Now, look at these three testswe have written for the max method. They look almost identical the only difference is in the values we're using in each test. So in this lecture I'm going to show you a cleaner way to rewrite these tests. In NUnit we have this concept called parameterized tests. So instead of these retest methods here. We can have one test method that takes parameters.

And then we can supply different arguments in that test. So, I'm going to change the name of the first test method to something more generic, so, max when called, it should return a greater argument. Alright? Now here I'm going to add three parameters. Integer A Integer E, and Integer expected result.

So instead of hard coding these values here, I want to pass them as arguments to this method. Okay? So, I'm going to replace 2 and 1, with A and B, and this is the expected result, alright? So now we have a generic, parameterized test method. In NUnit we have another useful attribute called test case, and with this we can supply different arguments with this test method. So, here we have three test cases, the first one is A is greater, so 2 and 1, and the expected result is 2. I'm going to duplicate this

for the second test case, we want the second argument to be greater.

So, 1 and 2, and the expected result is 2. And, the last test case, where, both arguments are equal. So, 1 on 1, and the expected result is 1. With this, we no longer need to write these additional tests. They are redundant. Isn't this cleaner? So, now let's run this test. Command and T, and Command and R,

We can see the test we have written for the Max method now has three test cases. And they're all passing, if one of them fails, you can simply click, and look at the details of the failed test here. And by the way, this is one of those areas where NUnit is different from MS Test, in MS Test, unfortunately we don't have a simple solution for parameterizing tests. You have to create an Excel spreadsheet or a CSS file, put it in your test project, and then reference that here, and the path to that file (?) specify the worksheet, it's just too complex.

And it surprises me that after all these years, MS Test still doesn't have this feature. NUnit has

been supporting this for a long time. But as you probably know, this is the case with Microsoft, Microsoft is often 5 years behind the game. So, whatever other frameworks and other third party libraries do, Microsoft will adopt that after 5 years. So that's why NUnit is my preferred testing framework at the time of recording this video.