Tutorial 5 and 6 | SQLite Download Page | Home Page - Select or create a noteb | Tutorial5&6 - Jupyter Notebook | about:blank | Tutorial5&6

File | file:///D:/Downloads/Tutorial5&6.html

Google | C Compiler | Basic Workshop | Academia | Classes | Quiklrn | SRM KTR | Registration | admissions.srmist.e... | HackerRank | Online Hostel Roo... | Men's Hostel Floor... | Tkinter Course - Cr... | microsoft/PowerToy... | Other bookmarks

In [5]:
```
import re
vowels=['a','e','i','o','u','A','I','E','O','U'] #list of vowels
str = input("Enter a string:--")
slist=re.split("\s",str)
a,b=slist[0],slist[1] #extracting first and second words from given string
if (a[len(a)-1] in vowels and b[0]in vowels): #checking given condition
    print(str,"->  True")
else:
    print(str,"-> False")
```

```
Enter a stringI am SRM university
I am SRM university ->  True
```

In [6]:
```
vowels=['a','e','i','o','u','A','I','E','O','U']
str = input("Enter a string:--")
slist=str.split(" ")
a,b=slist[0],slist[1]
if (a[len(a)-1] in vowels and b[0]in vowels):
    print(str,"->  True")
else:
    print(str,"-> False")
```

```
Enter a string:--I am at SRM university
I am at SRM university ->  True
```

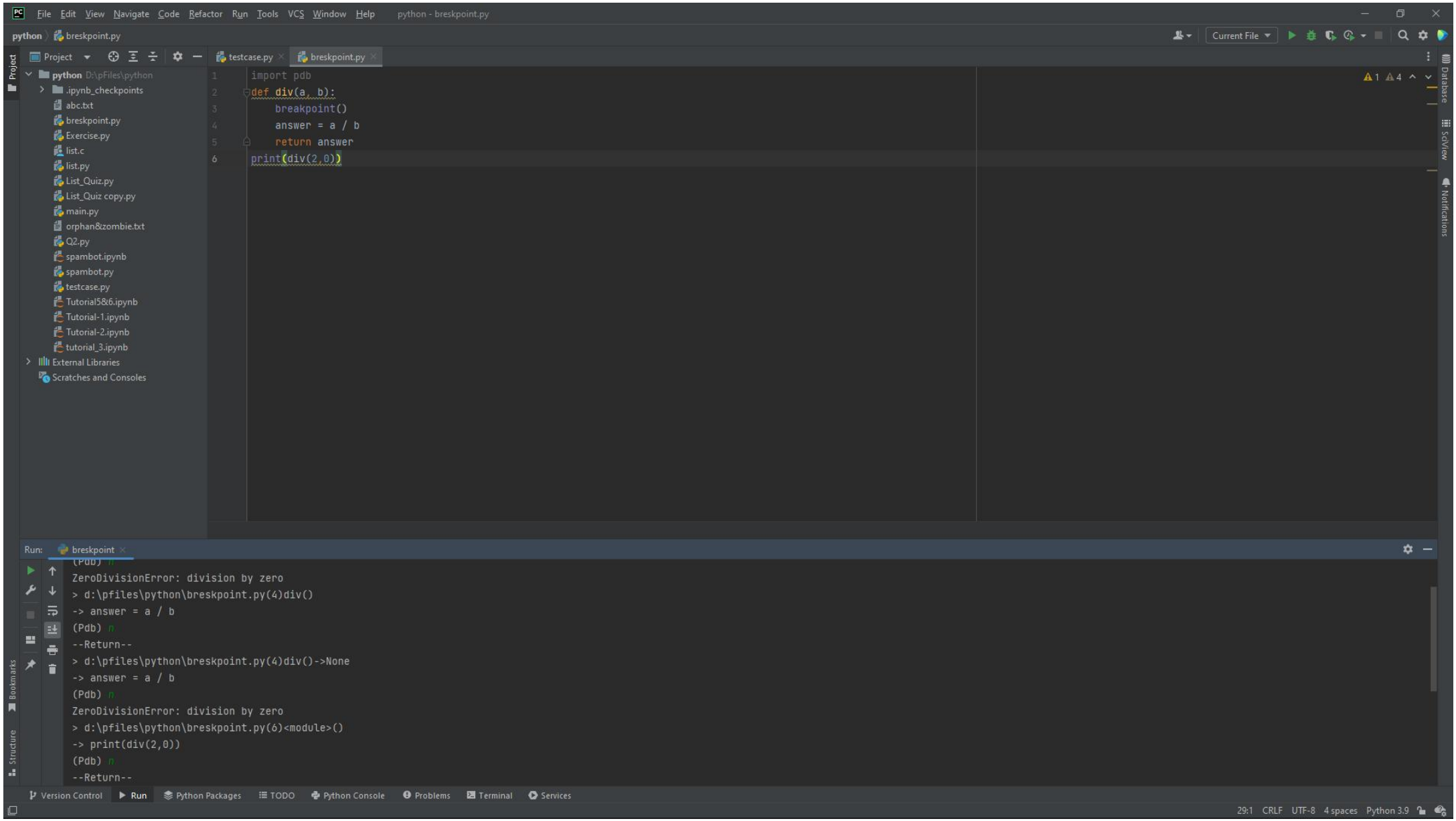
In [7]:
```
import re
txt = input("Enter a string:--")
print("Original string:")
print(txt)
new_txt = re.sub(r"(?<=\d)\s(?=\d)", '', txt)
print('\nAfter concatenating the consecutive numbers in the said string:')
print(new_txt)
```

