بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

The Islamia University of Bahawalpur

Baghdad-ul-Jadheed campus

Submitted to:

S Rehan Faheem

Submitted by:

Alia Saeed

Program:

BS CS Eve

Semester:

7thth

Subject:

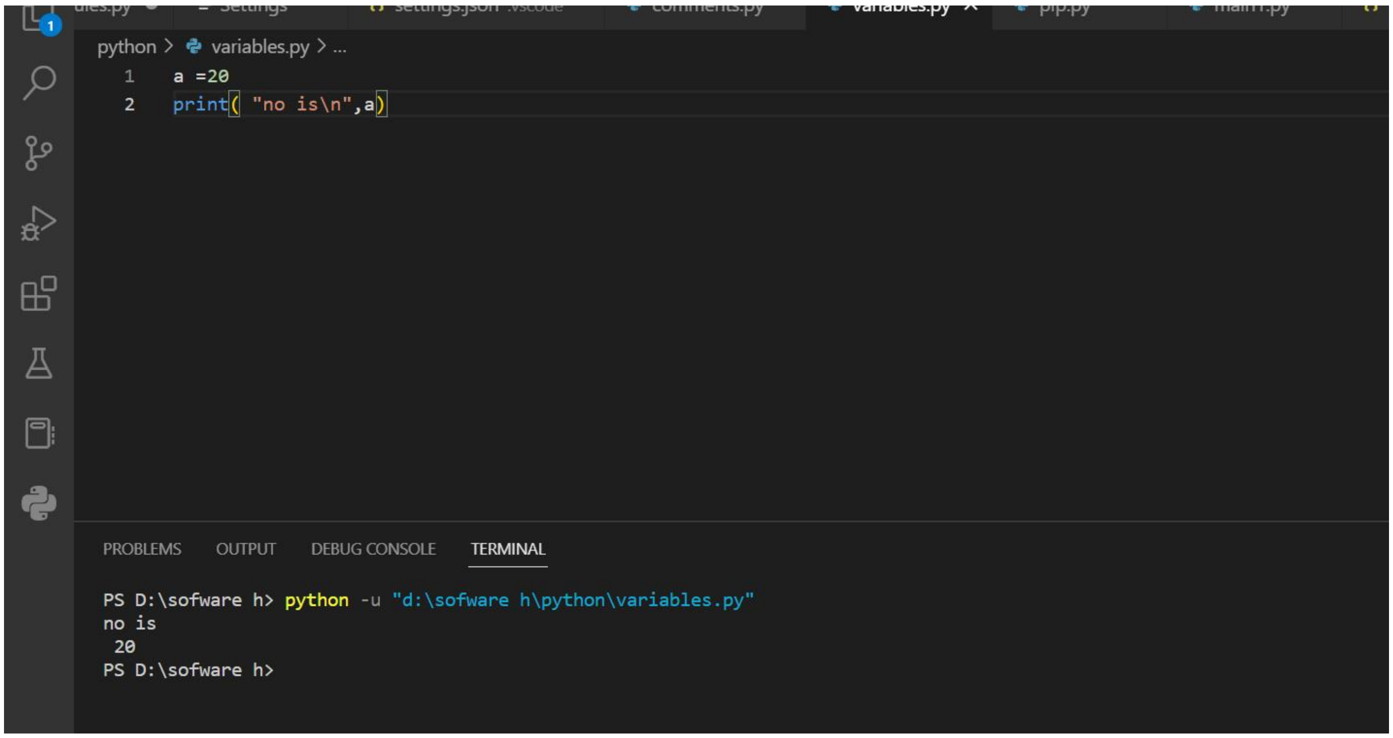
Web design & framework

Roll no: **SP20M2BB024**

Chapter # 2

➢ Variables:

A Python variable is a reserved memory location to store values.



```
python > variables.py > ...
a =20
print( "no is\n",a)
```

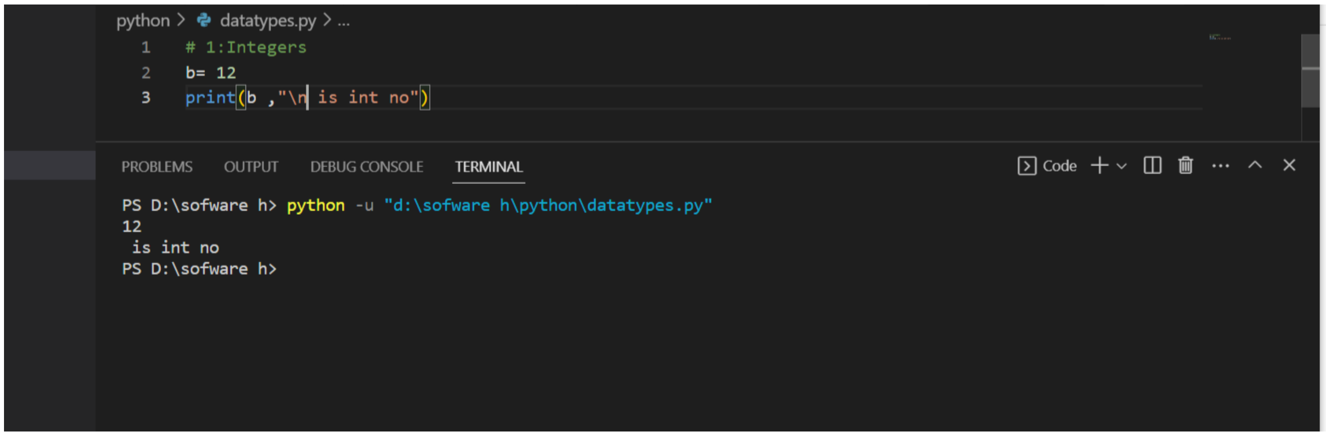
```
PS D:\sofware h> python -u "d:\sofware h\python\variables.py"
no is
 20
PS D:\sofware h>
```

➢ Data types in Python:

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data

PYTHON DATA TYPES:

