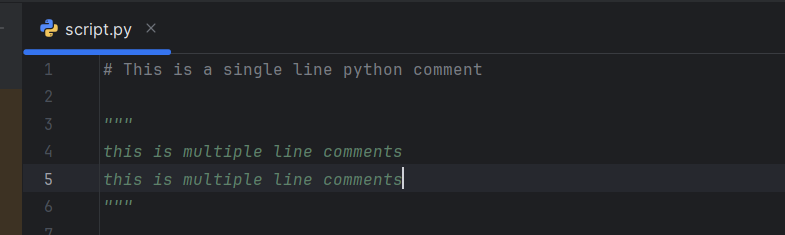
Core Python

***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

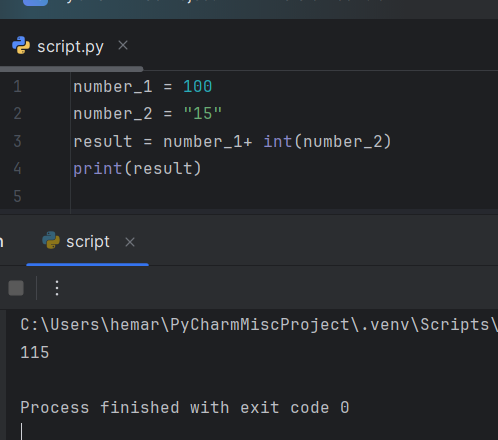
***salary: float*** = 9000.23

***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))

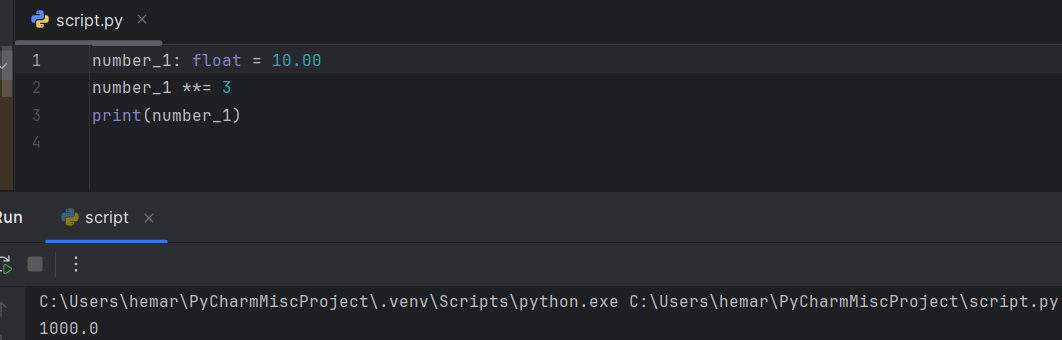
***and , or , not*** operator also used in python

***==*** 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:

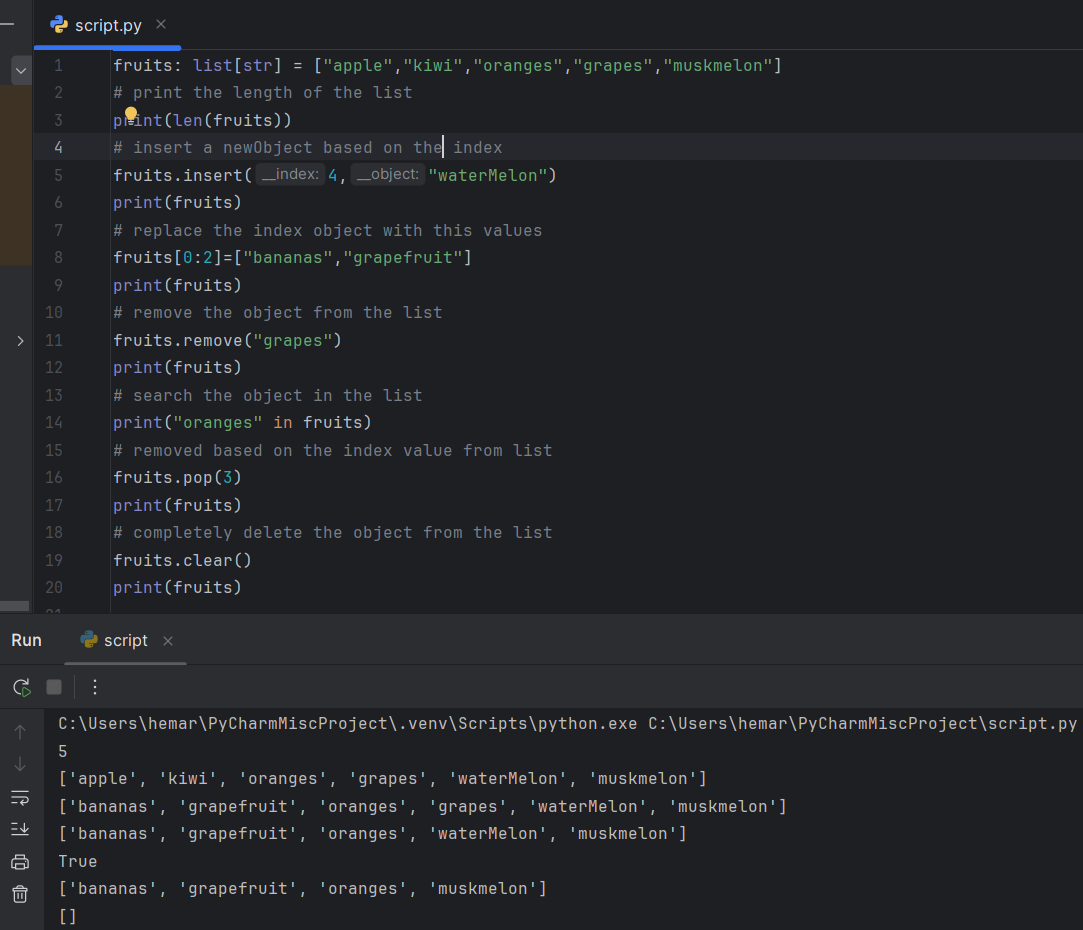
--> In this we can multiple dataTypes without any issue

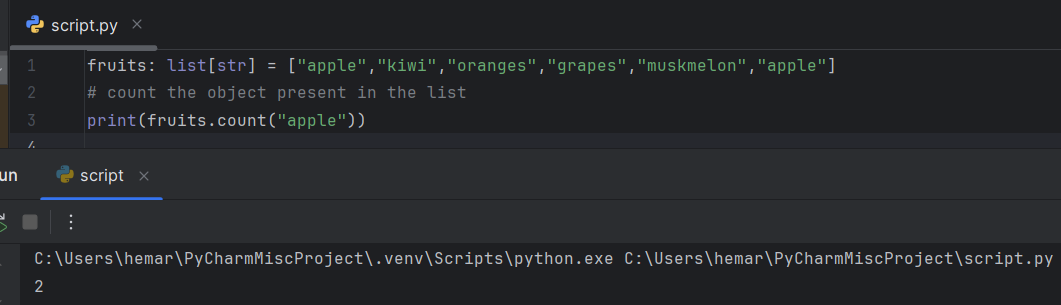
--> Even we can restrict the list to store only one DataType

--> 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:***

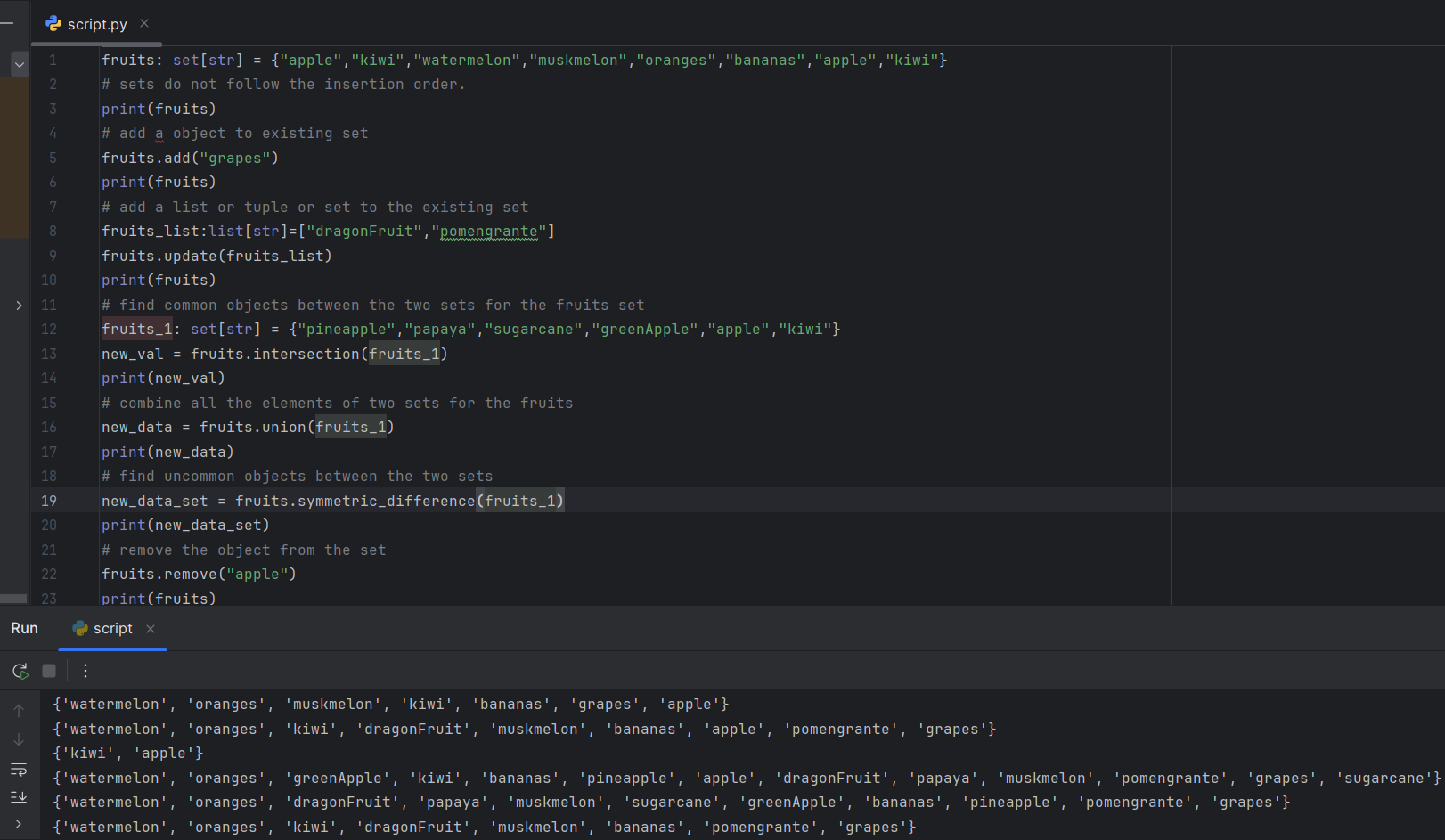
--> It does not follow any insertion order

