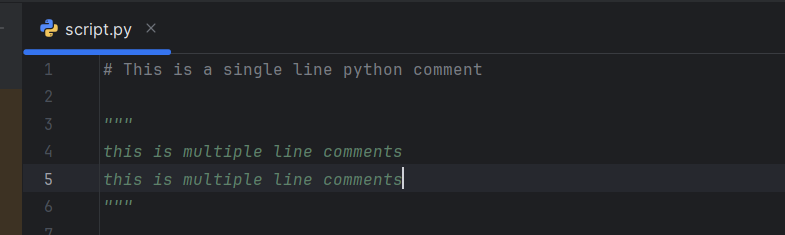
Core Python

***How to add comments in python code?***



***Rules Of naming conventions and DataType:***

\*) Python ***supports snake case*** which is used for the ***naming conventions*** to ***cleanly separate the words in the variable***

Example:

***product\_name*** = “gameBoy”

\*) In Python remember the ***constants*** are always declared as ***captial letters variable***

Example:

***PI*** = 3.14

Python supports following dataType ***int, string, float , complex, boolean, list, tuple, set, dictionaries.***

***How type ints works in python?***

Binding variable along with the dataType.

Synatax:

***Varaible:dataType = value***

Example:

***name: str =*** “Jhon Wick”

***isAdult: bool =*** True

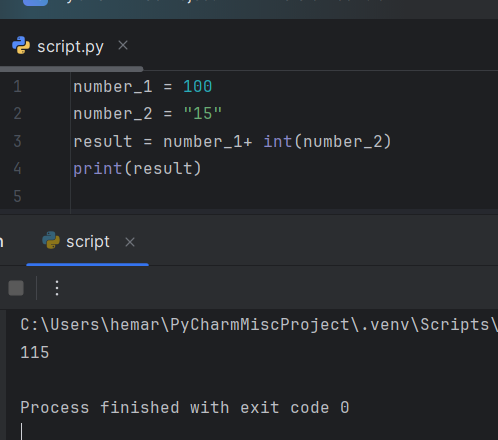
***salary: float*** = 9000.23

***emp\_id:int*** = 136



***How Type conversion works in python?***

This mainly to convert from ***one datatype to another datatype*** by mentioning the ***dataType before the variable***.



***What operators Python supports?***

***+***  --> addition operator

***-***  --> subtractor operator

***/***  --> division operator

***%*** --> modulo division operator

**\***  --> multiply operator

**\*\*** --> power (example 10 \*\* = 3 that means 10 power 3 (10^3 = 1000))

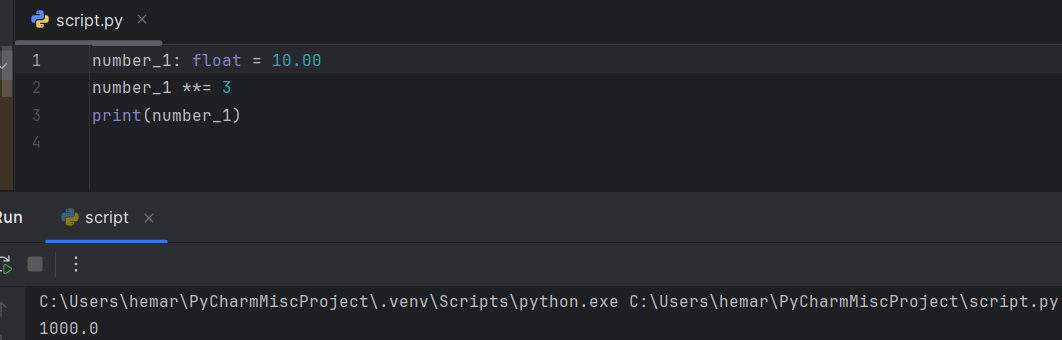
***and , or , not*** operator also used in python

***==*** equal to operator checks only value if the variables

***is*** operator is used to check the memory of the variables.



Power operator:



***Collection Types DATATYPE Python:***

List:

