ANSWER: ORGANIZE TESTS WITH SUITES

test/bargs.spec.js

```
const assert = require('assert');
const {parse} = require('bargs');

describe('when opts.expectsValue is an object', function () {
  it('should throw a TypeError', function () {
    assert.throws(() => {
      parse({
        expectsValue: {},
      }),
        {
          name: 'TypeError',
          message: /object is not iterable/,
        };
    });
  });
});

describe('when opts.expectsValue is a nonzero number',
function () {
  it('should throw a TypeError', function () {
    assert.throws(() => {
      parse({
        expectsValue: 1,
      }),
        {
          name: 'TypeError',
          message: /number 1 is not iterable/,
        };
    });
  });
});

describe('when passed a single argument "--foo"', function ()
{
  it('should return an object having property foo: true',
function () {
   assert.deepStrictEqual(parse(['--foo']), {_: [], foo:
true});
  });
});
```

HOOKS IN MOCHA

- A **hook** is code that runs *before all*, *after all*, *before each*, or *after each* every test in a suite
  - Has a `body` function
  - Has an *optional* `title` string
- AKA “setup” and “tear down” functions

- Global API:
  - `before([title], body)`
  - `beforeEach([title], body)`
  - `after([title], body)`
  - `afterEach([title], body)`