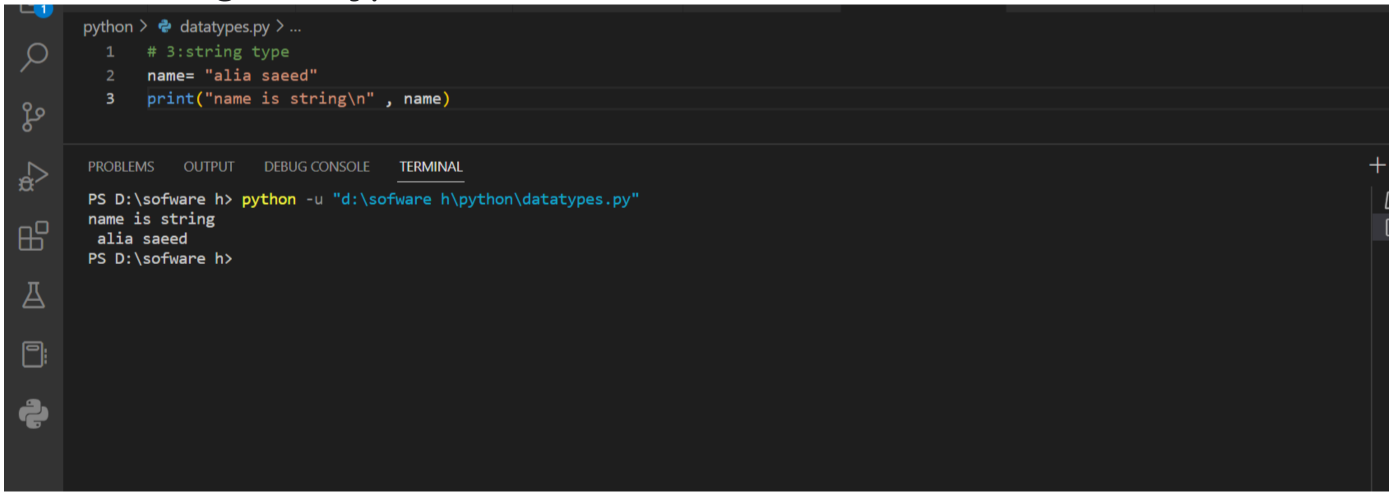
- **Numeric data types: int, float, complex.**



```python
# 1:Integers
b= 12
print(b ,"\n is int no")
```

```
PS D:\sofware h> python -u "d:\sofware h\python\datatypes.py"
12
 is int no
PS D:\sofware h>
```

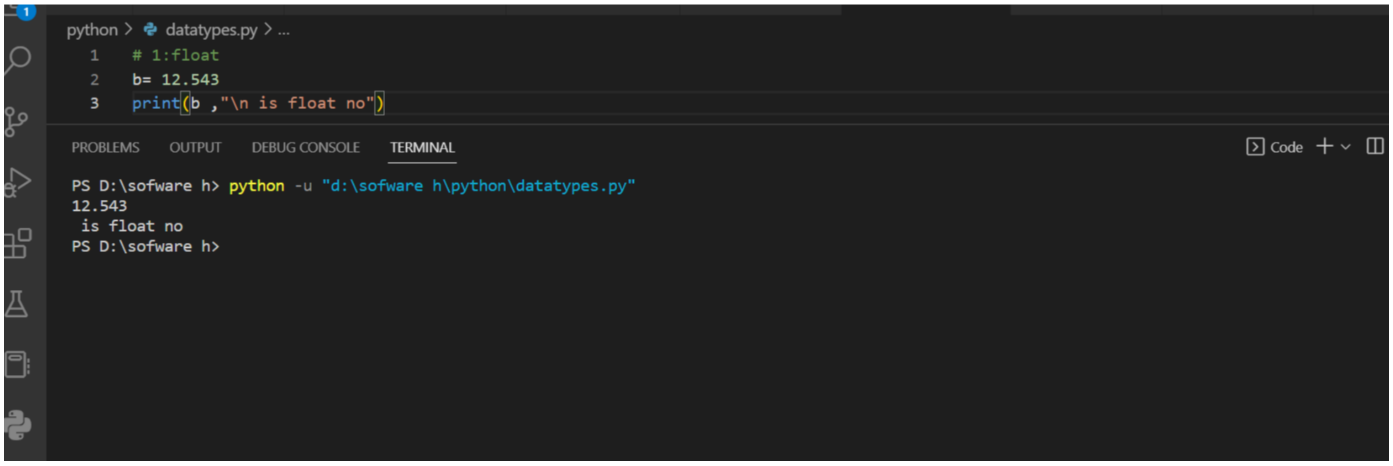
- **String data types: str.**



```python
# 3:string type
name= "alia saeed"
print("name is string\n" , name)
```

```
PS D:\sofware h> python -u "d:\sofware h\python\datatypes.py"
name is string
 alia saeed
PS D:\sofware h>
```