--> 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}***



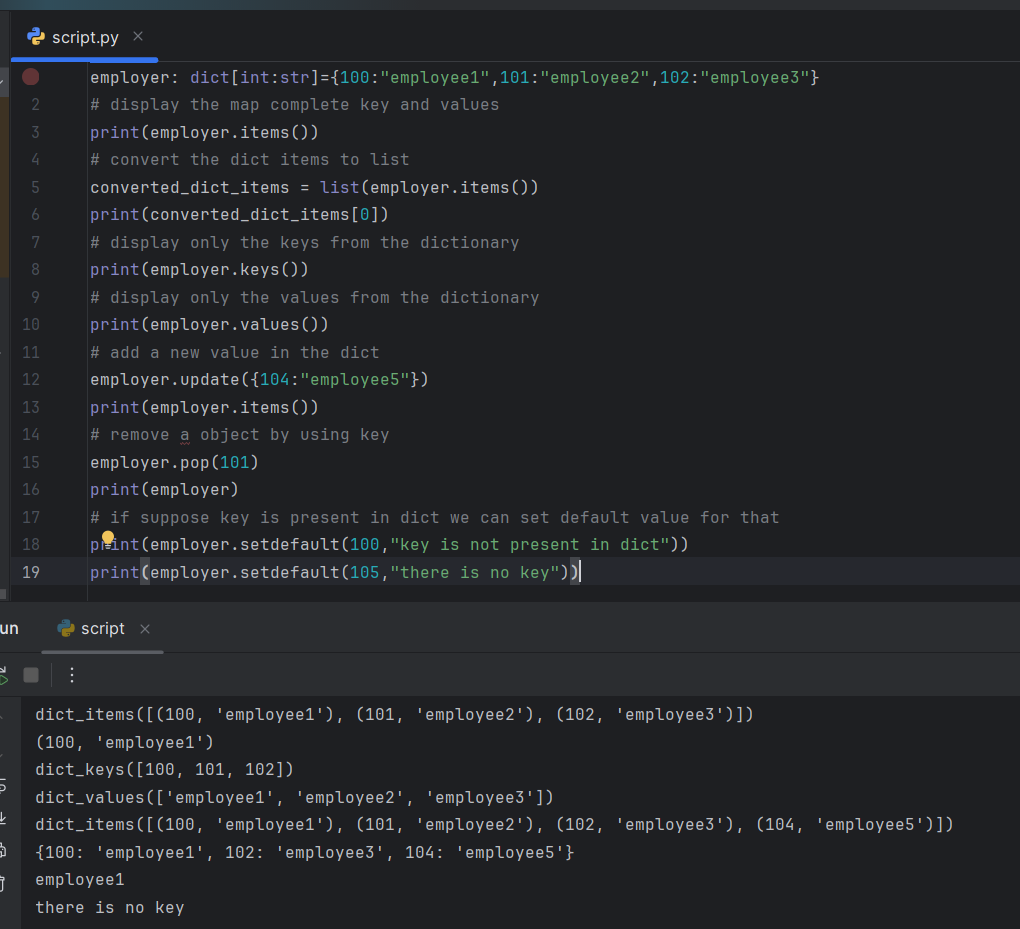
***Dictionary:***

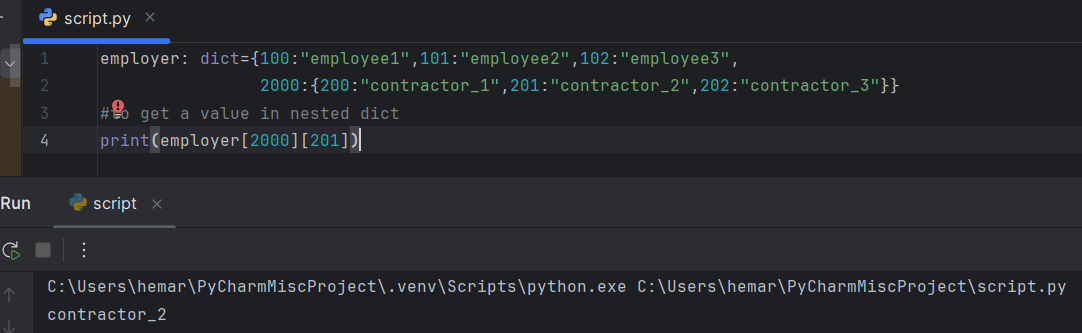
--> it stores the value in the form of key and value.

Syntax:

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

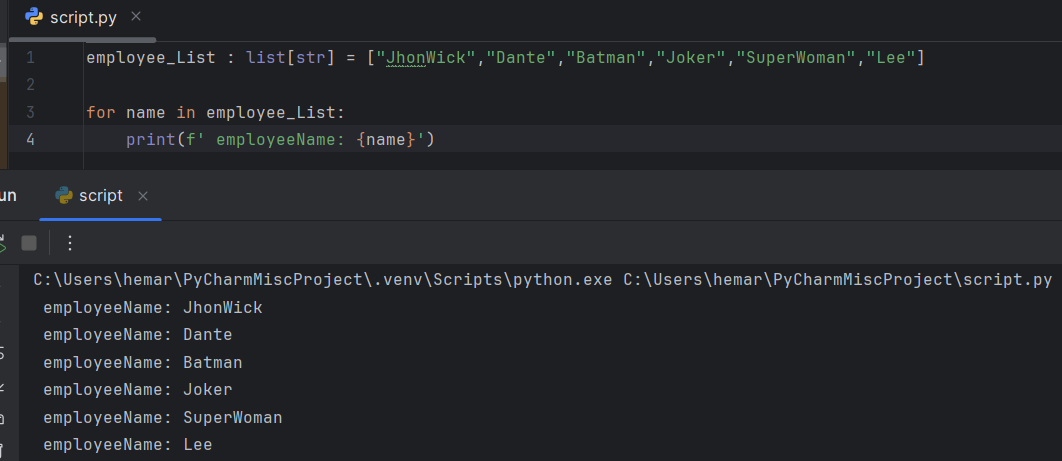
***Variable = {}***

******

******

***Control Flows Python:***

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



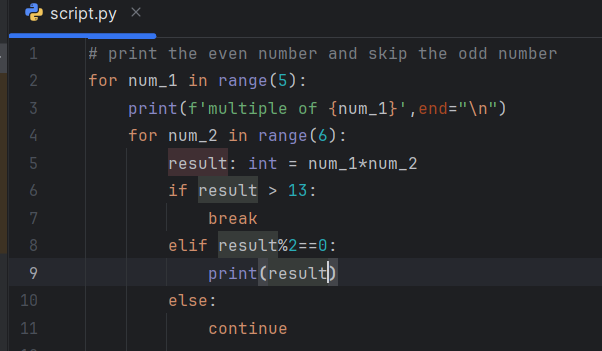
***For nested loop example:***



***While loop example:***

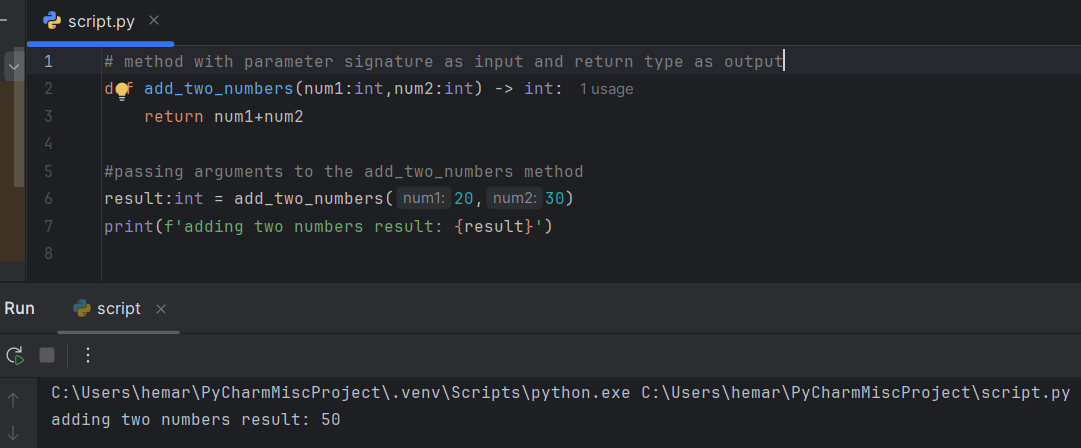


***Break and Continue:***

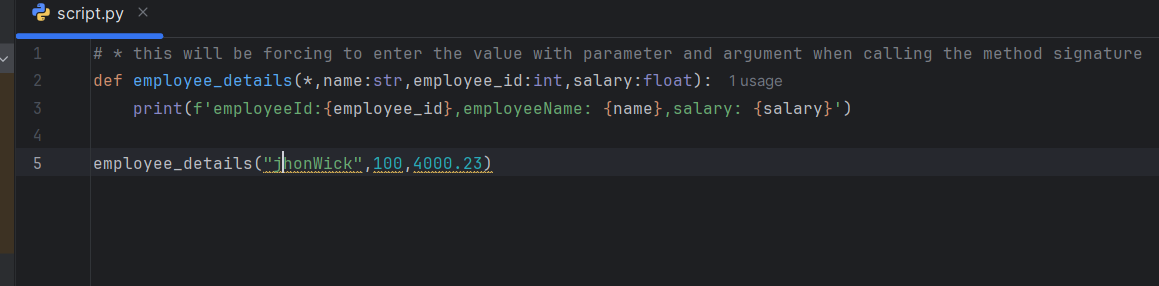


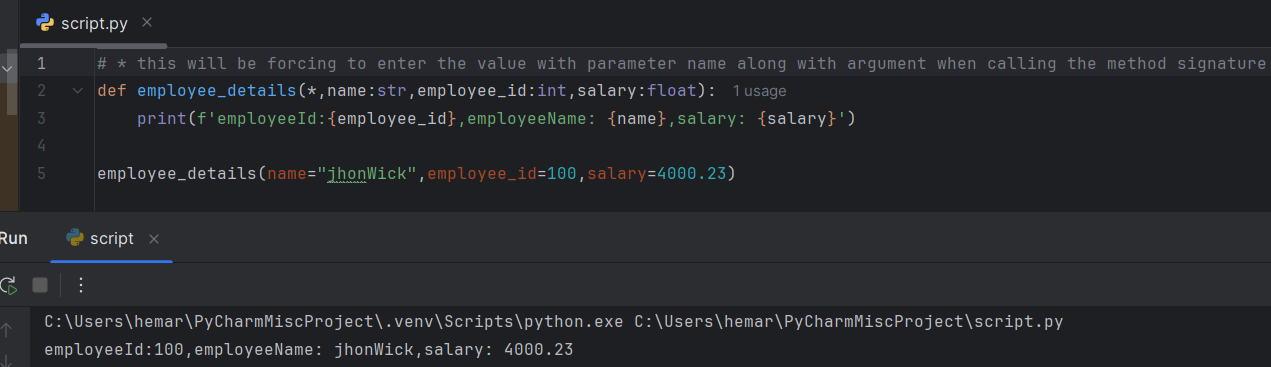
***Python Method Signature:***

*Method signature with the return type:*

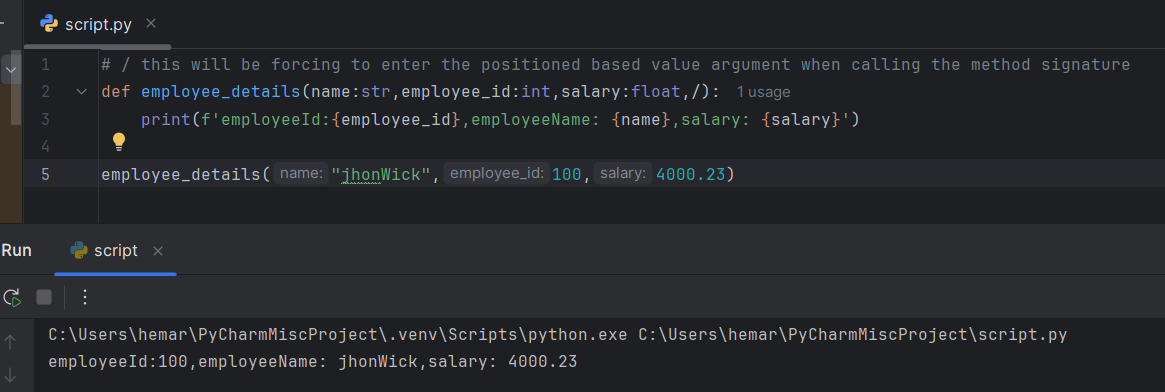


method signature which as ***\* in front of the*** ***method*** which means when you invoke the method you need to ***pass the parameter name along with arguments(values)*** if you don’t pass the parameter name method will throw error with yellow line

