

```

In [4]: #1) Explain with an example each when to use a for loop and a while loop.
#1) For loop
#ANS) for loop is a block of code to run for a certain number of times. It is used when the number of iterations is known.
#syntax:
# for x in range(start, stop, step):
#     statements
#Example:
for i in range(5):
    print(i)

print("-----")

#2) while loop
#Ans) In python while loop is a block of code to iterate until the condition is met. It is used when the number of iterations is not known.
#syntax:
# while condition:
#     statements
#Example:
name = "Python"
while i < 10:
    i += 1
    print(name)

```

```

0
1
2
3
4
-----
Python
Python
Python
Python
Python
Python

```

```
In [5]: #2)Write a python program to print the sum and product of the first 10 natural
# using function with for Loop
def natural_numbers():
    sum=0
    product=1
    for i in range(1,10):
        sum+=i
        product*=i
        print(f'sum of the first natural numbers {sum}')
        print("#####")
        print(f'The product of the first natural numbers {product}')
natural_numbers()
print("-----")
# using function with while loop
def natural():
    sum1=0
    prod=1
    n=1
    while n<10:
        sum1+=n
        prod=prod*n
        n+=1
        print(f'sum of the first natural numbers {sum}')
        print('-----')
        print(f'The product of the first natural numbers {product}')
natural()
```

```

sum of the first natural numbers 1
#####
The product of the first natural numbers 1
sum of the first natural numbers 3
#####
The product of the first natural numbers 2
sum of the first natural numbers 6
#####
The product of the first natural numbers 6
sum of the first natural numbers 10
#####
The product of the first natural numbers 24
sum of the first natural numbers 15
#####
The product of the first natural numbers 120
sum of the first natural numbers 21
#####
The product of the first natural numbers 720
sum of the first natural numbers 28
#####
The product of the first natural numbers 5040
sum of the first natural numbers 36
#####
The product of the first natural numbers 40320
sum of the first natural numbers 45
#####
The product of the first natural numbers 362880
-----
sum of the first natural numbers <built-in function sum>
-----

```

```

-----
NameError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_10284\1625695998.py in <module>
    24         print('-----')
    25         print(f'The product of the first natural numbers {product}')
---> 26 natural()

~\AppData\Local\Temp\ipykernel_10284\1625695998.py in natural()
    23         print(f'sum of the first natural numbers {sum}')
    24         print('-----')
---> 25         print(f'The product of the first natural numbers {product}')
    26 natural()

NameError: name 'product' is not defined

```

```
In [6]: #4)Write a program to filter count vowels in the below-given string.
string = "I want to become a data analyst"
count=0
for char in string:
    if char in ["a","i","e","o","u","A","I","O","U","E"]:
        count+=1
print(f'The count of the vowels is {count}')
```

The count of the vowels is 11

```
In [7]: #3)Create a List of numbers from 1 to 100. Use for loop and while loop to calculate cubes
cubes=[]
for x in range(1,101):
    x=x**3
    if x%4==0 or x%5==0:
        cubes.append(x)
print(cubes)
```

[8, 64, 125, 216, 512, 1000, 1728, 2744, 3375, 4096, 5832, 8000, 10648, 13824, 15625, 17576, 21952, 27000, 32768, 39304, 42875, 46656, 54872, 64000, 74088, 85184, 91125, 97336, 110592, 125000, 140608, 157464, 166375, 175616, 195112, 216000, 238328, 262144, 274625, 287496, 314432, 343000, 373248, 405224, 421875, 438976, 474552, 512000, 551368, 592704, 614125, 636056, 681472, 729000, 778688, 830584, 857375, 884736, 941192, 1000000]

```
In [8]: #5)Which keyword is used to create a function? Create a function to return a list of odd numbers.
#Ans) A function can be created by the using of the keyword def
def odd():
    z=[]
    for x in range(1,26):
        if x%2!=0:
            z.append(x)
    print(z)
odd()
```

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25]

In [9]: *#6)Why *args and **kwargs is used in some functions? Create a function each for*
*# *args is keyword which is accept number of positional arguments in the funct*
`def func(*args):`
 for name in args:
 print(name)
`func("Rk", "Yash", "Kiran")`
`print("-----")`
*# **kwargs allows a function to accept a variable number of keyword arguments.*
`def fun(**kwargs):`
 for key,values in kwargs.items():
 print(key,values)
`fun(name="Alic",age=25)`

```
Rk
Yash
Kiran
-----
name Alic
age 25
```

In [10]: *#7)What is an iterator in python? Name the method used to initialise the iterat*
`list=[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]`
`myit=iter(list)`
`print(next(myit))`
`print(next(myit))`
`print(next(myit))`
`print(next(myit))`
`print(next(myit))`

```
2
4
6
8
10
```

In [11]: *#8)What is a generator function in python? Why yield keyword is used? Give an e*
#In Python, a generator is a function that returns an iterator that produces a
`def generator():`
 yield 1
 yield 2
 yield 3
`for y in generator():`
 print(y)

```
1
2
3
```

```
In [ ]: #9)Create a generator function for prime numbers less than 1000. Use the next(),
def primes_generator():
    num = 2
    while num < 1000:
        if all(num % i != 0 for i in range(2, num)):
            yield num
        num += 1

# Create a generator object
primes_gen = primes_generator()

# Print the first 20 prime numbers using next() function
for i in range(20):
    print(next(primes_gen))
```

In [13]: *#10)Write a code to print odd numbers from 1 to 100 using List comprehension. I*

```
li=[x for x in range(1,100) if x%2!=0]  
li
```

Out[13]: [1,
3,
5,
7,
9,
11,
13,
15,
17,
19,
21,
23,
25,
27,
29,
31,
33,
35,
37,
39,
41,
43,
45,
47,
49,
51,
53,
55,
57,
59,
61,
63,
65,
67,
69,
71,
73,
75,
77,
79,
81,
83,
85,
87,
89,
91,
93,
95,
97,
99]

In []: