Quiz 2 - Chai Assertion Library

Félicitations !

Question 1 :

Suggestion: Have the Chai Assertion Library documentation: http://www.chaijs.com/api/bdd/ opened during this quiz.

Given this assertion from the Chai documentation.

```
expect(false).to.be.false;
```

How can you run in it Postman?

```
expect(false).to.be.false
```

```
pm.expect(false).to.be.false;
```

[✓]
```
pm.test("test", function () {
    pm.expect(false).to.be.false;
});
```

Félicitations !

Question 2 :

Have a look at the following assertion inside the test:

```
pm.test("test", function () {
    let number = '5';
    pm.expect(number).to.eql(5);
});
```

What will be the outcome of the test?

The test will succeed, because 5 equals 5.

[✓ The test will fail, because the two values have different data types. '5' is a string and 5 is a number.]

Félicitations !

Question 3 :

Consider the assertion from the test below:

```
pm.test("test", function () {
    let number;
    pm.expect(number).to.eql(null);
});
```

What will be the outcome of this test?

● It will fail, because number is *undefined* and *undefined* does not equal *null*.

○ It will succeed, because number is not defined, so it is by default null.

Félicitations !

Strings in JavaScript need to be surrounded by simple or double quotes. It the quotes are missing, JavaScript will try to resolve the names as variables.

Question 4 :

You want to check a value (apple) against multiple allowed values (orange, apple, pineapple). How can you do that?

○
```
pm.test("test", function () {
    pm.expect('apple').to.eql(['orange', 'apple', 'pineapple']);
});
```

● `pm.expect('apple').to.be.oneOf(['orange', 'apple', 'pineapple']);`

○ `pm.expect(apple).to.be.oneOf([orange, apple, pineapple]);`

○
```
pm.test("test", function () {
    pm.expect('apple').to.eql('orange');
    pm.expect('apple').to.eql('apple');
    pm.expect('apple').to.eql('pineapple');
});
```