Jasmine - 4

Organizing your specs

- describe -> groups specs
  -> suite

- