

Annotations

@Test	
@BeforeEach	Prepare Single
@AfterEach	Clean Single
@BeforeAll	Prepare All
@AfterAll	Cleanup All

Assertions

```

assert True(b oolean b)
assert False( boolean b)
assert Not Null(Obj ect o)
assert Null(Obj ect o)
assert Not Same(Obj ect unexpe cted, actual)
assert Equals( ex pected, actual)
assert Not Equals( unexpe cted, actual)
assert ArrayE qua ls( Obj[] expected, actual)
assert It e rab leE quals( expected, actual)
assert Tim eou t(D uration , Executable lambda)
assert Thr ows (Class <T extends Throwa ble> exClass, Executable lam bda)
fail(S tring message)

```

Assumptions

```

assume True(b oolean assumption)
assume Fal se( boolean assumption, String msg)

```

* Failed assumptions **aborts** test, not *failure*

* Assumptions are typically used whenever it does not make sense to continue execution of a given test method

Sample

```

@org.junit.jupiter.api.Test
void exampleTest() {
    Ass ert ion s.a sse rtT rue (tr ueB ool);
    Ass ert ion s.a sse rtF als e(f als eBool);
    Ass ert ion s.a sse rtN otN ull (no tNu -
    llS tring);
    Ass ert ion s.a sse rtN ull (no tNu llS -
    tring);
    Ass ertions
        .as ser tNo tSa me( ori gin alO -
    bject, otherO bject);
    Ass ert ion s.a sse rtE qua ls(4, 4);
    Ass ert ion s.a sse rtN otE qua ls(3, 2);
    Ass ertions
        .as ser tAr ray Equals(
        new int[]{ 1,2,3},
        new int[]{ 1,2,3},
        " Array Equal Test");
    It e rab le< Int ege r> listOne =
        new ArrayL ist <>( Arr ays.as Lis -
    t(1, 2, 3, 4));
    It e rab le< Int ege r> listTwo =
        new ArrayL ist <>( Arr ays.as Lis -
    t(1 ,2, 3,4));
    Ass ert ion s.a sse rtI ter abl eEq ual -
    s(listOne, listTwo);
    Ass ert ion s.a sse rtT ime out (Du rat -
    ion.of Mil lis (100), () -> {
        Thr ead.sl eep (50);
        return " res ult ";
    });
    Thr owable exception = Assertions
        .as ser tTh row s(I lle gal Arg ume -
    ntE xce pti on.c lass,
        () -> {
            throw
                new Illega lAr gum ent Exc -
    eption(
                " error messag e");
        });
    Ass ert ion s.f ail ("You shall not parse □
    □");
}

```

