**TestNG Vs JUnit**

Comparison code written here: <https://github.com/Amit-GitRepository/JUnitVsTestNG>

|  |  |  |
| --- | --- | --- |
| **Feature** | **JUnit** | **TestNG** |
| Skipping a test | Skipping test is done by using @Ignored above @Test annotation  Skipped test is highlighted in console | Skipped by using  @Test(enabled = false)  Skipped tests are not highlighted in console |
| Method Dependency | No such feature | Can be achieved using @Test(dependsOnMethods="test2")  Dependent method is skipped if prior method fails |
| Suite test | Test all classes can be run using @RunWith | Suits can be executed in testNG using xml file  However using TestNG it creates an xml for failed scenarios, which can be used to re-run tests |
| Parameterization | Test can be passed parameter using  @RunWith(value = Parameterized.class) | We can pass parameters from TestNG.xml to tests using @Parameter annotation |
| Running | Cannot run TestNG test using JUnit | JUnit test can be run using testNG.xml adding junit=true in test |
| Annotations | 1. Lesser number of annotations in comparison to TestNG 2. In JUnit 4, we have to declare “@BeforeClass” and “@AfterClass” method as static method. 3. In JUnit 4, the annotation naming convention is a bit confusing, e.g “Before”, “After” and “Expected”, we do not really understand what is “Before” and “After” do, and what we “Expected” from test method | 1. Extra annotations include @BeforeSuite, BeforeTest and BeforeGroup 2. TestNG is more flexible in method declaration, it does not have this constraints. 3. TestiNG is easier to understand, it uses “BeforeMethod”, “AfterMethod” and “ExpectedException” instead. |