

- Assertions
- TestNG.xml →
 - ↳ Depends On Methods
- parametrization

→ testng.xml → Convert to TestNG

→ To execute multiple classes at the same time

Depend on Method

m1() - 1 → fail ✓
m2() - 2 → skip
m3()

prioritization

m1 - 1, FAIL
m2 - 2 = FAIL

Depends On Methods →

@Test (depends On Methods = " ")
m1 () {

3

Assertions :- is used for validation purpose

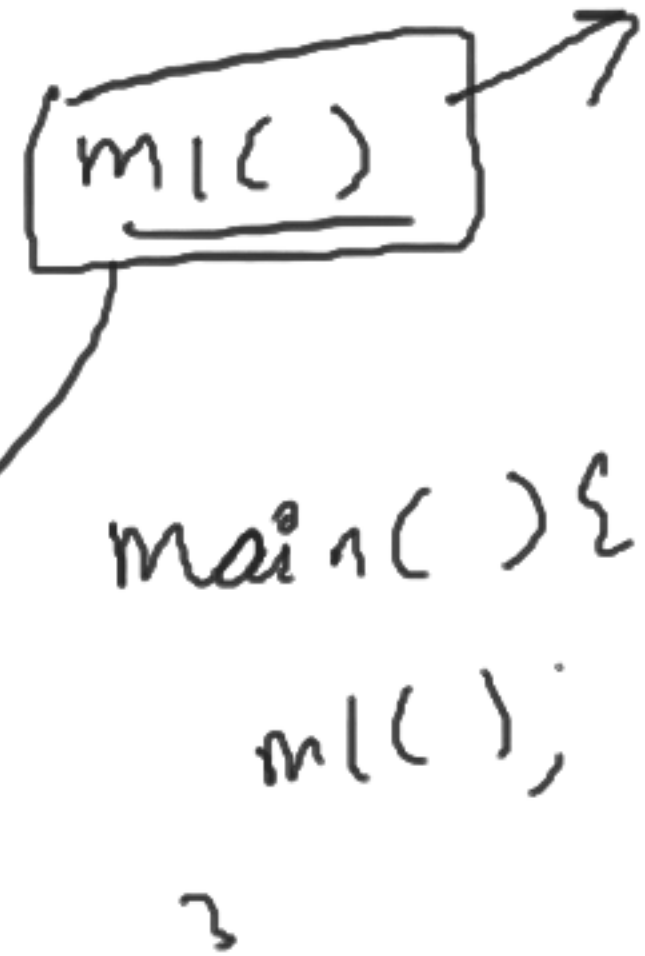
Compare actual result with expected

In TestNG -

- Hard Assertion
- Soft Assertion

@Test
✓✓

```
main() {  
    ml();  
}
```



SoftAssertion → If condition/validation is failed, still we can continue with

Assert ✓ → E2E execution along with Error Msg.
↑
Soft Assert ✓ → child Message - Pass, FAIL
Report

E2E

Class Assert { ^{Hard Assertion}
public static void assertEquals()
{
}
}

Assert.assertEquals();

^{Soft Assertion}
Class ~~Soft~~ Assert extends

Assert { ^{N.S}
public void assert
Equals() {
}

{
public void assertNull()
{
}

SoftAssert obj →
obj →

S.A

B known
App
login
valid → Fail → stop ✓

TC FAIL

X Hard Asu

Test Case → by providing some Arp

ml (intX) {

}

ml (100) }

→ Parallel execution / Seq. Execution
Parametrisation

TestNG Class

|

XML

@Test

→ Report → ^{not} Correct result

if () {

{else {

Match

testing.xml

<parameter>

name

@Parameter(' - ')

@BeforeClass

@BeforeTest ✓

@BeforeSuite

method ()

testing.xml

<suite>

<parameter>

<test>

↳ <parameter>

TestNG class
@parameters ("browser")

R @Test / @BeforeTest

@BeforeSuite

<Suite>

<test>

- <parameter>

<classes>

<class>

</class> </classes>

</test>

<test>
</test> <parameter>

XML

<test>

<parameter Tag>

name

value

<Suite> parallel="test"

<parameter>