Writing Tests Quiz

🔥 3 minutes   ✅ Completed

Score
You have submitted your quiz.
You got 5 questions correct, and 0 question(s) still being graded.
Your score so far is 100 out of 100.

Retake Quiz

Writing Tests Quiz

1.

Identify which function is invoked before every test within the same `describe` block?

○ `before`
✓ ● `beforeEach`
○ `afterEach`
○ `after`

> **EXPLANATION**
> The `beforeEach` Mocha hook with be invoked before each of the tests within the same block.

2.

Identify which function is invoked after every test within the same `describe` block?

○ `before`
✓ ● `afterEach`
○ `after`
○ `beforeEach`

> **EXPLANATION**
> The `afterEach` Mocha hook with be invoked *after* each of the tests within the same block.

```
describe("sandwichMaker()", function() {
  context("given an invalid argument", function() {
    it("should throw a TypeError when given an invalid argument", function()
      assert.throws(sandwichMaker(14), TypeError);
    });
  });
});
```

3.

Why does the above code snippet throw an error when run?

○ The `context` function's callback isn't being passed in a syntactically correct way.

○ The `assert.throws` method requires the error's message as well as the type of error.

✓ ● The `sandwichMaker` function is being invoked with an argument that will throw an error - halting program execution.

> **EXPLANATION**
> The `sandwichMaker` function is being invoked with an argument that will throw an error - halting program execution. To avoid this we could wrap the `sandwichMaker` function within another function.

```
describe("animalMakers", function() {
  before(function() {
    console.log("after");
  });

  describe("penguinMaker", function() {
    it("should make penguins", () => {});
  });

  describe("catMaker", function() {
    it("should make cats", () => {});
  });
});
```

4.

How many times will the above `before` be invoked?

○ 4
✓ ● 1
○ 2
○ 3

> **EXPLANATION**

The `before` Mocha hook with be invoked *once* before all tests within the same block are run.

```
describe("makePopcorn()", function() {
  describe("when we have kernels", () => {
    it("should make popcorn", function() {
      // etc.
    });
  });
  describe("when we don't have kernels", () => {
    it("should throw an error", function() {
      // etc.
    });
  });
});
```

5.

What about the above code snippet is not according to Mocha writing best practices?

○ The above code snippet improperly nests `it` blocks within `describe` blocks.

○ There should be an additional `describe` block wrapped around the entire test dictating how we are testing.

✓ ● We are using `describe` blocks instead of `context` blocks when signifying the state of the function.

EXPLANATION

In the above code snippet we are setting two states (`have kernels` vs. `don't have kernels`) using two `describe` blocks. We should instead use the alias for `describe` , `context` , to signify we are testing two different contexts.

[Retake Quiz]   [Continue To Next Page →]

Click Retake Quiz to attempt the quiz again!