**Name:** Mujahid Ali Rattar

**Class:** BSIT-3A

**Section:** A

**Lab Title:**  Artificial Intelligence

**Instructor:** Mam Aqsa Umar

**Objectives 01: Make 2 Programs on each data type.**

**01. Numeric Types:**

**-Int**

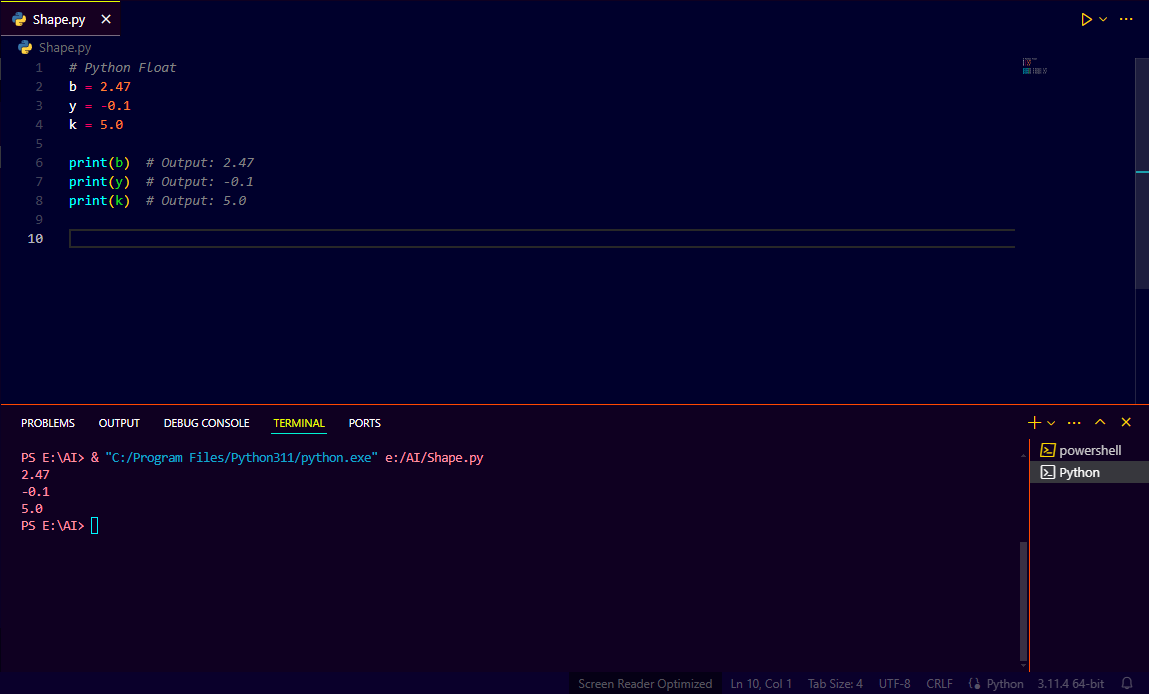
# Python Integer  
a = 7  
y = -1  
c = 0  
  
