ename=input('Enter a Employee Name:')
```

```
bsal=float(input('Enter Employee Basic Salary:'))
if(bsal<0):
          print('Invalid Salary No need to work')
          sys.exit()
elif(0<bsal<=10,000):
   da=bsal*(10/100)
   ta=bsal*(8/100)
   hra=bsal*(6/100)
   ma=bsal*(1/100)
   lic=bsal*(1/100)
   gpf=bsal*(1.5/100)
netsal=(da+ta+hra+ma)-(lic+gpf)
print('='*50)
print('PAYSLIP EMPLOYEE SALARY DATA')
print('='*50)
print('Employee Number {}:'.format(eno))
print('Employee Name {}:'.format(ename))
print('Employee Basic Salary {}|:'.format(bsal))
print('Empoyee DA {}:'.format(da))
print('Employee TA {}:'.format(ta))
print('Employee HRA {}:'.format(hra))
print('Employee MA {}:'.format(ma))
print('Deduction')
print('Employee LIC {}:'.format(lic))
print('Employee GPF {}:'.format(gpf))
print('='*50)
print('Employee NETSALARY {}:'.format(netsal))
```

match case is a feature in python 3.10 version match is used to take in deciding pre-designed conditions in menu driven app.

In [ ]:
```
#match case for week days
wk=input('Enter any week day')
match(wk):
    case 'MONDAY'|'TUESDAY'|'WEDNESDAY'|'THURSDAY'|'FRIDAY':
        print(' {} is week day ,work'.format(wk))
    case 'SATURDAY':
        print('{} is week day,packup'.format(wk))
    case 'SUNDAY':
        print('{} is holiday,enjoy'.format(wk))
```

In [ ]:
```
#match case for week days
wk=input('Enter any week day').upper()
match(wk):
    case 'MONDAY'|'TUESDAY'|'WEDNESDAY'|'THURSDAY'|'FRIDAY':
        print(' {} is week day ,work'.format(wk))
    case 'SATURDAY':
        print('{} is week day,packup'.format(wk))
    case 'SUNDAY':
        print('{} is holiday,enjoy'.format(wk))
```

In [ ]:
```
#match case for week days
wk=input('Enter any week day').upper()
match(wk[0:3]):
    case 'MON'|'TUE'|'WED'|'THU'|'FRI':
        print(' {} is week day ,work'.format(wk))
    case 'SAT':
        print('{} is week day,packup'.format(wk))
    case 'SUN':
        print('{} is holiday,enjoy'.format(wk))
```

In [ ]:
```
while(True):
    print('='*50)
    print('\t1.Additon')
    print('\t2.Multiplication')
    print('\t3.Divsion')
    print('\t4.Floor_divison')
    ch=int(input('enter your choice'))
    match(ch):
       case 1:
           print('enter a values for Addition')
           a,b=int(input('enter first value')),int(input('enter second value'))
           print('sum {} and {} ={}'.format(a,b,a+b))
       case 2:
           print('enter a values for multiplication')
           a,b=int(input()),int(input())
           print('multiplication {} and {} ={}'.format(a,b,a*b))
       case 3:
           print('enter a values for Division')
           a,b=int(input()),int(input())
           print('divison of {} and {}={}'.format(a,b,a/b))
       case 4:
           print('enter a values of floor_divison')
           a,b=int(input()),int(input())
```

```
            print('floor_divison of {} and {}={}'.format(a,b,a//b))
            break
        case _:
            print('your selection operation is wrong please try again')
```

Looping statements

To perform certain operations repeatedly for finite time until test cond becomes True.

while:used when you do not know how many times to execute. for:used when you know exactly how many times to execute.

In [ ]:
```
n=int(input('enter a value for execution'))
if(n<0):
      print('invalid selection')
else:
      i=1 #initluzation part
      while(i<=n):  #conditional part
         print('\t{}'.format(i))
         i=i+1 #updation part
```

In [40]:
```
#voting age
age=int(input('enter your age'))
if(age<18):
    print('Your are not elligable for vote please try another year ,age={}'.format(age))
else:
    i=age
    while(i>=18):
      print('please vote!!! yours age is {}'.format(age))
      i=i+1
      break
    else:
      print('try again')
```

```
please vote!!! yours age is 19
```

In [3]:
```
# Initialize a variable to start from 1
num = 1

# Use a while loop to iterate until num reaches 20
while num <= 20:
    # Check if the number is even
    if num % 2 == 0:
        # Print the even number
        print(num)
    # Increment the number by 1
    num += 1
```