- **Float types:**

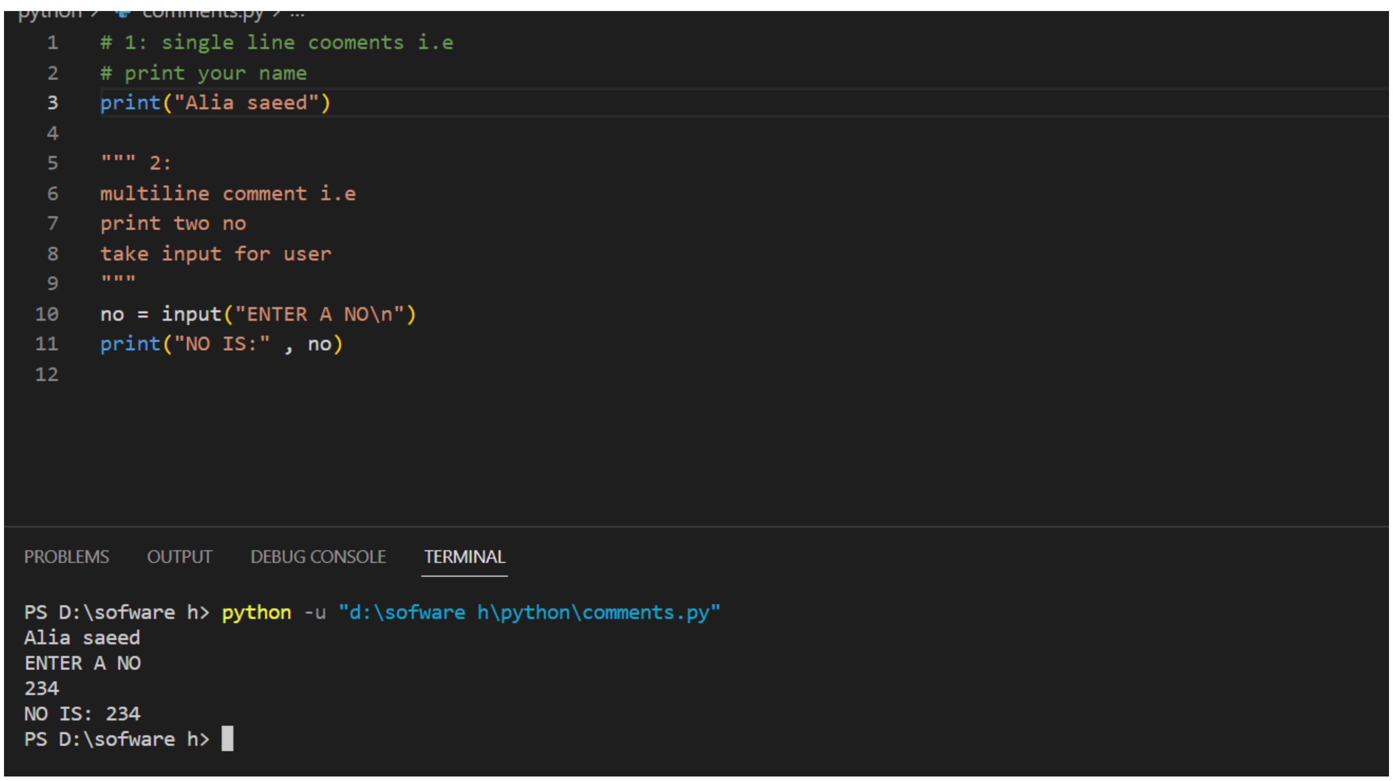


```python
# 1:float
b= 12.543
print(b ,"\n is float no")
```

```
PS D:\sofware h> python -u "d:\sofware h\python\datatypes.py"
12.543
 is float no
PS D:\sofware h>
```

➢ Rules for defining variable:

- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive

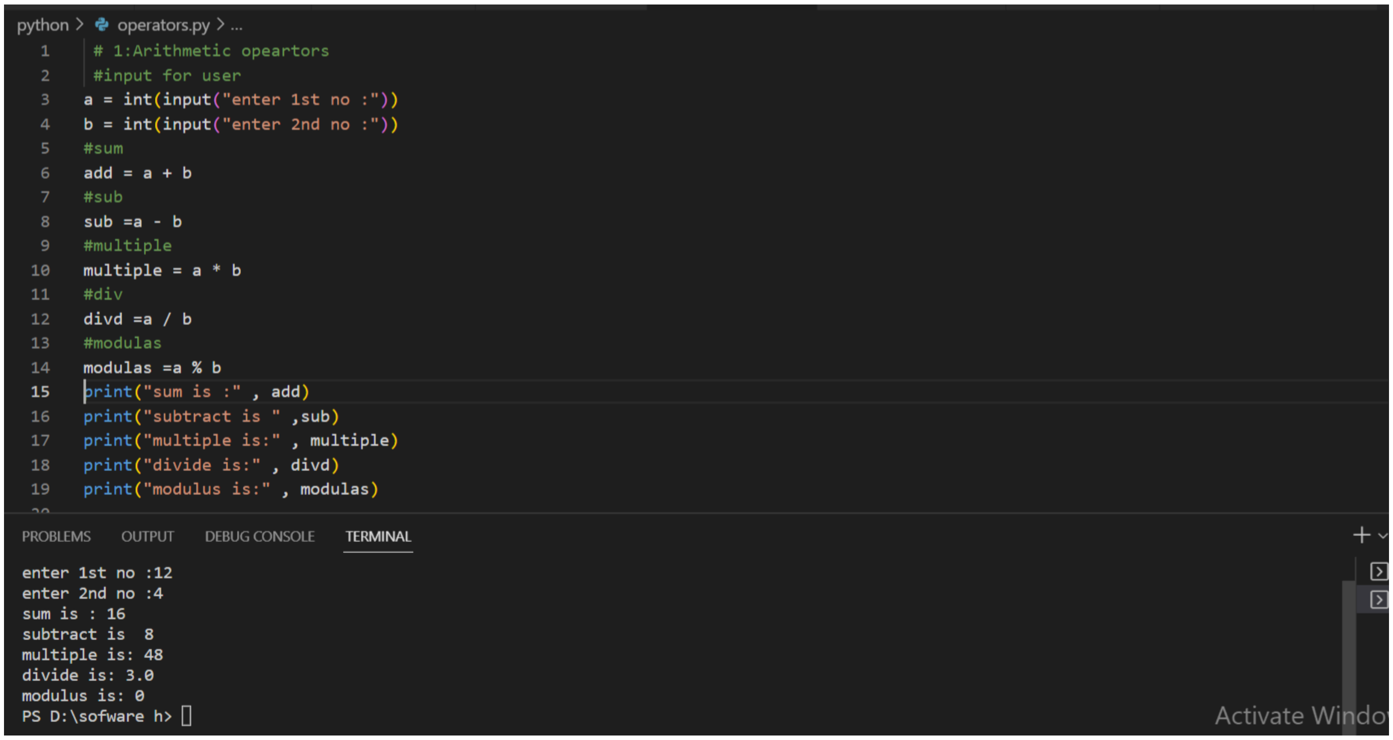


```python
# 1: single line cooments i.e
# print your name
print("Alia saeed")

""" 2:
multiline comment i.e
print two no
take input for user
"""
no = input("ENTER A NO\n")
print("NO IS:" , no)
```

```
PS D:\sofware h> python -u "d:\sofware h\python\comments.py"
Alia saeed
ENTER A NO
234
NO IS: 234
PS D:\sofware h>
```