print(a)  # Output: 7  
print(y)  # Output: -1  
print(c)  # Output: 0



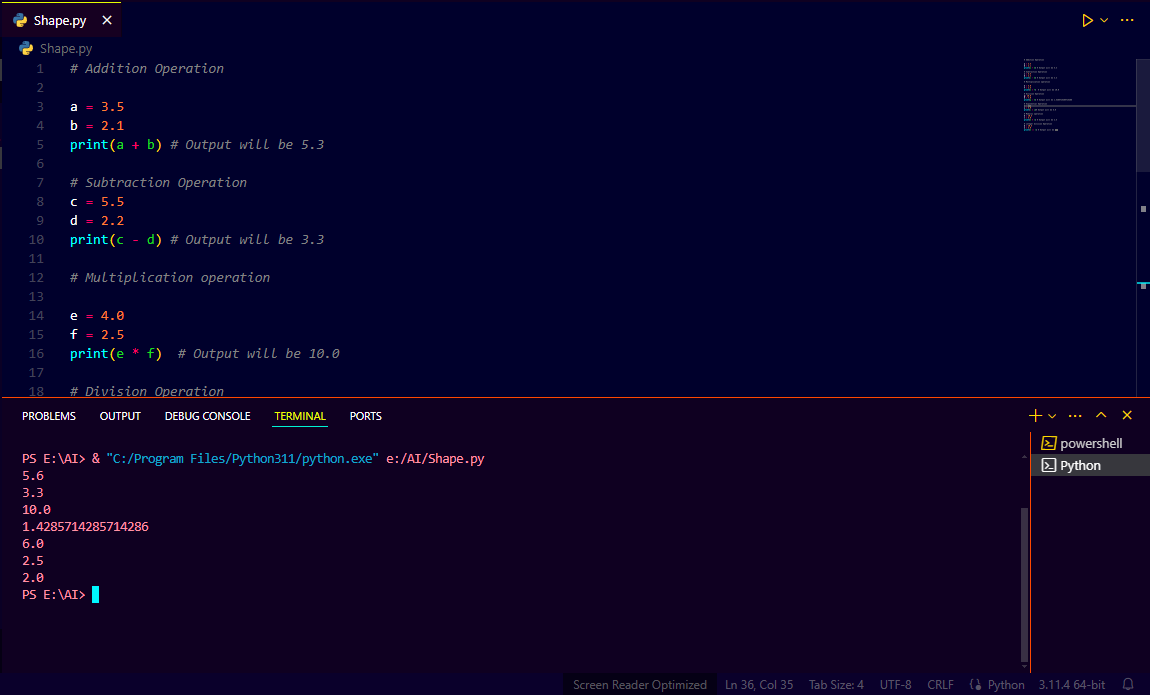
# Addition Operation +  
  
addition = 8 + 3  
print("Addition:", addition)  # Output: 11  
  
# Subtraction operation -  
subtraction = [9 - 4](E:/AI/tel:9 - 4)  
print("Subtraction:", subtraction)  # Output: 5  
  
# Multiplication operation \*  
multiplication = 10 \* 2  
print("Multiplication:", multiplication)  # Output: 20  
  
# Division operation /  
division = 10 / 6  
print("Division:", division)  # Output: [1.6666666666666667](E:/AI/tel:1.6666666666666667)  
  
# Integer Division operation //  
integer\_division = 10 // 2  
print("Integer Division:", integer\_division)  # Output: 5  
  
# Modulus operation %  
modulus = 10 % 5  
print("Modulus:", modulus)  # Output: 0  
  
# Exponentiation operation \*\*  
exponentiation = 2 \*\* 6  
print("Exponentiation:", exponentiation)  # Output: 64



-**Flaot**

# Python Float  
b = [2.47](E:/AI/tel:2.47)  
y = -[0.1](E:/AI/tel:0.1)  
k = [5.0](E:/AI/tel:5.0)  
  
print(b)  # Output: [2.47](E:/AI/tel:2.47)  
print(y)  # Output: -[0.1](E:/AI/tel:0.1)  
print(k)  # Output: [5.0](E:/AI/tel:5.0)

# Addition Operation  
a = [3.5](E:/AI/tel:3.5)  
b = [2.1](E:/AI/tel:2.1)  
print(a + b) # Output will be [5.3](E:/AI/tel:5.3)  
# Subtraction Operation  
c = [5.5](E:/AI/tel:5.5)  
d = [2.2](E:/AI/tel:2.2)  
print(c - d) # Output will be [3.3](E:/AI/tel:3.3)  
# Multiplication operation  
e = [4.0](E:/AI/tel:4.0)  
f = [2.5](E:/AI/tel:2.5)  
print(e \* f)  # Output will be [10.0](E:/AI/tel:10.0)  
# Division Operation  
g= [10.0](E:/AI/tel:10.0)  
h = [7.0](E:/AI/tel:7.0)  
print(g / h) # Output will be [1.4285714285714286](E:/AI/tel:1.4285714285714286)  
# Exponential Operation  
i = [2.0](E:/AI/tel:2.0)  
j = [3.0](E:/AI/tel:3.0)  
print(i \* j)# Output will be [6.0](E:/AI/tel:6.0)  
# Modulus operation  
k = [10.5](E:/AI/tel:10.5)  
l = [4.0](E:/AI/tel:4.0)  
print(k % l) # Output will be [2.5](E:/AI/tel:2.5)  
# integer division Operation  
m = [10.5](E:/AI/tel:10.5)  
n = [3.0](E:/AI/tel:3.0)  
print(k // l) # Output will be [2.0](E:/AI/tel:2.0)



**Sequence Type:**

# List creation  
  
the\_list = [1, 2, 4, 5]  
  
  
# creating a mixed data list  
multiple\_data\_list = [1, 'hi', [2.57](E:/AI/tel:2.57), False]  
print(the\_list[0])   # Output: 1  
print(multiple\_data\_list[2])  # Output:[2.57](E:/AI/tel:2.57)  
# Mutable feature of list  
  
the\_list[0] = 10          # Modify the first element  
[the\_list.append](http://the_list.append)(9)        # Append a new element  
[the\_list.extend](http://the_list.extend)([5, 4])   # Extend the list with another list  
[the\_list.remove](http://the_list.remove)(2)        # Remove an element by value  
del the\_list[0]           # Remove an element by index