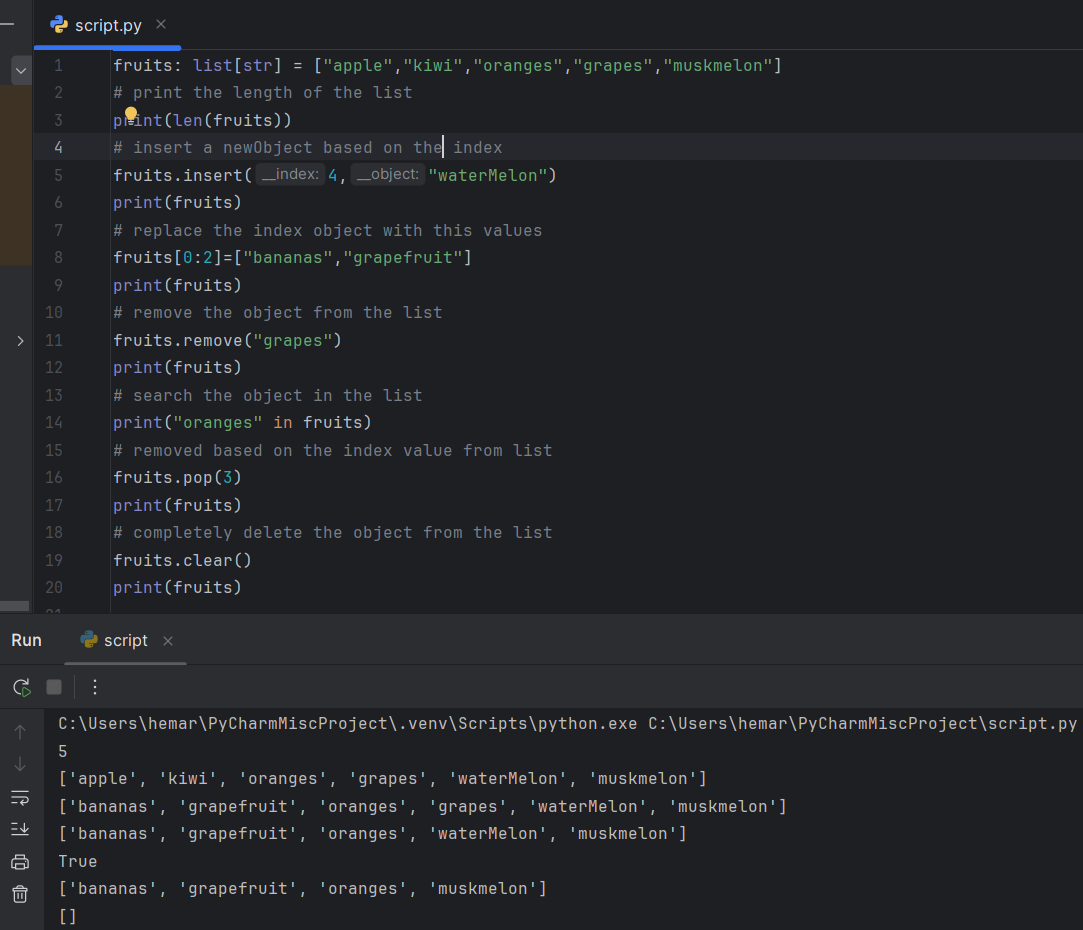
--> In this we can multiple dataTypes without any issue

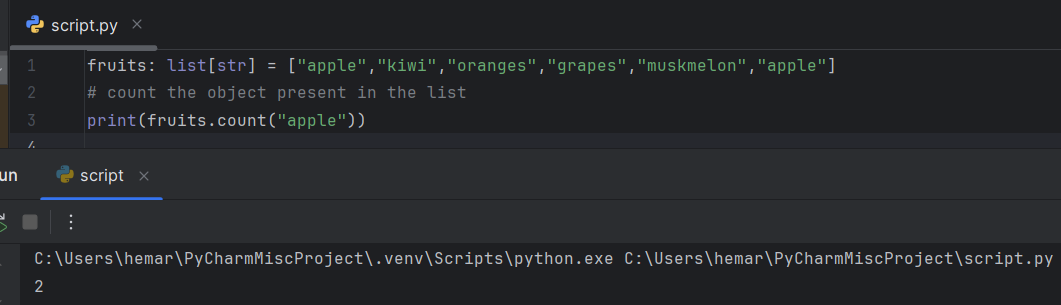
--> Even we can restrict the list to store only one DataType

--> It always follow insertion order

***Syntaxes:***

***Variable:list[dataType] = [mention objects]***

******

******

******

***Tuple:***

--> You cannot add , update or delete the objects that is present in the tuple.