```
2
4
6
8
10
12
14
16
18
20
```

In [33]:
```
n=int(input('enter  a number to generate'))
if n<=0:
    print('invalid {}'.format(n))
else:
    s=0
    i=1
    while(i<=n):
        s=s+i
        print('{}'.format(i),end='')
        i=i+1
    else:
        print('='*50)
        print('{}'.format(s),end='')
```

```
12345678910111213141516171819 20==================================================
210
```

In [17]:
```
n=int(input('enterr a number you want to generate'))
if n<=0:
    print("invalid")
else:
    i=1
    while(i<=n):
        print('multiplication is {} * {} ={}'.format(n,i,n*i))
```

```
        i=i+1
```

```
multiplication is 10 * 1 =10
multiplication is 10 * 2 =20
multiplication is 10 * 3 =30
multiplication is 10 * 4 =40
multiplication is 10 * 5 =50
multiplication is 10 * 6 =60
multiplication is 10 * 7 =70
multiplication is 10 * 8 =80
multiplication is 10 * 9 =90
multiplication is 10 * 10 =100
```

In [45]:
```
f=input('enter a number')
for i in f:
    print('{}'.format(i))
```

```
h
e
l
l
o
```

In [47]:
```
f=input('enter a number')
for i in f:
    print('{}'.format(i))
```

```
1
0
```

In [88]:
```
#range in for loop
n=int(input("enter a number"))
if n<0:
    print("invalid".format(n))
else:
    f=1
    for i in range(1,n+1):
        f=f*i
        print('{}\t'.format(i))
    else:
        print('{} \t {}'.format(n,f))
```

```
1
2
3
4
5
6
7
8
9
10
10      3628800
```

In [3]:
```
n=int(input('enter a number to generate'))
if n<=0:
    print('invalid')
else:
    f=1
    i=1
    while(i<=n):
        f=f+i
        i=f*i
    else:
        print('{} {}'.format(n,f))
```

```
10 12
```

In [11]:
```
n=int(input('enter how many number you want to generate'))
if n<0:
    print('invalid')
else:
    print('='*50)
    print('numbers 1 to {}'.format(n))
    print('='*50)
    for i in range(1,n+1):
        print(i)
    print('='*50)
    print('numbers {} to 1'.format(n))
    print('='*50)
    for i in range(n,0,-1):
        print(i)
```

```
================================================
numbers 1 to 10
================================================
1
2
3
4
5
6
7
8
9
10
================================================
numbers 10 to 1
================================================
10
9
8
7
6
5
4
3
2
1
```

In [23]:
```
#program for square and cube
n=int(input('enter a number you want to generate'))
if n<=0:
    print('invalid')
else:
    s=0
    ss=0
    sc=0
    for i in range(n,n+1):
        print('{}\t\t{}\t{}\t'.format(i,i**2,i**3))
        s=s+i
        ss=ss+i**2
        sc=sc+i**3
```

```
5               25      125
```

Transfer flow control

used to transfer the contro of pvm one part of the program to another part it has 4 types 1.break :break the loop/break the conti 2.continue :skip the current iteration of the program 3.pass 4.return

In [77]:
```
n='python'
for ch in n:
    print(ch)
print('='*50)
for ch in n:
    if ch=='h':
        break
    else:
        print(ch)
print('='*50)
for ch in n:
    if ch=='o':
        break
    else:
        print(ch)
```

```
p
y
t
h
o
n
================================================
p
y
t
================================================
p
y
t
h
```

In [1]:
```
n='gosling'
i=0
```

```
while(i<len(n)):
    print(n[i])
    i=i+1
```

```
g
o
s
l
i
n
g
```

In [1]:
```
n='python'
i=0
while(i<len(n)):
    print(n[i])
    i=i+1
```

```
p
y
t
h
o
n
```

In [3]:
```
k='travis'
i=0
while(i<len(k)):
    print(k[i])
    i=i+1
print('='*50)
i=0
while(i<len(k)):
    if k[i]=='i':
        break
    else:
        print(k[i])
        i=i+1
```

```
t
r
a
v
i
s
==================================================
t
r
a
v
```

In [5]:
```
s='mississim'
for i in s:
    print(s[i])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 3
      1 s='mississim'
      2 for i in s:
----> 3     print(s[i])

TypeError: string indices must be integers, not 'str'
```

In [7]:
```
s='mississim'
for i in s:
    print(i)
```

