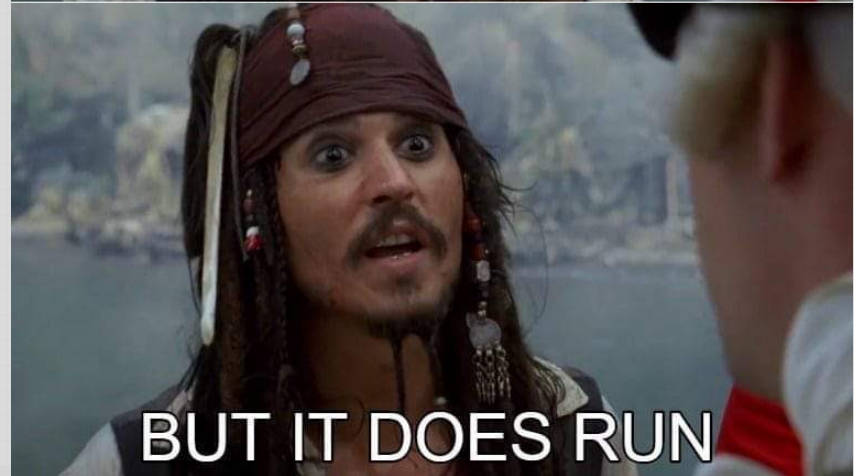
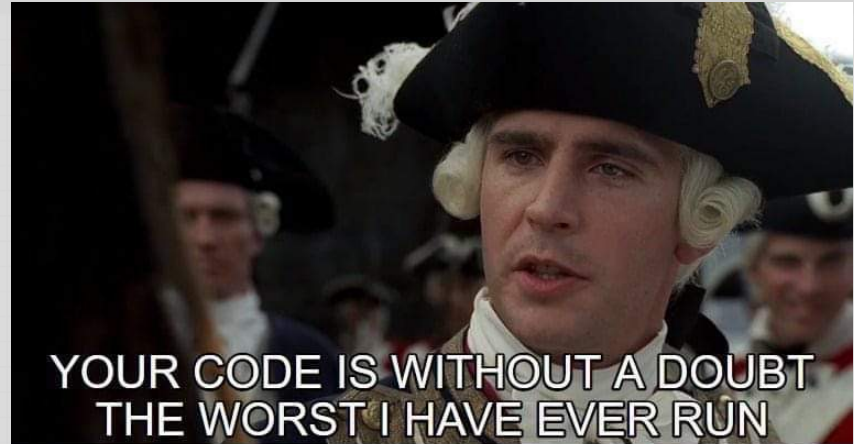




TestNG Introduction





Topics to cover

- What is TestNG?
- Why are we using TestNG?
- 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



Unit Testing Tools

- 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

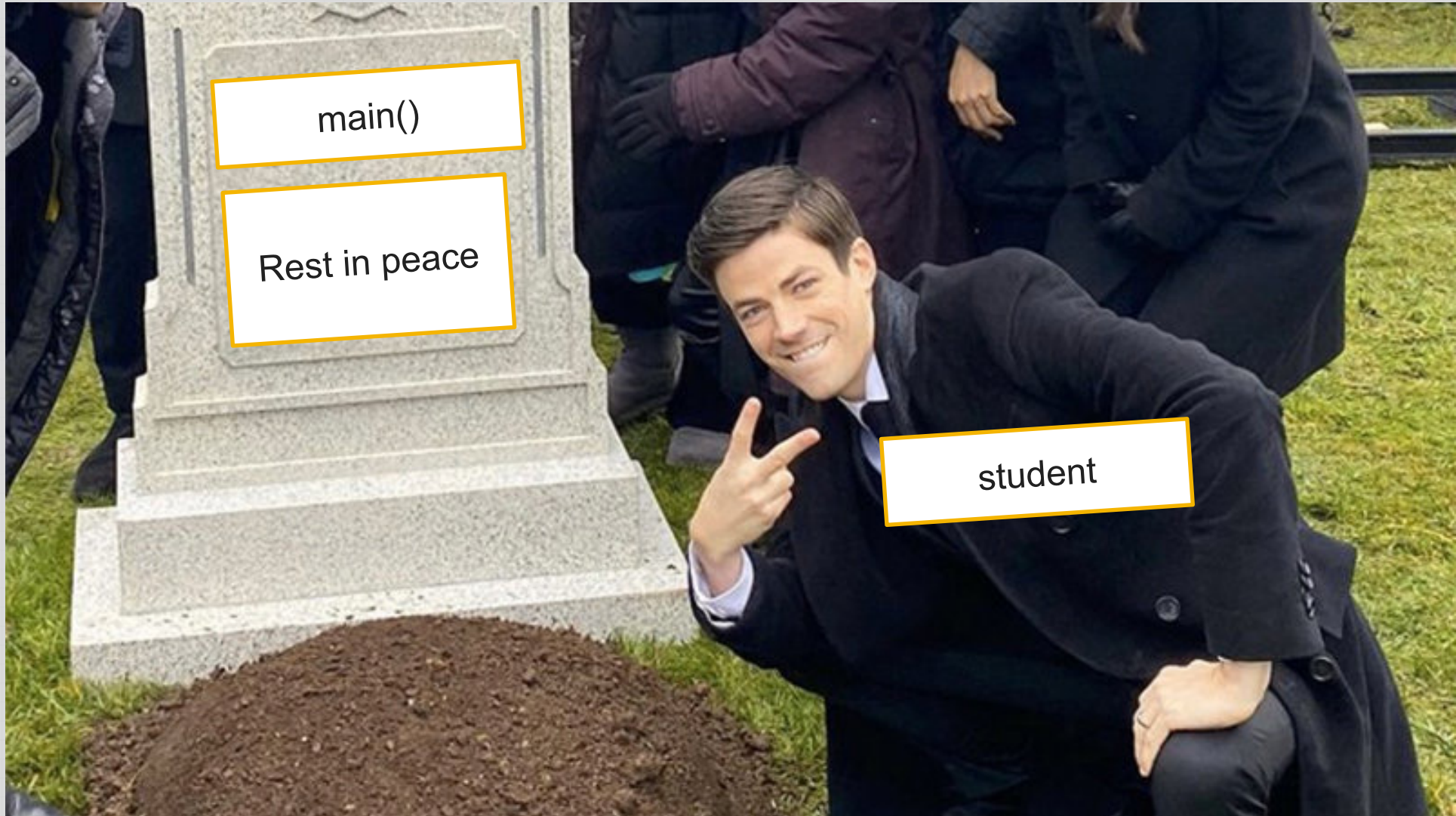


Let's implement TestNG in our project

1. Go to mvnrepository to get the dependency
2. Search for TestNG
3. Add the dependency into the pom.xml file
4. Refresh your project to download the dependency



From today we will not be using main method



@Test annotation

- Indicates that method is a test method.
- In a test method, we can do assertions.

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


@Test annotation

- “@Test” is the main annotation in TestNG, where we create the “test”
- All of the other annotations we will be learning later is there to create and support certain structure around “@Test” annotations.



@BeforeMethod annotation

- Method with this annotation will always run **ONCE** before **EVERY** @Test
- Usually used for setting up repeating steps for each test.

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

@AfterMethod annotation

- Basically runs **ONCE** before **AFTER** @Test
- Usually used for closing down repeating steps for each test.

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

@BeforeClass annotation

- Method with this annotation will always run **ONCE** before **all other methods**.
- **Runs ONCE only. Not repeated for each @Test.**
- Usually used for setups.

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

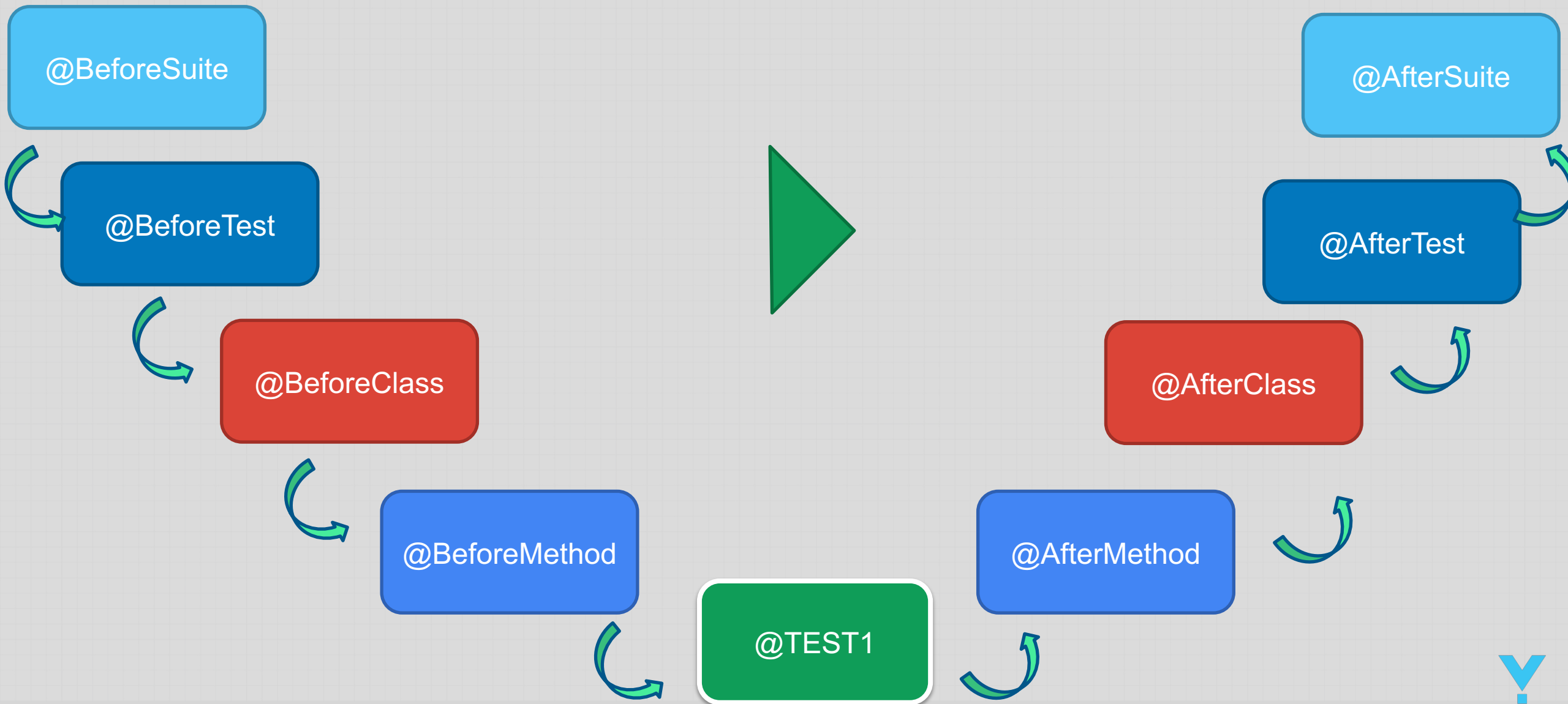


@AfterClass annotation

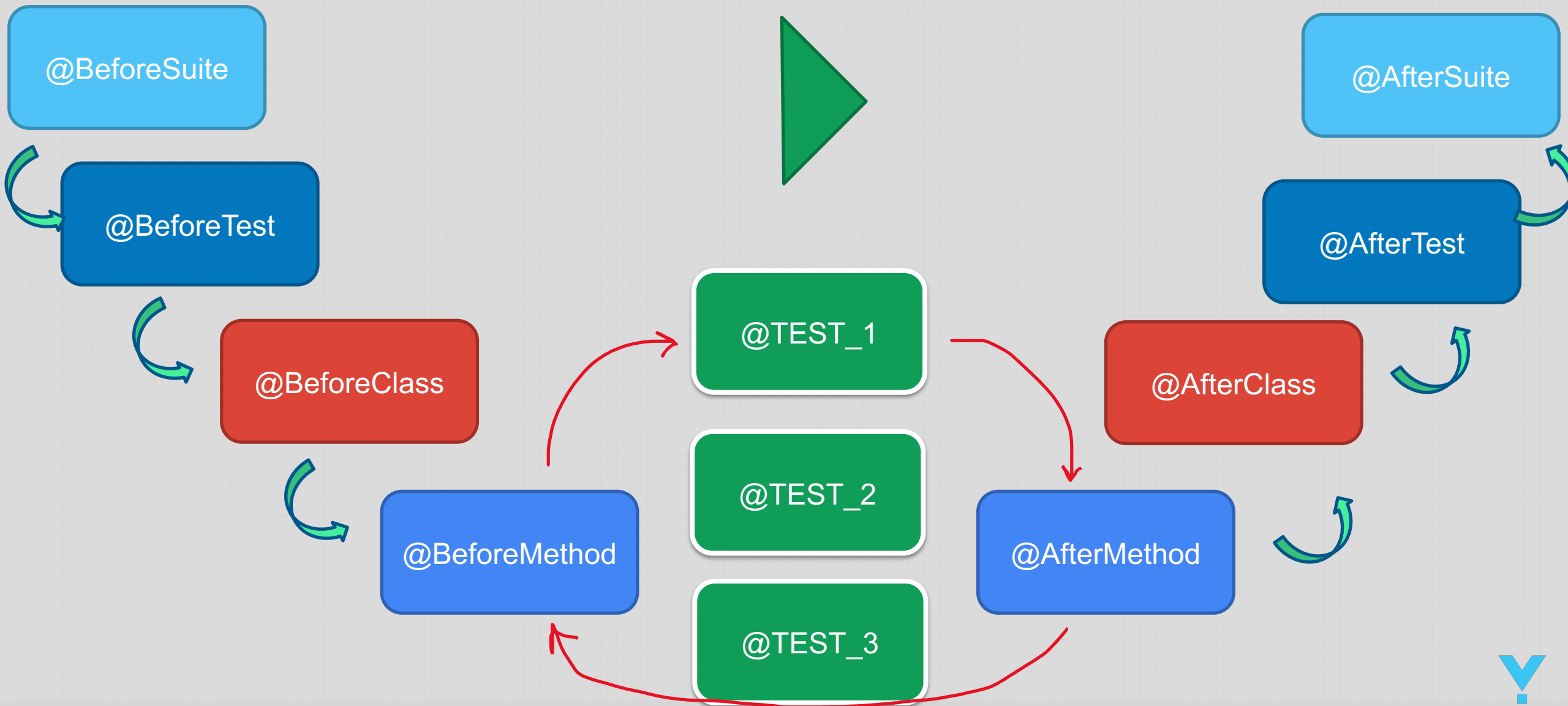
- Basically will run **ONCE** after **all other methods**.
- **Runs ONCE only. Not repeated for each @Test.**
- Usually used for teardowns, closing browser, killing instances, closing DB connections.

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

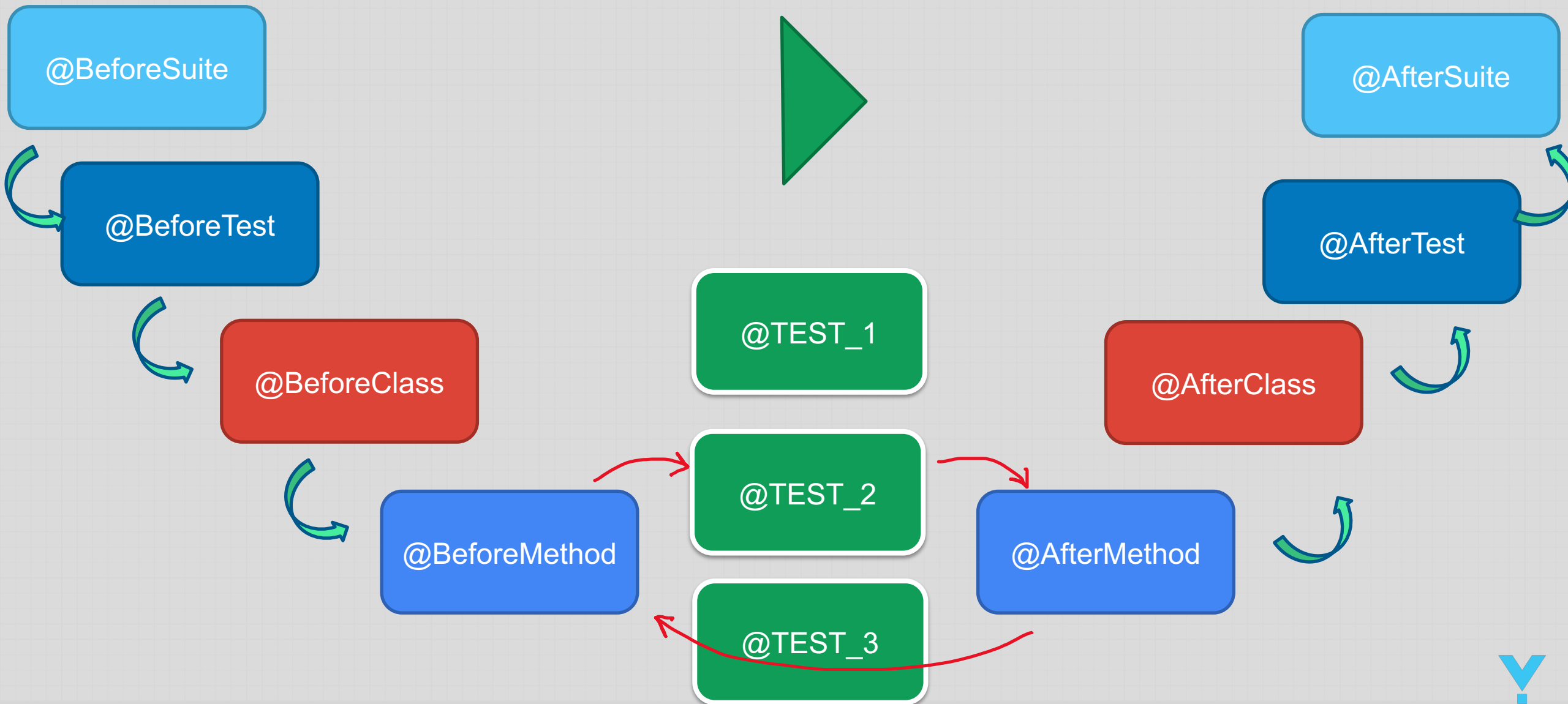

TestNG annotations in the order they are executed



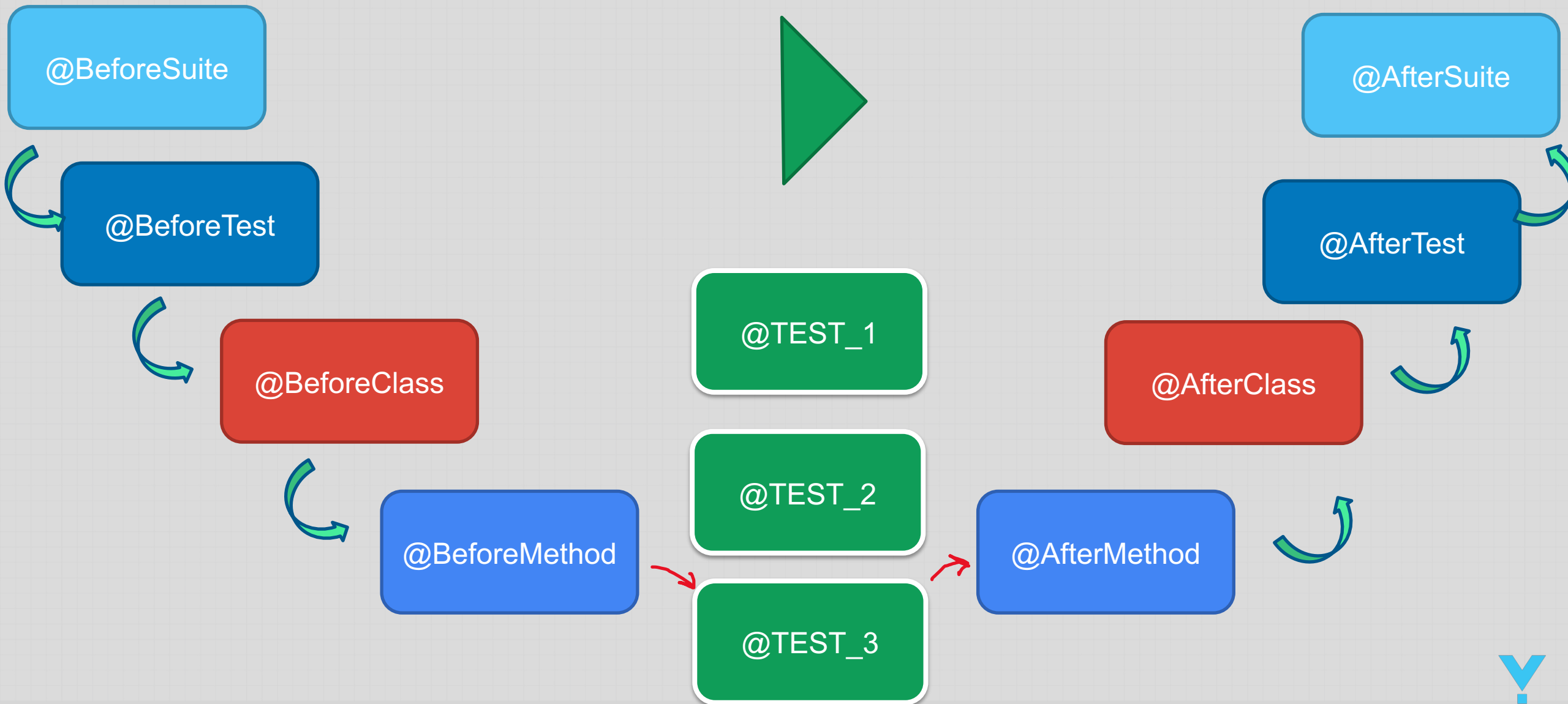
TestNG annotations in the order they are executed



TestNG annotations in the order they are executed



TestNG annotations in the order they are executed



Assertions annotation

- Assert class methods are used to do verifications with TestNG.
- All methods coming from Assert class is static.
- So we can just call the class name and methods we want to use.



assertTrue(); method

- Accepts a boolean value.
- If boolean is **true**, test **passes**.
- If boolean is **false**, test **fails**.

assertTrue()

```
assertTrue(true);
```

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

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



assertEquals(); method

- Accepts 2 arguments of the same type, and compares them.
- If the values we provide **equal**, test **passes**.
- If the values we provide **not equal**, test **fails**.

assertEquals()

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

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

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



assertFail(); method

- Its only job is to fail a test when its called.