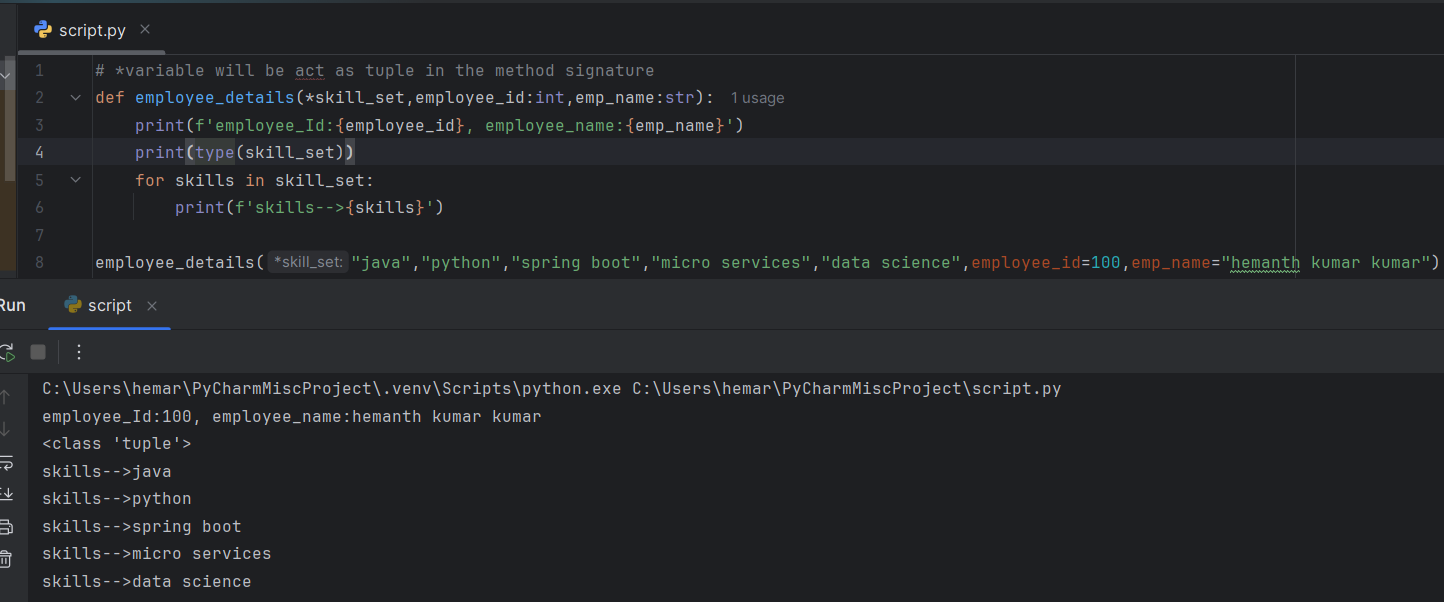




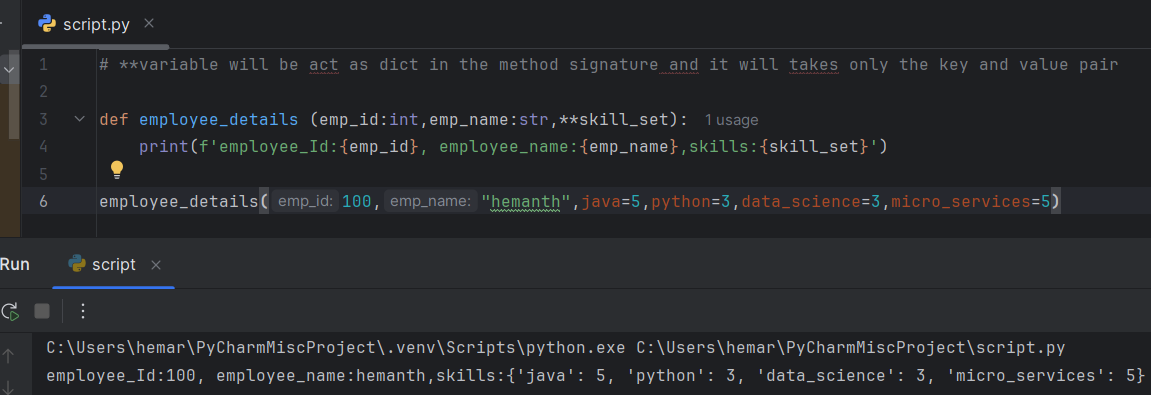
Method signature with ***/ at the end of the method which*** means you invoke the method arguments ***needs to pass in the positional value.***

******

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.

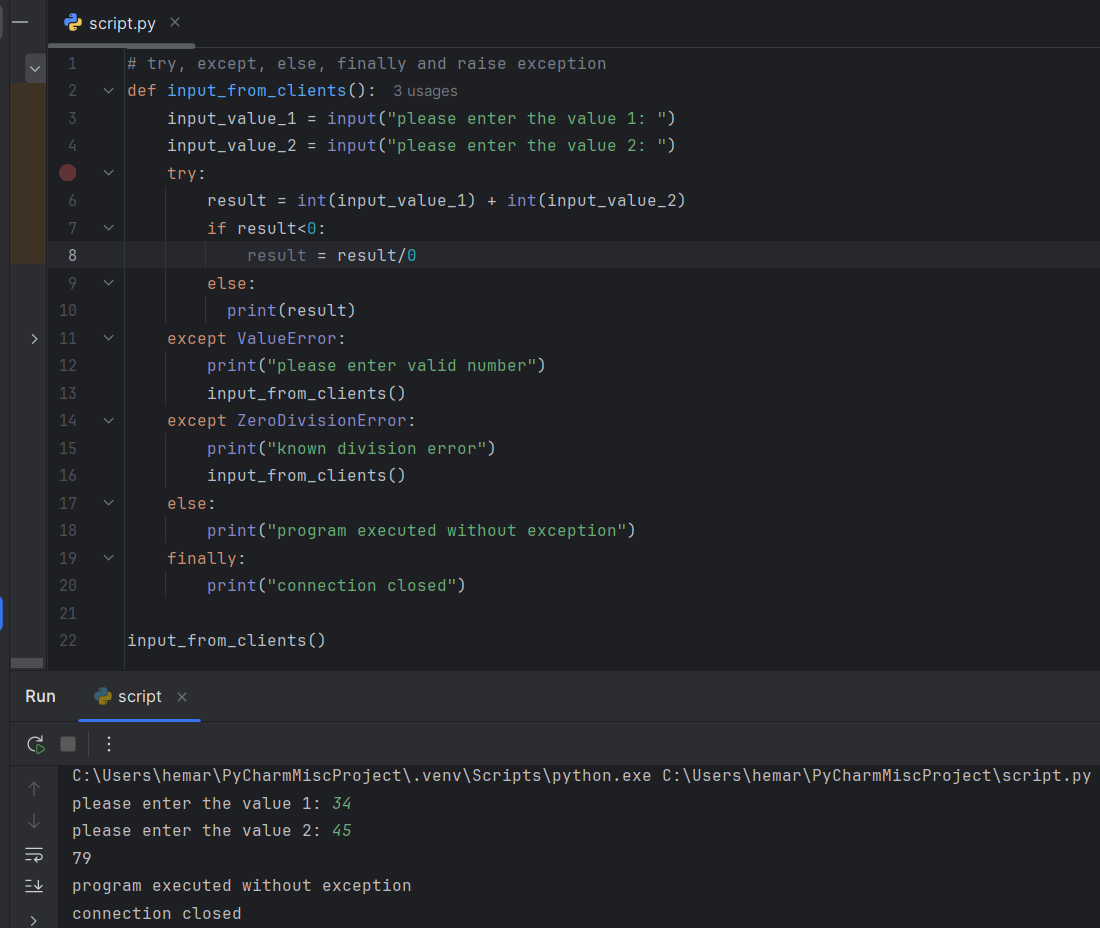


Method signature with ***\*\*variable in the method*** which means that ***value be takes as the dict***. While invoking the method you need to ***pass the value*** In the form of ***key and value pair*** like parameter with argument in to it.

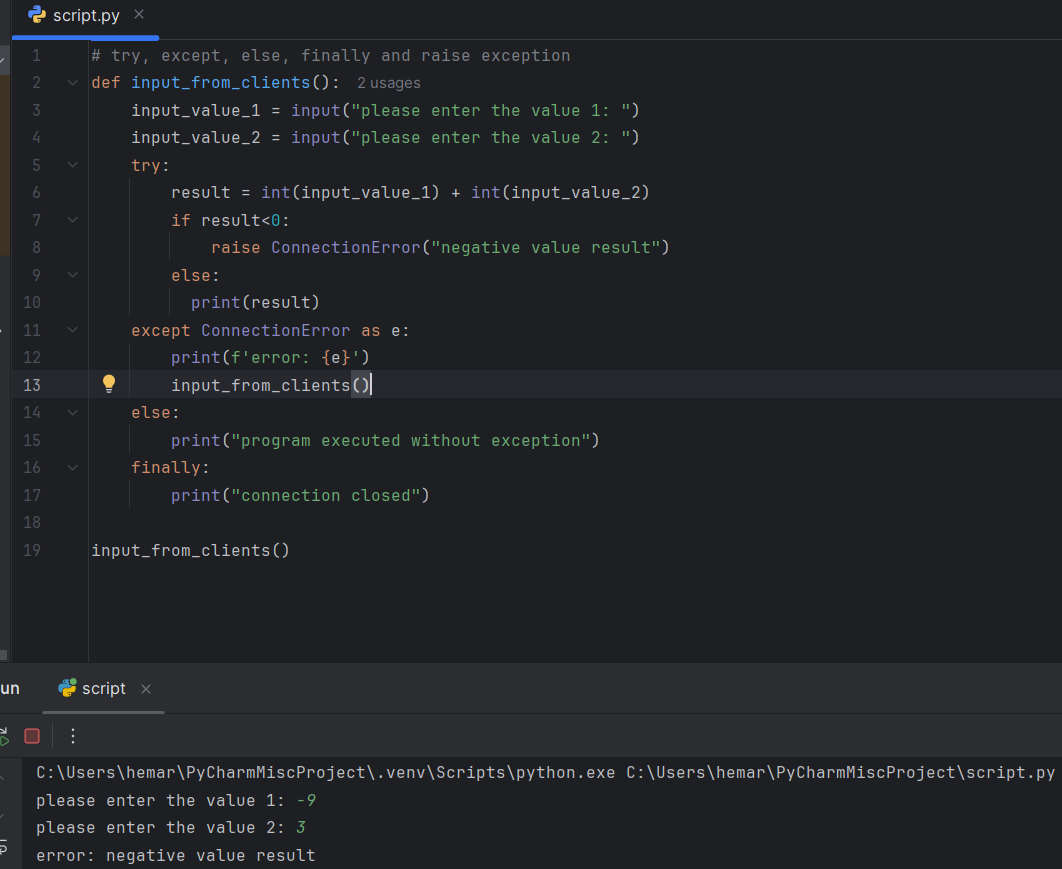


***Try , except, Finally and raise***

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 :***



***How to Import module and packages in python:***

***Import a module (file)***

\*) *import module\_name as alias\_name*

***module\_name*** is said to be the ***file name*** once you declared it you can able to access the methods or functions that is available in the module (file)

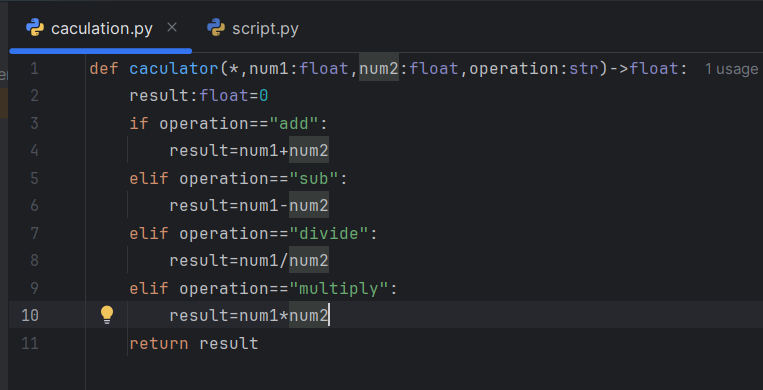
***Import a package from that import module:***

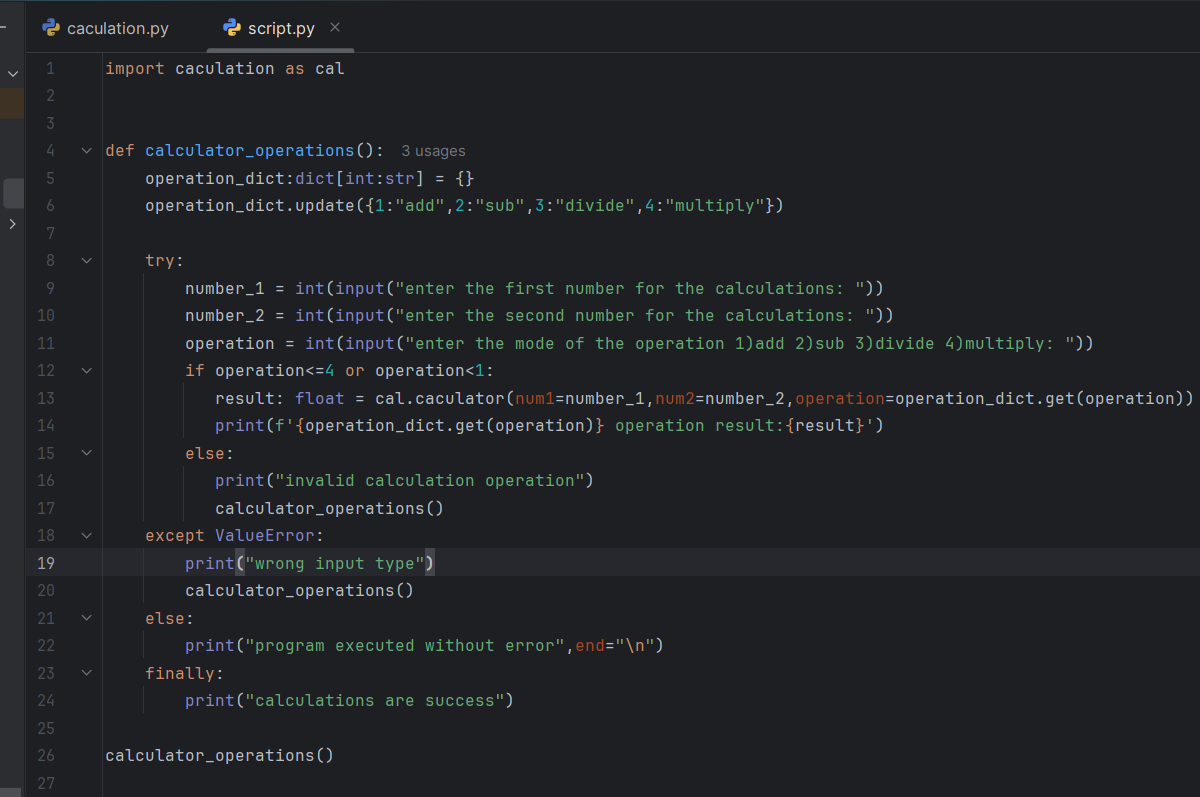
\*) *from package\_name import module\_name\_1 as alias\_name, module\_name\_2 as alias\_name*

\*) *import package\_name.module\_name\_1 as alias\_name, package\_name.module\_name\_2 as alias\_name*

Package as multiple modules (files)