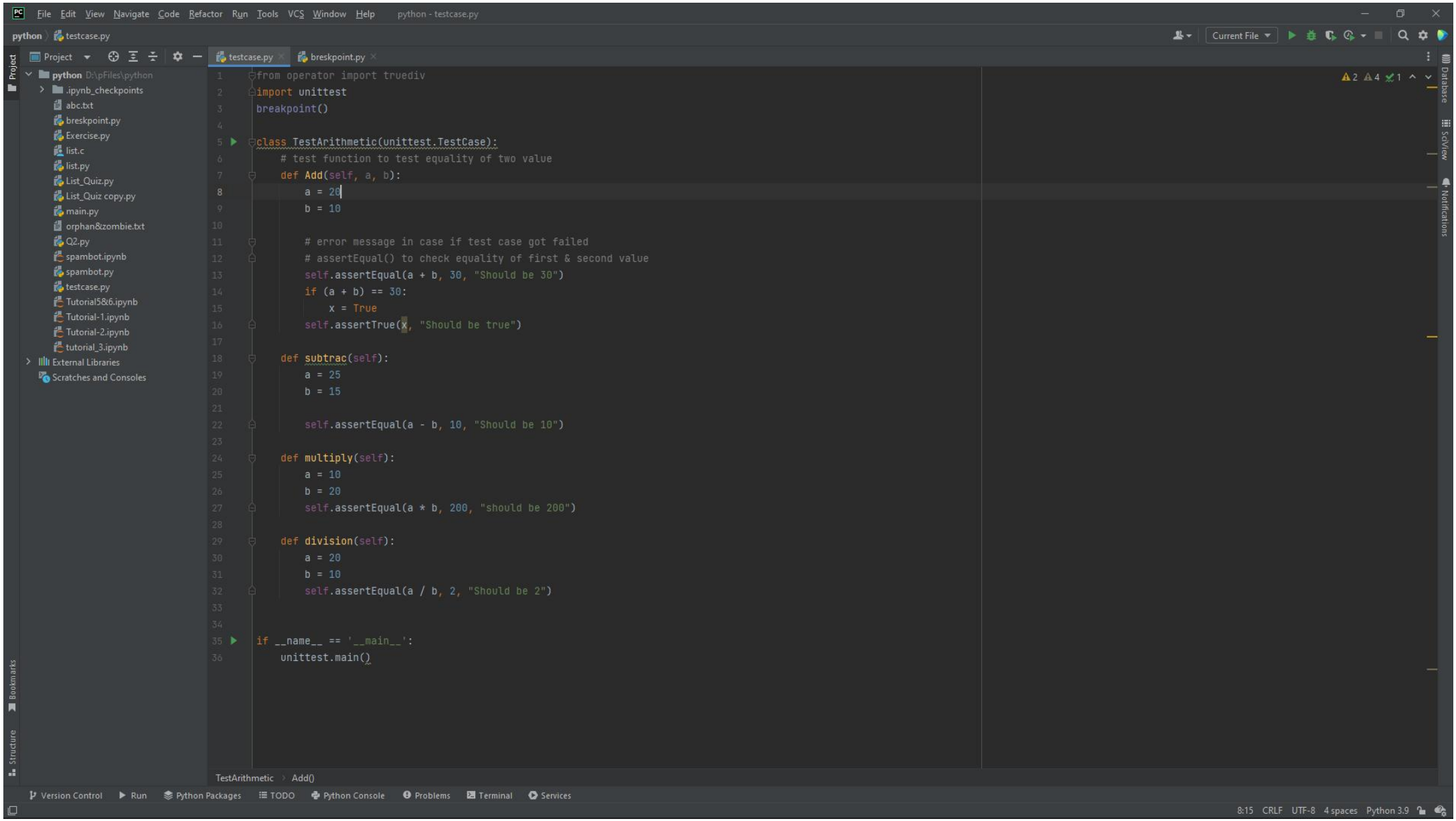
```
Enter a string:--Enter at 1 20 Kearny Street. The security desk can direct you to floor 1 6. Please have your identification ready.
Original string:
Enter at 1 20 Kearny Street. The security desk can direct you to floor 1 6. Please have your identification ready.

After concatenating the consecutive numbers in the said string:
Enter at 120 Kearny Street. The security desk can direct you to floor 16. Please have your identification ready.
```

