In [1]:
```
# FOR LOOP                  # for keyword is the start of a loop
# Looping over a List
fruits=["banana","apple","mango","grapes"]
for fruit in fruits:        # in keyword: Separates the loop variable from the sequence.
    print(fruit)
```

```
banana
apple
mango
grapes
```

In [3]:
```
#Looping over a Range
for i in range(5):
    print(i,end="")
```

```
01234
```

In [33]:
```
#Looping over a String
n="pavani"
for char in n:
    print(char)
```

```
p
a
v
a
n
i
```

In [8]:
```
#Looping over a Dictionary
person={"name":"pavani","age":30,"city":"vijayawada"}
for key,value in person.items():           #item is the syntax
    print(f"{key}:{value}")
```

```
name:pavani
age:30
city:vijayawada
```

In [15]:
```
#nested loops
for i in range(3):
    for j in range(2):          # how many times have to repeat
        print(f"({i},{j})")
```

```
(0,0)
(0,1)
(1,0)
(1,1)
(2,0)
(2,1)
```

In [16]:
```
# Specifying Start, Stop, and Step:
for i in range(2,8,2):
    print(i)
```

```
2
4
6
```

In [17]:
```
# using in list Comprehension:
list=[x**2 for x in range(5)]
print(list)
```

```
[0, 1, 4, 9, 16]
```

In [26]:
```
list=[20,22,35,67,89]
sum=0
for i in list:
    sum=sum+i
print("sum=",sum)
print("average=",sum/len(list))
```

```
sum= 233
average= 46.6
```

In [38]:
```
num = 4

for i in range(1, 11):
    mul = num * i
    print(num,"*",i,"=",mul)
```

```
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
```

In [40]:
```
#break
for i in range(1,10):
    if(i==6):
        break
    print(i)
```

```
1
2
3
4
5
```

In [41]:
```
#continue
for i in range(1,10):
    if(i==6):
        continue          #skip the value
    print(i)
```

```
1
2
3
4
5
7
8
9
```

In [49]:
```
num = [1, 2, 5, 7]
cube = []

for i in num:
    cube.append(i**3)

print(num)
print(cube)
```

```
[1, 2, 5, 7]
[1, 8, 125, 343]
```

In [54]:
```
n=int(input())
for i in range(0,n):
    for j in range(0,i+1):
        print("*",end="")
    print()
```

```
4
*
**
***
****
```

In [68]:
```
#while loop
i=0
while i<10:
    i=i+1
    print(i)
```

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

In [72]:
```
count=0
while count<=5:
    print("pavani is good")
    count=count+1
print("end")
```

```
pavani is good
pavani is good
pavani is good
pavani is good
pavani is good
pavani is good
end
```

In [75]:
```
while True:
    print("your name")
    name=input()
    if name=='pavani':
        break
print("thank you")
```

```
your name
pradeep
your name
pavani
thank you
```

In [79]:
```
#LCM of 4 and 7                          #break
x=0
while True:
    x+=1
    if not(x%4 or x%25):
        break
print(x,"is divisible by both 4 and 7")
```

```
100 is divisible by both 4 and 7
```

In [81]:
```
i=0                             #continue
while i<10:
    i+=1
    if i==6:
        continue
    print(i)
```

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

In [83]:
```
#avg of positive numbers
num=0
sum=0
count=0
while num>=0:
    num=int(input())
    if num>0:
        count=count+1
        sum=sum+num
avg=sum/count
print("sum=",sum,"average=",avg)
```

```
5
-1
sum= 5 average= 5.0
```

In [84]:
```
#user defined functions
def word():
    print("hello pavani")
word()
```

```
hello pavani
```

In [85]:
```
def word():
    name=str(input())
    print("hello"+name)
word()
```

```
india
helloindia
```

In [87]:
```
def ft_to_cm(ft):
    cent=ft*30.48
    return centi
ft_to_cm(6)
```

Out[87]:
```
182.88
```

In [89]:
```
def factorial(x):
    if x==0:
        return 1
    else:
        return x*factorial(x-1)
factorial(5)
```

Out[89]:
```
120
```

In [90]:
```
#default arguments
def word(name='pavani'):
    print("hello",name)
word()
```

```
hello pavani
```

In [91]:
```
#scope
def fun():                  #local variable
    z=7
    print(z)
fun()
```

```
7
```

In [92]:
```
z=7                             #global variable
def fun():
    print(z)
fun()
```

```
7
```

In [93]:
```
z
```

Out[93]:
```
7
```

In [94]:
```
#lambda function (anonymous)
add=lambda x:x+50
print(add(10))
```

```
60
```

In [96]:
```
(lambda a,b:a*b)(5,6)
```

Out[96]:
```
30
```

In [ ]:
```

```