



TestNG introduction

Topics:

- What is TestNG?
- Why are we using it?
- Testing frameworks overview
- TestNG basics
- TestNG assertions
- TestNG annotations
- Selenium testing with TestNG

After today's session you should be able to:

- Write tests using TestNG
- Integrate TestNG and Selenium
- Use assertions in tests
- Control test flow using annotations

The history of TestNG...

Testing Tools (Frameworks)

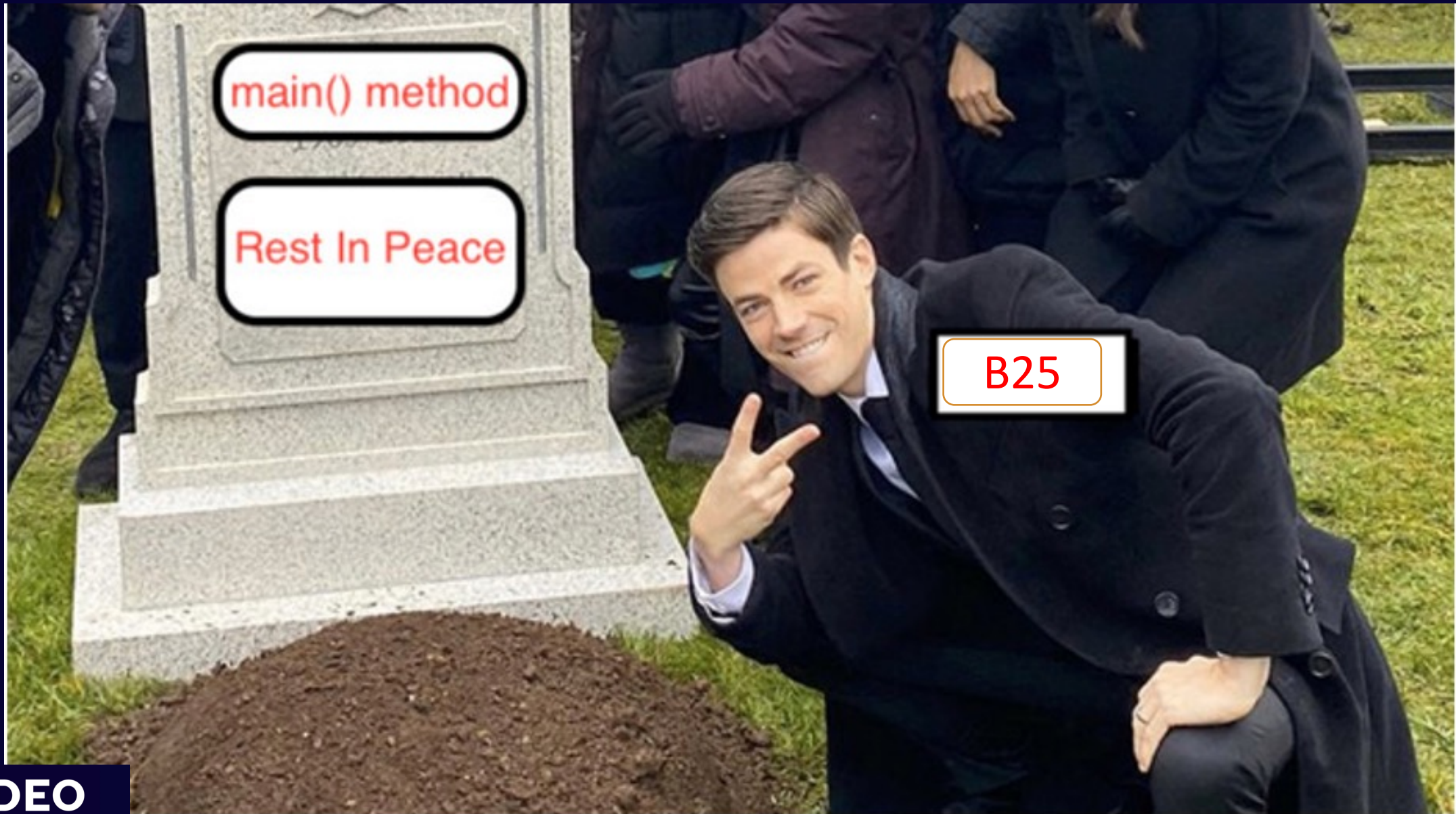
- TestNG, Junit, Nunit etc...
- These tools enables unit testing, functional testing...
- Provides assertions for verifying
- We can use annotation to control test flow
- We can generate some reports



Let's implement TestNG

- To do that we need to go to mvnrepository, and get the dependency
- Normally we also need to add the plug-in to the IDE as well but the IntelliJ comes with it, so we don't have to.
- If it was Eclipse we would have to.

From today we will not be using main method



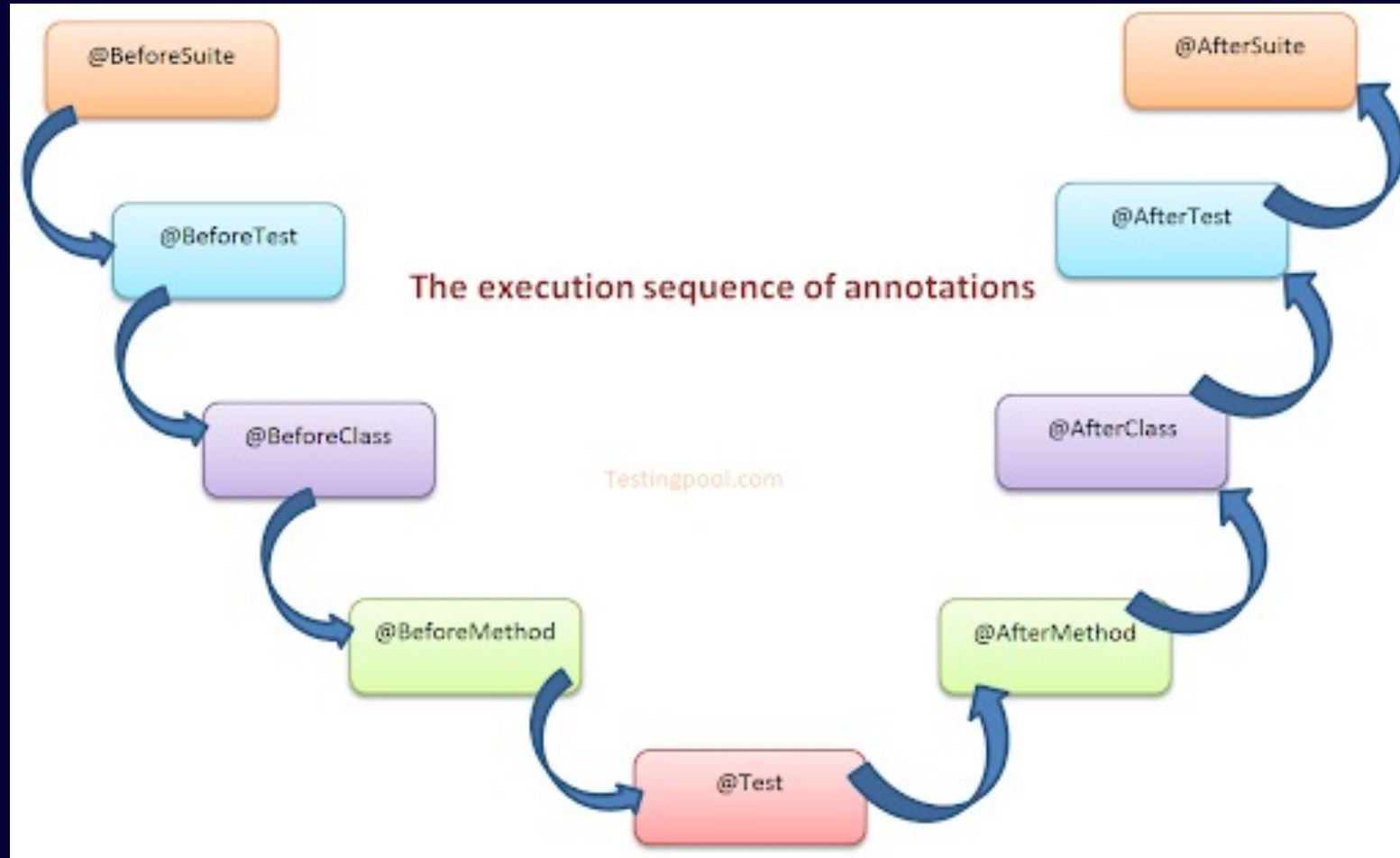
@Test

- Indicates that method is a test method. Test method where we do assertions.

```
@Test
public void titleTest(){
    assertEquals(driver.getTitle(), "Google");
}
```

- @Test is the main annotation where we create the test
- All of the other annotations are there to only create & support structure and flow around @Test annotation

All TestNG annotations in the order they are executed



@BeforeMethod

- Method with this annotation will always run once before every @Test method
- Usually used for setting up for every test

```
@BeforeMethod
public void setUp(){
    driver.get("http://google.com");
}
```

@AfterMethod

- Basically runs once after every @Test
- Usually used for closing down setups for every test

```
@AfterMethod  
public void tearDown(){  
    driver.close();  
}
```

@BeforeClass

- Method with this annotation will always run once before all of the other methods in this class
- Usually used for setups

```
@BeforeClass  
public void setUp(){  
    |    WebDriver driver = new ChromeDriver();  
}
```

@AfterClass

- Runs ONCE after all the methods are DONE.
- Usually used for tearDowns, closing browser, killing instances etc...

```
@AfterClass  
public void tearDown(){  
    driver.quit();  
}
```


Assertions

- Assert class methods are used to do verifications with TestNG

assertTrue();

assertTrue()

```
assertTrue(true);
```

```
assertTrue(1==1);
```

```
assertTrue(element.isDisplayed());
```

assertEquals() → Checks 2 arguments equal or not

assertEquals()

```
assertEquals(1, 1)
```

```
assertEquals("a", "a")
```

```
assertEquals(driver.getTitle(), "Google")
```

Assert.fail();

- If you want to fail your test intentionally...