Operator in python:

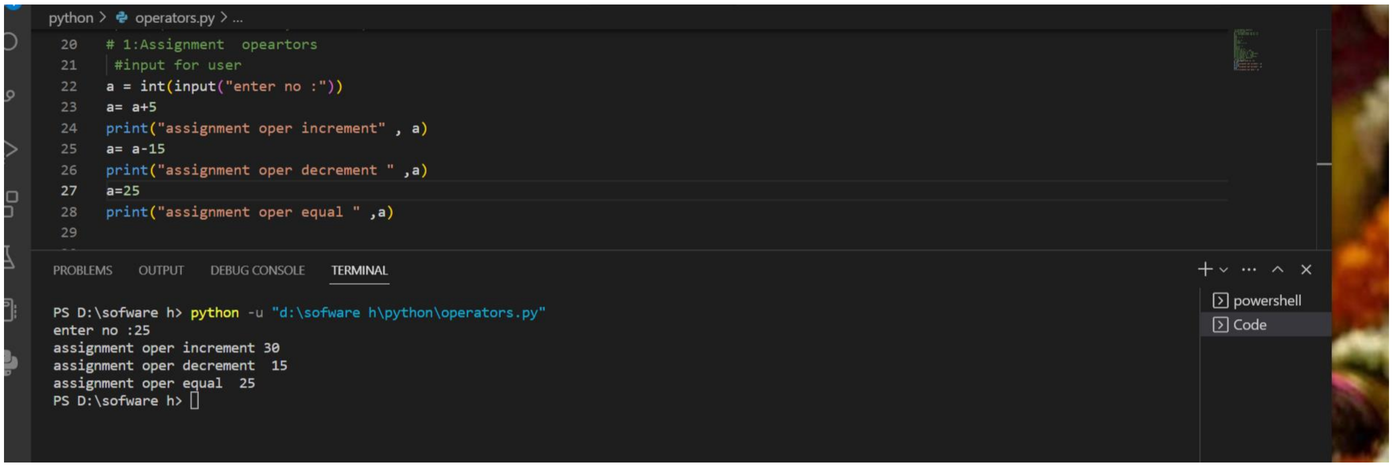
- **Arithmetic operators**



```
 # 1:Arithmetic opeartors
 #input for user
a = int(input("enter 1st no :"))
b = int(input("enter 2nd no :"))
#sum
add = a + b
#sub
sub =a - b
#multiple
multiple = a * b
#div
divd =a / b
#modulas
modulas =a % b
print("sum is :" , add)
print("subtract is " ,sub)
print("multiple is:" , multiple)
print("divide is:" , divd)
print("modulus is:" , modulas)
```

```
enter 1st no :12
enter 2nd no :4
sum is : 16
subtract is  8
multiple is: 48
divide is: 3.0
modulus is: 0
PS D:\sofware h> 
```

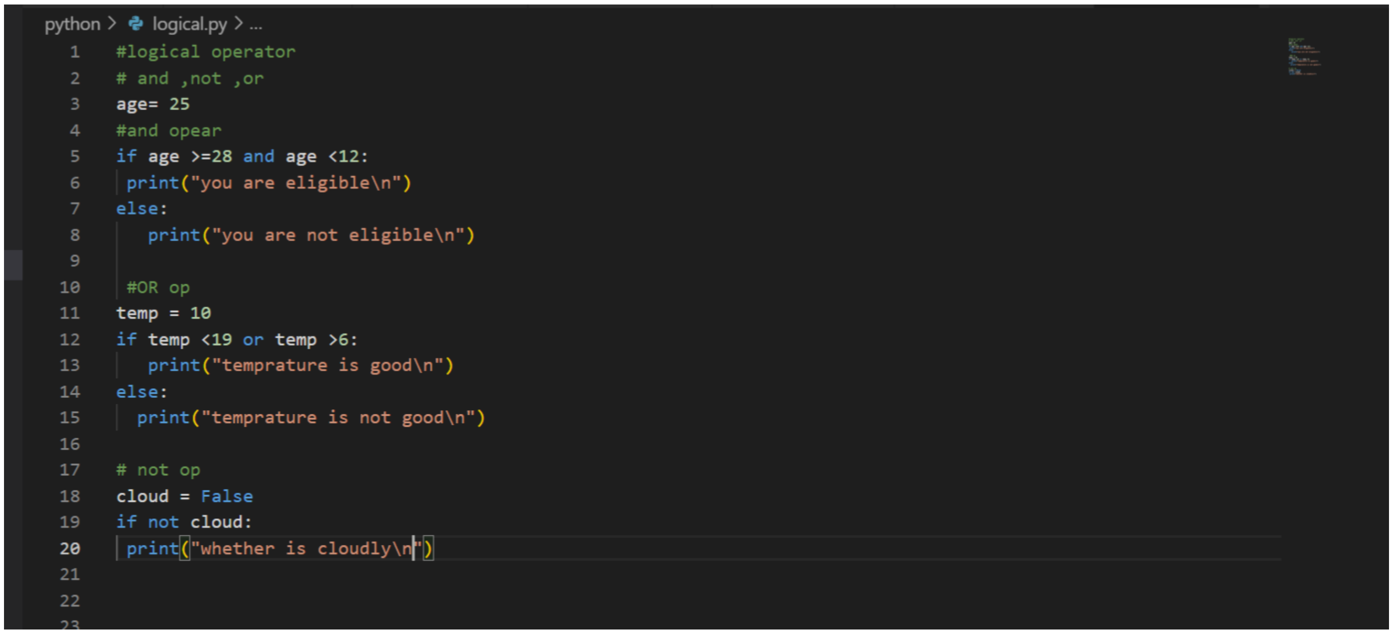
- **Assignment operator**



```
# 1:Assignment  opeartors
 #input for user
a = int(input("enter no :"))
a= a+5
print("assignment oper increment" , a)
a= a-15
print("assignment oper decrement " ,a)
a=25
print("assignment oper equal " ,a)
```