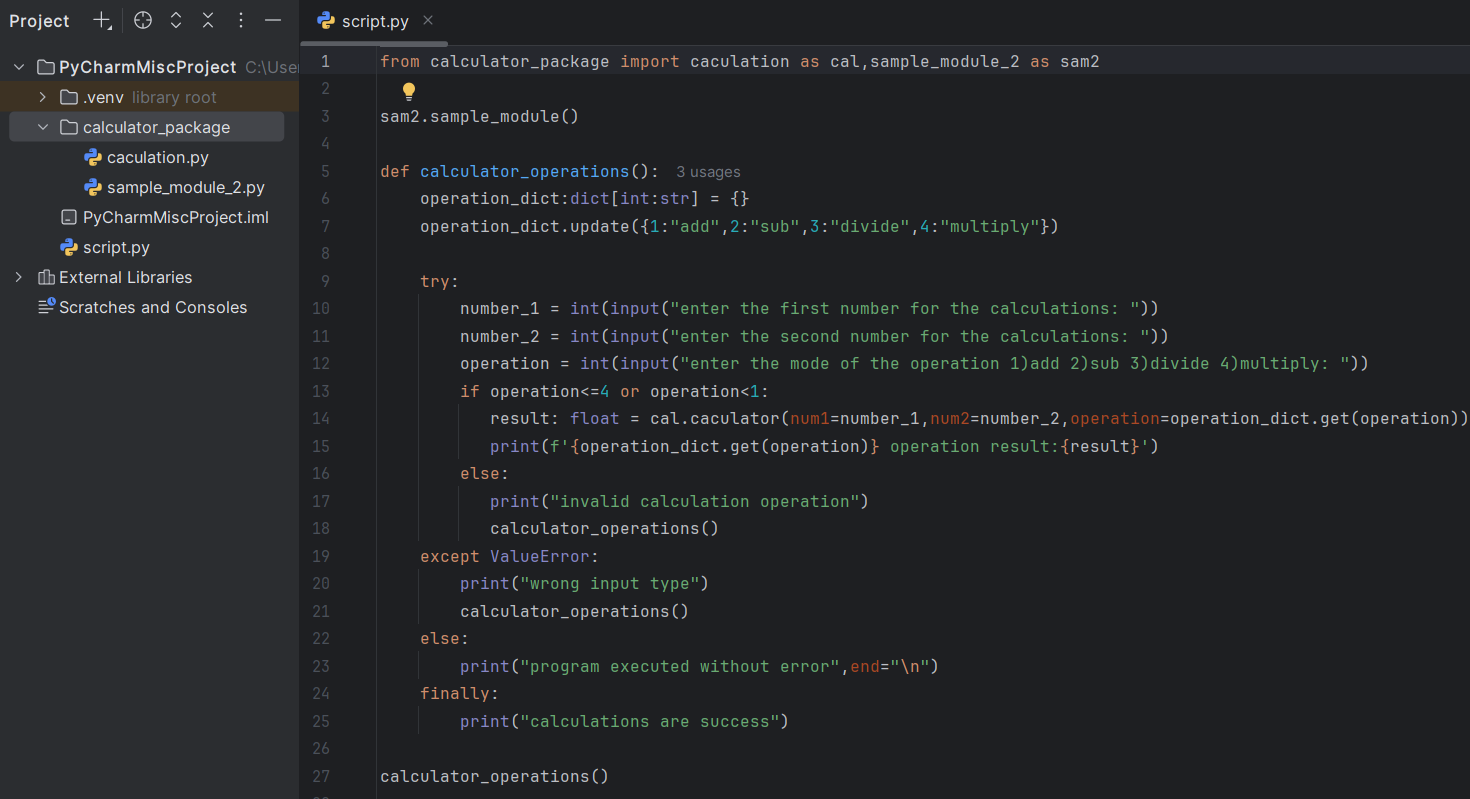
***Import as module example:***





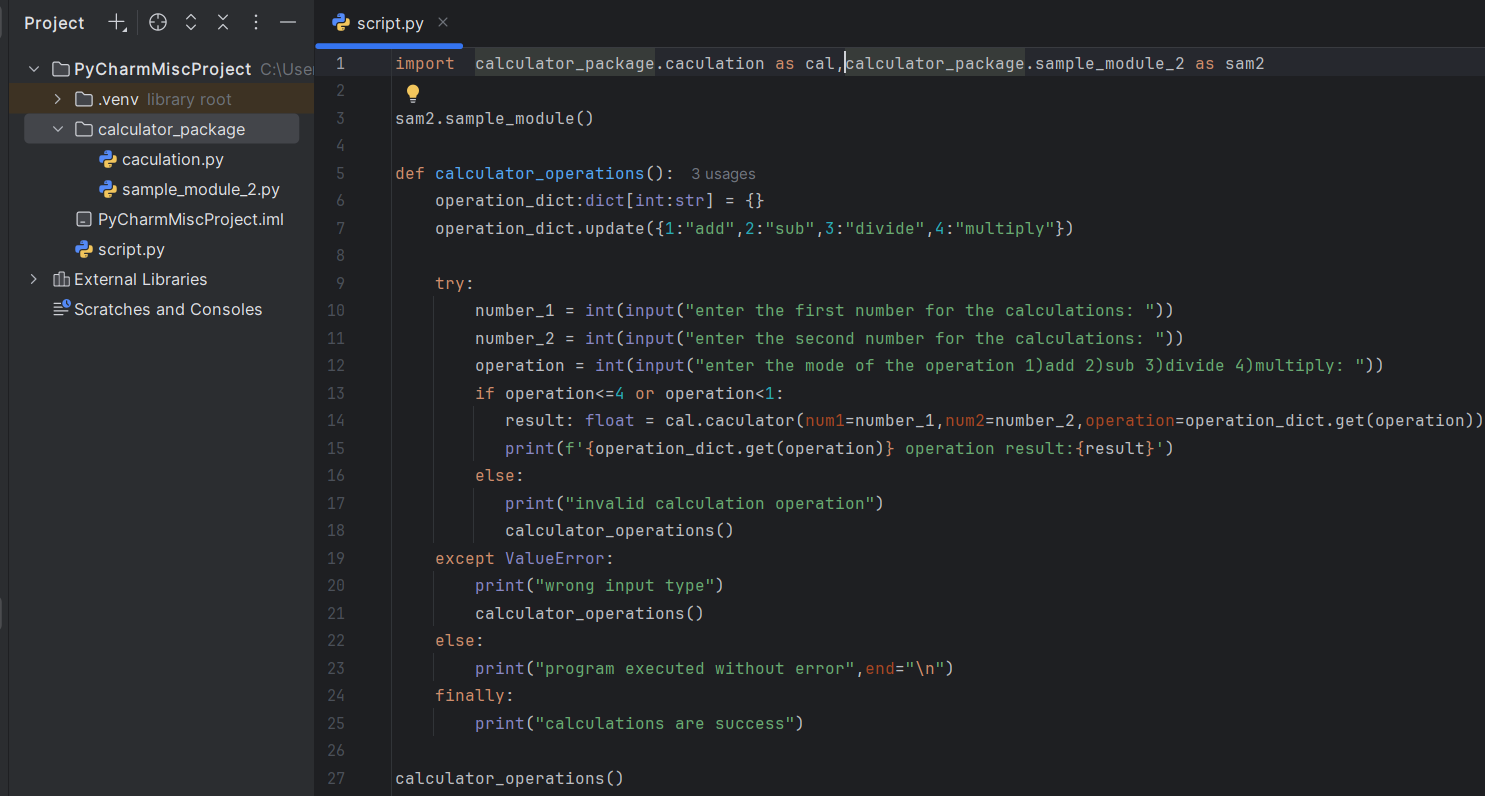
***Import package example in python:***

*from calculator\_package import caculation as cal,sample\_module\_2 as sam2*

******

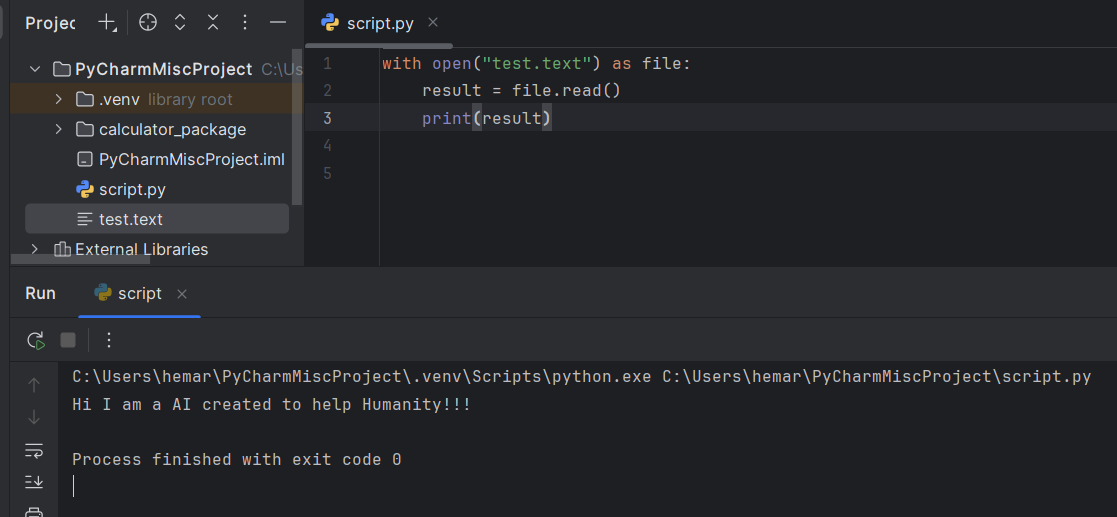
***Import package example-2 in python:***

*import calculator\_package.caculation as cal,calculator\_package.sample\_module\_2 as sam2*

**

***Usage Of With KeyWord in python programming language:***

***With*** keyword mainly used in the file system where it can automatically close the file we do not need to manually call the ***file.close()*** method

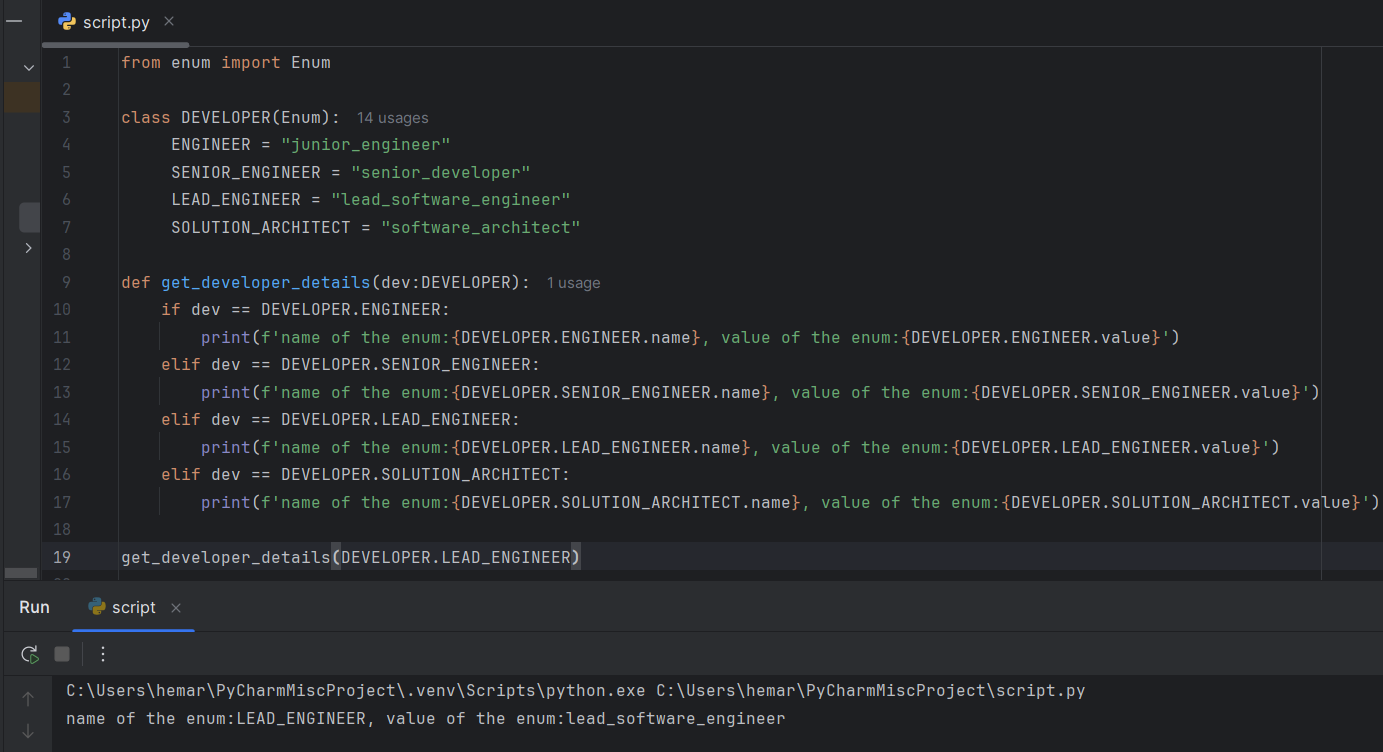


***How to decalre a enum and use the enum in python?***

***Enum*** is ***kind of constant*** through out the ***program*** . Enum as two methods one is name() and other is value(). Name() provides the exact of the Enum in string value and Value() provides the enum value what we declared in the progam.