--> it follows the insertion order

--> This is mainly used for the memory optimization and exceutes the python code faster

***Synatax:***

**Variable: tuple[dataType] = (mention the object values)**



***SET:***

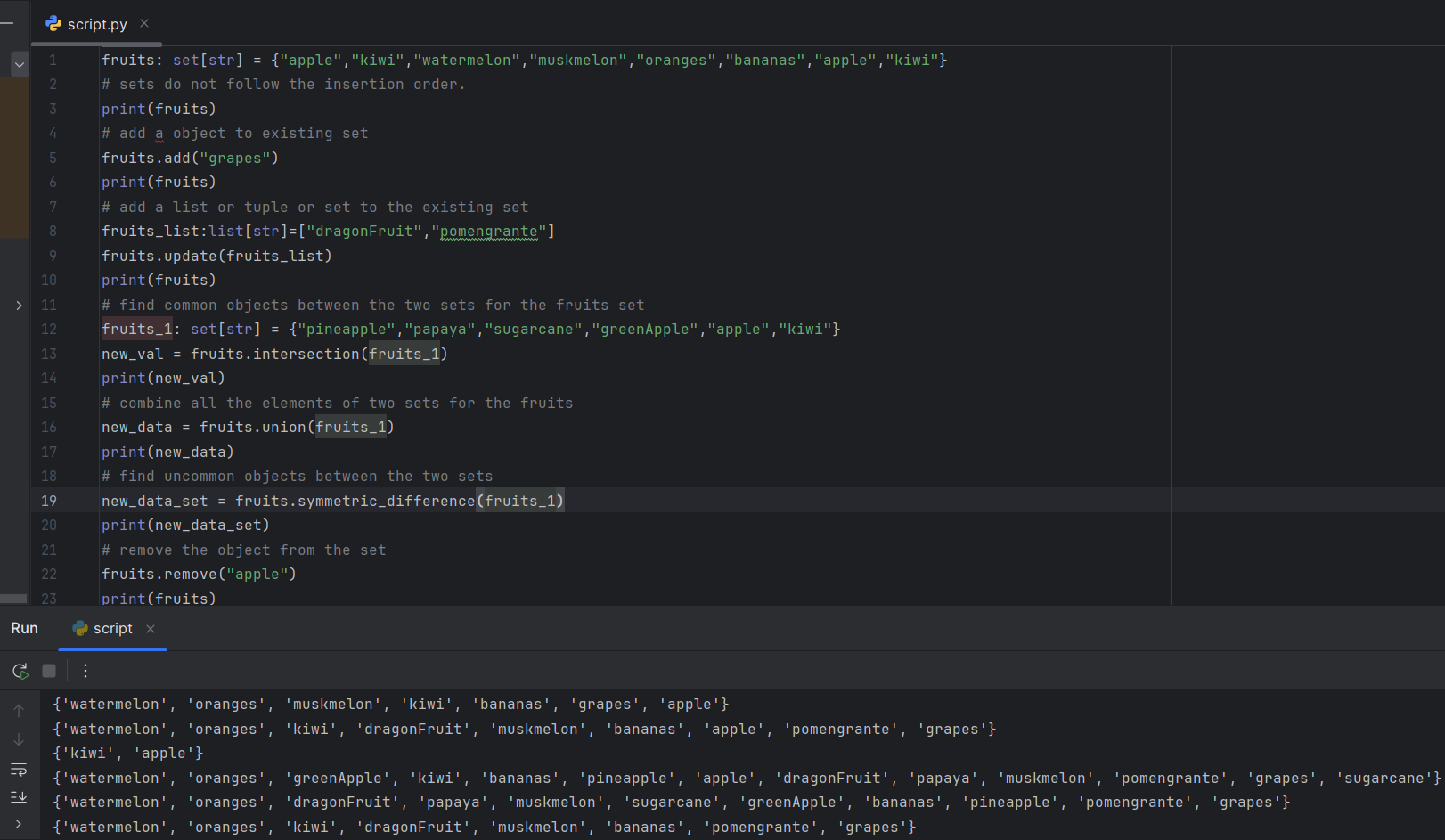
--> It does not follow any insertion order

--> it won’t allow the duplicates get stored in the sets.