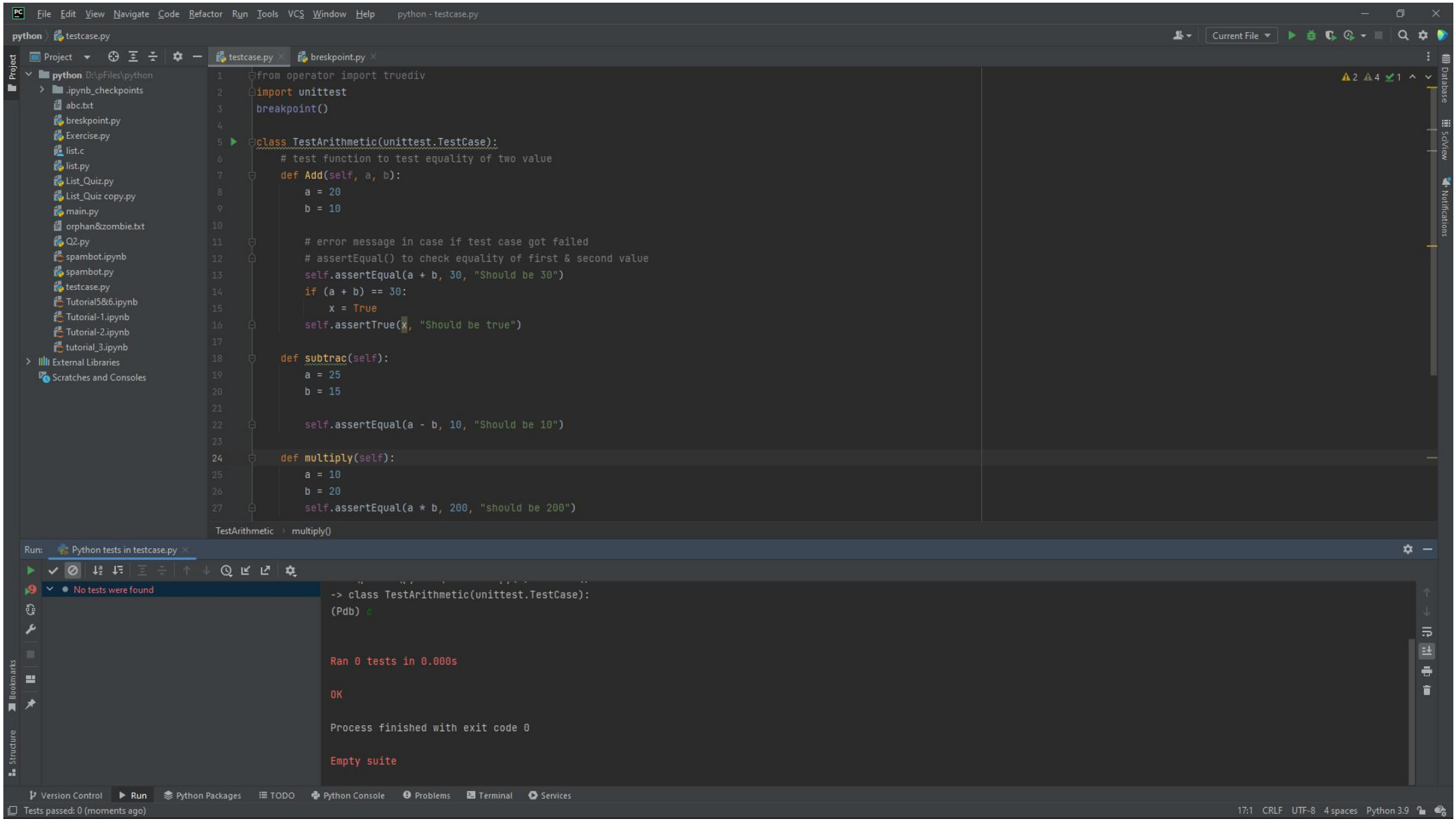
In [8]:
```
from re import sub
def snake_case(s):
  return '_'.join(
    sub('([A-Z][a-z]+)', r' \1',
    sub('([A-Z]+)', r' \1',
    s.replace('-', ' '))).split()).lower()

print(snake_case('iPhoneScript'))
```

```
i_phone_script
```