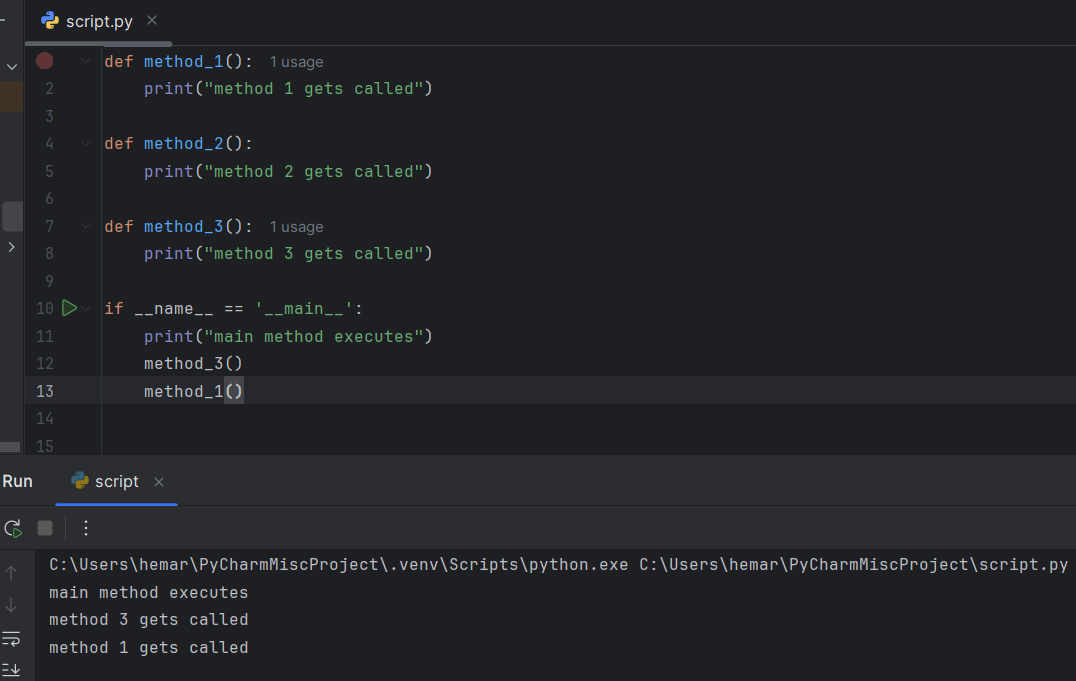
Step 1 --> from enum import Enum ***(from enum pacakage import Enum module which means file)***

Step 2 --> Declare a class and pass the Enum in the method signature



***What is the purpose of using the main name check in the python?***

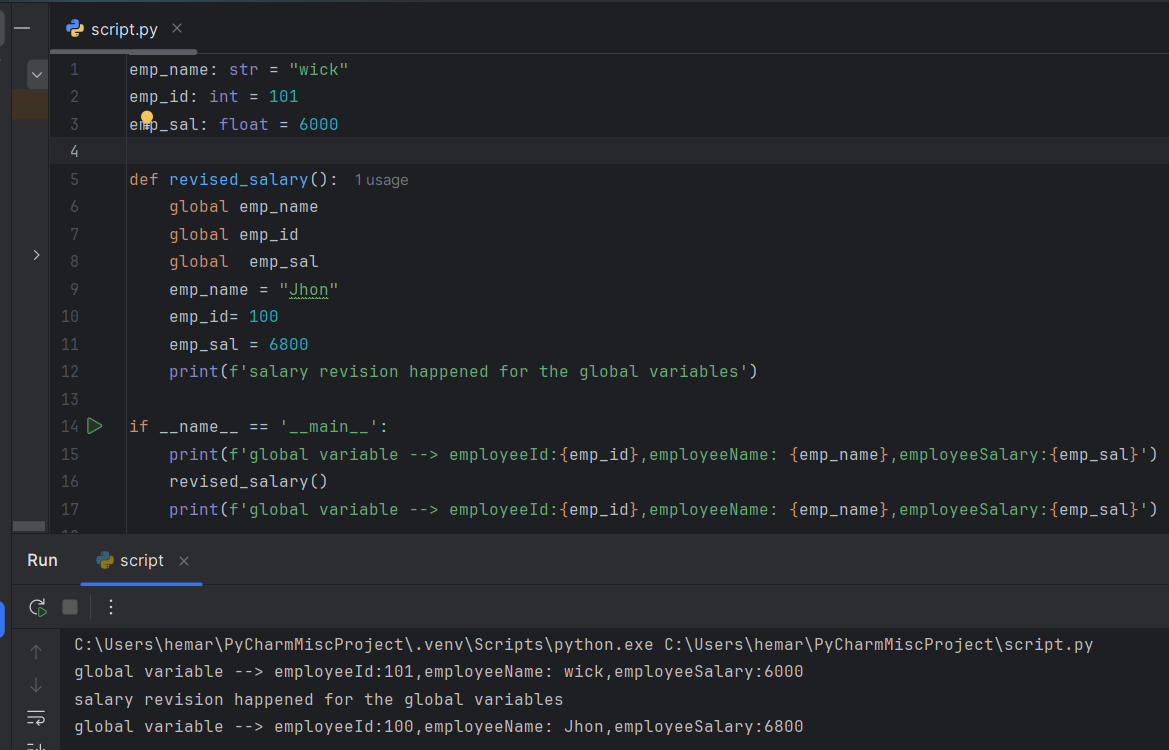
Once you implemented a main check in the python whatever method is invoked inside the main method that method only be called not all the methods or imports(from other modules or packages) used inside the module (file)



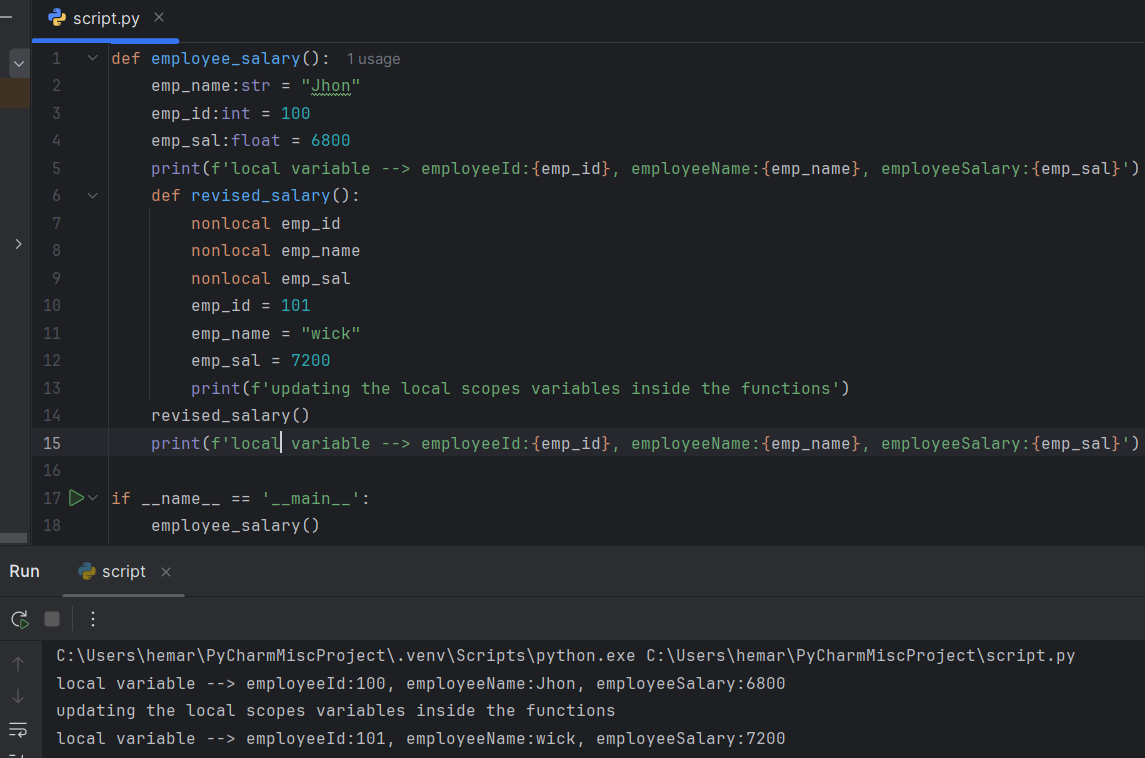
***What are the scopes in the Python?***

***#***Remember there are two scopes in python ***global scopes*** and ***nonlocal scopes:***

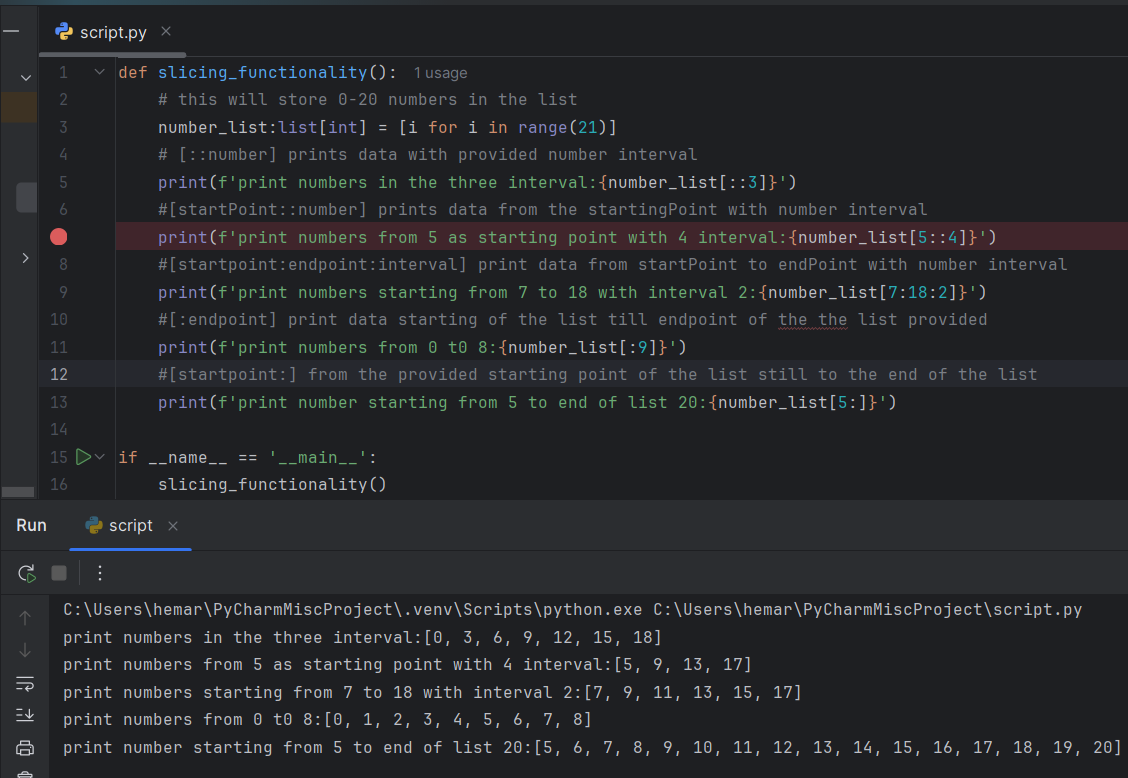
inside the ***functions the scope is always considered to be local*** but we can override the ***global variables values inside the functions*** by providing ***global before the variable***.



In the ***Python functions variable will be acting as local scope*** nested function inside the ***variable can replace this local scope by overriding the variable*** by mentioning

.

***List Operation methods:***



***# remember*** when ever you ***do a modify the list in a loop*** store the ***updated result in the another new list***.

