

Functions

Defining and calling functions

```
#Example
def greet(name):
    print("hello, "+name+"!")
greet("alice")

hello, alice!
```

FUNCTIONAL ARGUEMENTS

positional arguments

```
def add(a,b):
    return a + b
print(add(5,3))

8
```

Keyword arguments

```
def greet(name,message):
    print(message + ", " + name + "!")
greet(name="alice",message="Good morning")

Good morning,alice!
```

Default arguments

```
def greet(name,message="hello"):
    print(message + ", " + name + "!")
greet("Alice") # uses default argument
greet("bob", "Hi")

hello,Alice!
Hi,bob!
```

Variable-length arguments

```
def sum_numbers(*numbers):
    return sum(numbers)
print(sum_numbers(1, 2, 3, 4))
```

10

```
def prafful(**kwargs):  
    for key, value in kwargs.items():  
        print(f"{key}: {value}")  
prafful(name="pranjal", age="25", city="Paris")  
  
name: pranjal  
age: 25  
city: Paris
```

Return statement

```
def square(num):  
    return num * num  
square(5)  
  
25  
  
result = square(4)  
print("square is:", result)  
  
square is: 16
```

Modules

1.Importing modules

```
import math  
print(math.sqrt(16))  
  
4.0  
  
from math import pi, sin  
print(pi)  
print(sin(math.radians(90)))  
  
3.141592653589793  
1.0  
  
def greet(name):  
    print("Hello, " + name)  
  
def add(a,b):  
    return a + b  
  
import my_module
```

```
my_module,greet("Alice")
print(my_module.add(5,3))
```

```
Hello, Alice
8
Hello, Alice
8
```

Write a Function to check if a number is prime

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2,int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True
number = int(input("Enter the number: "))
if is_prime(number):
    print("The number is prime.")
else:
    print("The number is not prime. ")
```

```
Enter the number: 5
The number is prime.
```

create a function to generate Fibonacci Sequence

```
def fibonacci(n):
    sequence = []
    a,b=0,1
    for i in range(n):
        sequence.append(a)
        a,b=b,a+b
    return sequence

terms=int(input("Enter the number of terms: "))
print("Fibonacci sequence: ",fibonacci(terms))
```

```
Enter the number of terms: 5
Fibonacci sequence:  [0, 1, 1, 2, 3]
```

Use the math module to solve a problem

```
import math

angle=float(input("Enter an angle in degrees:"))
radian=math.radians(angle)
```

```
print("sin of angle:",math.sin(radian))
print("cosine of angle:",math.cos(radian))
```

```
Enter an angle in degrees:45
sin of angle: 0.7071067811865475
cosine of angle: 0.7071067811865476
```

Problem solving

```
#factorial using recursion
```

```
def factorial(n):
    if n ==0:
        return 1
    else:
        return n * factorial(n-1)
```

```
num=int(input("Enter the number:"))
print("factorial:",factorial(num))
```

```
Enter the number:8
factorial: 40320
```

```
#Fibonnaci series using functions
```

```
def fibonacci(n):
    a,b=0,1
    for i in range(n):
        print(a,end=" ")
        a,b=b,a+b
count=int(input("Enter the number:"))
fibonacci(count)
```

```
Enter the number:5
0 1 1 2 3
```

```
#Reverse a string using a function
```

```
def reverse_string(s):
    return s[::-1]

text=input("Enter a string:")
print("Resevsed string:",reverse_string(text))
```

```
Enter a string:pranju
Resevsed string: ujnarp
```

```
#Find GCD of two numbers using a function
```

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

num1=int(input("Enter the first number:"))
```

```
num2=int(input("Enter the second number:"))  
print("GCD:",gcd(num1,num2))
```

```
Enter the first number:12  
Enter the second number:24  
GCD: 12
```