

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



Baghdad-ul-Jadheed campus

Submitted to:

Sir Reehan Faheem

Submitted by:

Faria Safdar

Program:

BS CS Eve, A

Semester:

7th

Subject:

Web Design & Framework

Roll no:

SP20M2BB041

Chapter-2

Variables & Datatypes

➤ Variables:

```
9
10 # Creating Variables
11
12 x = 4          # x is of type int
13 x = "Faria"    # x is now of type str
14 print(x)
15
16 # Get the Type
17
18 x = 5
19 y = "John"
20 print(type(x))
21 print(type(y))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + - [] [] ... ^ X

```
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python311/python.exe" c:/Users/Administrator/Desktop/python/comments.py
Faria
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python311/python.exe" c:/Users/Administrator/Desktop/python/comments.py
Faria
<class 'int'>
<class 'str'>
PS C:\Users\Administrator\Desktop\python>
```

➤ Datatypes 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.

- String Datatype:

```
23
24 # In Python, the data type is set when you assign a value to a variable:
25
26
27 x = "Faria Safdar"
28
29 #display x:
30 print(x)
31
32 #display the data type of x:
33 print(type(x))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Administrator\Desktop\python> & "C:/Program Files/Python311/python.exe" c:/Users/Administrator/Desktop/python/comments.py
Faria Safdar
<class 'str'>
PS C:\Users\Administrator\Desktop\python>
```

- Float Datatype:

```
python > datatypes.py > ...
1 # 1:float
2 b= 12.543
3 print(b, "\n is float no")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code + v

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

Operator in python:

- Arithmetic operator

```
python > operators.py > ...
1  # 1:Arithmetic opeartors
2  #input for user
3  a = int(input("enter 1st no :"))
4  b = int(input("enter 2nd no :"))
5  #sum
6  add = a + b
7  #sub
8  sub = a - b
9  #multiple
10 multiple = a * b
11 #div
12 divd =a / b
13 #modulas
14 modulas =a % b
15 print("sum is : " , add)
16 print("subtract is " ,sub)
17 print("multiple is:" , multiple)
18 print("divide is:" , divd)
19 print("modulus is:" , modulas)
20

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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:\software h> 
```

- Assignment operator

```
python > operators.py > ...
20 # 1:Assignment opeartors
21 #input for user
22 a = int(input("enter no :"))
23 a= a+5
24 print("assignment oper increment" , a)
25 a= a-15
26 print("assignment oper decrement " ,a)
27 a=25
28 print("assignment oper equal " ,a)
29
--
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\software h> python -u "d:\software h\python\operators.py"
enter no :25
assignment oper increment 30
assignment oper decrement 15
assignment oper equal 25
PS D:\software h> 
```

- Logical operator:

```
python > logical.py > ...
1  #logical operator
2  # and ,not ,or
3  age= 25
4  #and opear
5  if age >=28 and age <12:
6      print("you are eligible\n")
7  else:
8      print("you are not eligible\n")
9
10 #OR op
11 temp = 10
12 if temp <19 or temp >6:
13     print("temprature is good\n")
14 else:
15     print("temprature is not good\n")
16
17 # not op
18 cloud = False
19 if not cloud:
20     print("whether is cloudy\n")
21
22
23
```

Output:

```
python > logical.py > ...
10  #OR op
11  temp = 10
12  if temp <19 or temp >6:
13      print("temprature is good\n")
14  else:
15      print("temprature is not good\n")
16
17  # not op
18  cloud = False
19  if not cloud:
20      print("whether is cloudy\n")
21
22
23
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\software h> python -u "d:\software h\python\logical.py"
you are not eligible

temprature is good

whether is cloudy

PS D:\software h>
```

- **Comparison operator:**

```
python > comparison.py > ...
1  # comparison opearator
2  # >, >=, <, <=, ==, !=
3  #we compare the value will be boolean data type ( true or false )
4  value = 25
5  #less than
6  print(value >23)
7
8  # less than equal
9  print (value >=12)
10
11 # greater than
12 print (value < 26)
13
14 # greater than equal to
15 print (value >=22)
16
17 # equal to
18 print (value == 26)
19 |
20 # != equal to
21 print (value != 45)
22
```