***OOPS In Python Programming Language***

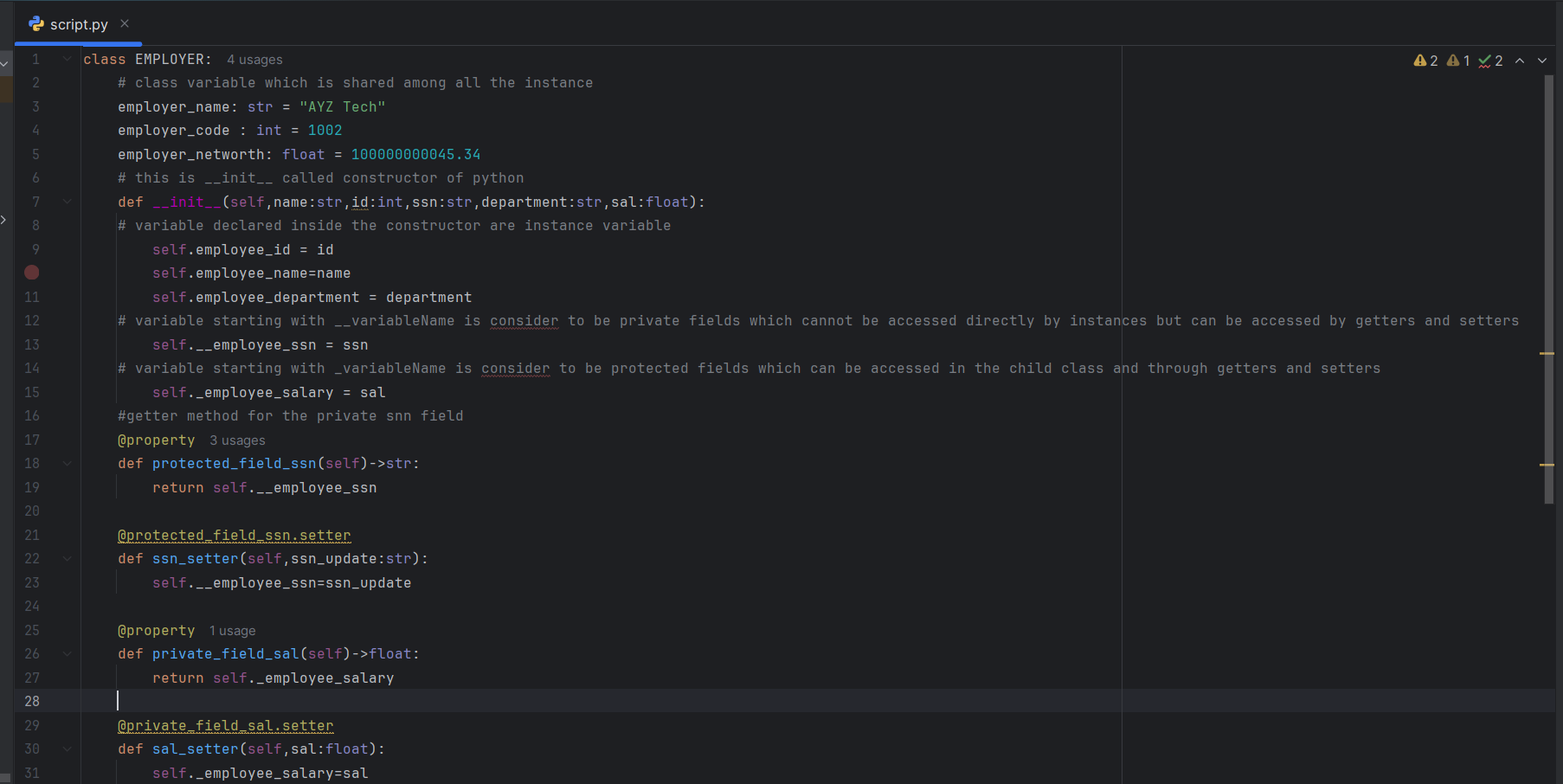
Variable declaration in python can be done in two ways.

1. One is declaring the ***class variables*** that is ***being shared across all the instances***.
2. Other is declaring the ***instance variable*** which ***will be unqiue for the each instance*** (this will be declared inside the constructor (\_\_init\_\_ method))

\* ***\_variableName*** this is will be consider a***s protected field*** which can be used in the ***current class and also the child class***. ***Outside the current class*** you can access it by ***setters and getters*** method

\* ***\_\_variableName*** this will be consider as ***private field*** which can be used in the ***current class*** only .Outside the current class you can access it by setters and getters method

***\_\_init\_\_ method*** is the constructor and ***self*** will be denoting the current instance simliar to ***this in java.***

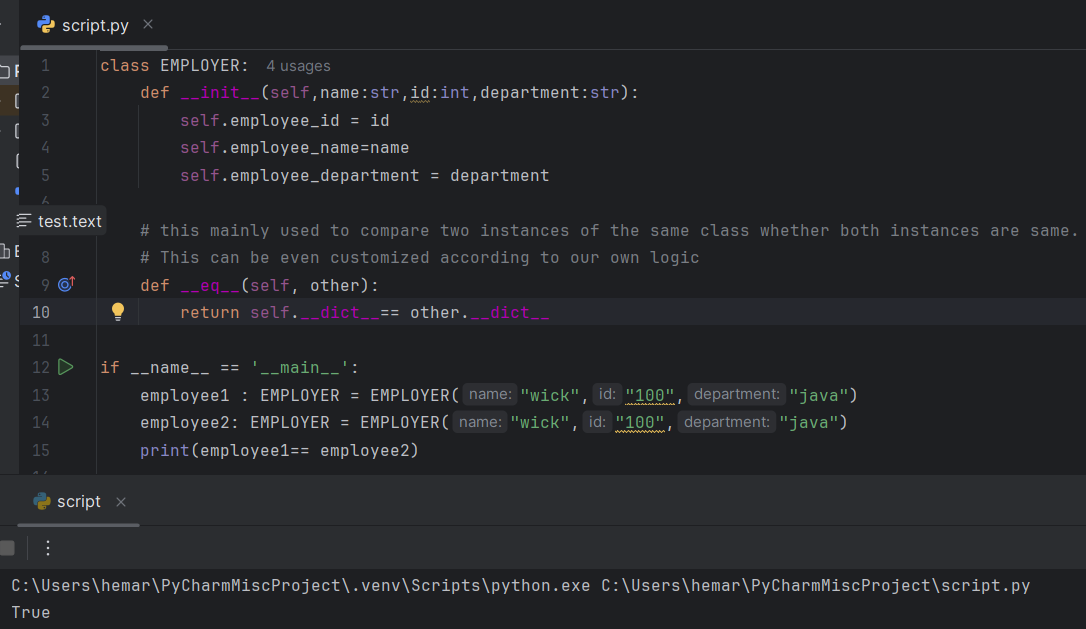
******

******

***How to Compare the instances of the same class?***

***def \_\_eq\_\_ (self,other)***

***# your own custom logic to compare the instances***

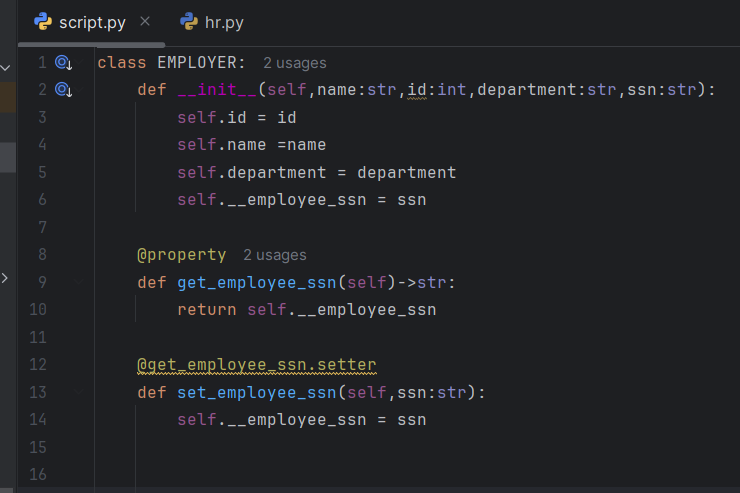


***How to Implement inheritance for the python?***

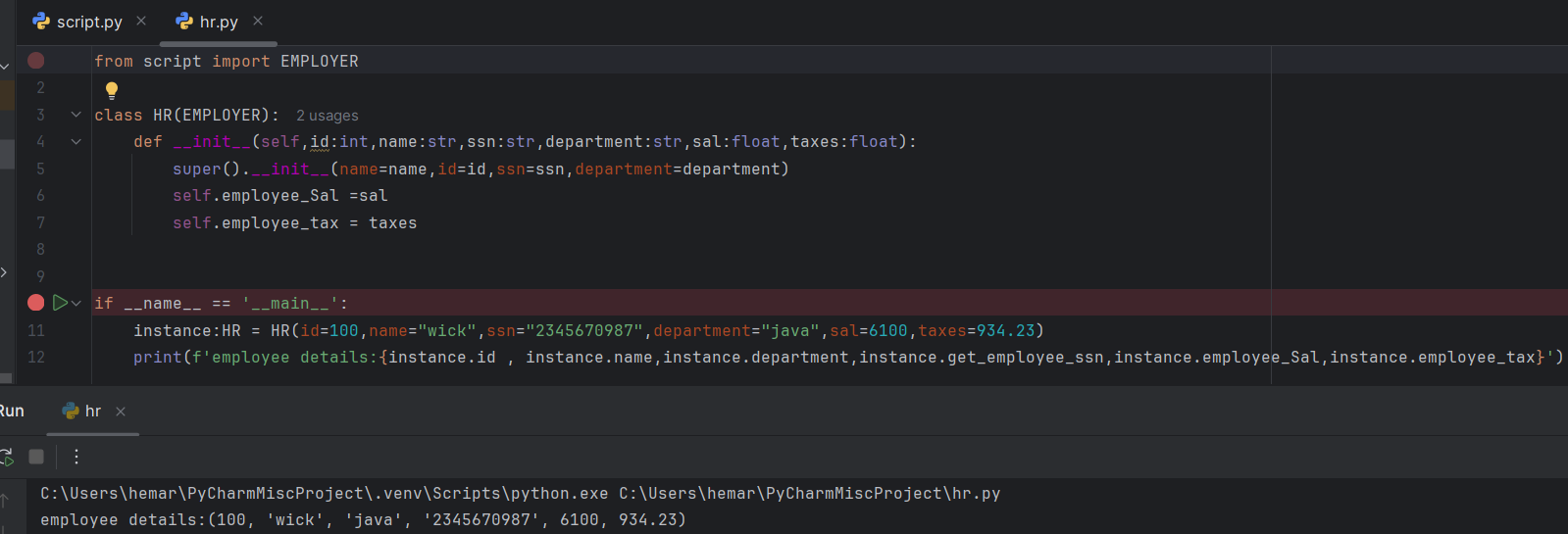
In the ***child class inside*** the init we need to initailize the ***parent constructor***

***Super().init(\_,\_,\_,…..)*** super refers to the parent class and super().init(\_,\_,\_…..) refers to the constructor of the parent class.

***Parent class:***

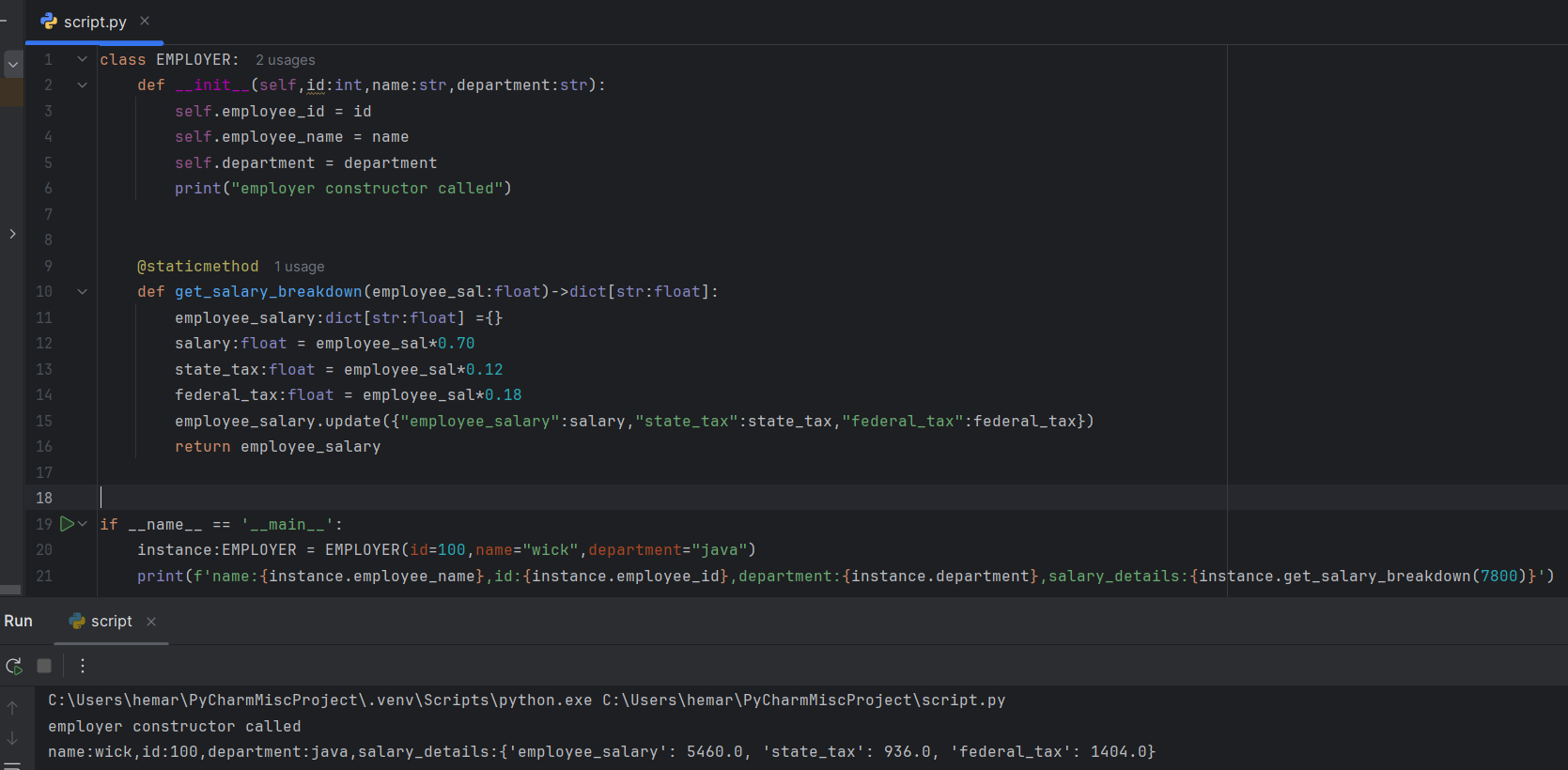


***Child class:***



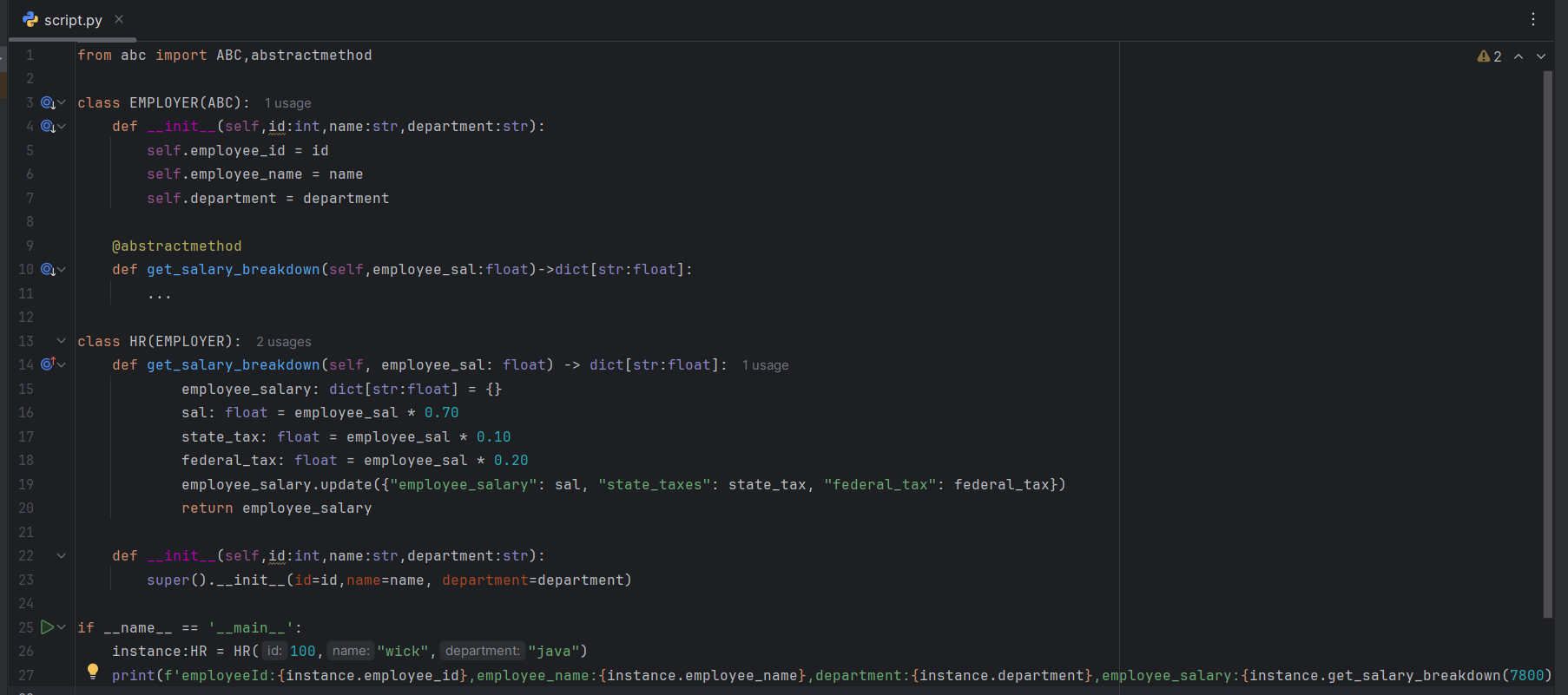
By ***inheriting the employer parent class in the HR child class*** we can able to see through the ***HR instance*** we can able to ***access the employer parent class methods and variables***.

***How to Implement static method?***



Even without the instance creation you can access the static method by using the format ***EMPLOYER.get\_salary\_breakdown(7800) (classname.staticmethod())***

***How To implement abstract method implementation?***



Step 1: import the respective pacakage details

***from abc import ABC,abstractmethod***

***Step 2:*** Make sure to the ***parent class which has the abstract method*** needs to ***inherit the ABC class.***

***Step3:*** Make sure the ***child class inherits the parent class which as the abstract method*** and implement the abstract method with logic.

***Python Built-in Function:***

***min()*** - compare the two values and retruns the minimum value

***max()*** - compare the two values and retruns the max value among them

***round(value,decimal)*** - rounding off the number

***all() -*** If all the conditions is satisfied it will return true otherwise it will false

***any()*** - If any conditions is satisfied it will return true

***sort(***) - This will sort based on the numbers , strings and class

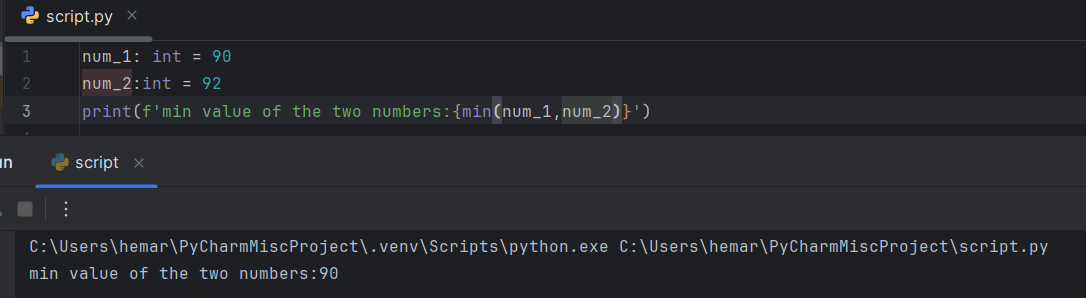
***filter()*** - This will filter the based on our custom logic

***map()*** - This will map the values based on our custom logic

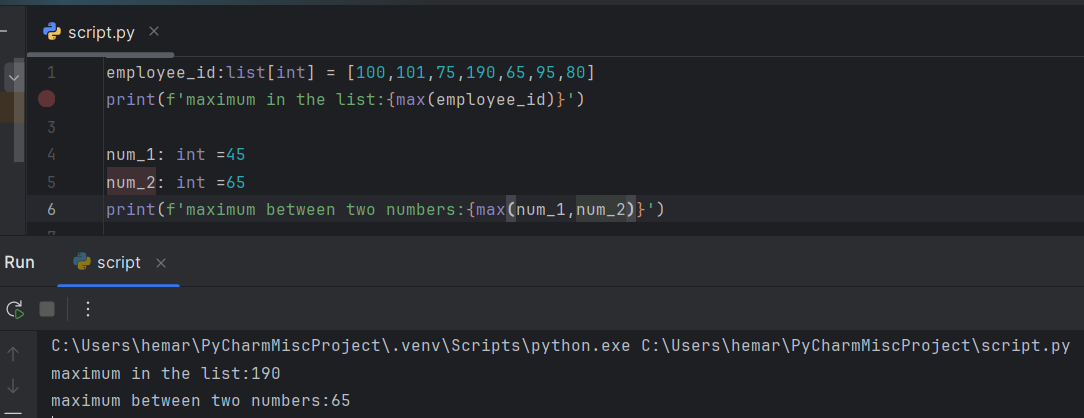
***sorted()*** - this will sort the class value based on the custom logic

***eval()*** - this will convert the mathematical statements

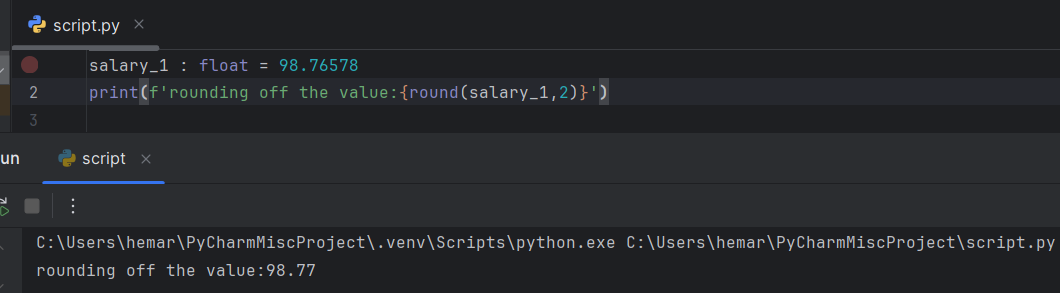
***Min:***

’

***Max:***

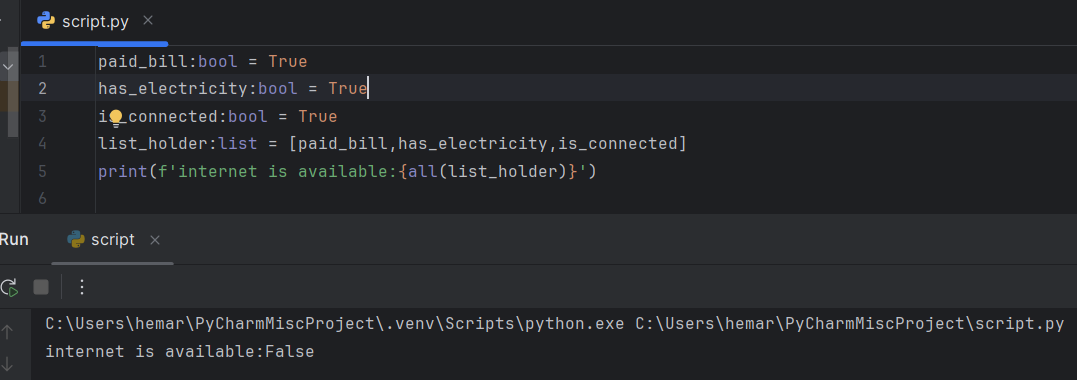


***Round():***

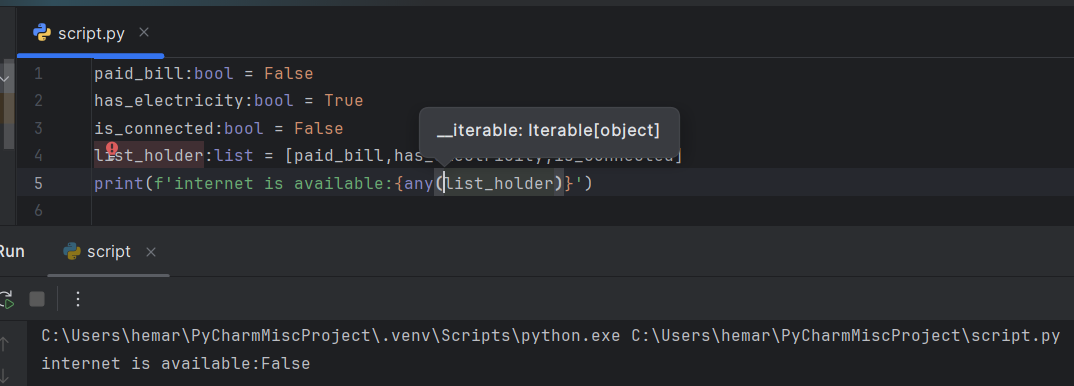
******

***All:***

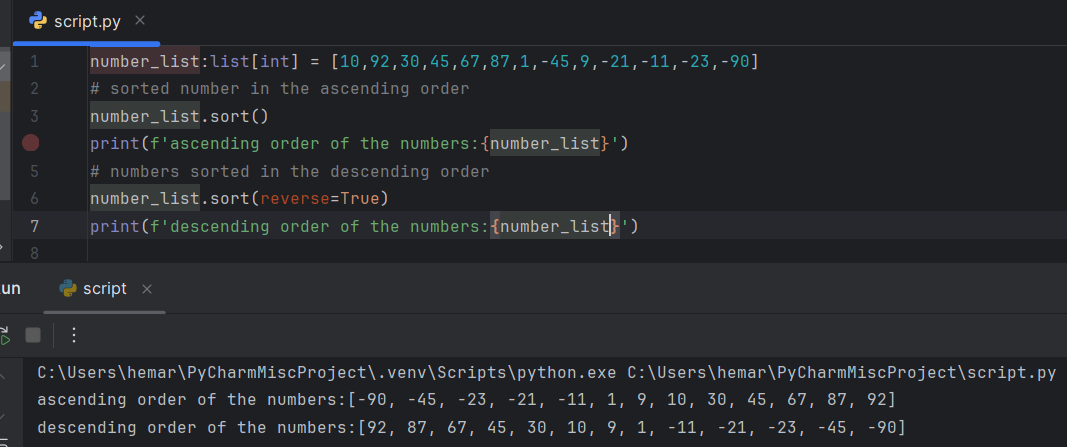
******

******

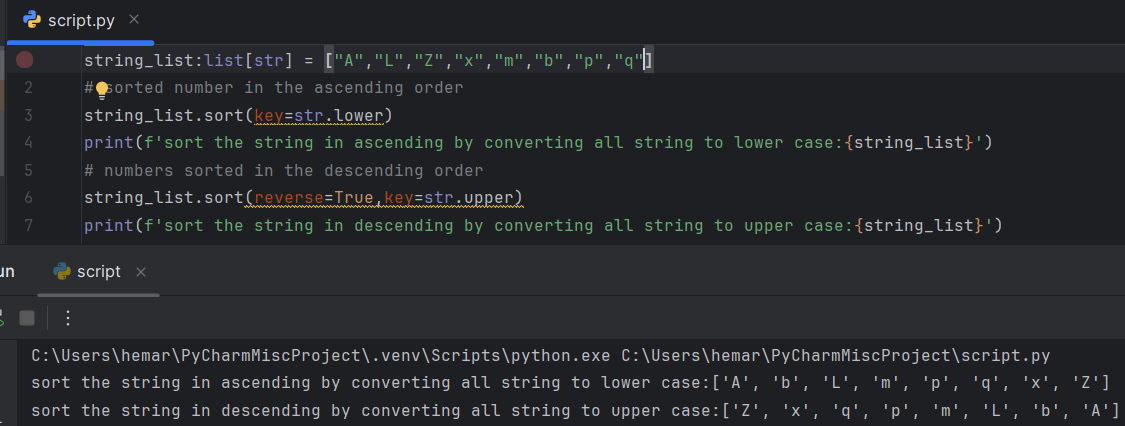
***Any:***

******

***Sort based on integer:***

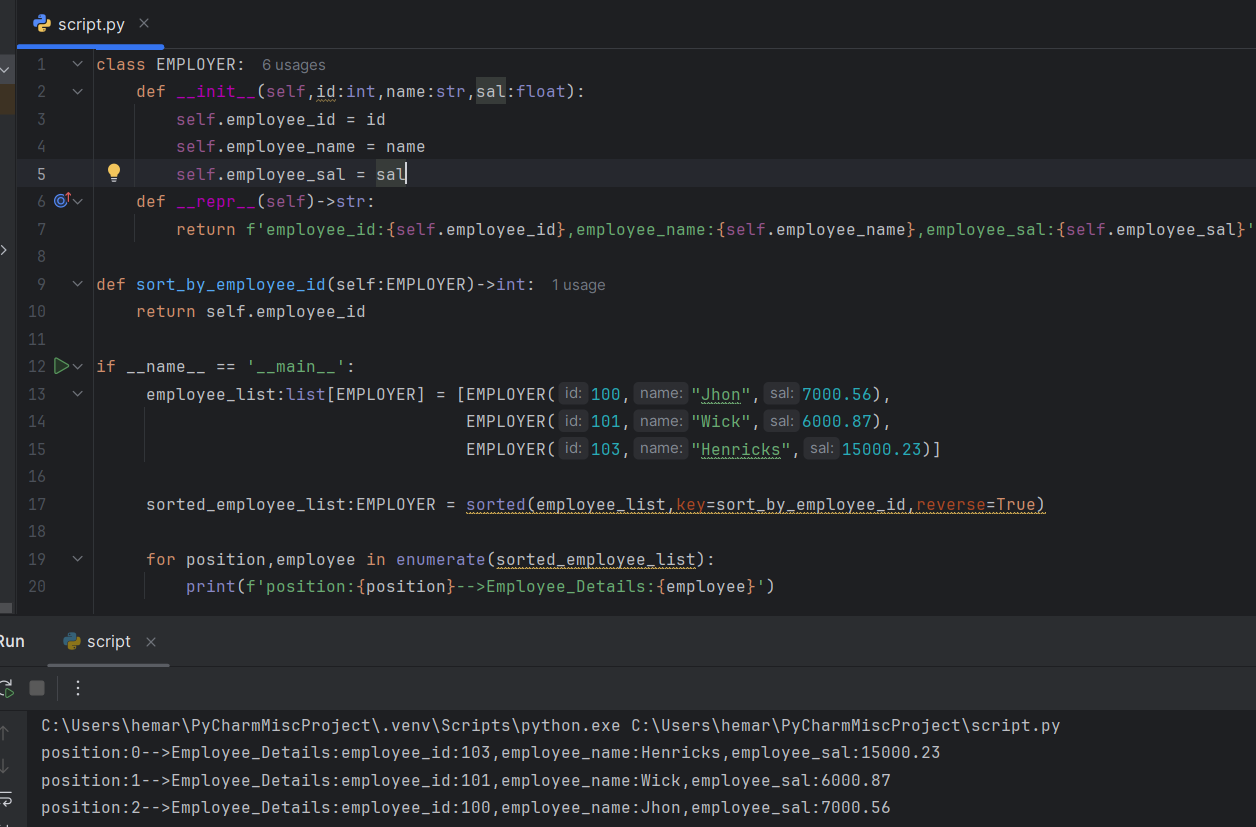
******

***Sort based on String:***

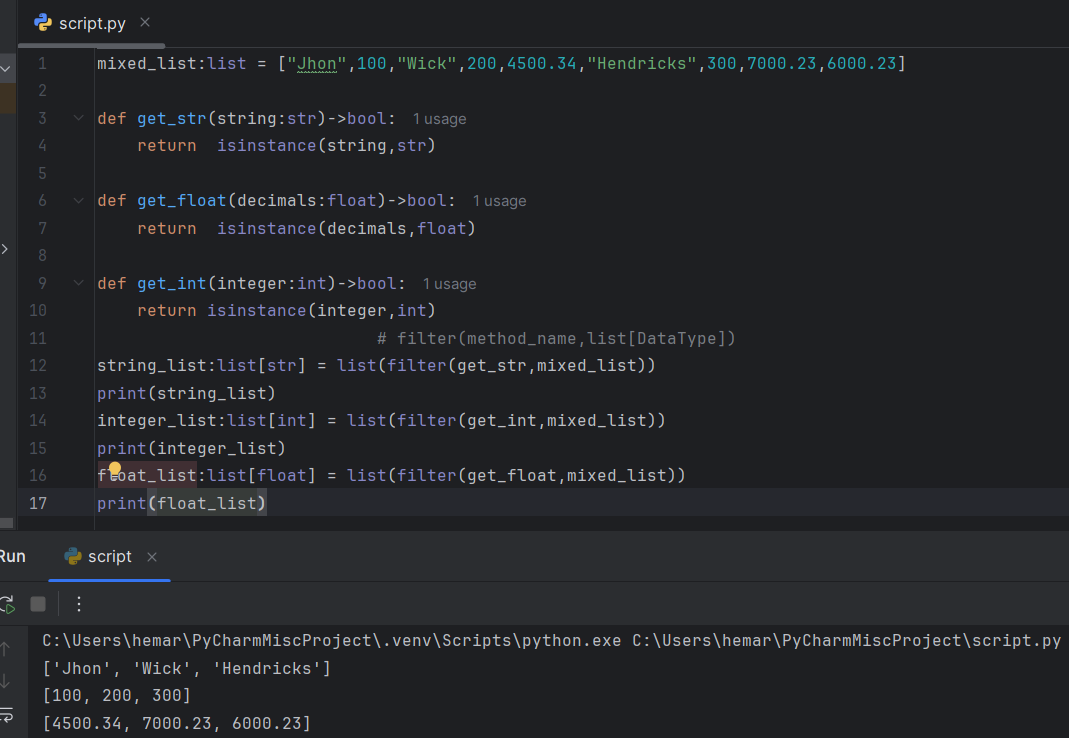
******

***Sort the class values based on the custom logic:***

***Sorted(list\_value,key=function\_name,reverse=true or false)***

******

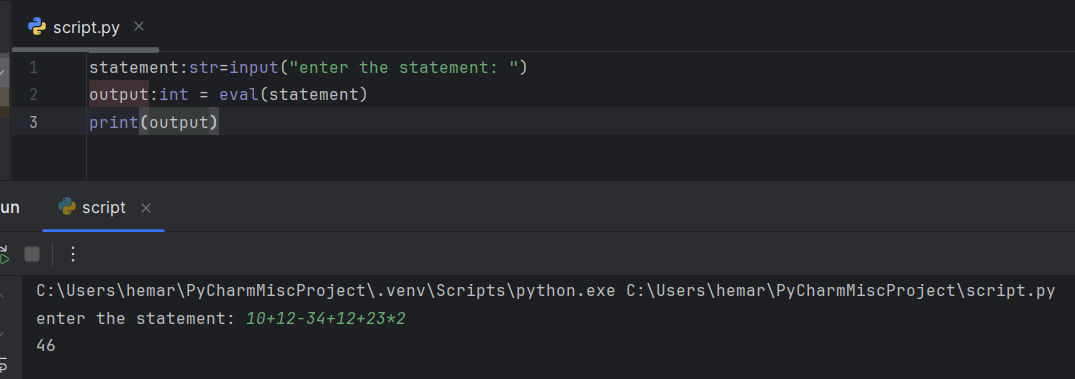
***Filter:***

******

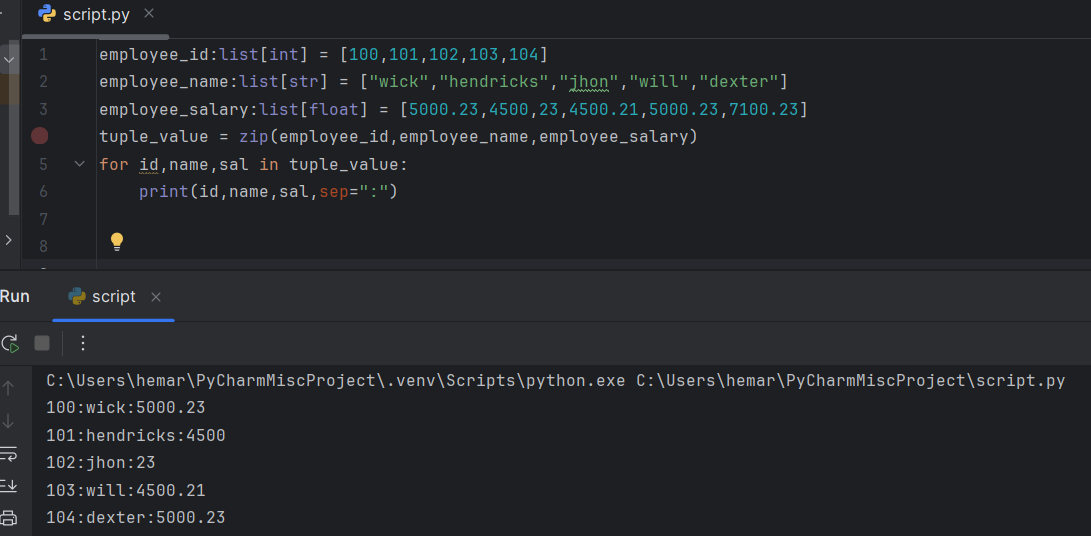
***Map:***



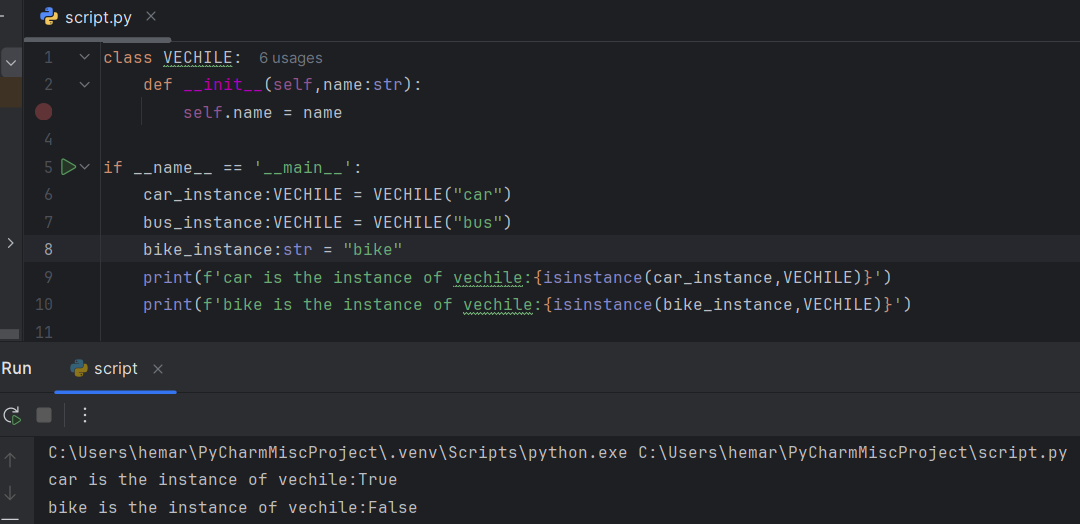
***eval():***

******

***Zip:***

******

***Instance:***

******