

# The Minitest Cheat Sheet

## ***Test file skeleton.***

```
require 'gems_or_files_under_test'
gem 'minitest'
require 'minitest/autorun'

class MyAwesomeTester < Minitest::Test
  def setup
    #This is run before each test.
  end

  def test_something_or_other
    #Test functionality with assertions and refutations.
  end

  #etc etc etc...
end
```

## ***Standard Assertions and Refutations (replace assert with refute)***

assert(test, msg = nil)	Fails unless test is a true value.
assert_block(msg = nil) { ... }	Fails unless block returns a true value.
assert_empty(obj, msg = nil)	Fails unless obj is empty?
assert_equal(expected, actual, msg = nil)	Fails unless expected == actual.
assert_in_delta(expected, actual, delta=0.001, msg=nil)	For comparing Floats. Fails unless exp and act are within delta of each other.
assert_includes(collection, obj, msg = nil)	Fails unless collection includes obj.
assert_instance_of(klass, obj, msg = nil)	Fails unless obj is an instance of klass
assert_kind_of(klass, obj, msg = nil)	Fails unless obj is a kind of klass
assert_match(regex, obj, msg = nil)	Fails unless regex =~ obj
assert_nil(obj, msg = nil)	Fails unless obj is nil
assert_output(stdout = nil, stderr = nil)	Fails if stdout or stderr do not output the expected results. Pass in nil to ignore that streams output. Pass in an empty string if you require it to be silent. Pass in a regexp if you want to pattern match.
assert_raises(*exceptions) { ... }	Fails unless a listed exception is raised by the block.
assert_silent()	Fails if the block outputs anything to stderr or stdout.

## ***Tools for creating customized tests, assertions, and refutations.***

flunk(msg = nil)	Fails with message.
pass(msg = nil)	Count a passing test.
skip(msg = nil)	Skip over a test.