In [ ]:


File Edit View Navigate Code Refactor Run Tools VCS Window Help
python - breskpoint.py
python breskpoint.py
Current File
Project
testcase.py
breskpoint.py
python D:\pFiles\python
.ipynb_checkpoints
abc.txt
breskpoint.py
Exercise.py
list.c
list.py
List_Quiz.py
List_Quiz copy.py
main.py
orphan&zombie.txt
Q2.py
spambot.ipynb
spambot.py
testcase.py
Tutorial5&6.ipynb
Tutorial-1.ipynb
Tutorial-2.ipynb
tutorial_3.ipynb
External Libraries
Scratches and Consoles
1 import pdb
2 def div(a, b):
3 breakpoint()
4 answer = a / b
5 return answer
6 print(div(2,0))
Run: breskpoint
(Pdb) n
ZeroDivisionError: division by zero
> d:\pfiles\python\breskpoint.py(4)div()
-> answer = a / b
(Pdb) n
--Return--
> d:\pfiles\python\breskpoint.py(4)div()->None
-> answer = a / b
(Pdb) n
ZeroDivisionError: division by zero
> d:\pfiles\python\breskpoint.py(6)<module>()
-> print(div(2,0))
(Pdb) n
--Return--
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
29:1 CRLF UTF-8 4 spaces Python 3.9


File Edit View Navigate Code Refactor Run Tools VCS Window Help
python - testcase.py
python testcase.py
Current File
Project
testcase.py
breskpoint.py
python D:\pFiles\python
.ipynb_checkpoints
abc.txt
breskpoint.py
Exercise.py
list.c
list.py
List_Quiz.py
List_Quiz copy.py
main.py
orphan&zombie.txt
Q2.py
spambot.ipynb
spambot.py
testcase.py
Tutorial5&6.ipynb
Tutorial-1.ipynb
Tutorial-2.ipynb
tutorial_3.ipynb
External Libraries
Scratches and Consoles
from operator import truediv
import unittest
breakpoint()
class TestArithmetic(unittest.TestCase):
# test function to test equality of two value
def Add(self, a, b):
a = 20
b = 10
# error message in case if test case got failed
# assertEqual() to check equality of first & second value
self.assertEqual(a + b, 30, "Should be 30")
if (a + b) == 30:
x = True
self.assertTrue(x, "Should be true")
def subtrac(self):
a = 25
b = 15
self.assertEqual(a - b, 10, "Should be 10")
def multiply(self):
a = 10
b = 20
self.assertEqual(a * b, 200, "should be 200")
def division(self):
a = 20
b = 10
self.assertEqual(a / b, 2, "Should be 2")
if __name__ == '__main__':
unittest.main()
TestArithmetic › Add()
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
8:15 CRLF UTF-8 4 spaces Python 3.9
Database
SciView
Notifications
Bookmarks
Structure


File Edit View Navigate Code Refactor Run Tools VCS Window Help
python - testcase.py
python testcase.py
Current File
Project
testcase.py
breskpoint.py
python D:\pFiles\python
.ipynb_checkpoints
abc.txt
breskpoint.py
Exercise.py
list.c
list.py
List_Quiz.py
List_Quiz copy.py
main.py
orphan&zombie.txt
Q2.py
spambot.ipynb
spambot.py
testcase.py
Tutorial5&6.ipynb
Tutorial-1.ipynb
Tutorial-2.ipynb
tutorial_3.ipynb
External Libraries
Scratches and Consoles
from operator import truediv
import unittest
breakpoint()
class TestArithmetic(unittest.TestCase):
# test function to test equality of two value
def Add(self, a, b):
a = 20
b = 10
# error message in case if test case got failed
# assertEqual() to check equality of first & second value
self.assertEqual(a + b, 30, "Should be 30")
if (a + b) == 30:
x = True
self.assertTrue(x, "Should be true")
def subtrac(self):
a = 25
b = 15
self.assertEqual(a - b, 10, "Should be 10")
def multiply(self):
a = 10
b = 20
self.assertEqual(a * b, 200, "should be 200")
TestArithmetic › multiply()
Run: Python tests in testcase.py
No tests were found
-> class TestArithmetic(unittest.TestCase):
(Pdb) c
Ran 0 tests in 0.000s
OK
Process finished with exit code 0
Empty suite
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
Tests passed: 0 (moments ago)
17:1 CRLF UTF-8 4 spaces Python 3.9