testify: (https://github.com/stretchr/testify) github.com/stretchr/testify (/github.com/stretchr/testify)/assert

Index | Examples | Files

package assert

```
import "github.com/stretchr/testify/assert"
```

Package assert provides a set of comprehensive testing tools for use with the normal Go testing system.

Example Usage

The following is a complete example using assert in a standard test function:

```
import (
  "testing"
  "github.com/stretchr/testify/assert"
)

func TestSomething(t *testing.T) {

  var a string = "Hello"
  var b string = "Hello"

  assert.Equal(t, a, b, "The two words should be the same.")

}
```

if you assert many times, use the format below:

```
import (
  "testing"
  "github.com/stretchr/testify/assert"
)

func TestSomething(t *testing.T) {
  assert := assert.New(t)

  var a string = "Hello"
  var b string = "Hello"

  assert.Equal(a, b, "The two words should be the same.")
}
```

Assertions

Assertions allow you to easily write test code, and are global funcs in the `assert` package. All assertion functions take, as the first argument, the `*testing.T` object provided by the testing framework. This allows the assertion funcs to write the failings and other details to the correct place.

Every assertion function also takes an optional string message as the final argument, allowing custom error messages to be appended to the message the assertion method outputs.

Index

func Equal(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool
func EqualError(t TestingT, theError error, errString string, msgAndArgs ...interface{}) bool
func EqualErrorf(t TestingT, theError error, errString string, msg string, args ...interface{}) bool
func EqualValues(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool
func EqualValuesf(t TestingT, expected interface{}, actual interface{}, msg string, args ...interface{}) bool
func Equalf(t TestingT, expected interface{}, actual interface{}, msg string, args ...interface{}) bool
func Error(t TestingT, err error, msgAndArgs ...interface{}) bool
func Errorf(t TestingT, err error, msg string, args ...interface{}) bool
func Exactly(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool
func Exactlyf(t TestingT, expected interface{}, actual interface{}, msg string, args ...interface{}) bool
func Fail(t TestingT, failureMessage string, msgAndArgs ...interface{}) bool
func FailNow(t TestingT, failureMessage string, msgAndArgs ...interface{}) bool
func FailNowf(t TestingT, failureMessage string, msg string, args ...interface{}) bool
func Failf(t TestingT, failureMessage string, msg string, args ...interface{}) bool
func False(t TestingT, value bool, msgAndArgs ...interface{}) bool
func Falsef(t TestingT, value bool, msg string, args ...interface{}) bool
func FileExists(t TestingT, path string, msgAndArgs ...interface{}) bool
func FileExistsf(t TestingT, path string, msg string, args ...interface{}) bool
func Greater(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
func GreaterOrEqual(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
func GreaterOrEqualf(t TestingT, e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
func Greaterf(t TestingT, e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
func HTTPBody(handler http.HandlerFunc, method, url string, values url.Values) string
func HTTPBodyContains(t TestingT, handler http.HandlerFunc, method, url string, values url.Values, str interface{}, msgAndArgs ...interface{}) bool
func HTTPBodyContainsf(t TestingT, handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msg string, args ...interface{}) bool
func HTTPBodyNotContains(t TestingT, handler http.HandlerFunc, method, url string, values url.Values, str interface{}, msgAndArgs ...interface{}) bool
func HTTPBodyNotContainsf(t TestingT, handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msg string, args ...interface{}) bool
func HTTPError(t TestingT, handler http.HandlerFunc, method, url string, values url.Values, msgAndArgs ...interface{}) bool
func HTTPErrorf(t TestingT, handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
func HTTPRedirect(t TestingT, handler http.HandlerFunc, method, url string, values url.Values, msgAndArgs ...interface{}) bool
func HTTPRedirectf(t TestingT, handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
func HTTPSuccess(t TestingT, handler http.HandlerFunc, method, url string, values url.Values, msgAndArgs ...interface{}) bool
func HTTPSuccessf(t TestingT, handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
func Implements(t TestingT, interfaceObject interface{}, object interface{}, msgAndArgs ...interface{}) bool
func Implementsf(t TestingT, interfaceObject interface{}, object interface{}, msg string, args ...interface{}) bool
func InDelta(t TestingT, expected, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
func InDeltaMapValues(t TestingT, expected, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
func InDeltaMapValuesf(t TestingT, expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
func InDeltaSlice(t TestingT, expected, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
func InDeltaSlicef(t TestingT, expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
func InDeltaf(t TestingT, expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
func InEpsilon(t TestingT, expected, actual interface{}, epsilon float64, msgAndArgs ...interface{}) bool
func InEpsilonSlice(t TestingT, expected, actual interface{}, epsilon float64, msgAndArgs ...interface{}) bool
func InEpsilonSlicef(t TestingT, expected interface{}, actual interface{}, epsilon float64, msg string, args ...interface{}) bool
func InEpsilonf(t TestingT, expected interface{}, actual interface{}, epsilon float64, msg string, args ...interface{}) bool
func IsType(t TestingT, expectedType interface{}, object interface{}, msgAndArgs ...interface{}) bool
func IsTypef(t TestingT, expectedType interface{}, object interface{}, msg string, args ...interface{}) bool
func JSONEq(t TestingT, expected string, actual string, msgAndArgs ...interface{}) bool
func JSONEqf(t TestingT, expected string, actual string, msg string, args ...interface{}) bool

func Len(t TestingT, object interface{}, length int, msgAndArgs ...interface{}) bool
func Lenf(t TestingT, object interface{}, length int, msg string, args ...interface{}) bool
func Less(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
func LessOrEqual(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
func LessOrEqualf(t TestingT, e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
func Lessf(t TestingT, e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
func Nil(t TestingT, object interface{}, msgAndArgs ...interface{}) bool
func Nilf(t TestingT, object interface{}, msg string, args ...interface{}) bool
func NoError(t TestingT, err error, msgAndArgs ...interface{}) bool
func NoErrorf(t TestingT, err error, msg string, args ...interface{}) bool
func NotContains(t TestingT, s, contains interface{}, msgAndArgs ...interface{}) bool
func NotContainsf(t TestingT, s interface{}, contains interface{}, msg string, args ...interface{}) bool
func NotEmpty(t TestingT, object interface{}, msgAndArgs ...interface{}) bool
func NotEmptyf(t TestingT, object interface{}, msg string, args ...interface{}) bool
func NotEqual(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool
func NotEqualf(t TestingT, expected interface{}, actual interface{}, msg string, args ...interface{}) bool
func NotNil(t TestingT, object interface{}, msgAndArgs ...interface{}) bool
func NotNilf(t TestingT, object interface{}, msg string, args ...interface{}) bool
func NotPanics(t TestingT, f PanicTestFunc, msgAndArgs ...interface{}) bool
func NotPanicsf(t TestingT, f PanicTestFunc, msg string, args ...interface{}) bool
func NotRegexp(t TestingT, rx interface{}, str interface{}, msgAndArgs ...interface{}) bool
func NotRegexpf(t TestingT, rx interface{}, str interface{}, msg string, args ...interface{}) bool
func NotSubset(t TestingT, list, subset interface{}, msgAndArgs ...interface{}) (ok bool)
func NotSubsetf(t TestingT, list interface{}, subset interface{}, msg string, args ...interface{}) bool
func NotZero(t TestingT, i interface{}, msgAndArgs ...interface{}) bool
func NotZerof(t TestingT, i interface{}, msg string, args ...interface{}) bool
func ObjectsAreEqual(expected, actual interface{}) bool
func ObjectsAreEqualValues(expected, actual interface{}) bool
func Panics(t TestingT, f PanicTestFunc, msgAndArgs ...interface{}) bool
func PanicsWithValue(t TestingT, expected interface{}, f PanicTestFunc, msgAndArgs ...interface{}) bool
func PanicsWithValuef(t TestingT, expected interface{}, f PanicTestFunc, msg string, args ...interface{}) bool
func Panicsf(t TestingT, f PanicTestFunc, msg string, args ...interface{}) bool
func Regexp(t TestingT, rx interface{}, str interface{}, msgAndArgs ...interface{}) bool
func Regexpf(t TestingT, rx interface{}, str interface{}, msg string, args ...interface{}) bool
func Same(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool
func Samef(t TestingT, expected interface{}, actual interface{}, msg string, args ...interface{}) bool
func Subset(t TestingT, list, subset interface{}, msgAndArgs ...interface{}) (ok bool)
func Subsetf(t TestingT, list interface{}, subset interface{}, msg string, args ...interface{}) bool
func True(t TestingT, value bool, msgAndArgs ...interface{}) bool
func Truef(t TestingT, value bool, msg string, args ...interface{}) bool
func WithinDuration(t TestingT, expected, actual time.Time, delta time.Duration, msgAndArgs ...interface{}) bool
func WithinDurationf(t TestingT, expected time.Time, actual time.Time, delta time.Duration, msg string, args ...interface{}) bool
func Zero(t TestingT, i interface{}, msgAndArgs ...interface{}) bool
func Zerof(t TestingT, i interface{}, msg string, args ...interface{}) bool
type Assertions

- func New(t TestingT) *Assertions
- func (a *Assertions) Condition(comp Comparison, msgAndArgs ...interface{}) bool
- func (a *Assertions) Conditionf(comp Comparison, msg string, args ...interface{}) bool
- func (a *Assertions) Contains(s interface{}, contains interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Containsf(s interface{}, contains interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) DirExists(path string, msgAndArgs ...interface{}) bool
- func (a *Assertions) DirExistsf(path string, msg string, args ...interface{}) bool
- func (a *Assertions) ElementsMatch(listA interface{}, listB interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) ElementsMatchf(listA interface{}, listB interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Empty(object interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Emptyf(object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Equal(expected interface{}, actual interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) EqualError(theError error, errString string, msgAndArgs ...interface{}) bool
- func (a *Assertions) EqualErrorf(theError error, errString string, msg string, args ...interface{}) bool
- func (a *Assertions) EqualValues(expected interface{}, actual interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) EqualValuesf(expected interface{}, actual interface{}, msg string, args ...interface{}) bool

- func (a *Assertions) Equalf(expected interface{}, actual interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Error(err error, msgAndArgs ...interface{}) bool
- func (a *Assertions) Errorf(err error, msg string, args ...interface{}) bool
- func (a *Assertions) Exactly(expected interface{}, actual interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Exactlyf(expected interface{}, actual interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Fail(failureMessage string, msgAndArgs ...interface{}) bool
- func (a *Assertions) FailNow(failureMessage string, msgAndArgs ...interface{}) bool
- func (a *Assertions) FailNowf(failureMessage string, msg string, args ...interface{}) bool
- func (a *Assertions) Failf(failureMessage string, msg string, args ...interface{}) bool
- func (a *Assertions) False(value bool, msgAndArgs ...interface{}) bool
- func (a *Assertions) Falsef(value bool, msg string, args ...interface{}) bool
- func (a *Assertions) FileExists(path string, msgAndArgs ...interface{}) bool
- func (a *Assertions) FileExistsf(path string, msg string, args ...interface{}) bool
- func (a *Assertions) Greater(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) GreaterOrEqual(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) GreaterOrEqualf(e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Greaterf(e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) HTTPBodyContains(handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) HTTPBodyContainsf(handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) HTTPBodyNotContains(handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) HTTPBodyNotContainsf(handler http.HandlerFunc, method string, url string, values url.Values, str interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) HTTPError(handler http.HandlerFunc, method string, url string, values url.Values, msgAndArgs ...interface{}) bool
- func (a *Assertions) HTTPErrorf(handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
- func (a *Assertions) HTTPRedirect(handler http.HandlerFunc, method string, url string, values url.Values, msgAndArgs ...interface{}) bool
- func (a *Assertions) HTTPRedirectf(handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
- func (a *Assertions) HTTPSuccess(handler http.HandlerFunc, method string, url string, values url.Values, msgAndArgs ...interface{}) bool
- func (a *Assertions) HTTPSuccessf(handler http.HandlerFunc, method string, url string, values url.Values, msg string, args ...interface{}) bool
- func (a *Assertions) Implements(interfaceObject interface{}, object interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Implementsf(interfaceObject interface{}, object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) InDelta(expected interface{}, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
- func (a *Assertions) InDeltaMapValues(expected interface{}, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
- func (a *Assertions) InDeltaMapValuesf(expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
- func (a *Assertions) InDeltaSlice(expected interface{}, actual interface{}, delta float64, msgAndArgs ...interface{}) bool
- func (a *Assertions) InDeltaSlicef(expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
- func (a *Assertions) InDeltaf(expected interface{}, actual interface{}, delta float64, msg string, args ...interface{}) bool
- func (a *Assertions) InEpsilon(expected interface{}, actual interface{}, epsilon float64, msgAndArgs ...interface{}) bool
- func (a *Assertions) InEpsilonSlice(expected interface{}, actual interface{}, epsilon float64, msgAndArgs ...interface{}) bool
- func (a *Assertions) InEpsilonSlicef(expected interface{}, actual interface{}, epsilon float64, msg string, args ...interface{}) bool
- func (a *Assertions) InEpsilonf(expected interface{}, actual interface{}, epsilon float64, msg string, args ...interface{}) bool
- func (a *Assertions) IsType(expectedType interface{}, object interface{}, msgAndArgs ...interface{}) bool

- func (a *Assertions) IsTypef(expectedType interface{}, object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) JSONEq(expected string, actual string, msgAndArgs ...interface{}) bool
- func (a *Assertions) JSONEqf(expected string, actual string, msg string, args ...interface{}) bool
- func (a *Assertions) Len(object interface{}, length int, msgAndArgs ...interface{}) bool
- func (a *Assertions) Lenf(object interface{}, length int, msg string, args ...interface{}) bool
- func (a *Assertions) Less(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) LessOrEqual(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) LessOrEqualf(e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Lessf(e1 interface{}, e2 interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Nil(object interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Nilf(object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NoError(err error, msgAndArgs ...interface{}) bool
- func (a *Assertions) NoErrorf(err error, msg string, args ...interface{}) bool
- func (a *Assertions) NotContains(s interface{}, contains interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotContainsf(s interface{}, contains interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotEmpty(object interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotEmptyf(object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotEqual(expected interface{}, actual interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotEqualf(expected interface{}, actual interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotNil(object interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotNilf(object interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotPanics(f PanicTestFunc, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotPanicsf(f PanicTestFunc, msg string, args ...interface{}) bool
- func (a *Assertions) NotRegexp(rx interface{}, str interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotRegexpf(rx interface{}, str interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotSubset(list interface{}, subset interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotSubsetf(list interface{}, subset interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) NotZero(i interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) NotZerof(i interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Panics(f PanicTestFunc, msgAndArgs ...interface{}) bool
- func (a *Assertions) PanicsWithValue(expected interface{}, f PanicTestFunc, msgAndArgs ...interface{}) bool
- func (a *Assertions) PanicsWithValuef(expected interface{}, f PanicTestFunc, msg string, args ...interface{}) bool
- func (a *Assertions) Panicsf(f PanicTestFunc, msg string, args ...interface{}) bool
- func (a *Assertions) Regexp(rx interface{}, str interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Regexpf(rx interface{}, str interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Same(expected interface{}, actual interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Samef(expected interface{}, actual interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) Subset(list interface{}, subset interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Subsetf(list interface{}, subset interface{}, msg string, args ...interface{}) bool
- func (a *Assertions) True(value bool, msgAndArgs ...interface{}) bool
- func (a *Assertions) Truef(value bool, msg string, args ...interface{}) bool
- func (a *Assertions) WithinDuration(expected time.Time, actual time.Time, delta time.Duration, msgAndArgs ...interface{}) bool
- func (a *Assertions) WithinDurationf(expected time.Time, actual time.Time, delta time.Duration, msg string, args ...interface{}) bool
- func (a *Assertions) Zero(i interface{}, msgAndArgs ...interface{}) bool
- func (a *Assertions) Zerof(i interface{}, msg string, args ...interface{}) bool

type BoolAssertionFunc
type Comparison
type ComparisonAssertionFunc
type ErrorAssertionFunc
type PanicTestFunc
type TestingT
type ValueAssertionFunc

Examples

BoolAssertionFunc
ComparisonAssertionFunc
ErrorAssertionFunc

ValueAssertionFunc

Package Files (https://github.com/stretchr/testify/tree/master/assert)

assertion_format.go (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go)
assertion_forward.go (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go)
assertion_order.go (https://github.com/stretchr/testify/blob/master/assert/assertion_order.go) assertions.go
(https://github.com/stretchr/testify/blob/master/assert/assertions.go) doc.go
(https://github.com/stretchr/testify/blob/master/assert/doc.go) errors.go
(https://github.com/stretchr/testify/blob/master/assert/errors.go) forward_assertions.go
(https://github.com/stretchr/testify/blob/master/assert/forward_assertions.go) http_assertions.go
(https://github.com/stretchr/testify/blob/master/assert/http_assertions.go)

Variables

```
var AnError = errors (/errors).New (/errors#New)("assert.AnError general error for testing")
```

AnError is an error instance useful for testing. If the code does not care about error specifics, and only needs to return the error for example, this error should be used to make the test code more readable.

func CallerInfo (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L103)

```
func CallerInfo() []string (/builtin#string)
```

CallerInfo returns an array of strings containing the file and line number of each stack frame leading from the current test to the assert call that failed.

func Condition (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L887)

```
func Condition(t TestingT, comp Comparison, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Condition uses a Comparison to assert a complex condition.

func Conditionf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Conditionf(t TestingT, comp Comparison, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

Conditionf uses a Comparison to assert a complex condition.

func Contains (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L701)

```
func Contains(t TestingT, s, contains interface{}, msgAndArgs ...interface{}) bool (/builtin#b
ool)
```

Contains asserts that the specified string, list(array, slice...) or map contains the specified substring or element.

```
assert.Contains(t, "Hello World", "World")
assert.Contains(t, ["Hello", "World"], "World")
assert.Contains(t, {"Hello": "World"}, "Hello")
```

func Containsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Containsf(t TestingT, s interface{}, contains interface{}, msg string (/builtin#string),
 args ...interface{}) bool (/builtin#bool)
```

Containsf asserts that the specified string, list(array, slice...) or map contains the specified substring or element.

```
assert.Containsf(t, "Hello World", "World", "error message %s", "formatted")
assert.Containsf(t, ["Hello", "World"], "World", "error message %s", "formatted")
assert.Containsf(t, {"Hello": "World"}, "Hello", "error message %s", "formatted")
```

func DirExists (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1335

```
func DirExists(t TestingT, path string (/builtin#string), msgAndArgs ...interface{}) bool (/bu
iltin#bool)
```

DirExists checks whether a directory exists in the given path. It also fails if the path is a file rather a directory or there is an error checking whether it exists.

func DirExistsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func DirExistsf(t TestingT, path string (/builtin#string), msg string (/builtin#string), args
 ...interface{}) bool (/builtin#bool)
```

DirExistsf checks whether a directory exists in the given path. It also fails if the path is a file rather a directory or there is an error checking whether it exists.

func ElementsMatch (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L834)

```
func ElementsMatch(t TestingT, listA, listB interface{}, msgAndArgs ...interface{}) (ok bool
 (/builtin#bool))
```

ElementsMatch asserts that the specified listA(array, slice...) is equal to specified listB(array, slice...) ignoring the order of the elements. If there are duplicate elements, the number of appearances of each of them in both lists should match.

assert.ElementsMatch(t, [1, 3, 2, 3], [1, 3, 3, 2])

func ElementsMatchf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func ElementsMatchf(t TestingT, listA interface{}, listB interface{}, msg string (/builtin#str
ing), args ...interface{}) bool (/builtin#bool)
```

ElementsMatchf asserts that the specified listA(array, slice...) is equal to specified listB(array, slice...) ignoring the order of the elements. If there are duplicate elements, the number of appearances of each of them in both lists should match.

assert.ElementsMatchf(t, [1, 3, 2, 3], [1, 3, 3, 2], "error message %s", "formatted")

func Empty (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L531)

```
func Empty(t TestingT, object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Empty asserts that the specified object is empty. I.e. nil, "", false, 0 or either a slice or a channel with len == 0.

```
assert.Empty(t, obj)
```

func Emptyf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Emptyf(t TestingT, object interface{}, msg string (/builtin#string), args ...interface{})
bool (/builtin#bool)
```

Emptyf asserts that the specified object is empty. I.e. nil, "", false, 0 or either a slice or a channel with len == 0.

```
assert.Emptyf(t, obj, "error message %s", "formatted")
```

func Equal (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L332)

```
func Equal(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool (/builtin
#bool)
```

Equal asserts that two objects are equal.

```
assert.Equal(t, 123, 123)
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses). Function equality cannot be determined and will always fail.

func EqualError (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1226

```
func EqualError(t TestingT, theError error (/builtin#error), errString string (/builtin#strin
g), msgAndArgs ...interface{}) bool (/builtin#bool)
```

EqualError asserts that a function returned an error (i.e. not `nil`) and that it is equal to the provided error.

```
actualObj, err := SomeFunction()
assert.EqualError(t, err,  expectedErrorString)
```

func EqualErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func EqualErrorf(t TestingT, theError error (/builtin#error), errString string (/builtin#strin
g), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

EqualErrorf asserts that a function returned an error (i.e. not `nil`) and that it is equal to the provided error.

```
actualObj, err := SomeFunction()
assert.EqualErrorf(t, err,  expectedErrorString, "error message %s", "formatted")
```

func EqualValues (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L404)

```
func EqualValues(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool (/b
uiltin#bool)
```

EqualValues asserts that two objects are equal or convertable to the same types and equal.

```
assert.EqualValues(t, uint32(123), int32(123))
```

func EqualValuesf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func EqualValuesf(t TestingT, expected interface{}, actual interface{}, msg string (/builtin#s
tring), args ...interface{}) bool (/builtin#bool)
```

EqualValuesf asserts that two objects are equal or convertable to the same types and equal.

```
assert.EqualValuesf(t, uint32(123, "error message %s", "formatted"), int32(123))
```

func Equalf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Equalf(t TestingT, expected interface{}, actual interface{}, msg string (/builtin#string)
, args ...interface{}) bool (/builtin#bool)
```

Equalf asserts that two objects are equal.

```
assert.Equalf(t, 123, 123, "error message %s", "formatted")
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses). Function equality cannot be determined and will always fail.

func Error (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1209

```
func Error(t TestingT, err error (/builtin#error), msgAndArgs ...interface{}) bool (/builtin#b
ool)
```

Error asserts that a function returned an error (i.e. not `nil`).

```
  actualObj, err := SomeFunction()
  if assert.Error(t, err) {
          assert.Equal(t, expectedError, err)
  }
```

func Errorf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Errorf(t TestingT, err error (/builtin#error), msg string (/builtin#string), args ...inte
rface{}) bool (/builtin#bool)
```

Errorf asserts that a function returned an error (i.e. not `nil`).

```
  actualObj, err := SomeFunction()
  if assert.Errorf(t, err, "error message %s", "formatted") {
          assert.Equal(t, expectedErrorf, err)
  }
```

func Exactly (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L424)

```
func Exactly(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool (/built
in#bool)
```

Exactly asserts that two objects are equal in value and type.

```
assert.Exactly(t, int32(123), int64(123))
```

func Exactlyf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Exactlyf(t TestingT, expected interface{}, actual interface{}, msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

Exactlyf asserts that two objects are equal in value and type.

```
assert.Exactlyf(t, int32(123, "error message %s", "formatted"), int64(123))
```

func Fail (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L239)

```
func Fail(t TestingT, failureMessage string (/builtin#string), msgAndArgs ...interface{}) bool
(/builtin#bool)
```

Fail reports a failure through

func FailNow (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L218)

```
func FailNow(t TestingT, failureMessage string (/builtin#string), msgAndArgs ...interface{}) b
ool (/builtin#bool)
```

FailNow fails test

func FailNowf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func FailNowf(t TestingT, failureMessage string (/builtin#string), msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

FailNowf fails test

func Failf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Failf(t TestingT, failureMessage string (/builtin#string), msg string (/builtin#string),
 args ...interface{}) bool (/builtin#bool)
```

Failf reports a failure through

func False (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L620)

```
func False(t TestingT, value bool (/builtin#bool), msgAndArgs ...interface{}) bool (/builtin#b
ool)
```

False asserts that the specified value is false.

```
assert.False(t, myBool)
```

func Falsef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Falsef(t TestingT, value bool (/builtin#bool), msg string (/builtin#string), args ...inte
rface{}) bool (/builtin#bool)
```

Falsef asserts that the specified value is false.

```
assert.Falsef(t, myBool, "error message %s", "formatted")
```

func FileExists (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1317

```
func FileExists(t TestingT, path string (/builtin#string), msgAndArgs ...interface{}) bool (/b
uiltin#bool)
```

FileExists checks whether a file exists in the given path. It also fails if the path points to a directory or there is an error when trying to check the file.

func FileExistsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func FileExistsf(t TestingT, path string (/builtin#string), msg string (/builtin#string), args
...interface{}) bool (/builtin#bool)
```

FileExistsf checks whether a file exists in the given path. It also fails if the path points to a directory or there is an error when trying to check the file.

func Greater (https://github.com/stretchr/testify/blob/master/assert/assertion_order.go#l

```
func Greater(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool (/bui
ltin#bool)
```

Greater asserts that the first element is greater than the second

```
assert.Greater(t, 2, 1)
assert.Greater(t, float64(2), float64(1))
assert.Greater(t, "b", "a")
```

func GreaterOrEqual (https://github.com/stretchr/testify/blob/master/assert/assertion_order.go#l

```
func GreaterOrEqual(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) boo
l (/builtin#bool)
```

GreaterOrEqual asserts that the first element in greater or equal than the second

```
assert.GreaterOrEqual(t, 2, 1)
assert.GreaterOrEqual(t, 2, 2)
assert.GreaterOrEqual(t, "b", "a")
assert.GreaterOrEqual(t, "b", "b")
```

func GreaterOrEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func GreaterOrEqualf(t TestingT, e1 interface{}, e2 interface{}, msg string (/builtin#string),
args ...interface{}) bool (/builtin#bool)
```

GreaterOrEqualf asserts that the first element in greater or equal than the second

```
assert.GreaterOrEqualf(t, 2, 1, "error message %s", "formatted")
assert.GreaterOrEqualf(t, 2, 2, "error message %s", "formatted")
assert.GreaterOrEqualf(t, "b", "a", "error message %s", "formatted")
assert.GreaterOrEqualf(t, "b", "b", "error message %s", "formatted")
```

func Greaterf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Greaterf(t TestingT, e1 interface{}, e2 interface{}, msg string (/builtin#string), args
 ...interface{}) bool (/builtin#bool)
```

Greaterf asserts that the first element is greater than the second

```
assert.Greaterf(t, 2, 1, "error message %s", "formatted")
assert.Greaterf(t, float64(2, "error message %s", "formatted"), float64(1))
assert.Greaterf(t, "b", "a", "error message %s", "formatted")
```

func HTTPBody (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPBody(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), method, url string
(/builtin#string), values url (/net/url).Values (/net/url#Values)) string (/builtin#string)
```

HTTPBody is a helper that returns HTTP body of the response. It returns empty string if building a new request fails.

func HTTPBodyContains (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPBodyContains(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc)
, method, url string (/builtin#string), values url (/net/url).Values (/net/url#Values), str in
terface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPBodyContains asserts that a specified handler returns a body that contains a string.

```
assert.HTTPBodyContains(t, myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky")
```

Returns whether the assertion was successful (true) or not (false).

func HTTPBodyContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func HTTPBodyContainsf(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFun
c), method string (/builtin#string), url string (/builtin#string), values url (/net/url).Value
s (/net/url#Values), str interface{}, msg string (/builtin#string), args ...interface{}) bool
 (/builtin#bool)
```

HTTPBodyContainsf asserts that a specified handler returns a body that contains a string.

```
assert.HTTPBodyContainsf(t, myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky", "error
```

Returns whether the assertion was successful (true) or not (false).

func HTTPBodyNotContains (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPBodyNotContains(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFu
nc), method, url string (/builtin#string), values url (/net/url).Values (/net/url#Values), str
interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPBodyNotContains asserts that a specified handler returns a body that does not contain a string.

```
assert.HTTPBodyNotContains(t, myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky")
```

Returns whether the assertion was successful (true) or not (false).

func HTTPBodyNotContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func HTTPBodyNotContainsf(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerF
unc), method string (/builtin#string), url string (/builtin#string), values url (/net/url).Val
ues (/net/url#Values), str interface{}, msg string (/builtin#string), args ...interface{}) boo
l (/builtin#bool)
```

HTTPBodyNotContainsf asserts that a specified handler returns a body that does not contain a string.

```
assert.HTTPBodyNotContainsf(t, myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky", "er
```

Returns whether the assertion was successful (true) or not (false).

func HTTPError (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPError(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), metho
d, url string (/builtin#string), values url (/net/url).Values (/net/url#Values), msgAndArgs
 ...interface{}) bool (/builtin#bool)
```

HTTPError asserts that a specified handler returns an error status code.

```
assert.HTTPError(t, myHandler, "POST", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true) or not (false).

func HTTPErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func HTTPErrorf(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), meth
od string (/builtin#string), url string (/builtin#string), values url (/net/url).Values (/net/
url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPErrorf asserts that a specified handler returns an error status code.

```
assert.HTTPErrorf(t, myHandler, "POST", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true, "error message %s", "formatted") or not (false).

func HTTPRedirect (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPRedirect(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), me
thod, url string (/builtin#string), values url (/net/url).Values (/net/url#Values), msgAndArgs
...interface{}) bool (/builtin#bool)
```

HTTPRedirect asserts that a specified handler returns a redirect status code.

```
assert.HTTPRedirect(t, myHandler, "GET", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true) or not (false).

func HTTPRedirectf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func HTTPRedirectf(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), m
ethod string (/builtin#string), url string (/builtin#string), values url (/net/url).Values (/n
et/url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPRedirectf asserts that a specified handler returns a redirect status code.

```
assert.HTTPRedirectf(t, myHandler, "GET", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true, "error message %s", "formatted") or not (false).

func HTTPSuccess (https://github.com/stretchr/testify/blob/master/assert/http_assertions.go#L

```
func HTTPSuccess(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), met
hod, url string (/builtin#string), values url (/net/url).Values (/net/url#Values), msgAndArgs
 ...interface{}) bool (/builtin#bool)
```

HTTPSuccess asserts that a specified handler returns a success status code.

```
assert.HTTPSuccess(t, myHandler, "POST", "http://www.google.com (http://www.google.com)", nil)
```

Returns whether the assertion was successful (true) or not (false).

func HTTPSuccessf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func HTTPSuccessf(t TestingT, handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), me
thod string (/builtin#string), url string (/builtin#string), values url (/net/url).Values (/ne
t/url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPSuccessf asserts that a specified handler returns a success status code.

```
assert.HTTPSuccessf(t, myHandler, "POST", "http://www.google.com (http://www.google.com)", nil,
```

Returns whether the assertion was successful (true) or not (false).

func Implements (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L296)

```
func Implements(t TestingT, interfaceObject interface{}, object interface{}, msgAndArgs ...int
erface{}) bool (/builtin#bool)
```

Implements asserts that an object is implemented by the specified interface.

```
assert.Implements(t, (*MyInterface)(nil), new(MyObject))
```

func Implementsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Implementsf(t TestingT, interfaceObject interface{}, object interface{}, msg string (/bui
ltin#string), args ...interface{}) bool (/builtin#bool)
```

Implementsf asserts that an object is implemented by the specified interface.

```
assert.Implementsf(t, (*MyInterface, "error message %s", "formatted")(nil), new(MyObject))
```

func InDelta (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L102

```
func InDelta(t TestingT, expected, actual interface{}, delta float64 (/builtin#float64), msgAn
dArgs ...interface{}) bool (/builtin#bool)
```

InDelta asserts that the two numerals are within delta of each other.

```
assert.InDelta(t, math.Pi, (22 / 7.0), 0.01)
```

func InDeltaMapValues (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1082

```
func InDeltaMapValues(t TestingT, expected, actual interface{}, delta float64 (/builtin#float6
4), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InDeltaMapValues is the same as InDelta, but it compares all values between two maps. Both maps must have exactly the same keys.

func InDeltaMapValuesf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func InDeltaMapValuesf(t TestingT, expected interface{}, actual interface{}, delta float64 (/b
uiltin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaMapValuesf is the same as InDelta, but it compares all values between two maps. Both maps must have exactly the same keys.

func InDeltaSlice (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1058

```
func InDeltaSlice(t TestingT, expected, actual interface{}, delta float64 (/builtin#float64),
 msgAndArgs ...interface{}) bool (/builtin#bool)
```

InDeltaSlice is the same as InDelta, except it compares two slices.

func InDeltaSlicef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func InDeltaSlicef(t TestingT, expected interface{}, actual interface{}, delta float64 (/built
in#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaSlicef is the same as InDelta, except it compares two slices.

func InDeltaf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func InDeltaf(t TestingT, expected interface{}, actual interface{}, delta float64 (/builtin#fl
oat64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaf asserts that the two numerals are within delta of each other.

```
assert.InDeltaf(t, math.Pi, (22 / 7.0, "error message %s", "formatted"), 0.01)
```

func InEpsilon (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1142

```
func InEpsilon(t TestingT, expected, actual interface{}, epsilon float64 (/builtin#float64), m
sgAndArgs ...interface{}) bool (/builtin#bool)
```

InEpsilon asserts that expected and actual have a relative error less than epsilon

func InEpsilonSlice (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1159

```
func InEpsilonSlice(t TestingT, expected, actual interface{}, epsilon float64 (/builtin#float6
4), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InEpsilonSlice is the same as InEpsilon, except it compares each value from two slices.

func InEpsilonSlicef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func InEpsilonSlicef(t TestingT, expected interface{}, actual interface{}, epsilon float64 (/b
uiltin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InEpsilonSlicef is the same as InEpsilon, except it compares each value from two slices.

func InEpsilonf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func InEpsilonf(t TestingT, expected interface{}, actual interface{}, epsilon float64 (/builti
n#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InEpsilonf asserts that expected and actual have a relative error less than epsilon

func IsType (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L313)

```
func IsType(t TestingT, expectedType interface{}, object interface{}, msgAndArgs ...interface
{}) bool (/builtin#bool)
```

IsType asserts that the specified objects are of the same type.

func IsTypef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func IsTypef(t TestingT, expectedType interface{}, object interface{}, msg string (/builtin#st
ring), args ...interface{}) bool (/builtin#bool)
```

IsTypef asserts that the specified objects are of the same type.

func JSONEq (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1355

```
func JSONEq(t TestingT, expected string (/builtin#string), actual string (/builtin#string), ms
gAndArgs ...interface{}) bool (/builtin#bool)
```

JSONEq asserts that two JSON strings are equivalent.

```
assert.JSONEq(t, `{"hello": "world", "foo": "bar"}`, `{"foo": "bar", "hello": "world"}`)
```

func JSONEqf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func JSONEqf(t TestingT, expected string (/builtin#string), actual string (/builtin#string), m
sg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

JSONEqf asserts that two JSON strings are equivalent.

```
assert.JSONEqf(t, `{"hello": "world", "foo": "bar"}`, `{"foo": "bar", "hello": "world"}`, "error
```

func Len (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L581)

```
func Len(t TestingT, object interface{}, length int (/builtin#int), msgAndArgs ...interface{})
bool (/builtin#bool)
```

Len asserts that the specified object has specific length. Len also fails if the object has a type that len() not accept.

```
assert.Len(t, mySlice, 3)
```

func Lenf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Lenf(t TestingT, object interface{}, length int (/builtin#int), msg string (/builtin#stri
ng), args ...interface{}) bool (/builtin#bool)
```

Lenf asserts that the specified object has specific length. Lenf also fails if the object has a type that len() not accept.

```
assert.Lenf(t, mySlice, 3, "error message %s", "formatted")
```

func Less (https://github.com/stretchr/testify/blob/master/assert/assertion_order.go#L

```
func Less(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool (/builti
n#bool)
```

Less asserts that the first element in less than the second

```
assert.Less(t, 1, 2)
assert.Less(t, float64(1), float64(2))
assert.Less(t, "a", "b")
```

func LessOrEqual (https://github.com/stretchr/testify/blob/master/assert/assertion_order.go#L

```
func LessOrEqual(t TestingT, e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
 (/builtin#bool)
```

LessOrEqual asserts that the first element in greater or equal than the second

```
assert.LessOrEqual(t, 1, 2)
assert.LessOrEqual(t, 2, 2)
assert.LessOrEqual(t, "a", "b")
assert.LessOrEqual(t, "b", "b")
```

func LessOrEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func LessOrEqualf(t TestingT, e1 interface{}, e2 interface{}, msg string (/builtin#string), ar
gs ...interface{}) bool (/builtin#bool)
```

LessOrEqualf asserts that the first element in greater or equal than the second

```
assert.LessOrEqualf(t, 1, 2, "error message %s", "formatted")
assert.LessOrEqualf(t, 2, 2, "error message %s", "formatted")
assert.LessOrEqualf(t, "a", "b", "error message %s", "formatted")
assert.LessOrEqualf(t, "b", "b", "error message %s", "formatted")
```

func Lessf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Lessf(t TestingT, e1 interface{}, e2 interface{}, msg string (/builtin#string), args ...i
nterface{}) bool (/builtin#bool)
```

Lessf asserts that the first element in less than the second

```
assert.Lessf(t, 1, 2, "error message %s", "formatted")
assert.Lessf(t, float64(1, "error message %s", "formatted"), float64(2))
assert.Lessf(t, "a", "b", "error message %s", "formatted")
```

func Nil (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L489)

```
func Nil(t TestingT, object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Nil asserts that the specified object is nil.

```
assert.Nil(t, err)
```

func Nilf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Nilf(t TestingT, object interface{}, msg string (/builtin#string), args ...interface{}) b
ool (/builtin#bool)
```

Nilf asserts that the specified object is nil.

```
assert.Nilf(t, err, "error message %s", "formatted")
```

func NoError (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1192

```
func NoError(t TestingT, err error (/builtin#error), msgAndArgs ...interface{}) bool (/builtin
#bool)
```

NoError asserts that a function returned no error (i.e. `nil`).

```
  actualObj, err := SomeFunction()
  if assert.NoError(t, err) {
          assert.Equal(t, expectedObj, actualObj)
  }
```

func NoErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NoErrorf(t TestingT, err error (/builtin#error), msg string (/builtin#string), args ...in
terface{}) bool (/builtin#bool)
```

NoErrorf asserts that a function returned no error (i.e. `nil`).

```
  actualObj, err := SomeFunction()
  if assert.NoErrorf(t, err, "error message %s", "formatted") {
          assert.Equal(t, expectedObj, actualObj)
  }
```

func NotContains (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L724)

```
func NotContains(t TestingT, s, contains interface{}, msgAndArgs ...interface{}) bool (/builti
n#bool)
```

NotContains asserts that the specified string, list(array, slice...) or map does NOT contain the specified substring or element.

```
assert.NotContains(t, "Hello World", "Earth")
assert.NotContains(t, ["Hello", "World"], "Earth")
assert.NotContains(t, {"Hello": "World"}, "Earth")
```

func NotContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotContainsf(t TestingT, s interface{}, contains interface{}, msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

NotContainsf asserts that the specified string, list(array, slice...) or map does NOT contain the specified substring or element.

```
assert.NotContainsf(t, "Hello World", "Earth", "error message %s", "formatted")
assert.NotContainsf(t, ["Hello", "World"], "Earth", "error message %s", "formatted")
assert.NotContainsf(t, {"Hello": "World"}, "Earth", "error message %s", "formatted")
```

func NotEmpty (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L551)

```
func NotEmpty(t TestingT, object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

NotEmpty asserts that the specified object is NOT empty. I.e. not nil, "", false, 0 or either a slice or a channel with len == 0.

```
if assert.NotEmpty(t, obj) {
  assert.Equal(t, "two", obj[1])
}
```

func NotEmptyf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotEmptyf(t TestingT, object interface{}, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

NotEmptyf asserts that the specified object is NOT empty. I.e. not nil, "", false, 0 or either a slice or a channel with len == 0.

```
if assert.NotEmptyf(t, obj, "error message %s", "formatted") {
  assert.Equal(t, "two", obj[1])
}
```

func NotEqual (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L639)

```
func NotEqual(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool (/buil
tin#bool)
```

NotEqual asserts that the specified values are NOT equal.

```
assert.NotEqual(t, obj1, obj2)
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses).

func NotEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotEqualf(t TestingT, expected interface{}, actual interface{}, msg string (/builtin#stri
ng), args ...interface{}) bool (/builtin#bool)
```

NotEqualf asserts that the specified values are NOT equal.

```
assert.NotEqualf(t, obj1, obj2, "error message %s", "formatted")
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses).

func NotNil (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L443)

```
func NotNil(t TestingT, object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

NotNil asserts that the specified object is not nil.

```
assert.NotNil(t, err)
```

func NotNilf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotNilf(t TestingT, object interface{}, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

NotNilf asserts that the specified object is not nil.

```
assert.NotNilf(t, err, "error message %s", "formatted")
```

func NotPanics (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L962)

```
func NotPanics(t TestingT, f PanicTestFunc, msgAndArgs ...interface{}) bool (/builtin#bool)
```

NotPanics asserts that the code inside the specified PanicTestFunc does NOT panic.

```
assert.NotPanics(t, func(){ RemainCalm() })
```

func NotPanicsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotPanicsf(t TestingT, f PanicTestFunc, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

NotPanicsf asserts that the code inside the specified PanicTestFunc does NOT panic.

```
assert.NotPanicsf(t, func(){ RemainCalm() }, "error message %s", "formatted")
```

func NotRegexp (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L128(

```
func NotRegexp(t TestingT, rx interface{}, str interface{}, msgAndArgs ...interface{}) bool (/
builtin#bool)
```

NotRegexp asserts that a specified regexp does not match a string.

```
assert.NotRegexp(t, regexp.MustCompile("starts"), "it's starting")
assert.NotRegexp(t, "^start", "it's not starting")
```

func NotRegexpf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotRegexpf(t TestingT, rx interface{}, str interface{}, msg string (/builtin#string), arg
s ...interface{}) bool (/builtin#bool)
```

NotRegexpf asserts that a specified regexp does not match a string.

```
assert.NotRegexpf(t, regexp.MustCompile("starts", "error message %s", "formatted"), "it's starti
assert.NotRegexpf(t, "^start", "it's not starting", "error message %s", "formatted")
```

func NotSubset (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L789)

```
func NotSubset(t TestingT, list, subset interface{}, msgAndArgs ...interface{}) (ok bool (/bui
ltin#bool))
```

NotSubset asserts that the specified list(array, slice...) contains not all elements given in the specified subset(array, slice...).