--> used for the memory optimization and exceutes the python code faster

***Syntax:***

***Varaible: set[dataType] = {mention the object values}***



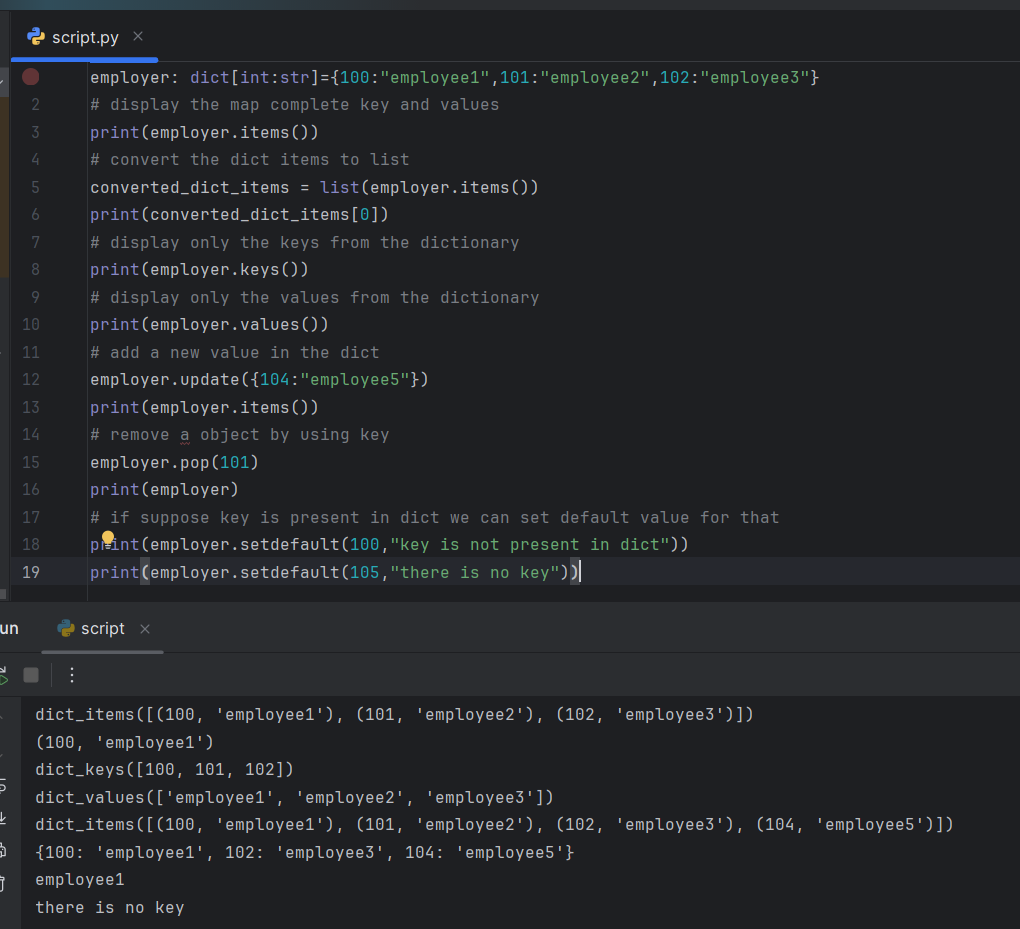
***Dictionary:***

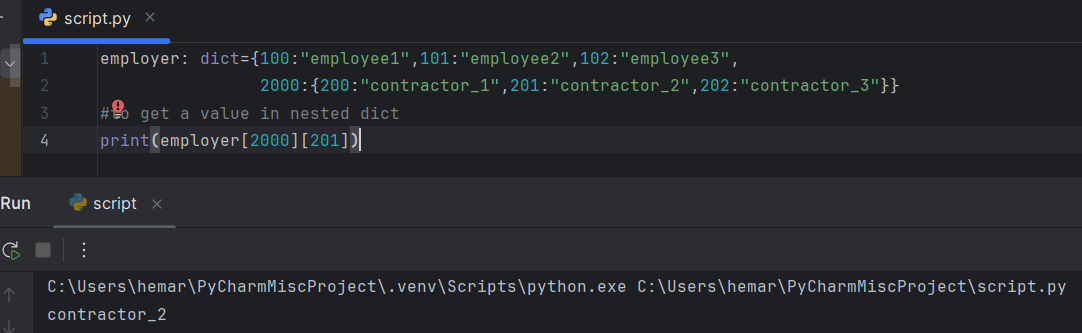
--> it stores the value in the form of key and value.

