



NUnit Cheat Sheet

Simple Test

```
class UserTests
{
    [Test]
    public void isAdult_AgeLessThan18_False()
    {
        User paul = new User(17);
        Assert.That(paul.isAdult(), Is.False);
    }
}
```

Basic Assertions

```
Assert.That(condition, Is.True);
Assert.That(condition, Is.False);
Assert.That(actual, Is.EqualTo(expected)); // Equal
Assert.That(actual, Is.Not.EqualTo(expected)); // Not equal
Assert.That(actual, Is.LessThan(expected)); // value1 < value2
Assert.That(actual, Is.GreaterThan(expected)); // value1 > value2
```

Exception Assertions

```
Assert.Throws<expectedExceptionType>(method);
Assert.DoesNotThrow(method);

[Test]
public void crash_Normal_ShouldThrow()
{
    Car car = new Car();
    Assert.Throws<NotImplementedException>(car.crash);
}
```



inovex

Test Fixture

```
[TestFixture]
public class UserTests
{
    [SetUp]
    public void SetUp() { /* runs before each test */ }

    [TearDown]
    public void TearDown() { /* runs after each test */ }

    User paul = new User(5);

    [Test]
    public void isAdult_AgeLessThan18_False()
    {
        Assert.That(paul.isAdult(), Is.False);
    }
}
```

Value Parameterized Tests

```
[TestFixture]
class UserTests
{
    private static IEnumerable<int> adultAges()
    {
        yield return 18;
        yield return 20;
        yield return 60;
    }

    [Test, TestCaseSource(nameof(adultAges))]
    public void isAdult_AgeLessThan18_False2(int age)
    {
        User paul = new User(age);
        Assert.That(paul.isAdult(), Is.True);
    }
}
```