```
m
i
s
s
i
s
s
i
m
```

In [37]:
```
s='mississim'
for i in range(0,len(s)):
     print(s[i])
print('='*50)
for i in range(0,len(s)):
```

```
    if i==4:
      break
    print(s[i])
```

```
m
i
s
s
i
s
s
i
m
==================================================
m
i
s
s
```

In [43]:
```
s='mississim'
for i in range(0,len(s)):
     print(s[i])
print('='*50)
for i in range(0,len(s)):
    if i==4:
      continue
    print(s[i])
```

```
m
i
s
s
i
s
s
i
m
==================================================
m
i
s
s
s
s
i
m
```

In [1]:
```
a='alina'
i=0
while(i<len(a)):
    print(a[i])
    i=i+1
print('='*50)
i=0
while(i<len(a)):
    if a[i]=='n':
      break
    else:
     print(a[i])
     i=i+1
```

```
a
l
i
n
a
==================================================
a
l
i
```

In [ ]:
```
a='tonight'
i=0
while(i<len(a)):
    if a[i]=='g':
         continue
    else:
        print(a[i])
        i=i+1
```

```
t
o
n
i
```

In [17]:
```
s='python'
for i in s:
    if i=='y':
        continue
    print('{}'.format(i))
```

```
p
t
h
o
n
```

In [19]:
```
s='how are you'
for i in s:
    if i=='e':
        continue
    print(i)
```

```
h
o
w

a
r

y
o
u
```

In [35]:
```
#find the factor of a given number
n=int(input('enter a number'))
if n<=0:
    print('invalidd')
else:
    print("factorial of number {}".format(n))
    for i in range(1,(n//2)+1):
          if n%i==0:
              print(i)
```

```
factorial of number 10
1
2
5
```

In [43]:
```
n=int(input('enter a number'))
for i in range(1,(n//2)+1):
    if n%i==0:
        print(i,end='-')
```

```
1-2-4-5-10-20-25-50-
```

In [49]:
```
s='how are you'
for i in s:
    if i=='e':
        pass
    print(i)
```

```
h
o
w

a
r
e

y
o
u
```

Nested loop

1.for loop inside for loop 2.while loop inside while loop 3.for loop inside while loop 4.while loop inside for loop

In [81]:
```
for i in range(1,6):
    print('='*50)
    print(i)
    print('='*50)
```

```
    for j in range(1,4):
        print('i am for in loop {}'.format(j))
```

```
==================================================
1
==================================================
i am for in loop 1
i am for in loop 2
i am for in loop 3
==================================================
2
==================================================
i am for in loop 1
i am for in loop 2
i am for in loop 3
==================================================
3
==================================================
i am for in loop 1
i am for in loop 2
i am for in loop 3
==================================================
4
==================================================
i am for in loop 1
i am for in loop 2
i am for in loop 3
==================================================
5
==================================================
i am for in loop 1
i am for in loop 2
i am for in loop 3
```

In [93]:
```
i=1
while(i<=5):
    print('='*50)
    print(i)
    print('='*50)
    j=1
    while(j<=4):
        print('i am from in loop {}'.format(j))
        j=j+1
    i=i+1
```

```
==================================================
1
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
2
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
3
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
4
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
5
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
```

In [109…
```
for i in range(1,5):
    print('='*50)
    print(i)
    print('='*50)
    j=4
    while(j>1):
      j=j-1
      print('i am from in loop {}'.format(j))
```

```
==================================================
1
==================================================
i am from in loop 3
i am from in loop 2
i am from in loop 1
==================================================
2
==================================================
i am from in loop 3
i am from in loop 2
i am from in loop 1
==================================================
3
==================================================
i am from in loop 3
i am from in loop 2
i am from in loop 1
==================================================
4
==================================================
i am from in loop 3
i am from in loop 2
i am from in loop 1
```

In [113…
```
for i in range(1,5):
    print('='*50)
    print(i)
    print('='*50)
    j=0
    while(j<4):
      j=j+1
      print('i am from in loop {}'.format(j))
```

```
==================================================
1
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
2
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
3
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
==================================================
4
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
i am from in loop 4
```

In [115…
```
i=0
while(i<=4):
    print('='*50)
    print(i)
    print('='*50)
    i=i+1
    for j in range(1,4):
        print('i am from in loop {}'.format(j))
```

```
==================================================
0
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
==================================================
1
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
==================================================
2
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
==================================================
3
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
==================================================
4
==================================================
i am from in loop 1
i am from in loop 2
i am from in loop 3
```

In [ ]: