

GoogleTest

Fatal assertion	Nonfatal assertion
ASSERT_TRUE(condition);	EXPECT_TRUE(condition);
ASSERT_FALSE(condition);	EXPECT_FALSE(condition);
ASSERT_EQ(expected, actual);	EXPECT_EQ(expected, actual);
ASSERT_NE(val1, val2);	EXPECT_NE(val1, val2);
ASSERT_LT(val1, val2);	EXPECT_LT(val1, val2);
ASSERT_LE(val1, val2);	EXPECT_LE(val1, val2);
ASSERT_GT(val1, val2);	EXPECT_GT(val1, val2);
ASSERT_GE(val1, val2);	EXPECT_GE(val1, val2);
ASSERT_STREQ(expected, actual);	EXPECT_STREQ(expected, actual);
ASSERT_STRNE(str1, str2);	EXPECT_STRNE(str1, str2);
ASSERT_STRCASEEQ(expected, actual);	EXPECT_STRCASEEQ(expected, actual);
ASSERT_STRCASENE(str1, str2);	EXPECT_STRCASENE(str1, str2);

```
#include "this/package/foo.h"
#include "gtest/gtest.h"

namespace {

class FooTest : public ::testing::Test {
protected:
    FooTest() { }
    virtual ~FooTest(){ }
    virtual void SetUp(){ }
    virtual void TearDown() { }
    // Objects declared here can be used by all tests in the test case for Foo.
};

TEST_F(FooTest, DoesXyz) {

}

} // namespace

int main(int argc, char **argv) {
    ::testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```