Day 6

User Defined Functions

```
In [7]:  def greet():
             print("Hello, welcome to Python!")
```

```
In [9]:  # Function Call
         greet()
```

```
Hello, welcome to Python!
```

```
In [11]:  def details_bill():
              print("The last date of paying the bill is 20 April")
              print("After deadline you need o pay fine of Rs 1000")
              print("Pay your bill soon")
```

```
In [13]:  for i in range(3):
              a = input("Enter the name of the customer")
              n = int(input("enter the units of electricity consumed"))
              bill = n*10
              print("Total amount paid for the electricity bill", bill)
              details_bill()
```

```
Total amount paid for the electricity bill 5000
The last date of paying the bill is 20 April
After deadline you need o pay fine of Rs 1000
Pay your bill soon
Total amount paid for the electricity bill 2540
The last date of paying the bill is 20 April
After deadline you need o pay fine of Rs 1000
Pay your bill soon
Total amount paid for the electricity bill 360
The last date of paying the bill is 20 April
After deadline you need o pay fine of Rs 1000
Pay your bill soon
```

return

```
In [ ]:  The return statement is essential in Python functions to return results for furt
         It enhances reusability, modularity, and efficiency in programming.
```

```
In [47]:  def add(a, b):
              return a + b  # Returns the sum of a and b
```

```
In [49]:  # Calling the function and storing the result
          result = add(5, 10)
          print("Sum:", result)  # Output: Sum: 15
```

```
Sum: 15
```

Arguments

create a function that checks if a number is even or odd.

```
In [17]:  #Creating a function

          def check_even_odd(number):
              """This function checks if a number is even or odd."""
              if number % 2 == 0:
                  print(f"{number} is even.")
              else:
                  print(f"{number} is odd.")
```

```
In [19]:  check_even_odd(10)
```

10 is even.

```
In [21]:  check_even_odd(7)
```

7 is odd.

2 arguments

Checking Greater Number

```
In [25]:  def find_max(num1, num2):
              if num1 > num2:
                  return num1
              else:
                  return num2
```

```
In [27]:  print(find_max(10, 20))
```

20

```
In [29]:  print(find_max(50, 25))
```

50

```
In [ ]:   A company gives a bonus to employees based on their years of service:

          If an employee has worked for more than 10 years, they get a ₹50,000 bonus.
          If they have worked between 5 and 10 years, they get ₹25,000 bonus.
          If they have worked for less than 5 years, they get ₹10,000 bonus.
          💡 Write a function that takes the number of years an employee has worked and r
```

```
In [33]:  def calculate_bonus(years_of_service):
              if years_of_service > 10:
                  return 50000
              elif 5 <= years_of_service <= 10:
```

```
            return 25000
        else:
            return 10000

    # Example Usage
    years = int(input("Enter years of service: "))
    bonus = calculate_bonus(years)
    print(f"Bonus Amount: ₹{bonus}")
```

Bonus Amount: ₹50000

In [ ]: A smart water meter gives alerts based on daily water usage:

If usage **is** below 100 liters**, return** "Low Consumption"**.**
If usage **is** between 100 **and** 500 liters**, return** "Normal Consumption"**.**
If usage **is** above 500 liters**, return** "High Consumption! Save Water."**.**
💡 Write a function that takes water usage **in** liters **as** input **and** returns an al

In [35]:
```python
def water_consumption_alert(liters):
    if liters < 100:
        return "Low Consumption"
    elif 100 <= liters <= 500:
        return "Normal Consumption"
    else:
        return "High Consumption! Save Water."

# Example Usage
usage = int(input("Enter water usage in liters: "))
print("Alert:", water_consumption_alert(usage))
```

Alert: Low Consumption

In [ ]: A university offers a scholarship to students based on the following conditions:

If a student has a CGPA of 9.0 **or** above **and** their family income **is** below ₹5,00,0
If a student has a CGPA between 8.0 **and** 9.0 **and** their family income **is** below ₹8,
Otherwise**,** the student **is not** eligible **for** a scholarship**.**
💡 Task: Write a Python function that takes CGPA **and** family income **as** input **and**

In [38]:
```python
def scholarship_eligibility(cgpa, income):
    if cgpa >= 9.0 and income < 500000:
        return "100% Scholarship"
    elif 8.0 <= cgpa < 9.0 and income < 800000:
        return "50% Scholarship"
    else:
        return "Not Eligible for Scholarship"

# Example Usage
cgpa = float(input("Enter your CGPA: "))
income = int(input("Enter your family income: ₹"))
print("Scholarship Status:", scholarship_eligibility(cgpa, income))
```

Scholarship Status: 100% Scholarship

Homework

In [ ]: Bank Loan Eligibility
👉 Situation: A bank approves a loan based on the following conditions:

If the person's salary **is** ₹50,000 **or** more **and** their credit score **is** 700 **or** above
Otherwise, they are **not** eligible.
💡 Write a function that takes salary **and** credit score **as** input **and** returns whe

In [ ]: