In [2]:
```
# 1. Understanding
#(a) Define a variable in Python.Provide an example of how to create a variable that stores your name.
# In Python, a variable is like a container that holds information. You can store different types of data in a variable, such as numbers or words.
```

In [3]:
```
Name = "Vaishnavi"
```

In [4]:
```
#(b) What is the difference between a variable and constant? Can we have constant in Python?
# A constant is a piece of data that stays the same throughout the program and doesn't change while it's running.
```

In [5]:
```
#Example
My_name_is_Vaishnavi = "vaishnavi_Gupta"
```

In [6]:
```
#2. Working with Different Data Types
#(a). Create variables of the following types in Python:
```

In [7]:
```
#1. Integer
Num = 1215
print(Num)
```

```
1215
```

In [11]:
```
#2. Float
Num1 = 12.15
print(Num)
```

```
12.15
```

In [9]:
```
#3. String
Name = "Vaishnavi Gupta"
print(Name)
```

```
Vaishnavi Gupta
```

In [10]:
```
#4. Boolean
bo = 1<2
print(bo)
```

```
True
```

In [14]:
```
#b. Write a Python script to display the type of each variable you created.
integer_variable = 1215          # Integer
float_variable = 12.15           # Float
string_variable = "Vaishnavi Gupta"  # String
boolean_variable = True          # Boolean

print(f"The type of Num is: {type(integer_variable)}")
print(f"The type of Num1 is: {type(float_variable)}")
print(f"The type of Name is: {type(string_variable)}")
print(f"The type of bo is: {type(boolean_variable)}")
```

```
The type of Num is: <class 'int'>
The type of Num1 is: <class 'float'>
The type of Name is: <class 'str'>
The type of bo is: <class 'bool'>
```

In [15]:
```
#3. Arithmetic Operators
#a. Explain the following arithmetic operators with examples:
```

In [19]:
```
a = 12 #Addition (`+`) = Combines two values to get their total
b = 15
c = a+b
print(c)
```

```
27
```

In [20]:
```
a = 12 #Subtraction: Subtracts the second number from the first
b = 15
c = a-b
print(c)
```

```
-3
```

In [21]:
```
a = 12 #Multiplication (*): Multiplies two numbers.
b = 15
c = a*b
print(c)
```

```
180
```

In [23]:
```
a = 15 #Division (/): Divides the first number by the second and always returns a decimal result, even if both numbers are whole numbers.
b = 12
c = a/b
print(c)
```

```
1.25
```

In [24]:
```
a = 15 #Floor Division (//): Divides two numbers and returns the largest whole number that is less than or equal to the result.
b = 12
c = a//b
print(c)
```

```
1
```

In [25]:
```
a = 15 #Modulus (%): Returns the remainder when the first number is divided by the second.
b = 12
c = a%b
print(c)
```

```
3
```

In [27]:
```
a = 5 #Exponentiation (**): Raises the first number to the power of the second.
b = 3
c = a**b
print(c)
```

```
125
```

In [29]:
```
#b. Write a Python script to calculate the area of a rectangle using variables `length`and `width`
with values 5 and 10, respectively. Use the multiplication operator.
```

In [35]:
```
length = 5
width = 12
area = length*width
print(area)
```

```
60
```

In [1]:
```
#Comparison and Logical Operators
#a. Explain the following comparison operators with examples:
```

In [2]:
```
 #1.Equal to (==): Returns True if both values are the same, otherwise False.
x = 15
v = 15
y = 12
z = 22
print(x==v)
print(y==z)
```

```
True
False
```

In [3]:
```
#2. Not equal to (!=): Returns True if the values are different, otherwise False.
a = 15
b = 12
c = 15
print(a!= b)
print(a!= c)
```

```
True
False
```

In [4]:
```
#Greater than (>): Returns True if the left value is larger than the right, otherwise False
a = 15
b = 12
print(a>b)
```

```
True
```

In [5]:
```
#Less than (<): Returns True if the left value is smaller than the right, otherwise False.
a = 12
b = 15
print(a<b)
```

```
True
```

In [1]:
```
#Greater than or equal to (>=): Returns True if the left value is larger than or equal to the right, otherwise False.
a = 15
b = 12
print(a>=b)
```

```
True
```

In [3]:
```
#Less than or equal to (<=): Returns True if the left value is smaller than or equal to the right, otherwise False.
a = 6
b = 12
print(a<=b)
```

```
True
```

In [1]:
```
#b Using logical operators  (and, or, not), write a python script that check if a number is positive and even

a = int(input("Enter a number:"))

if  a > 0 and a % 2 == 0:
        print (f"{a} is positive and even")
elif a > 0 and a % 2 != 0:
        print (f"{a} is positive but not even")
else :
        print(f"{a} is neither positive nor even")
```

```
15 is positive but not even
```

In [3]:
```
# 5 Type Casting is Python
#a What is Type casting? explain the difference between implicit and explicit type casting with examples.
#Type casting is the process of changing data from one type to another. In implicit type casting,
#Python automatically changes the data type without needing any input from the user.
#In explicit type casting, the programmer manually changes the data type using
#Python's built-in functions like `int()`, `float()`, `str()`, and others.


#implicit type casting
a = 5
b = 5.6
c = a + b
print(c)
type(b)
```

```
10.6
```

Out[3]: float

In [4]:
```
#Explicit type casting
a = 88
b = int(a)
print (b)
type (b)
```

```
88
```

Out[4]: int

In [5]:
```
#b write a python script that:

#convert  a float to an interger
a = 48.2
b = int(a)
print (b)
type (b)
```

```
48
```

Out[5]: int

In [6]:
```
#convert  a interger to a String
a = 84
b = str(a)
print (b)
type (b)
```

```
84
```

Out[6]: str

In [7]:
```
#convert  a string to a float
a = "9.968"
b = float(a)
print (b)
type (b)
```

```
9.968
```

Out[7]: float

In [8]:
```
#6 Practical Exercise: Mini calculator
#write a python script that asks the user to input two number and then

num1 = int(input("Enter a number:"))
num2 = int(input("Enter a number:"))
add = num1 + num2
subs = num1 - num2
mult = num1 * num2
div = num1 / num2
print(add)
print(subs)
print(mult)
print(div)
```

```
1230
-1200
```

```
18225
0.012345679012345678
```

In [ ]: