TESTNG - BASIC ANNOTATIONS

http://www.tutorialspoint.com/testng/testng_basic_annotations.htm

The traditional way to indicate test methods in JUnit 3 is by prefixing their names with test. This is a very effective method for tagging certain methods in a class as having a special meaning, but the naming doesn’t scale very well *whatifwewanttoaddmoretagsfordifferentframeworks*? and is rather inflexible *whatifwewanttopassadditionalparameterstothetestingframework*?.

Annotations were formally added to the Java language in JDK 5, and TestNG made the choice to use annotations to annotate test classes.

Here is the list of annotations that TestNG supports:

Annotation	Description
@BeforeSuite	The annotated method will be run only once before all tests in this suite have run.
@AfterSuite	The annotated method will be run only once after all tests in this suite have run.
@BeforeClass	The annotated method will be run only once before the first test method in the current class is invoked.
@AfterClass	The annotated method will be run only once after all the test methods in the current class have run.
@BeforeTest	The annotated method will be run before any test method belonging to the classes inside the <test> tag is run.
@AfterTest	The annotated method will be run after all the test methods belonging to the classes inside the <test> tag have run.
@BeforeGroups	The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
@AfterGroups	The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
@BeforeMethod	The annotated method will be run before each test method.
@AfterMethod	The annotated method will be run after each test method.

@DataProvider	Marks a method as supplying data for a test method. The annotated method must return an Object[ ][ ], where each Object[ ] can be assigned the parameter list of the test method. The @Test method that wants to receive data from this DataProvider needs to use a dataProvider name equals to the name of this annotation.
@Factory	Marks a method as a factory that returns objects that will be used by TestNG as Test classes. The method must return Object[ ].
@Listeners	Defines listeners on a test class.
@Parameters	Describes how to pass parameters to a @Test method.
@Test	Marks a class or a method as a part of the test.

Benefits of Using Annotations

Following are some of the benefits of using annotations:

- TestNG identifies the methods it is interested in, by looking up annotations. Hence, method names are not restricted to any pattern or format.
- We can pass additional parameters to annotations.
- Annotations are strongly typed, so the compiler will flag any mistakes right away.
- Test classes no longer need to extend anything $suchasTestCase, forJUnit3$.