Creating and running a Python unit test

What this tutorial is about

Here we'll see how PyCharm helps creating and running Python unit tests.

What this tutorial is not about

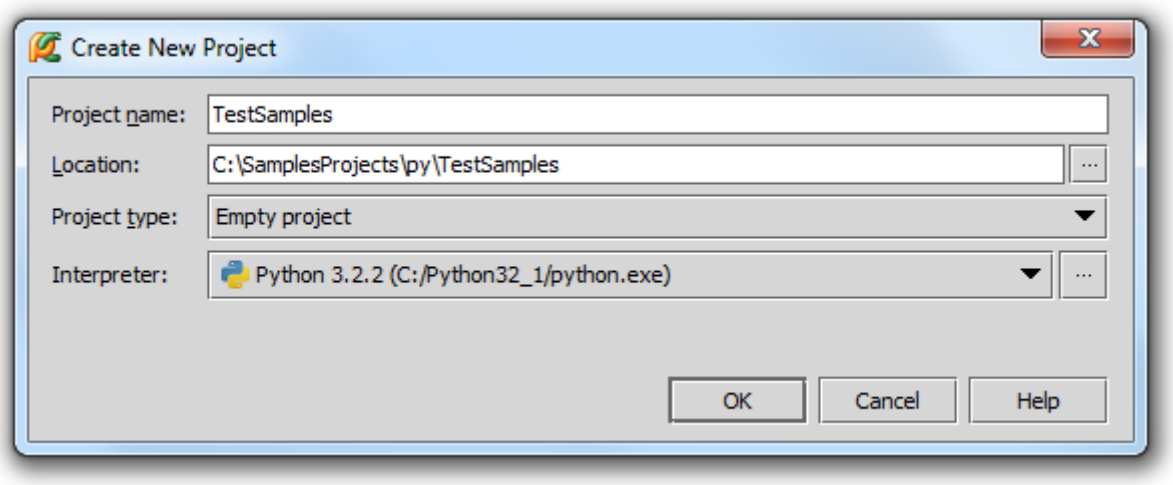
Python programming and writing Python unit tests is out of the scope of this tutorial.

Before you start...

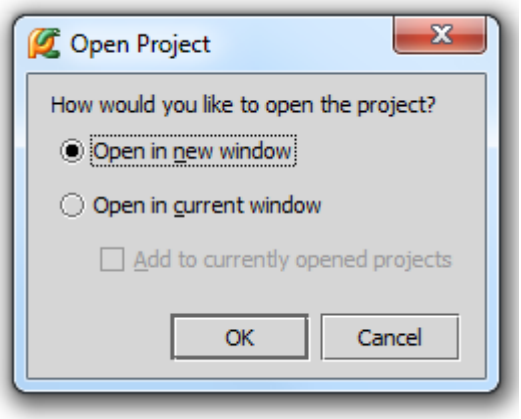
Make sure that at least one Python interpreter (versions from 2.4 to 3.3) is properly installed on your computer.

Creating a simple Python project

On the main menu, choose **File | New Project**.
In the Create New Project dialog box, specify the project name (let it be TestSamples), select the desired project type (here this is Empty project), and the Python interpreter you want to use.

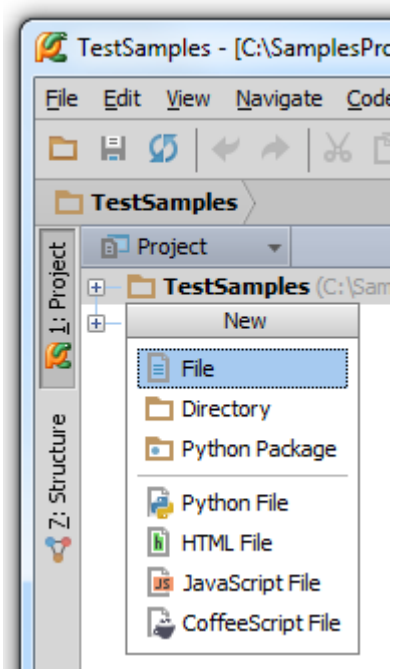


After you click OK, select the window to work with your new project in. In the Open Project dialog box, let us select the first option – open the new project in a separate window:

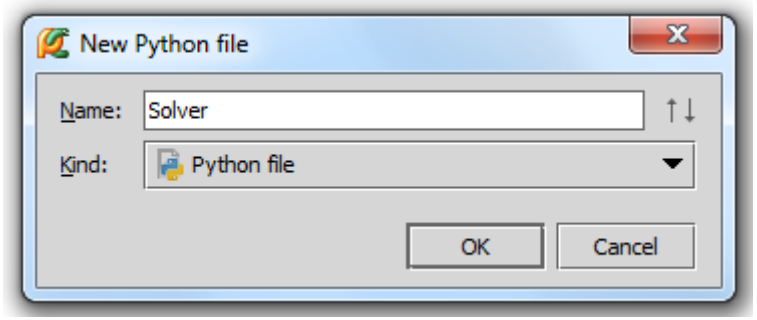


Creating a Python class

Press **Alt+Insert**, and then choose **Python file** from the pop-up window:



In the **New Python file** dialog box, type the new file name:

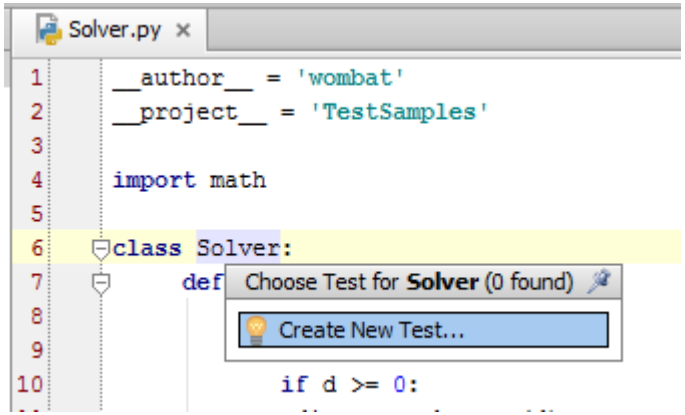


The new file opens in its tab in the editor, with the *__author__* and *__project__* variables already defined. Let's create a simple script that will solve a quadratic equation. Type the following code:

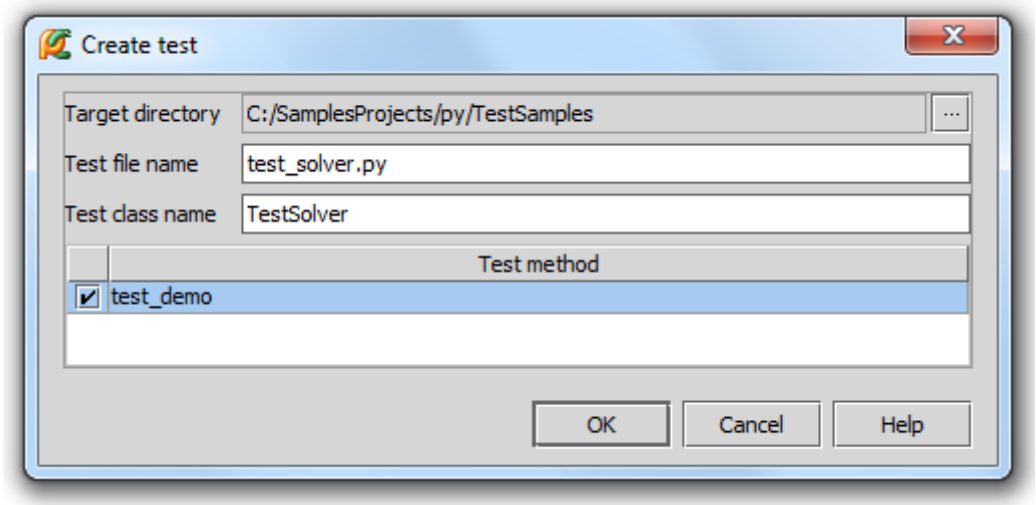
```
import math

class Solver:

     def demo(self, a, b, c):

            d = b ** 2 - 4 * a * c

            if d >= 0:

             disc = math.sqrt(d)

             root1 = (-b + disc) / (2 * a)

             root2 = (-b - disc) / (2 * a)

             print(root1, root2)

            else:

                raise Exception

Solver().demo(2,1,0)
```

Creating test

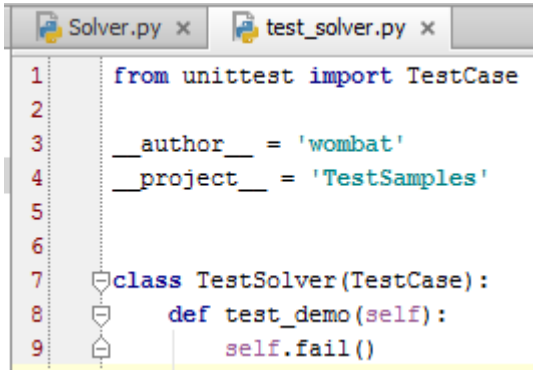
Right-click the class name, and choose **Go to | Test** on the context menu, or just press **Ctrl+Shift+T**:



In the **Create test** dialog box, just accept the suggested directory and names, and select the check box next to the suggested **test_demo** function:



The test case opens in the editor:

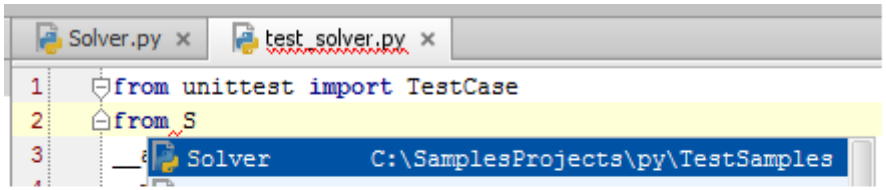


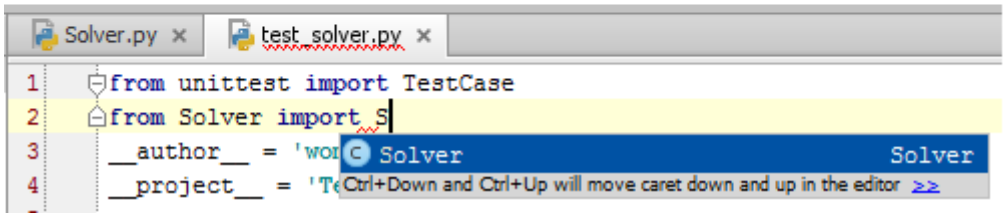
As you see, the created test case meets the requirements to the Python unit testing framework – it imports TestCase class from the unittest module, and the test function names begin with the string “test”.

However, this test case is a kind of common place and requires some fine tuning. Let us first of all import the class we’re going to test. To do that, add import statement:

```
from Solver import Solver
```

Use basic code completion while typing this statement: on pressing **Ctrl+Space**, PyCharm suggests suitable module and class names:

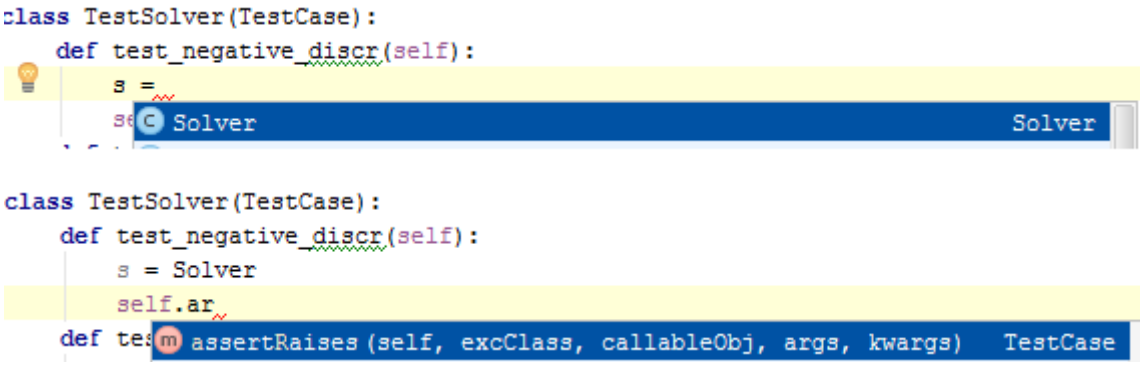




When this statement is complete, it is shown grayed out, which means that import is not used so far.

Next, let's create a test method that will assert throwing an exception, if the discriminant of a quadratic equation is negative. Add the following code to our test case:

```
def test_negative_discr(self):

      s = Solver()

      self.assertRaises(Exception,s.demo,2,1,2)
```



So, the resulting code is:

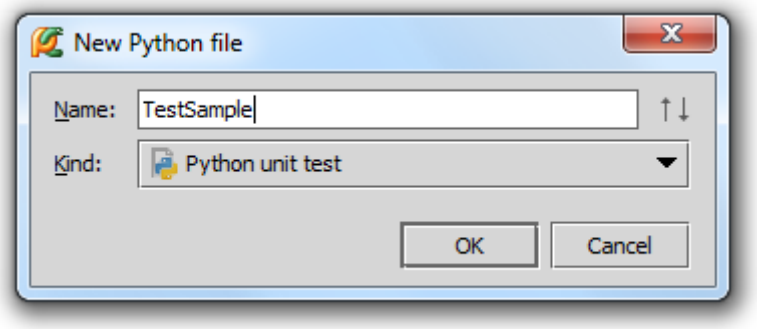
```
from unittest import TestCase

from Solver import Solver



class TestSolver(TestCase):

    def test_negative_discr(self):

        s = Solver()

        self.assertRaises(Exception,s.demo,2,1,2)

    def test_demo(self):

        self.fail()
```

Our test case contains two test methods: test_negative_discr and test_demo. The latter always fails.

Note that import statement is not gray any more – we've used the imported Solver class in the method test_negative_discr.

Alternative way of creating a test case

Try a different approach. Press Alt+Insert, choose **Python file** from the pop-up window, then, in the **New Python file** dialog box, select **Python unit test** from the **Kind** menu, and type the name of new test:



PyCharm create a test case with some initial code, and opens it in the editor:

TestSample.py

```
1  __author__ = 'wombat'
2
3  import unittest
4
5  class MyTestCase(unittest.TestCase):
6      def test_something(self):
7          self.assertEqual(True, False)
8
9
10 if __name__ == '__main__':
11     unittest.main()
```

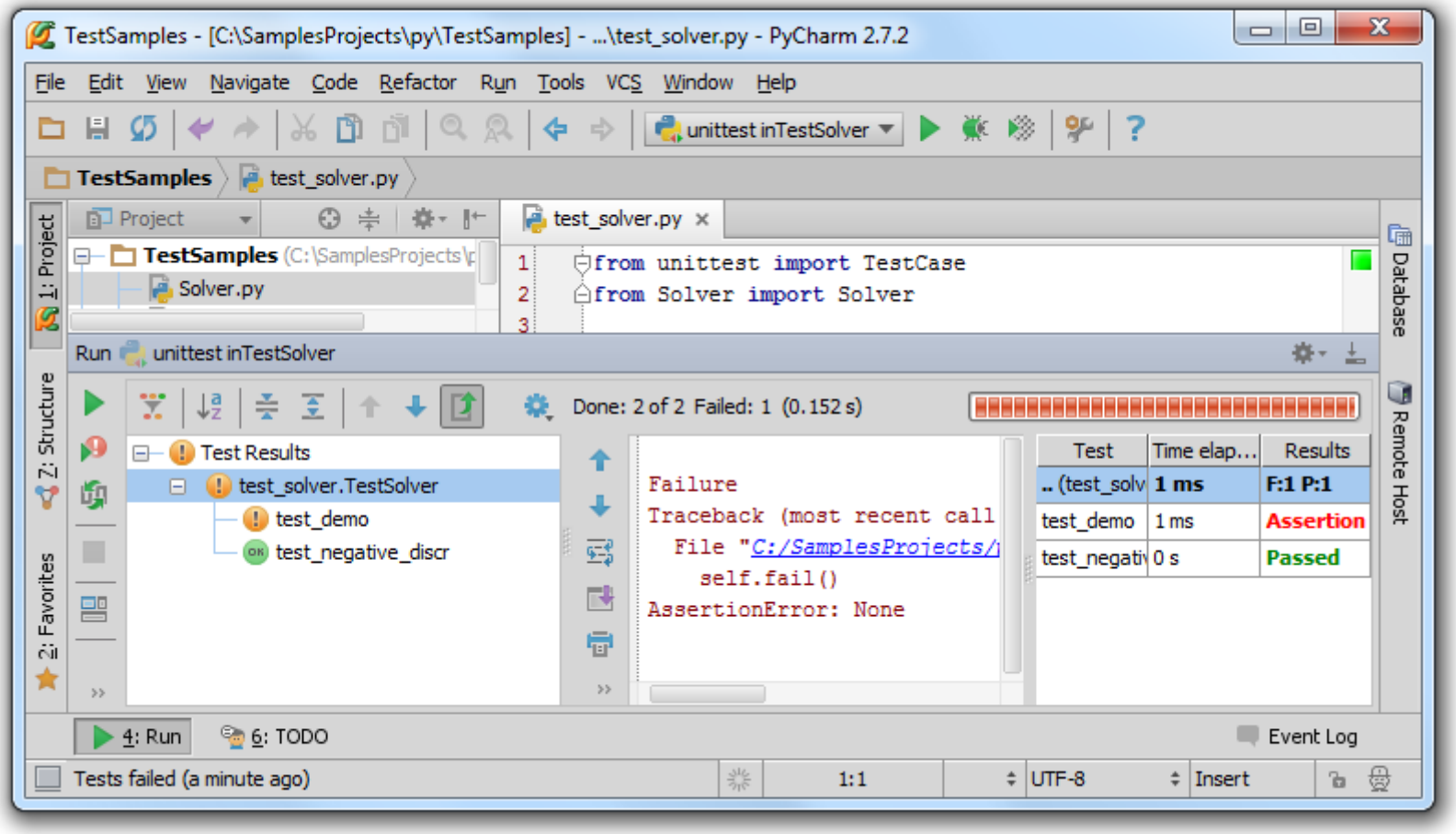
This is again nothing but a stub. So, we'll import the class to be tested, and add a test method. The resulting code is:

```
import unittest

from Solver import Solver



class MyTestCase(unittest.TestCase):

    def test_negative_discr(self):

        s = Solver()
        self.assertRaises(Exception)

    def test_something(self):

        self.assertEqual(True, False)



if __name__ == '__main__':

    unittest.main()
```

Running test

For executing our test case, PyCharm suggests a dedicated temporary run/debug configuration. It is quite ready to be used it “as is”: just press **Ctrl+Shift+F10,** or right-click somewhere within the class, and choose **Run unittests in test_solver** on the context menu.

The results are shown in the Run tool window:



Your Rating: Results: 122 rates

Graham White likes this

10 Comments

Graham White

This is a great page but sadly there is a small error in it. The following:

```
class TestSolver(TestCase):
  def test_demo(self):
     s = Solver
     self.assertRaises(Exception,s.demo,2,1,2)
```

Is missing the parentheses on the Solver class instantiation. It probably wasn't picked up because this causes the test to pass i.e. there is an exception, problem being you will also get an exception with any set of values e.g. 2,1,0.

What this should be is:

```
class TestSolver(TestCase):
  def test_demo(self):
     s = Solver()
     self.assertRaises(Exception,s.demo,2,1,2)
```

Also when you are looking for auto completion on the import Solver from Solver statements then you will find that if you have a project directory structure you will need to reflect that in the auto complete. For me this meant I ended up with:

```
from Chapter09.Solver import Solver
```

I hope this helps someone.

Irina Megorskaya

Thank you. consider it fixed. Note also that since the next release this info will be added to the PyCharm's online help in the Tutorial:Code Running Assistance.

Mansi Mehta

This page is really good. But getting error "No module named selfun" with this code.

```
import unittest
from dsvv_link.selfun import selfun

class MyTestCase(unittest.TestCase):
    def test_negative_discr(self):

        s=selfun()
        self.assertRaises(Exception)

    def test_something(self):
        self.assertEquals(True,False)

if __name__ == "__main__":
    unittest.main()
```

Could anyone help me for this ?

Graham White

For the code above to work you will need to be running it in a file which resides in the dsvv_link folder. That folder will also need to contain a file called selfun.py and contain a selfun class.

Mansi Mehta

I have created class selfun in dsvv_link folder,nut getting this error TypeError: unbound method demo() must be called with selfun instance as first argument (got WebDriver instance instead).

Here is the code for dsvv_link folder:

```
from selenium import webdriver

class selfun():
   def demo(self):

      driver= webdriver.Chrome()

      driver.get("http://www.dsvv.ac.in/")

      scroll_link= driver.find_element_by_link_text("Skill Development Workshops (CCAM) 2017-18")

      scroll_link.click()

      driver.close()

selfun.demo(webdriver.Chrome())

if __name__ == "__main__":
  selfun()
```

Could you please take a look for help me this ?

Graham White

You've defined selfun. demo() to take no arguments. Then you've tried to pass the webdriver.Chrome argument to it.

Replace selfun.demo(webdriver.Chrome()) with selfun.demo()

Mansi Mehta

I tried with replacing selfun.demo(webdriver.Chrome()) with selfun.demo() but it can't work form me.

When I pass webdriver.Chrome argument then browser opens for a while and shows error . So i think whenevr passing argument it tries to open the browser but code not complete after that

Graham White

Actually because you have an if __name__== "__main__" you can just remove the line above it. So remove selfun.demo(webdriver.Chrome())

Mansi Mehta

I tried with removing selfun.demo(webdriver.Chrome()) and code no 'TypeError:' shows but now it will shows "ImportError:".

May be this is because of import package module not installed ? I tried to install import packages but 1 import_path can't be installed.

Dominik Rappaport

Hi,

How can I run all unit tests at once? If I click on the project root and select "Run unittest in projectname" then it says, zero tests were found. It only works for me if I select a specific module.

Best regards, Dominik