Syntax:

***Variable:dict = {key1:value1,key2:value2,…….}***

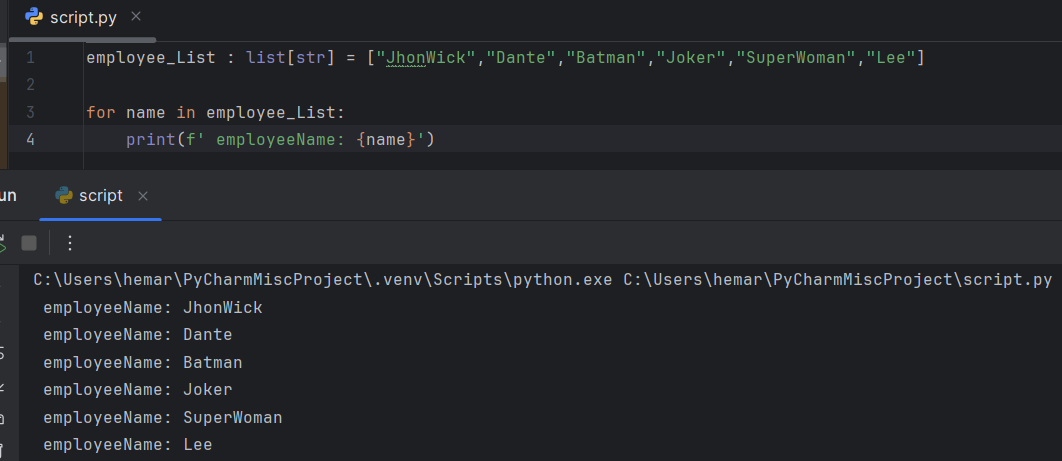
***Variable = {}***

******

******

***Control Flows Python:***

***For loop list iterate example:***



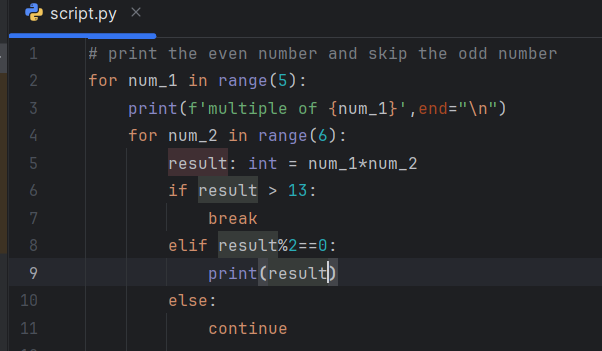
***For nested loop example:***



***While loop example:***

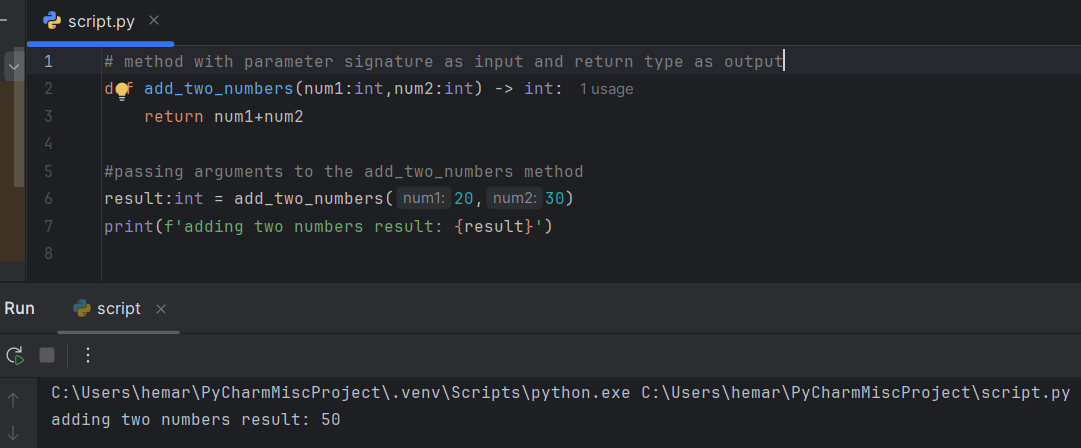


***Break and Continue:***

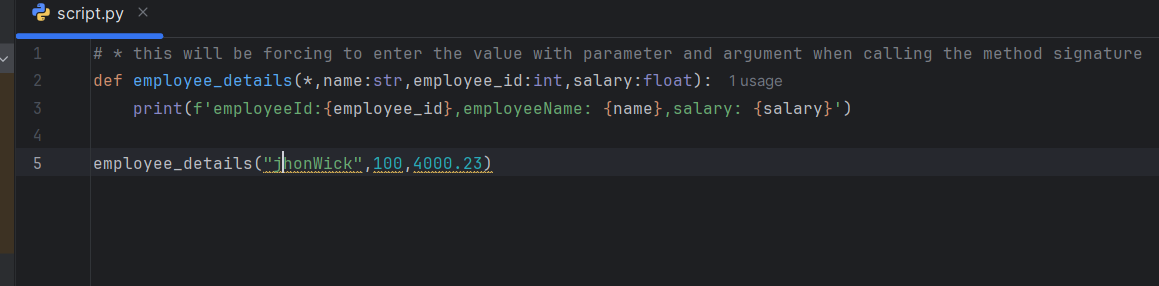


