Conditional Statements

Strings

Loops

Functions

Data Structures ¶

In [1]:
```
a = 10
b = 10
c = 50
if (a>b):
    if (a==c):
        print("Both are equal")
    else:
        print("Both are different")
elif (a<b):
    print(b)
else:
    print(a,b)
```

```
10 10
```

In [ ]:
```

```

In [2]:
```
a = "Kits college"
b = "Kits college of Engineering"
c = "Engineering collge"
d = "Kits college"
if (a ==b):
    print("Both a,b are equal")
elif(a==c):
    print("Both a,c are equal")
else:
    print("a,d are equal")
```

```
a,d are equal
```

In [3]:
```
# if(cond){
#     ...st
# }
# if cond:
#     ..st

```

In [4]:
```
a=10
b=4
if a>b:
    print("YES")
```

```
YES
```

In [5]:
```
#if else
# if cond:
#     ..stm
# else:
#     ...stms
```

In [6]:
```
if a<b:
    print("yes")
else:
    print("No")
```

```
No
```

In [7]:
```
# if ..elif..else..
if cond:
    ...stms
elif cond:
    ...stms
else:
    ..srtms
```

```
  File "<ipython-input-7-b54eeabbee20>", line 3
    ...stms
        ^
SyntaxError: invalid syntax
```

In [ ]:
```
#check the given number is even or odd
    #input:n=5
    #input:n=8
n=int(input())
if n%2==0:
    print("Even number")
else:
    print("not even  number")

```

In [ ]:
```
#check the given year is leap year or not
# year=2000-leap year
# ->2019-non_leap year
# ->1900-non_leap year
# ->2020-leap year
year=int(input())
if (year%400==0 or (year % 100!=0 and year%4==0)):
    print("leap year")
else:
    print("non leap year")

```

Membership Operators

- is, is not
- in, not in

In [ ]:
```
a = "kits college"
b = "kits college"
if (a ==b):
    print("same")

```

In [ ]:
```
a = "kits college"
b = "kits college"
if (a is b):
    print("Same")
else:
    print("different")
```

In [ ]:
```
print(id(a))
print(id(b))
```

In [ ]:
```
if ('k' in a):
    print("yes")
```

Strings

In [ ]:
```
s="kits college of engineering"
```

In [ ]:
```
s[0]# It will get the first index value
```

In [ ]:
```
s[1:3]# to get the start range of 1 and ending with 3 index
```

In [ ]:
```
s[5:12]
```

In [ ]:
```
ns="hello apssdc"
ns
```

In [ ]:
```
len(ns)#length of the string
```

In [ ]:
```
ns[2:]
```

In [ ]:
```
ns[:4]
```

In [ ]:
```
ns
```

In [ ]:
```
ns[::-1]
```

In [ ]:
```
p='madam'
p[::-1]
```

In [ ]:
```
if p==p[::-1]:
    print("palindrom")
else:
    print("no a palindrom")
```

In [ ]:
```
s="kits college of engineering"

s[2:9:3]
```

In [ ]:
```
s[-1:8:-1]
```

In [ ]:
```
s[-1::-1]#It is also string Reverse
```

In [ ]:
```
s[::2]#alternative number or two numbers increment#chik
```

In [ ]:
```
s1='Python'

```

In [ ]:
```
#Tasks
    # access the first char of given string
print("access the first char of given string :",s1[0])
    # Access the last two characters in reverse order
print("the last two characters in reverse order :",s1[-1:-3:-1])
    # Access the char 5th char to end of the string
print("the char 5th char to end of the string:",s1[4:])
    # Access the Alternative char of string
    #Print string like below:
            #p
            #y
            #t
            #o
            #n
    #Print the string reverse
    #print the ovels(aeiou) which present in the string

```

In [ ]:
```
print(dir(str))
```

In [ ]:
```
s="Apssdc"
d="hello"

```

In [ ]:
```
d=d.capitalize()
d
```

In [ ]:
```
d
```

In [ ]:
```
d.swapcase()
```

In [ ]:
```
d.count('L')
```

In [ ]:
```
s=" muni "
s.strip()
```

In [ ]:
```
s.rstrip()
```

In [ ]:
```
s.islower()
```

In [ ]:
```
d.islower()
```

In [ ]:
```
d
```

Control Statements

- Two types
  - For loop
  - While
    - break
    - continue

In [ ]:
```
#syntax in c-Lang
#     for(int i=0;i<=n;i++){

#     }

#Python for syntax
#     for cond:
#         ..stmts


```

In [ ]:
```
#Print the first 10 natural number
for number in range(1,11):
    print(number,end="")
```

In [ ]:
```
for i in range(1,101):
    if i%2==0:
        print(i,end=" ")
```

In [ ]:
```
for even in range(0,101,2):
    print(even,end=" ")
```

In [ ]:
```
for even in range(1,101,2):
    print(even,end=" ")
```

In [ ]:
```
s1
```

In [ ]:
```
for i in s1:
    print(i)
```

In [ ]:
```
https://bit.ly/2RvYOad
```

In [ ]:
```
# # Whlie
# while (cond){
#     ...stms
# }

```

In [ ]:
```
# in Python
#     while cond:
#         ..stm
```

In [ ]:
```
n=12345612#find the length of the number
count=0
while n>0:
    n=n//10#quotient
    count=count+1
print(count)