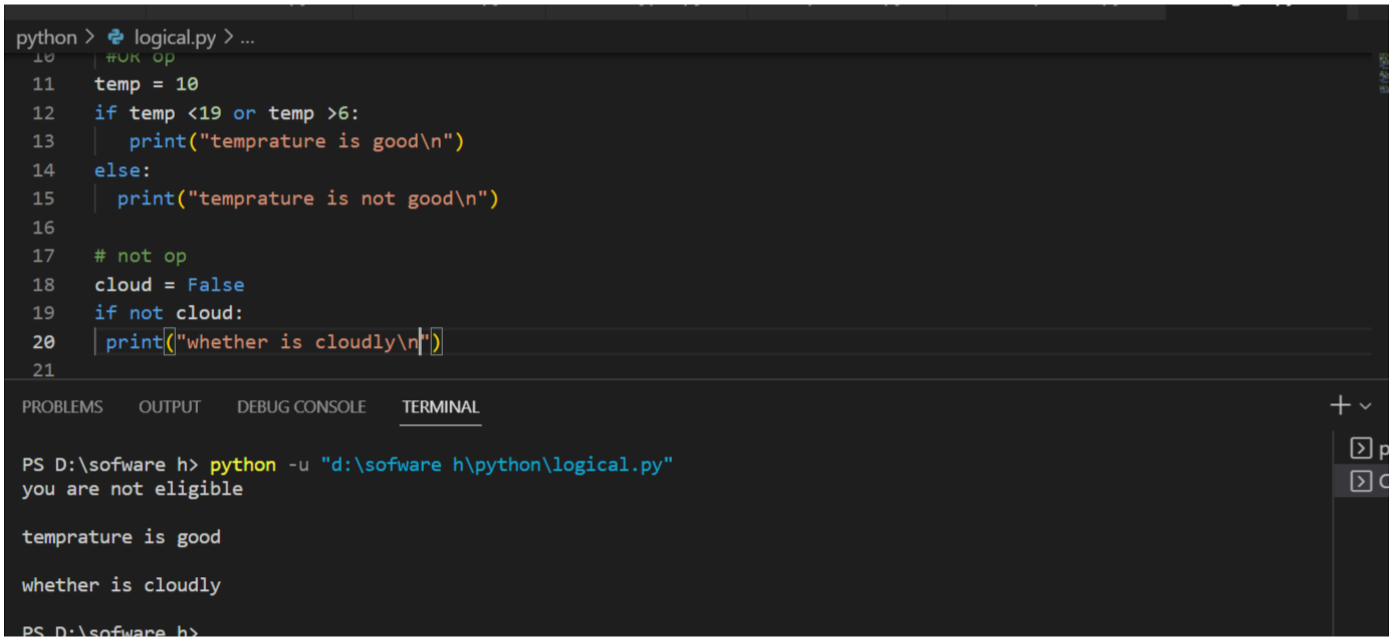
```
PS D:\sofware h> python -u "d:\sofware h\python\operators.py"
enter no :25
assignment oper increment 30
assignment oper decrement  15
assignment oper equal  25
PS D:\sofware h> 
```

- **Logical operator:**



```
python > logical.py > ...
#logical operator
# and ,not ,or
age= 25
#and opear
if age >=28 and age <12:
  print("you are eligible\n")
else:
    print("you are not eligible\n")

  #OR op
temp = 10
if temp <19 or temp >6:
    print("temprature is good\n")
else:
  print("temprature is not good\n")

# not op
cloud = False
if not cloud:
  print("whether is cloudly\n")
```

Output:

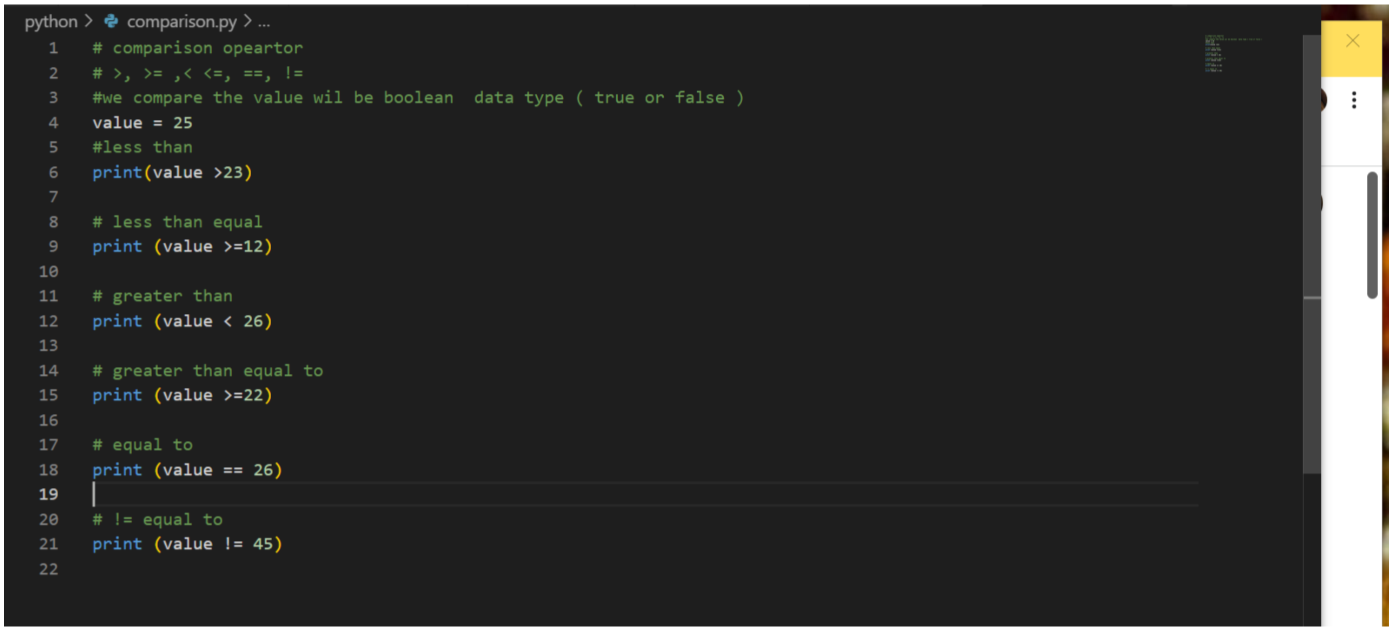


```
python > logical.py > ...
  #OR op
temp = 10
if temp <19 or temp >6:
    print("temprature is good\n")
else:
  print("temprature is not good\n")

# not op
cloud = False
if not cloud:
  print("whether is cloudly\n")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\sofware h> python -u "d:\sofware h\python\logical.py"
you are not eligible

temprature is good

whether is cloudly

PS D:\sofware h>
```