Output :

```
10
11 # greater than
12 print (value < 26)
13
14 # greater than equal to
15 print (value >=22)
16
17 # equal to
18 print (value == 26)
19 |
20 # != equal to
21 print (value != 45)
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\software h> python -u "d:\software h\python\comparison.py"
True
True
True
True
False
True
PS D:\software 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 > type casting.py > ...
1 a=3
2 print(type(a))
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

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

```
python > type casting.py > [c]
1 f=3.45
2 print(type(c))
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

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

```
File Edit Selection View Go Run Terminal Help
• type casting.py - software h - Visual Studio Code
js.json .vscode comments.py variables.py datatypes.py operators.py 7 type casting.py pip.py main1.py

python > type casting.py > ...
1 #convert typecasting
2 x="3"
3 print(int("3"))
```

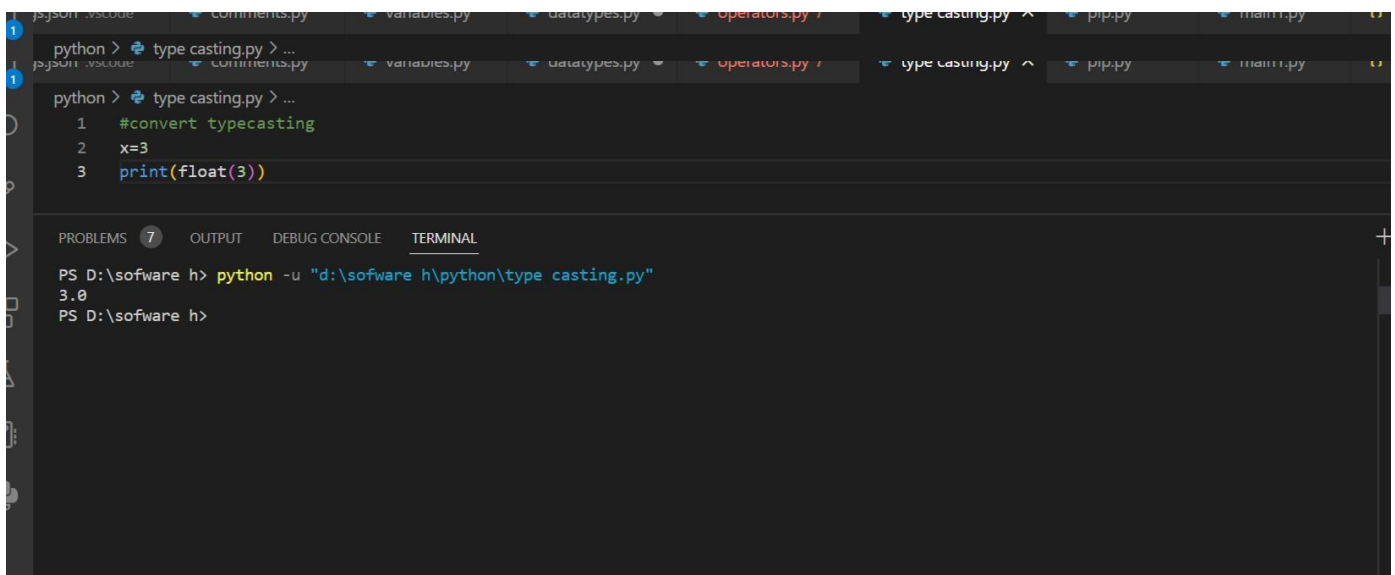
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\software h> python -u "d:\software h\python\type casting.py"
3
PS D:\software 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)
- **string to int casting**



The screenshot shows a Python IDE with a file explorer at the top displaying various Python files. The main editor window shows a script named `type casting.py` with the following code:

```
1 #convert typecasting
2 x=3
3 print(float(3))
```

Below the editor, the `TERMINAL` tab is active, showing the command `python -u "d:\software h\python\type casting.py"` being executed. The output of the script is `3.0`.