```

In [ ]:
```
n=int(input())# n=3 print("Going to home")
while True:
    if n==3:
        print("Going to home")
        break
```

In [ ]:
```
addtion()
```

Functions in python

- In built Functions
  - With in System contains
    - Ex:math,re,random,etc
- User Defiend Functions
  - Four Types

* 1.funtion with arg. with return value
* 2.funtion with arg. with out return value
* 3.funtion with out arg. with out return value
* 4.funtion with out arg. with return value

In [ ]:
```
#Funtion syntax
    #With out arg
        #     def fun_name():
        #     #     ...stmts
        #     #     ...stms
        #     fun_name()

   # With  arg
        #     def fun_name(a,b):
        #     #     ...stmts
        #     #     ...stms
        #     fun_name(a,b)

    # With  arg with return value
        #     def fun_name(a,b):
        #     #     return ...stmts
        #     fun_name(a,b)
```

In [ ]:
```
#funtion with out arg. with out return value
def addtion():
    print(a+b)
a=int(input("Enter the a value"))
b=int(input("Enter the b value"))
addtion()
```

In [ ]:
```
#funtion with arg. with return value
def subtraction(x,y):
    return x-y
x=9
y=3
subtraction(x,y)
```

In [ ]:
```
addtion()
```

In [ ]:
```
#funtion with arg. with out  return value
def multiplication(x,y):
    print(x*y)
multiplication(2,4)
```

In [ ]:
```
#funtion with out arg. with return value
def multiplication():
    return(x*y)
x=9
y=6
multiplication()
```

In [9]:
```
# Funtion for addtion oparation
def myAddition(k,j):
    print(k+j)
# k=int(input("Enter the k value for addition "))
# j=int(input("Enter the j value for addition "))
myAddition(4,5)
```

```
9
```

In [14]:
```
# Funtion for addtion oparation
def mysutraction(k,j):
    return k-j
k=int(input("Enter the k value for subtract "))
j=int(input("Enter the j value for subtract "))
mysutraction(k,j)
```

```
Enter the k value for subtract 4
Enter the j value for subtract 5
```

Out[14]:
```
-1
```

In [21]:
```
# Funtion for addtion oparation
def myMultiplication(k,j):
    return k*j
k=int(input("Enter the k value for multiplication "))
j=int(input("Enter the j value for multiplication "))
myMultiplication(k,j)
```

```
Enter the k value for multiplication 5
Enter the j value for multiplication 3
```

Out[21]:
```
15
```

In [26]:
```
# Funtion for addtion oparation
def myDivision(k,j):
    return k/j
k=int(input("Enter the k value for div "))
j=int(input("Enter the j value for div "))
myDivision(k,j)
```

```
Enter the k value for div 5
Enter the j value for div 4
```

Out[26]:
```
1.25
```

In [ ]:
```
1.add
2.sub
3.mul
4.div
#invalid option
5.exit


```

In [ ]:
```

```

In [10]:
```
#creating calculator app
def calculatorApp():
    print("1.add \n 2.sub \n 3.Mul \n 4.Div \n 5.Exit")
    userChoice=int(input("Enter the user choice"))
    if userChoice==1:
        myAddition(4,2)
#     elif userChoice==3:
#         myMultiplication(arg1,arg2)
#     elif userChoice==4:
#         myDivision(arg1,arg2)
#     elif userChoice==5:
#         return True
#     else:
#         print("invalid option ")

arg1=int(input())
arg2=int(input())
calculatorApp()

```

```
1
4
1.add
 2.sub
 3.Mul
 4.Div
 5.Exit
Enter the user choice4
```

In [11]:
```
myAddition(4,6)
```

```
10
```

In [ ]:
```
mysutraction(4,5)
```

In [15]:
```
mysutraction(4,2)
```

Out[15]:
```
2
```

In [30]:
```
def calC():
    uc=int(input())
        if uc==1:
            myAddition(1,5)
        elif uc==2:
            print(mysutraction(4,8))
        elif uc==3:
            print(myMultiplication(3,3))
        elif uc==4:
            print(myDivision(5,2))
        elif uc==5:
            print("EXIT")

calC()
```

```
4
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
2.5
```

In [37]:
```
s = "Engineering"
for i in range(len(s)):
    print(i, end = " ")

    print(s[i], end = " ")
```

```
0 E 1 n 2 g 3 i 4 n 5 e 6 e 7 r 8 i 9 n 10 g
```

In [39]:
```
for i in s:
    print(i, end = " ")
```

```
E n g i n e e r i n g
```

While

In [ ]:
```
while True:
    n = int(input())
    if n!=0:
        print(n)
n = int(input())

```

```
4
4
5
5
```

In [ ]:
```

```