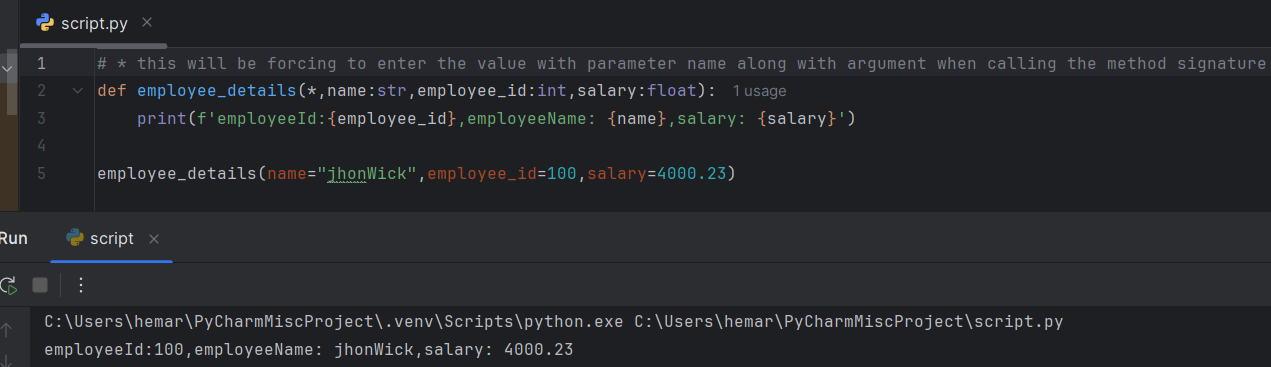
***Python Method Signature:***

*Method signature with the return type:*

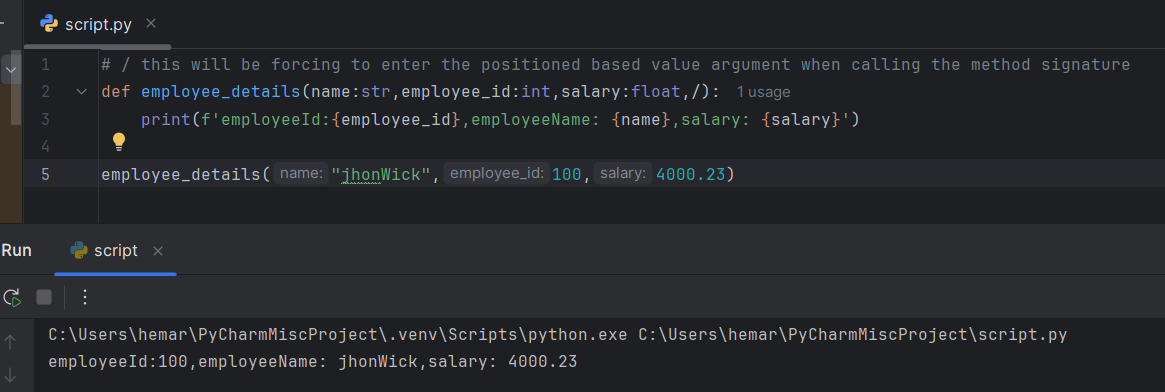


method signature which as ***\* in front of the*** ***method*** which means when you invoke the method you need to ***pass the parameter name along with arguments(values)*** if you don’t pass the parameter name method will throw error with yellow line

