

# Creating Test Suites

As was mentioned at the beginning, a test suite is just a collection of test cases, test suites or both. Most of the time, when you call **unittest.main()**, it will do the right thing and gather all the module's test cases for you before executing them. But sometimes you will want to be the one in control. In that circumstance, you can use the **TestSuite** class. Here's an example of how you might use it:

```
import unittest

from test_mymath import TestAdd

def my_suite():
    suite = unittest.TestSuite()
    result = unittest.TestResult()
    suite.addTest(unittest.makeSuite(TestAdd))
    runner = unittest.TextTestRunner()
    print(runner.run(suite))

my_suite()
```

Creating your own suite is a slightly convoluted process. First you need to create an instance of **TestSuite** and an instance of **TestResult**. The **TestResult** class just holds the results of the tests. Next we call **addTest** on our suite object. This is where it gets a bit weird. If you just pass in **TestAdd**, then it has to be an instance of **TestAdd** and **TestAdd** must also implement a **runTest** method. Since we didn't do that, we use **unittest's makeSuite** function to turn our **TestCase** class into a suite.

The last step is to actually run the suite, which means that we need a runner if we want nice output. Thus, we create an instance of **TextTestRunner** and have it run our suite. If you do that and you print out what it returns, you should get something like this printed out to your screen:

...

-----  
Ran 3 tests in 0.001s

OK

<unittest.runner.TextTestResult run=3 errors=0 failures=0>



An alternative is to just call **suite.run(result)** and print out its result.

However all that will give you is a `TestResult` object that looks very similar to that last line of output above. If you want the more usual output, then you will want to use a runner.