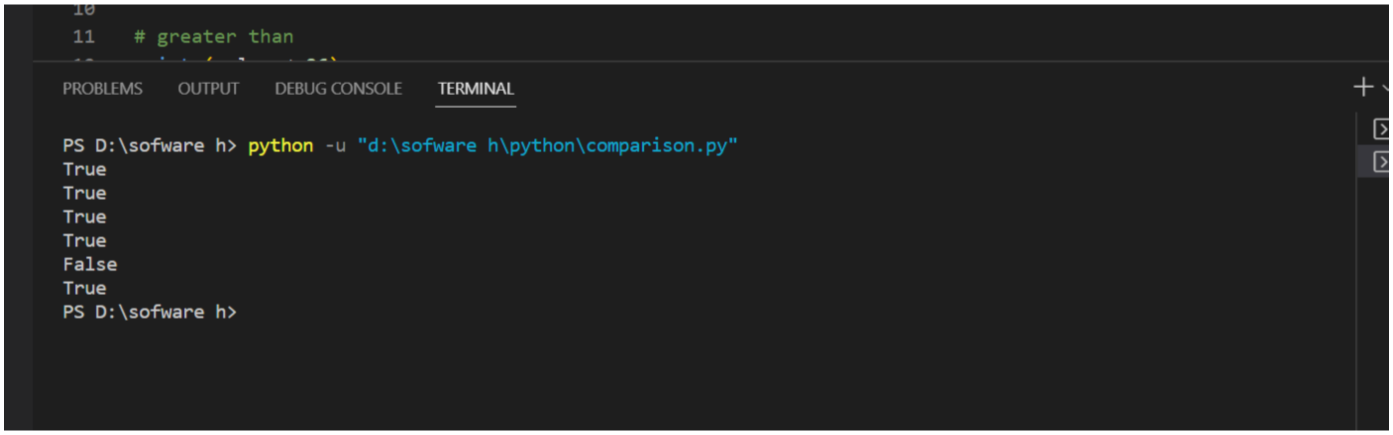
- **Comparison operator:**



python > comparison.py > ...

```python
# comparison opeartor
# >, >= ,< <=, ==, !=
#we compare the value wil be boolean  data type ( true or false )
value = 25
#less than
print(value >23)

# less than equal
print (value >=12)

# greater than
print (value < 26)

# greater than equal to
print (value >=22)

# equal to
print (value == 26)

# != equal to
print (value != 45)
```

Output :

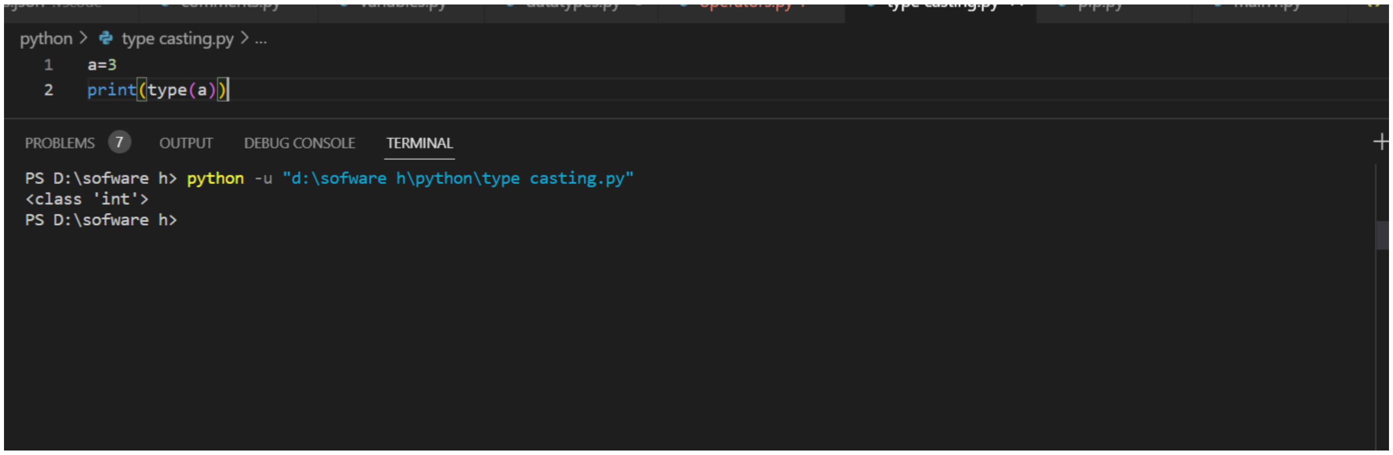


```
PS D:\sofware h> python -u "d:\sofware h\python\comparison.py"
True
True
True
True
False
True
PS D:\sofware h>
```

➢ Type () function:

Python type() is a built-in function that returns the type of the objects/data elements stored in any data type .