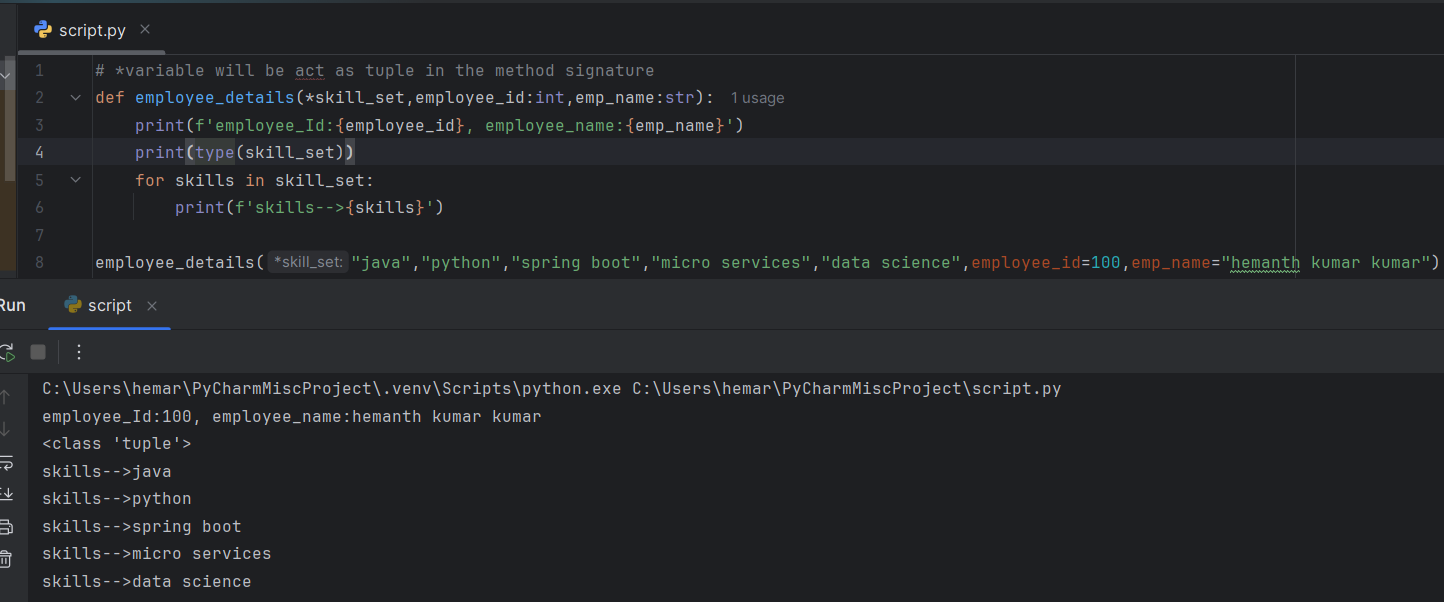




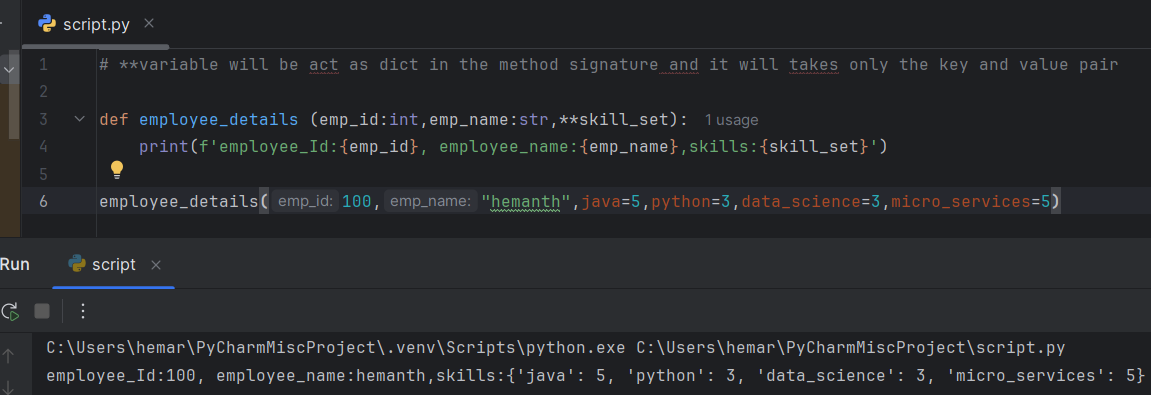
Method signature with ***/ at the end of the method which*** means you invoke the method arguments ***needs to pass in the positional value.***

******

Method signature with ***\*variable in the method*** which means that value will be taken ***as the tuple***. While invoking the method you can enter as many values.