```
assert.NotSubset(t, [1, 3, 4], [1, 2], "But [1, 3, 4] does not contain [1, 2]")
```

func NotSubsetf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotSubsetf(t TestingT, list interface{}, subset interface{}, msg string (/builtin#string)
, args ...interface{}) bool (/builtin#bool)
```

NotSubsetf asserts that the specified list(array, slice...) contains not all elements given in the specified subset(array, slice...).

```
assert.NotSubsetf(t, [1, 3, 4], [1, 2], "But [1, 3, 4] does not contain [1, 2]", "error message
```

func NotZero (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1306

```
func NotZero(t TestingT, i interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

NotZero asserts that i is not the zero value for its type.

func NotZerof (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func NotZerof(t TestingT, i interface{}, msg string (/builtin#string), args ...interface{}) bo
ol (/builtin#bool)
```

NotZerof asserts that i is not the zero value for its type.

func ObjectsAreEqual (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L56)

```
func ObjectsAreEqual(expected, actual interface{}) bool (/builtin#bool)
```

ObjectsAreEqual determines if two objects are considered equal.

This function does no assertion of any kind.

func ObjectsAreEqualValues (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L78)

```
func ObjectsAreEqualValues(expected, actual interface{}) bool (/builtin#bool)
```

ObjectsAreEqualValues gets whether two objects are equal, or if their values are equal.

func Panics (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L927)

```
func Panics(t TestingT, f PanicTestFunc, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Panics asserts that the code inside the specified PanicTestFunc panics.

```
assert.Panics(t, func(){ GoCrazy() })
```

func PanicsWithValue (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L943)

```
func PanicsWithValue(t TestingT, expected interface{}, f PanicTestFunc, msgAndArgs ...interfac
e{}) bool (/builtin#bool)
```

PanicsWithValue asserts that the code inside the specified PanicTestFunc panics, and that the recovered panic value equals the expected panic value.

```
assert.PanicsWithValue(t, "crazy error", func(){ GoCrazy() })
```

func PanicsWithValuef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func PanicsWithValuef(t TestingT, expected interface{}, f PanicTestFunc, msg string (/builtin#
string), args ...interface{}) bool (/builtin#bool)
```

PanicsWithValuef asserts that the code inside the specified PanicTestFunc panics, and that the recovered panic value equals the expected panic value.

```
assert.PanicsWithValuef(t, "crazy error", func(){ GoCrazy() }, "error message %s", "formatted")
```

func Panicsf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Panicsf(t TestingT, f PanicTestFunc, msg string (/builtin#string), args ...interface{}) b
ool (/builtin#bool)
```

Panicsf asserts that the code inside the specified PanicTestFunc panics.

```
assert.Panicsf(t, func(){ GoCrazy() }, "error message %s", "formatted")
```

func Regexp (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L1262

```
func Regexp(t TestingT, rx interface{}, str interface{}, msgAndArgs ...interface{}) bool (/bui
ltin#bool)
```

Regexp asserts that a specified regexp matches a string.

```
assert.Regexp(t, regexp.MustCompile("start"), "it's starting")
assert.Regexp(t, "start...$", "it's not starting")
```

func Regexpf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Regexpf(t TestingT, rx interface{}, str interface{}, msg string (/builtin#string), args
 ...interface{}) bool (/builtin#bool)
```

Regexpf asserts that a specified regexp matches a string.

```
assert.Regexpf(t, regexp.MustCompile("start", "error message %s", "formatted"), "it's starting")
assert.Regexpf(t, "start...$", "it's not starting", "error message %s", "formatted")
```

func Same (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L359)

```
func Same(t TestingT, expected, actual interface{}, msgAndArgs ...interface{}) bool (/builtin#
bool)
```

Same asserts that two pointers reference the same object.

```
assert.Same(t, ptr1, ptr2)
```

Both arguments must be pointer variables. Pointer variable sameness is determined based on the equality of both type and value.

func Samef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Samef(t TestingT, expected interface{}, actual interface{}, msg string (/builtin#string),
args ...interface{}) bool (/builtin#bool)
```

Samef asserts that two pointers reference the same object.

```
assert.Samef(t, ptr1, ptr2, "error message %s", "formatted")
```

Both arguments must be pointer variables. Pointer variable sameness is determined based on the equality of both type and value.

func Subset (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L745)

```
func Subset(t TestingT, list, subset interface{}, msgAndArgs ...interface{}) (ok bool (/builti
n#bool))
```

Subset asserts that the specified list(array, slice...) contains all elements given in the specified subset(array, slice...).

```
assert.Subset(t, [1, 2, 3], [1, 2], "But [1, 2, 3] does contain [1, 2]")
```

func Subsetf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Subsetf(t TestingT, list interface{}, subset interface{}, msg string (/builtin#string), a
rgs ...interface{}) bool (/builtin#bool)
```

Subsetf asserts that the specified list(array, slice...) contains all elements given in the specified subset(array, slice...).

```
assert.Subsetf(t, [1, 2, 3], [1, 2], "But [1, 2, 3] does contain [1, 2]", "error message %s", "f
```

func True (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L599)

```
func True(t TestingT, value bool (/builtin#bool), msgAndArgs ...interface{}) bool (/builtin#bo
ol)
```

True asserts that the specified value is true.

```
assert.True(t, myBool)
```

func Truef (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Truef(t TestingT, value bool (/builtin#bool), msg string (/builtin#string), args ...inter
face{}) bool (/builtin#bool)
```

Truef asserts that the specified value is true.

```
assert.Truef(t, myBool, "error message %s", "formatted")
```

func WithinDuration (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L977)

```
func WithinDuration(t TestingT, expected, actual time (/time).Time (/time#Time), delta time (/
time).Duration (/time#Duration), msgAndArgs ...interface{}) bool (/builtin#bool)
```

WithinDuration asserts that the two times are within duration delta of each other.

```
assert.WithinDuration(t, time.Now(), time.Now(), 10*time.Second)
```

func WithinDurationf (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func WithinDurationf(t TestingT, expected time (/time).Time (/time#Time), actual time (/time).
Time (/time#Time), delta time (/time).Duration (/time#Duration), msg string (/builtin#string),
args ...interface{}) bool (/builtin#bool)
```

WithinDurationf asserts that the two times are within duration delta of each other.

```
assert.WithinDurationf(t, time.Now(), time.Now(), 10*time.Second, "error message %s", "formatted
```

func Zero (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L129

```
func Zero(t TestingT, i interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Zero asserts that i is the zero value for its type.

func Zerof (https://github.com/stretchr/testify/blob/master/assert/assertion_format.go#

```
func Zerof(t TestingT, i interface{}, msg string (/builtin#string), args ...interface{}) bool
 (/builtin#bool)
```

Zerof asserts that i is the zero value for its type.

type Assertions (https://github.com/stretchr/testify/blob/master/assert/forward_assertions.g

```
type Assertions struct {
    // contains filtered or unexported fields
}
```

Assertions provides assertion methods around the TestingT interface.

func New (https://github.com/stretchr/testify/blob/master/assert/forward_assertions.go#L10)

```
func New(t TestingT) *Assertions
```

New makes a new Assertions object for the specified TestingT.

func (*Assertions) Condition (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L15)

```
func (a *Assertions) Condition(comp Comparison, msgAndArgs ...interface{}) bool (/builtin#boo
l)
```

Condition uses a Comparison to assert a complex condition.

func (*Assertions) Conditionf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L23)

```
func (a *Assertions) Conditionf(comp Comparison, msg string (/builtin#string), args ...interfa
ce{}) bool (/builtin#bool)
```

Conditionf uses a Comparison to assert a complex condition.

func (*Assertions) Contains (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L36)

```
func (a *Assertions) Contains(s interface{}, contains interface{}, msgAndArgs ...interface{})
bool (/builtin#bool)
```

Contains asserts that the specified string, list(array, slice...) or map contains the specified substring or element.

```
a.Contains("Hello World", "World")
a.Contains(["Hello", "World"], "World")
a.Contains({"Hello": "World"}, "Hello")
```

func (*Assertions) Containsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L49)

```
func (a *Assertions) Containsf(s interface{}, contains interface{}, msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

Containsf asserts that the specified string, list(array, slice...) or map contains the specified substring or element.

```
a.Containsf("Hello World", "World", "error message %s", "formatted")
a.Containsf(["Hello", "World"], "World", "error message %s", "formatted")
a.Containsf({"Hello": "World"}, "Hello", "error message %s", "formatted")
```

func (*Assertions) DirExists (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L57)

```
func (a *Assertions) DirExists(path string (/builtin#string), msgAndArgs ...interface{}) bool
 (/builtin#bool)
```

DirExists checks whether a directory exists in the given path. It also fails if the path is a file rather a directory or there is an error checking whether it exists.

func (*Assertions) DirExistsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L65)

```
func (a *Assertions) DirExistsf(path string (/builtin#string), msg string (/builtin#string), a
rgs ...interface{}) bool (/builtin#bool)
```

DirExistsf checks whether a directory exists in the given path. It also fails if the path is a file rather a directory or there is an error checking whether it exists.

func (*Assertions) ElementsMatch (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L77)

```
func (a *Assertions) ElementsMatch(listA interface{}, listB interface{}, msgAndArgs ...interfa
ce{}) bool (/builtin#bool)
```

ElementsMatch asserts that the specified listA(array, slice...) is equal to specified listB(array, slice...) ignoring the order of the elements. If there are duplicate elements, the number of appearances of each of them in both lists should match.

a.ElementsMatch([1, 3, 2, 3], [1, 3, 3, 2])

func (*Assertions) ElementsMatchf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L89)

```
func (a *Assertions) ElementsMatchf(listA interface{}, listB interface{}, msg string (/builtin
#string), args ...interface{}) bool (/builtin#bool)
```

ElementsMatchf asserts that the specified listA(array, slice...) is equal to specified listB(array, slice...) ignoring the order of the elements. If there are duplicate elements, the number of appearances of each of them in both lists should match.

a.ElementsMatchf([1, 3, 2, 3], [1, 3, 3, 2], "error message %s", "formatted")

func (*Assertions) Empty (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L100)

```
func (a *Assertions) Empty(object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Empty asserts that the specified object is empty. I.e. nil, "", false, 0 or either a slice or a channel with len == 0.

```
a.Empty(obj)
```

func (*Assertions) Emptyf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L111)

```
func (a *Assertions) Emptyf(object interface{}, msg string (/builtin#string), args ...interfac
e{}) bool (/builtin#bool)
```

Emptyf asserts that the specified object is empty. I.e. nil, "", false, 0 or either a slice or a channel with len == 0.

```
a.Emptyf(obj, "error message %s", "formatted")
```

func (*Assertions) Equal (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L125)

```
func (a *Assertions) Equal(expected interface{}, actual interface{}, msgAndArgs ...interface
{}) bool (/builtin#bool)
```

Equal asserts that two objects are equal.

```
a.Equal(123, 123)
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses). Function equality cannot be determined and will always fail.

func (*Assertions) EqualError (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L137)

```
func (a *Assertions) EqualError(theError error (/builtin#error), errString string (/builtin#st
ring), msgAndArgs ...interface{}) bool (/builtin#bool)
```

EqualError asserts that a function returned an error (i.e. not `nil`) and that it is equal to the provided error.

```
actualObj, err := SomeFunction()
a.EqualError(err,  expectedErrorString)
```

func (*Assertions) EqualErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L149)

```
func (a *Assertions) EqualErrorf(theError error (/builtin#error), errString string (/builtin#s
tring), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

EqualErrorf asserts that a function returned an error (i.e. not `nil`) and that it is equal to the provided error.

```
actualObj, err := SomeFunction()
a.EqualErrorf(err,  expectedErrorString, "error message %s", "formatted")
```

func (*Assertions) EqualValues (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L160)

```
func (a *Assertions) EqualValues(expected interface{}, actual interface{}, msgAndArgs ...inter
face{}) bool (/builtin#bool)
```

EqualValues asserts that two objects are equal or convertable to the same types and equal.

```
a.EqualValues(uint32(123), int32(123))
```

func (*Assertions) EqualValuesf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L171)

```
func (a *Assertions) EqualValuesf(expected interface{}, actual interface{}, msg string (/built
in#string), args ...interface{}) bool (/builtin#bool)
```

EqualValuesf asserts that two objects are equal or convertable to the same types and equal.

```
a.EqualValuesf(uint32(123, "error message %s", "formatted"), int32(123))
```

func (*Assertions) Equalf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L185)

```
func (a *Assertions) Equalf(expected interface{}, actual interface{}, msg string (/builtin#str
ing), args ...interface{}) bool (/builtin#bool)
```

Equalf asserts that two objects are equal.

```
a.Equalf(123, 123, "error message %s", "formatted")
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses). Function equality cannot be determined and will always fail.

func (*Assertions) Error (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L198)

```
func (a *Assertions) Error(err error (/builtin#error), msgAndArgs ...interface{}) bool (/built
in#bool)
```

Error asserts that a function returned an error (i.e. not `nil`).

```
  actualObj, err := SomeFunction()
  if a.Error(err) {
          assert.Equal(t, expectedError, err)
  }
```

func (*Assertions) Errorf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L211)

```
func (a *Assertions) Errorf(err error (/builtin#error), msg string (/builtin#string), args ...
interface{}) bool (/builtin#bool)
```

Errorf asserts that a function returned an error (i.e. not `nil`).

```
  actualObj, err := SomeFunction()
  if a.Errorf(err, "error message %s", "formatted") {
          assert.Equal(t, expectedErrorf, err)
  }
```

func (*Assertions) Exactly (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L221)

```
func (a *Assertions) Exactly(expected interface{}, actual interface{}, msgAndArgs ...interface
{}) bool (/builtin#bool)
```

Exactly asserts that two objects are equal in value and type.

```
a.Exactly(int32(123), int64(123))
```

func (*Assertions) Exactlyf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L231)

```
func (a *Assertions) Exactlyf(expected interface{}, actual interface{}, msg string (/builtin#s
tring), args ...interface{}) bool (/builtin#bool)
```

Exactlyf asserts that two objects are equal in value and type.

```
a.Exactlyf(int32(123, "error message %s", "formatted"), int64(123))
```

func (*Assertions) Fail (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L239)

```
func (a *Assertions) Fail(failureMessage string (/builtin#string), msgAndArgs ...interface{})
bool (/builtin#bool)
```

Fail reports a failure through

func (*Assertions) FailNow (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L247)

```
func (a *Assertions) FailNow(failureMessage string (/builtin#string), msgAndArgs ...interface
{}) bool (/builtin#bool)
```

FailNow fails test

func (*Assertions) FailNowf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L255)

```
func (a *Assertions) FailNowf(failureMessage string (/builtin#string), msg string (/builtin#st
ring), args ...interface{}) bool (/builtin#bool)
```

FailNowf fails test

func (*Assertions) Failf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L263)

```
func (a *Assertions) Failf(failureMessage string (/builtin#string), msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

Failf reports a failure through

func (*Assertions) False (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L273)

```
func (a *Assertions) False(value bool (/builtin#bool), msgAndArgs ...interface{}) bool (/built
in#bool)
```

False asserts that the specified value is false.

```
a.False(myBool)
```

func (*Assertions) Falsef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L283)

```
func (a *Assertions) Falsef(value bool (/builtin#bool), msg string (/builtin#string), args ...
interface{}) bool (/builtin#bool)
```

Falsef asserts that the specified value is false.

```
a.Falsef(myBool, "error message %s", "formatted")
```

func (*Assertions) FileExists (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L291)

```
func (a *Assertions) FileExists(path string (/builtin#string), msgAndArgs ...interface{}) bool
(/builtin#bool)
```

FileExists checks whether a file exists in the given path. It also fails if the path points to a directory or there is an error when trying to check the file.

func (*Assertions) FileExistsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L299)

```
func (a *Assertions) FileExistsf(path string (/builtin#string), msg string (/builtin#string),
 args ...interface{}) bool (/builtin#bool)
```

FileExistsf checks whether a file exists in the given path. It also fails if the path points to a directory or there is an error when trying to check the file.

func (*Assertions) Greater (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L311)

```
func (a *Assertions) Greater(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool
 (/builtin#bool)
```

Greater asserts that the first element is greater than the second

```
a.Greater(2, 1)
a.Greater(float64(2), float64(1))
a.Greater("b", "a")
```

func (*Assertions) GreaterOrEqual (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L324)

```
func (a *Assertions) GreaterOrEqual(e1 interface{}, e2 interface{}, msgAndArgs ...interface{})
bool (/builtin#bool)
```

GreaterOrEqual asserts that the first element in greater or equal than the second

```
a.GreaterOrEqual(2, 1)
a.GreaterOrEqual(2, 2)
a.GreaterOrEqual("b", "a")
a.GreaterOrEqual("b", "b")
```

func (*Assertions) GreaterOrEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L337)

```
func (a *Assertions) GreaterOrEqualf(e1 interface{}, e2 interface{}, msg string (/builtin#stri
ng), args ...interface{}) bool (/builtin#bool)
```

GreaterOrEqualf asserts that the first element in greater or equal than the second

```
a.GreaterOrEqualf(2, 1, "error message %s", "formatted")
a.GreaterOrEqualf(2, 2, "error message %s", "formatted")
a.GreaterOrEqualf("b", "a", "error message %s", "formatted")
a.GreaterOrEqualf("b", "b", "error message %s", "formatted")
```

func (*Assertions) Greaterf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L349)

```
func (a *Assertions) Greaterf(e1 interface{}, e2 interface{}, msg string (/builtin#string), ar
gs ...interface{}) bool (/builtin#bool)
```

Greaterf asserts that the first element is greater than the second

```
a.Greaterf(2, 1, "error message %s", "formatted")
a.Greaterf(float64(2, "error message %s", "formatted"), float64(1))
a.Greaterf("b", "a", "error message %s", "formatted")
```

func (*Assertions) HTTPBodyContains (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L362)

```
func (a *Assertions) HTTPBodyContains(handler http (/net/http).HandlerFunc (/net/http#HandlerF
unc), method string (/builtin#string), url string (/builtin#string), values url (/net/url).Val
ues (/net/url#Values), str interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPBodyContains asserts that a specified handler returns a body that contains a string.

```
a.HTTPBodyContains(myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky")
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPBodyContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L375)

```
func (a *Assertions) HTTPBodyContainsf(handler http (/net/http).HandlerFunc (/net/http#Handler
Func), method string (/builtin#string), url string (/builtin#string), values url (/net/url).Va
lues (/net/url#Values), str interface{}, msg string (/builtin#string), args ...interface{}) bo
ol (/builtin#bool)
```

HTTPBodyContainsf asserts that a specified handler returns a body that contains a string.

```
a.HTTPBodyContainsf(myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky", "error message
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPBodyNotContains (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L388)

```
func (a *Assertions) HTTPBodyNotContains(handler http (/net/http).HandlerFunc (/net/http#Handl
erFunc), method string (/builtin#string), url string (/builtin#string), values url (/net/url).
Values (/net/url#Values), str interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPBodyNotContains asserts that a specified handler returns a body that does not contain a string.

```
a.HTTPBodyNotContains(myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky")
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPBodyNotContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L401)

```
func (a *Assertions) HTTPBodyNotContainsf(handler http (/net/http).HandlerFunc (/net/http#Hand
lerFunc), method string (/builtin#string), url string (/builtin#string), values url (/net/url)
.Values (/net/url#Values), str interface{}, msg string (/builtin#string), args ...interface{})
bool (/builtin#bool)
```

HTTPBodyNotContainsf asserts that a specified handler returns a body that does not contain a string.

```
a.HTTPBodyNotContainsf(myHandler, "GET", "www.google.com", nil, "I'm Feeling Lucky", "error mess
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPError (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L413)

```
func (a *Assertions) HTTPError(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc), m
ethod string (/builtin#string), url string (/builtin#string), values url (/net/url).Values (/n
et/url#Values), msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPError asserts that a specified handler returns an error status code.

```
a.HTTPError(myHandler, "POST", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L425)

```
func (a *Assertions) HTTPErrorf(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc),
 method string (/builtin#string), url string (/builtin#string), values url (/net/url).Values
 (/net/url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPErrorf asserts that a specified handler returns an error status code.

```
a.HTTPErrorf(myHandler, "POST", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true, "error message %s", "formatted") or not (false).

func (*Assertions) HTTPRedirect (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L437)

```
func (a *Assertions) HTTPRedirect(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc)
, method string (/builtin#string), url string (/builtin#string), values url (/net/url).Values
 (/net/url#Values), msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPRedirect asserts that a specified handler returns a redirect status code.

```
a.HTTPRedirect(myHandler, "GET", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPRedirectf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L449)

```
func (a *Assertions) HTTPRedirectf(handler http (/net/http).HandlerFunc (/net/http#HandlerFun
c), method string (/builtin#string), url string (/builtin#string), values url (/net/url).Value
s (/net/url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPRedirectf asserts that a specified handler returns a redirect status code.

```
a.HTTPRedirectf(myHandler, "GET", "/a/b/c", url.Values{"a": []string{"b", "c"}}
```

Returns whether the assertion was successful (true, "error message %s", "formatted") or not (false).

func (*Assertions) HTTPSuccess (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L461)

```
func (a *Assertions) HTTPSuccess(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc),
method string (/builtin#string), url string (/builtin#string), values url (/net/url).Values (/
net/url#Values), msgAndArgs ...interface{}) bool (/builtin#bool)
```

HTTPSuccess asserts that a specified handler returns a success status code.

```
a.HTTPSuccess(myHandler, "POST", "http://www.google.com (http://www.google.com)", nil)
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) HTTPSuccessf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L473)

```
func (a *Assertions) HTTPSuccessf(handler http (/net/http).HandlerFunc (/net/http#HandlerFunc)
, method string (/builtin#string), url string (/builtin#string), values url (/net/url).Values
 (/net/url#Values), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

HTTPSuccessf asserts that a specified handler returns a success status code.

```
a.HTTPSuccessf(myHandler, "POST", "http://www.google.com (http://www.google.com)", nil, "error m
```

Returns whether the assertion was successful (true) or not (false).

func (*Assertions) Implements (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L483)

```
func (a *Assertions) Implements(interfaceObject interface{}, object interface{}, msgAndArgs
 ...interface{}) bool (/builtin#bool)
```

Implements asserts that an object is implemented by the specified interface.

```
a.Implements((*MyInterface)(nil), new(MyObject))
```

func (*Assertions) Implementsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L493)

```
func (a *Assertions) Implementsf(interfaceObject interface{}, object interface{}, msg string
 (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

Implementsf asserts that an object is implemented by the specified interface.

```
a.Implementsf((*MyInterface, "error message %s", "formatted")(nil), new(MyObject))
```

func (*Assertions) InDelta (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L503)

```
func (a *Assertions) InDelta(expected interface{}, actual interface{}, delta float64 (/builtin
#float64), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InDelta asserts that the two numerals are within delta of each other.

```
a.InDelta(math.Pi, (22 / 7.0), 0.01)
```

func (*Assertions) InDeltaMapValues (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L511)

```
func (a *Assertions) InDeltaMapValues(expected interface{}, actual interface{}, delta float64
 (/builtin#float64), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InDeltaMapValues is the same as InDelta, but it compares all values between two maps. Both maps must have exactly the same keys.

func (*Assertions) InDeltaMapValuesf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L519)

```
func (a *Assertions) InDeltaMapValuesf(expected interface{}, actual interface{}, delta float64
(/builtin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaMapValuesf is the same as InDelta, but it compares all values between two maps. Both maps must have exactly the same keys.

func (*Assertions) InDeltaSlice (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L527)

```
func (a *Assertions) InDeltaSlice(expected interface{}, actual interface{}, delta float64 (/bu
iltin#float64), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InDeltaSlice is the same as InDelta, except it compares two slices.

func (*Assertions) InDeltaSlicef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L535)

```
func (a *Assertions) InDeltaSlicef(expected interface{}, actual interface{}, delta float64 (/b
uiltin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaSlicef is the same as InDelta, except it compares two slices.

func (*Assertions) InDeltaf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L545)

```
func (a *Assertions) InDeltaf(expected interface{}, actual interface{}, delta float64 (/builti
n#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InDeltaf asserts that the two numerals are within delta of each other.

```
a.InDeltaf(math.Pi, (22 / 7.0, "error message %s", "formatted"), 0.01)
```

func (*Assertions) InEpsilon (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L553)

```
func (a *Assertions) InEpsilon(expected interface{}, actual interface{}, epsilon float64 (/bui
ltin#float64), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InEpsilon asserts that expected and actual have a relative error less than epsilon

func (*Assertions) InEpsilonSlice (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L561)

```
func (a *Assertions) InEpsilonSlice(expected interface{}, actual interface{}, epsilon float64
 (/builtin#float64), msgAndArgs ...interface{}) bool (/builtin#bool)
```

InEpsilonSlice is the same as InEpsilon, except it compares each value from two slices.

func (*Assertions) InEpsilonSlicef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L569)

```
func (a *Assertions) InEpsilonSlicef(expected interface{}, actual interface{}, epsilon float64
(/builtin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InEpsilonSlicef is the same as InEpsilon, except it compares each value from two slices.

func (*Assertions) InEpsilonf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L577)

```
func (a *Assertions) InEpsilonf(expected interface{}, actual interface{}, epsilon float64 (/bu
iltin#float64), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

InEpsilonf asserts that expected and actual have a relative error less than epsilon

func (*Assertions) IsType (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L585)

```
func (a *Assertions) IsType(expectedType interface{}, object interface{}, msgAndArgs ...interf
ace{}) bool (/builtin#bool)
```

IsType asserts that the specified objects are of the same type.

func (*Assertions) IsTypef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L593)

```
func (a *Assertions) IsTypef(expectedType interface{}, object interface{}, msg string (/builti
n#string), args ...interface{}) bool (/builtin#bool)
```

IsTypef asserts that the specified objects are of the same type.

func (*Assertions) JSONEq (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L603)

```
func (a *Assertions) JSONEq(expected string (/builtin#string), actual string (/builtin#string)
, msgAndArgs ...interface{}) bool (/builtin#bool)
```

JSONEq asserts that two JSON strings are equivalent.

```
a.JSONEq(`{"hello": "world", "foo": "bar"}`, `{"foo": "bar", "hello": "world"}`)
```

func (*Assertions) JSONEqf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L613)

```
func (a *Assertions) JSONEqf(expected string (/builtin#string), actual string (/builtin#strin
g), msg string (/builtin#string), args ...interface{}) bool (/builtin#bool)
```

JSONEqf asserts that two JSON strings are equivalent.

```
a.JSONEqf(`{"hello": "world", "foo": "bar"}`, `{"foo": "bar", "hello": "world"}`, "error message
```

func (*Assertions) Len (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L624)

```
func (a *Assertions) Len(object interface{}, length int (/builtin#int), msgAndArgs ...interfac
e{}) bool (/builtin#bool)
```

Len asserts that the specified object has specific length. Len also fails if the object has a type that len() not accept.

```
a.Len(mySlice, 3)
```

func (*Assertions) Lenf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L635)

```
func (a *Assertions) Lenf(object interface{}, length int (/builtin#int), msg string (/builtin#
string), args ...interface{}) bool (/builtin#bool)
```

Lenf asserts that the specified object has specific length. Lenf also fails if the object has a type that len() not accept.

```
a.Lenf(mySlice, 3, "error message %s", "formatted")
```

func (*Assertions) Less (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L647)

```
func (a *Assertions) Less(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bool (/bu
iltin#bool)
```

Less asserts that the first element in less than the second

```
a.Less(1, 2)
a.Less(float64(1), float64(2))
a.Less("a", "b")
```

func (*Assertions) LessOrEqual (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L660)

```
func (a *Assertions) LessOrEqual(e1 interface{}, e2 interface{}, msgAndArgs ...interface{}) bo
ol (/builtin#bool)
```

LessOrEqual asserts that the first element in greater or equal than the second

```
a.LessOrEqual(1, 2)
a.LessOrEqual(2, 2)
a.LessOrEqual("a", "b")
a.LessOrEqual("b", "b")
```

func (*Assertions) LessOrEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L673)

```
func (a *Assertions) LessOrEqualf(e1 interface{}, e2 interface{}, msg string (/builtin#string)
, args ...interface{}) bool (/builtin#bool)
```

LessOrEqualf asserts that the first element in greater or equal than the second

```
a.LessOrEqualf(1, 2, "error message %s", "formatted")
a.LessOrEqualf(2, 2, "error message %s", "formatted")
a.LessOrEqualf("a", "b", "error message %s", "formatted")
a.LessOrEqualf("b", "b", "error message %s", "formatted")
```

func (*Assertions) Lessf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L685)

```
func (a *Assertions) Lessf(e1 interface{}, e2 interface{}, msg string (/builtin#string), args
 ...interface{}) bool (/builtin#bool)
```

Lessf asserts that the first element in less than the second

```
a.Lessf(1, 2, "error message %s", "formatted")
a.Lessf(float64(1, "error message %s", "formatted"), float64(2))
a.Lessf("a", "b", "error message %s", "formatted")
```

func (*Assertions) Nil (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L695)

```
func (a *Assertions) Nil(object interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Nil asserts that the specified object is nil.

```
a.Nil(err)
```

func (*Assertions) Nilf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L705)

```
func (a *Assertions) Nilf(object interface{}, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

Nilf asserts that the specified object is nil.

```
a.Nilf(err, "error message %s", "formatted")
```

func (*Assertions) NoError (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L718)

```
func (a *Assertions) NoError(err error (/builtin#error), msgAndArgs ...interface{}) bool (/bui
ltin#bool)
```

NoError asserts that a function returned no error (i.e. `nil`).

```
  actualObj, err := SomeFunction()
  if a.NoError(err) {
          assert.Equal(t, expectedObj, actualObj)
  }
```

func (*Assertions) NoErrorf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L731)

```
func (a *Assertions) NoErrorf(err error (/builtin#error), msg string (/builtin#string), args
 ...interface{}) bool (/builtin#bool)
```

NoErrorf asserts that a function returned no error (i.e. `nil`).

```
  actualObj, err := SomeFunction()
  if a.NoErrorf(err, "error message %s", "formatted") {
          assert.Equal(t, expectedObj, actualObj)
  }
```

func (*Assertions) NotContains (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L744)

```
func (a *Assertions) NotContains(s interface{}, contains interface{}, msgAndArgs ...interface
{}) bool (/builtin#bool)
```

NotContains asserts that the specified string, list(array, slice...) or map does NOT contain the specified substring or element.

```
a.NotContains("Hello World", "Earth")
a.NotContains(["Hello", "World"], "Earth")
a.NotContains({"Hello": "World"}, "Earth")
```

func (*Assertions) NotContainsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L757)

```
func (a *Assertions) NotContainsf(s interface{}, contains interface{}, msg string (/builtin#st
ring), args ...interface{}) bool (/builtin#bool)
```

NotContainsf asserts that the specified string, list(array, slice...) or map does NOT contain the specified substring or element.

```
a.NotContainsf("Hello World", "Earth", "error message %s", "formatted")
a.NotContainsf(["Hello", "World"], "Earth", "error message %s", "formatted")
a.NotContainsf({"Hello": "World"}, "Earth", "error message %s", "formatted")
```

func (*Assertions) NotEmpty (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L770)

```
func (a *Assertions) NotEmpty(object interface{}, msgAndArgs ...interface{}) bool (/builtin#bo
ol)
```

NotEmpty asserts that the specified object is NOT empty. I.e. not nil, "", false, 0 or either a slice or a channel with len == 0.

```
if a.NotEmpty(obj) {
  assert.Equal(t, "two", obj[1])
}
```

func (*Assertions) NotEmptyf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L783)

```
func (a *Assertions) NotEmptyf(object interface{}, msg string (/builtin#string), args ...inter
face{}) bool (/builtin#bool)
```

NotEmptyf asserts that the specified object is NOT empty. I.e. not nil, "", false, 0 or either a slice or a channel with len == 0.

```
if a.NotEmptyf(obj, "error message %s", "formatted") {
  assert.Equal(t, "two", obj[1])
}
```

func (*Assertions) NotEqual (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L796)

```
func (a *Assertions) NotEqual(expected interface{}, actual interface{}, msgAndArgs ...interfac
e{}) bool (/builtin#bool)
```

NotEqual asserts that the specified values are NOT equal.

```
a.NotEqual(obj1, obj2)
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses).

func (*Assertions) NotEqualf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L809)

```
func (a *Assertions) NotEqualf(expected interface{}, actual interface{}, msg string (/builtin#
string), args ...interface{}) bool (/builtin#bool)
```

NotEqualf asserts that the specified values are NOT equal.

```
a.NotEqualf(obj1, obj2, "error message %s", "formatted")
```

Pointer variable equality is determined based on the equality of the referenced values (as opposed to the memory addresses).

func (*Assertions) NotNil (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L819)

```
func (a *Assertions) NotNil(object interface{}, msgAndArgs ...interface{}) bool (/builtin#boo
l)
```

NotNil asserts that the specified object is not nil.

```
a.NotNil(err)
```

func (*Assertions) NotNilf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L829)

```
func (a *Assertions) NotNilf(object interface{}, msg string (/builtin#string), args ...interfa
ce{}) bool (/builtin#bool)
```

NotNilf asserts that the specified object is not nil.

```
a.NotNilf(err, "error message %s", "formatted")
```

func (*Assertions) NotPanics (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L839)

```
func (a *Assertions) NotPanics(f PanicTestFunc, msgAndArgs ...interface{}) bool (/builtin#boo
l)
```

NotPanics asserts that the code inside the specified PanicTestFunc does NOT panic.

```
a.NotPanics(func(){ RemainCalm() })
```

func (*Assertions) NotPanicsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L849)

```
func (a *Assertions) NotPanicsf(f PanicTestFunc, msg string (/builtin#string), args ...interfa
ce{}) bool (/builtin#bool)
```

NotPanicsf asserts that the code inside the specified PanicTestFunc does NOT panic.

```
a.NotPanicsf(func(){ RemainCalm() }, "error message %s", "formatted")
```

func (*Assertions) NotRegexp (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L860)

```
func (a *Assertions) NotRegexp(rx interface{}, str interface{}, msgAndArgs ...interface{}) boo
l (/builtin#bool)
```

NotRegexp asserts that a specified regexp does not match a string.

```
a.NotRegexp(regexp.MustCompile("starts"), "it's starting")
a.NotRegexp("^start", "it's not starting")
```

func (*Assertions) NotRegexpf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L871)

```
func (a *Assertions) NotRegexpf(rx interface{}, str interface{}, msg string (/builtin#string),
args ...interface{}) bool (/builtin#bool)
```

NotRegexpf asserts that a specified regexp does not match a string.

```
a.NotRegexpf(regexp.MustCompile("starts", "error message %s", "formatted"), "it's starting")
a.NotRegexpf("^start", "it's not starting", "error message %s", "formatted")
```

func (*Assertions) NotSubset (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L882)

```
func (a *Assertions) NotSubset(list interface{}, subset interface{}, msgAndArgs ...interface
{}) bool (/builtin#bool)
```

NotSubset asserts that the specified list(array, slice...) contains not all elements given in the specified subset(array, slice...).

```
a.NotSubset([1, 3, 4], [1, 2], "But [1, 3, 4] does not contain [1, 2]")
```

func (*Assertions) NotSubsetf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L893)

```
func (a *Assertions) NotSubsetf(list interface{}, subset interface{}, msg string (/builtin#str
ing), args ...interface{}) bool (/builtin#bool)
```

NotSubsetf asserts that the specified list(array, slice...) contains not all elements given in the specified subset(array, slice...).

```
a.NotSubsetf([1, 3, 4], [1, 2], "But [1, 3, 4] does not contain [1, 2]", "error message %s", "fo
```

func (*Assertions) NotZero (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L901)

```
func (a *Assertions) NotZero(i interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

NotZero asserts that i is not the zero value for its type.

func (*Assertions) NotZerof (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L909)

```
func (a *Assertions) NotZerof(i interface{}, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

NotZerof asserts that i is not the zero value for its type.

func (*Assertions) Panics (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L919)

```
func (a *Assertions) Panics(f PanicTestFunc, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Panics asserts that the code inside the specified PanicTestFunc panics.

```
a.Panics(func(){ GoCrazy() })
```

func (*Assertions) PanicsWithValue (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L930)

```
func (a *Assertions) PanicsWithValue(expected interface{}, f PanicTestFunc, msgAndArgs ...inte
rface{}) bool (/builtin#bool)
```

PanicsWithValue asserts that the code inside the specified PanicTestFunc panics, and that the recovered panic value equals the expected panic value.

```
a.PanicsWithValue("crazy error", func(){ GoCrazy() })
```

func (*Assertions) PanicsWithValuef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L941)

```
func (a *Assertions) PanicsWithValuef(expected interface{}, f PanicTestFunc, msg string (/buil
tin#string), args ...interface{}) bool (/builtin#bool)
```

PanicsWithValuef asserts that the code inside the specified PanicTestFunc panics, and that the recovered panic value equals the expected panic value.

```
a.PanicsWithValuef("crazy error", func(){ GoCrazy() }, "error message %s", "formatted")
```

func (*Assertions) Panicsf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L951)

```
func (a *Assertions) Panicsf(f PanicTestFunc, msg string (/builtin#string), args ...interface
{}) bool (/builtin#bool)
```

Panicsf asserts that the code inside the specified PanicTestFunc panics.

```
a.Panicsf(func(){ GoCrazy() }, "error message %s", "formatted")
```

func (*Assertions) Regexp (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L962)

```
func (a *Assertions) Regexp(rx interface{}, str interface{}, msgAndArgs ...interface{}) bool
 (/builtin#bool)
```

Regexp asserts that a specified regexp matches a string.

```
a.Regexp(regexp.MustCompile("start"), "it's starting")
a.Regexp("start...$", "it's not starting")
```

func (*Assertions) Regexpf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L973)

```
func (a *Assertions) Regexpf(rx interface{}, str interface{}, msg string (/builtin#string), ar
gs ...interface{}) bool (/builtin#bool)
```

Regexpf asserts that a specified regexp matches a string.

```
a.Regexpf(regexp.MustCompile("start", "error message %s", "formatted"), "it's starting")
a.Regexpf("start...$", "it's not starting", "error message %s", "formatted")
```

func (*Assertions) Same (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L986)

```
func (a *Assertions) Same(expected interface{}, actual interface{}, msgAndArgs ...interface{})
bool (/builtin#bool)
```

Same asserts that two pointers reference the same object.

```
a.Same(ptr1, ptr2)
```

Both arguments must be pointer variables. Pointer variable sameness is determined based on the equality of both type and value.

func (*Assertions) Samef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L999)

```
func (a *Assertions) Samef(expected interface{}, actual interface{}, msg string (/builtin#stri
ng), args ...interface{}) bool (/builtin#bool)
```

Samef asserts that two pointers reference the same object.

```
a.Samef(ptr1, ptr2, "error message %s", "formatted")
```

Both arguments must be pointer variables. Pointer variable sameness is determined based on the equality of both type and value.

func (*Assertions) Subset (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1010)

```
func (a *Assertions) Subset(list interface{}, subset interface{}, msgAndArgs ...interface{}) b
ool (/builtin#bool)
```

Subset asserts that the specified list(array, slice...) contains all elements given in the specified subset(array, slice...).

```
a.Subset([1, 2, 3], [1, 2], "But [1, 2, 3] does contain [1, 2]")
```

func (*Assertions) Subsetf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1021)

```
func (a *Assertions) Subsetf(list interface{}, subset interface{}, msg string (/builtin#strin
g), args ...interface{}) bool (/builtin#bool)
```

Subsetf asserts that the specified list(array, slice...) contains all elements given in the specified subset(array, slice...).

```
a.Subsetf([1, 2, 3], [1, 2], "But [1, 2, 3] does contain [1, 2]", "error message %s", "formatted
```

func (*Assertions) True (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1031)

```
func (a *Assertions) True(value bool (/builtin#bool), msgAndArgs ...interface{}) bool (/builti
n#bool)
```

True asserts that the specified value is true.

```
a.True(myBool)
```

func (*Assertions) Truef (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1041)

```
func (a *Assertions) Truef(value bool (/builtin#bool), msg string (/builtin#string), args ...i
nterface{}) bool (/builtin#bool)
```

Truef asserts that the specified value is true.

```
a.Truef(myBool, "error message %s", "formatted")
```

func (*Assertions) WithinDuration (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1051)

```
func (a *Assertions) WithinDuration(expected time (/time).Time (/time#Time), actual time (/tim
e).Time (/time#Time), delta time (/time).Duration (/time#Duration), msgAndArgs ...interface{})
bool (/builtin#bool)
```

WithinDuration asserts that the two times are within duration delta of each other.

```
a.WithinDuration(time.Now(), time.Now(), 10*time.Second)
```

func (*Assertions) WithinDurationf (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1061)

```
func (a *Assertions) WithinDurationf(expected time (/time).Time (/time#Time), actual time (/ti
me).Time (/time#Time), delta time (/time).Duration (/time#Duration), msg string (/builtin#stri
ng), args ...interface{}) bool (/builtin#bool)
```

WithinDurationf asserts that the two times are within duration delta of each other.

```
a.WithinDurationf(time.Now(), time.Now(), 10*time.Second, "error message %s", "formatted")
```

func (*Assertions) Zero (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1069)

```
func (a *Assertions) Zero(i interface{}, msgAndArgs ...interface{}) bool (/builtin#bool)
```

Zero asserts that i is the zero value for its type.

func (*Assertions) Zerof (https://github.com/stretchr/testify/blob/master/assert/assertion_forward.go#L1077)

```
func (a *Assertions) Zerof(i interface{}, msg string (/builtin#string), args ...interface{}) b
ool (/builtin#bool)
```

Zerof asserts that i is the zero value for its type.

type BoolAssertionFunc (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L40)

```
type BoolAssertionFunc func(TestingT, bool (/builtin#bool), ...interface{}) bool (/builtin#bool)
```

BoolAssertionFunc is a common function prototype when validating a bool value. Can be useful for table driven tests.

Example

type Comparison (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L47)

```
type Comparison func() (success bool (/builtin#bool))
```

Comparison a custom function that returns true on success and false on failure

type ComparisonAssertionFunc (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L32)

```
type ComparisonAssertionFunc func(TestingT, interface{}, interface{}, ...interface{}) bool (/bui
```

ComparisonAssertionFunc is a common function prototype when comparing two values. Can be useful for table driven tests.

Example

type ErrorAssertionFunc (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L44)

```
type ErrorAssertionFunc func(TestingT, error (/builtin#error), ...interface{}) bool (/builtin#bo
```

ErrorAssertionFunc is a common function prototype when validating an error value. Can be useful for table driven tests.

Example

type PanicTestFunc (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L900)

```
type PanicTestFunc func()
```

PanicTestFunc defines a func that should be passed to the assert.Panics and assert.NotPanics methods, and represents a simple func that takes no arguments, and returns nothing.

type TestingT (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L26)

```
type TestingT interface {
    Errorf(format string (/builtin#string), args ...interface{})
}
```

TestingT is an interface wrapper around *testing.T

type ValueAssertionFunc (https://github.com/stretchr/testify/blob/master/assert/assertions.go#L36)

```
type ValueAssertionFunc func(TestingT, interface{}, ...interface{}) bool (/builtin#bool)
```

ValueAssertionFunc is a common function prototype when validating a single value. Can be useful for table driven tests.

Example

Package assert imports 19 packages (?imports) (graph (?import-graph)) and is imported by 1809 packages (? importers). Updated 9 days ago. Refresh now. Tools (?tools) for package owners.