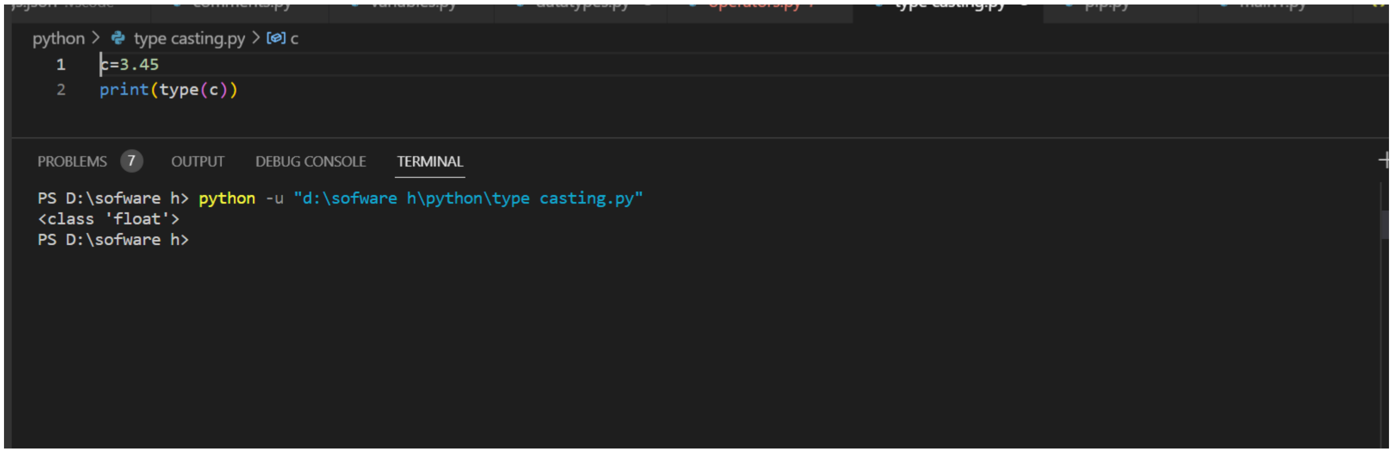
- EXAMPLE:



```python
a=3
print(type(a))
```

```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
<class 'int'>
PS D:\sofware h>
```

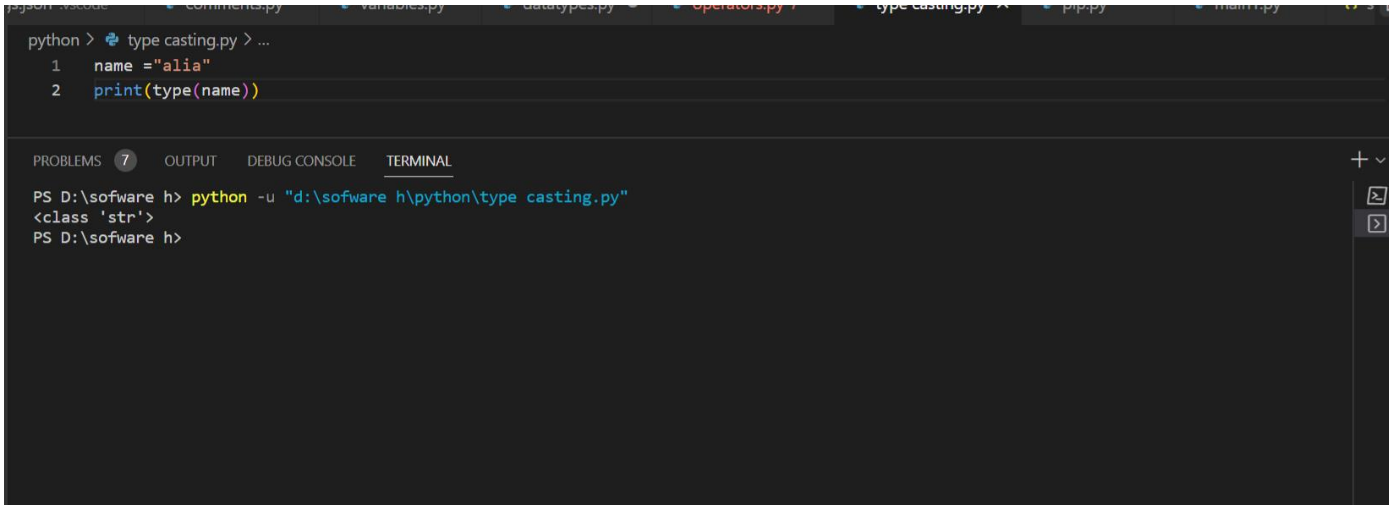
AND



```python
c=3.45
print(type(c))
```

```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
<class 'float'>
PS D:\sofware h>
```

AND



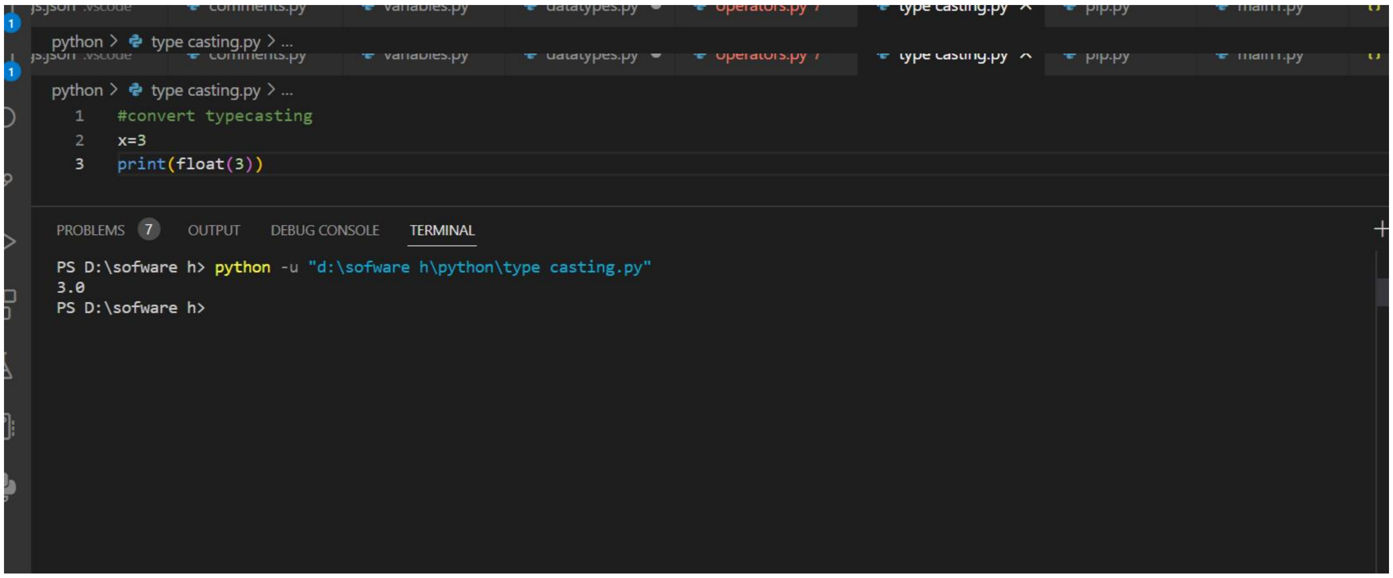
```python
name ="alia"
print(type(name))
```

```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
<class 'str'>
PS D:\sofware h>
```

➢ Type casting

Type Casting is the method to convert the variable data type into a certain data type

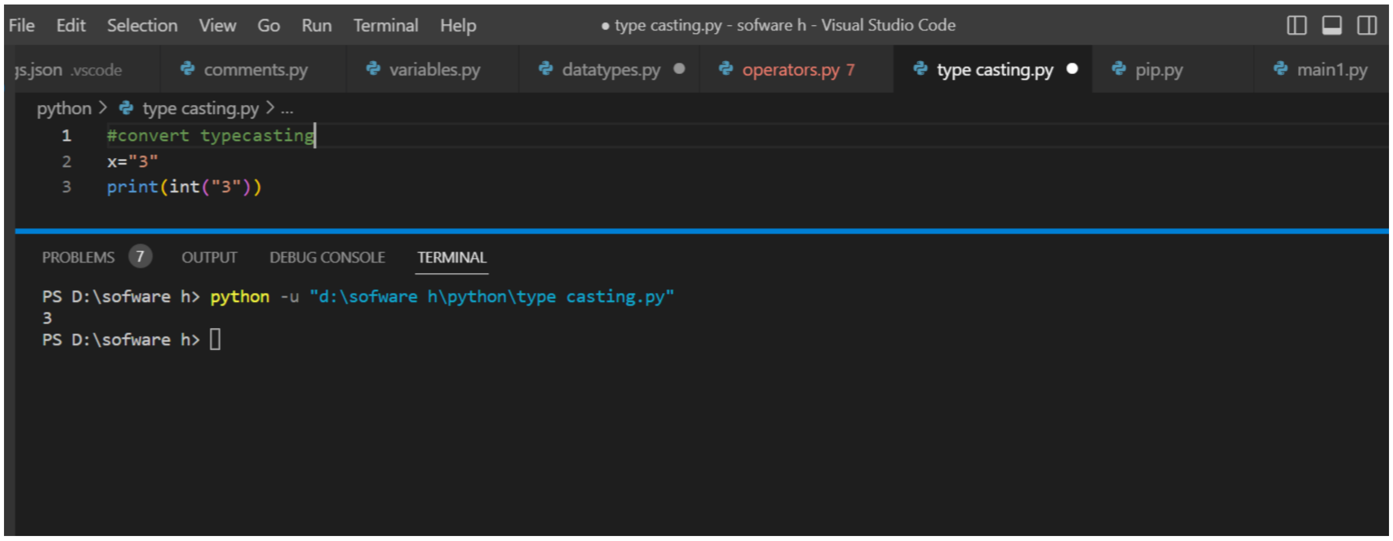
- **Types of casting:**
  I. - Explicit Conversion(Explicit type casting in python),
  II. Implicit Conversion(Implicit type casting in python)
- Int to float casting



```python
#convert typecasting
x=3
print(float(3))
```

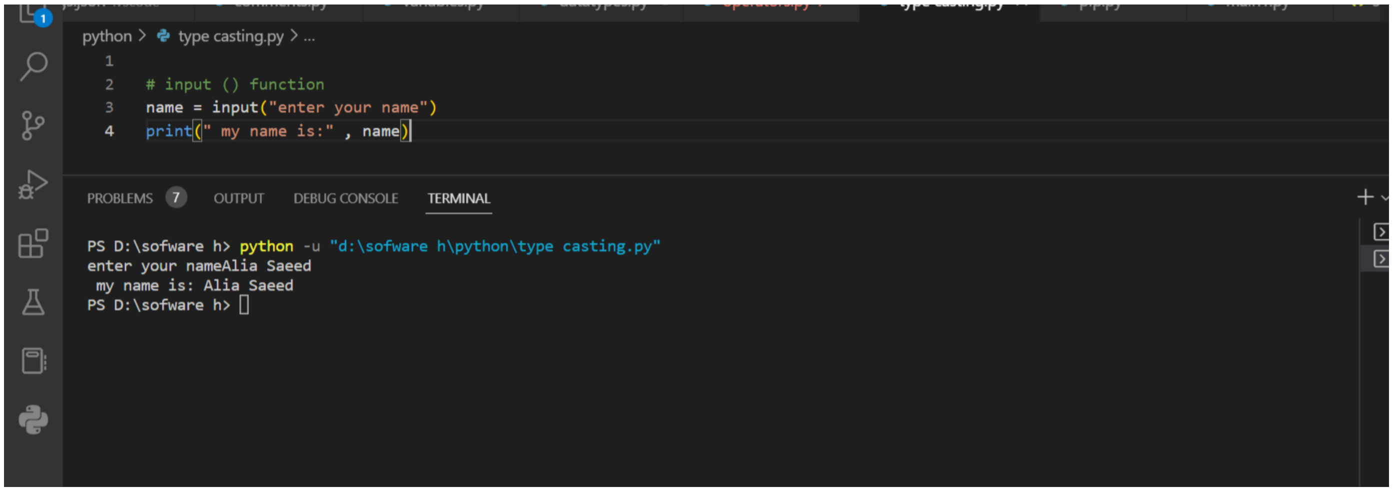
```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
3.0
PS D:\sofware h>
```

- string to int casting



```python
#convert typecasting
x="3"
print(int("3"))
```

```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
3
PS D:\sofware h>
```



```python

# input () function
name = input("enter your name")
print(" my name is:" , name)
```

```
PS D:\sofware h> python -u "d:\sofware h\python\type casting.py"
enter your nameAlia Saeed
 my name is: Alia Saeed
PS D:\sofware h> 
```