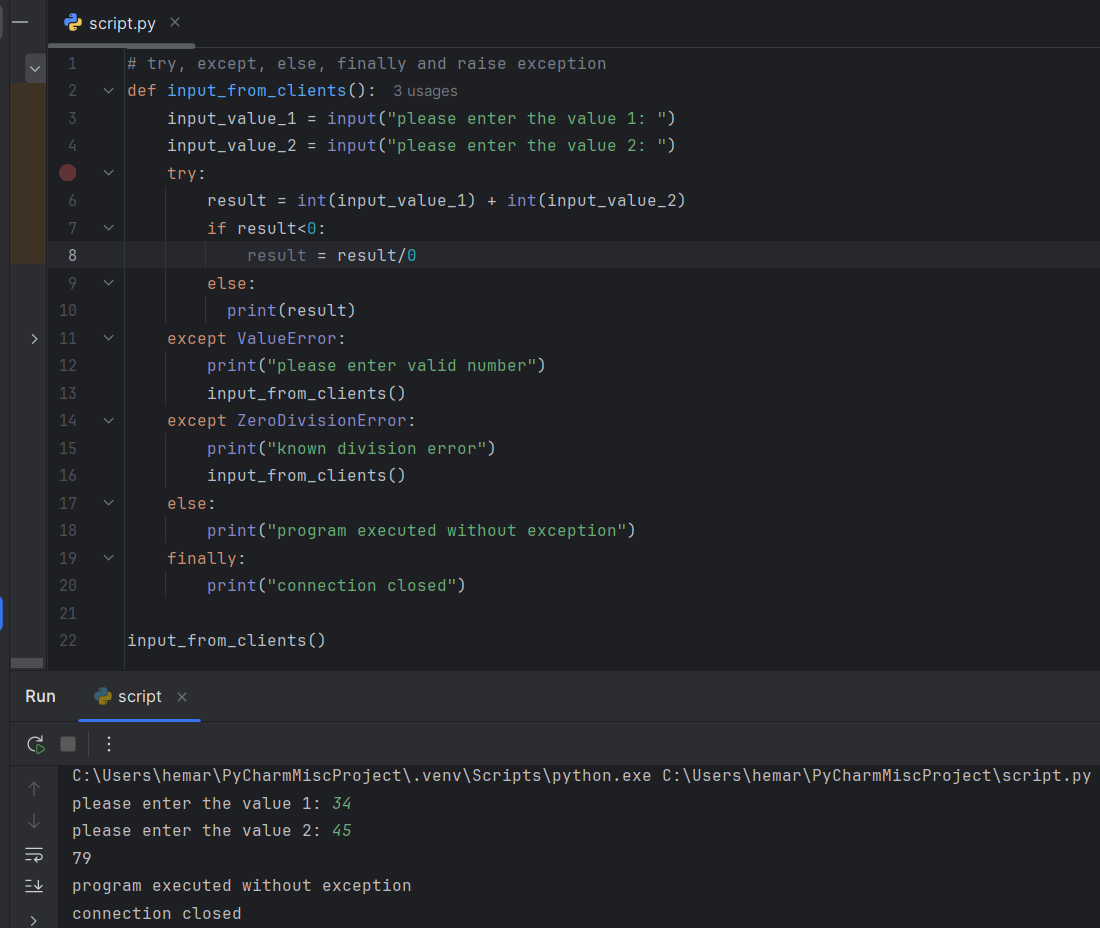


Method signature with ***\*\*variable in the method*** which means that ***value be takes as the dict***. While invoking the method you need to ***pass the value*** In the form of ***key and value pair*** like parameter with argument in to it.



***Try , except, Finally and raise***

In the ***try block*** execute ***your program logic*** and if any ***exception occurs*** it will be ***capture in the except block*** and ***finally*** will be execute ***no matter exception occurred or logic worked*** .



***Raise means throwing a custom exception :***