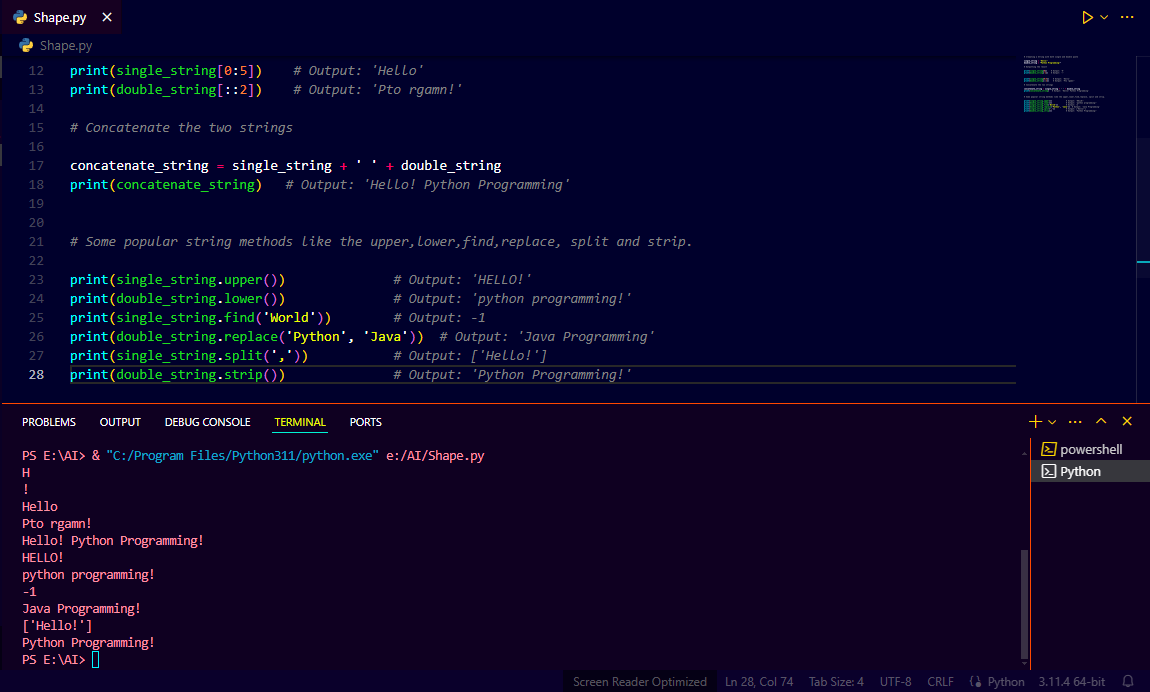
-**Truple**

# Empty tuple  
zilch\_tuple = ()    
# Tuple with a single element             
single\_tuple = (1,)  
# Tuple with multiple elements  
multiple\_tuple = (1,8,9,3, 5)  
# Tuple immutability  
single\_slice[0] = 5  
print(tuple\_slice) # TypeError: 'tuple' object does not support item assignment  
# Different elements tuples  
mixed\_tuple = (1, 'hello', [3.14](E:/AI/tel:3.14), True)  
# Nested tuple  
nested\_tuple = ('Orange', ('banana', 'Pineapple'), ["he", 'she', 'them'])  
# Concatenate tuples  
add\_tuple = multiple\_tuple + (6, 7, 8)  # Output: (1, 8, 9, 4, 3, 5, 7, 8)  
# Create a slice of the tuple  
tuple\_slice = add\_tuple[1:3]     # Output: (8, 9)



-**Str**

# Creaating a String with both single and double quote  
single\_string = 'Hello!'  
double\_string = "Python Programming!"  
# Outputting the result  
print(single\_string[0])    # Output: 'H'  
print(double\_string[-1])   # Output: '!'  
print(single\_string[0:5])    # Output: 'Hello'  
print(double\_string[::2])    # Output: 'Pto rgamn!'  
# Concatenate the two strings  
concatenate\_string = single\_string + ' ' + double\_string  
print(concatenate\_string)   # Output: 'Hello! Python Programming'  
# Some popular string methods like the upper,lower,find,replace, split and strip.  
print([single\_string.upper](http://single_string.upper)())              # Output: 'HELLO!'  
print([double\_string.lower](http://double_string.lower)())              # Output: 'python programming!'  
print([single\_string.find](http://single_string.find)('World'))        # Output: -1  
print([double\_string.replace](http://double_string.replace)('Python', 'Java'))  # Output: 'Java Programming'  
print([single\_string.split](http://single_string.split)(','))           # Output: ['Hello!']  
print([double\_string.strip](http://double_string.strip)())              # Output: 'Python Programming!'



**Mapping Type:**

person = {  
    "name": "Kamaldeen",  
    "age": 32,  
    "city": "Nigeria"  
}



**Set Types:**

# Creating curly braces set  
curly\_set = {1, 2, 6, 4, 9}  
# Creating  set() function set  
func\_set = set([1, 2, 6, 9, 5])

**Boolean Type:**

a = 10  
b = 15  
  
# Comparison operators  
print(a == b)  # False  
print(a < b)   # True  
# Logical operators  
print(a < 10 and b > 5)  # False  
print(a < 3 or b> 20)   # False  
print(not(a == b))        # True  
# Control Flow  
age = 75  
if age >= 18:  
    print("You are an adult.")  
else:  
    print("You are a minor.")  
# Functions return  
def is\_even(number):  
    return number % 2 ==   
print(is\_even(10))  # True  
print(is\_even(7))  # False

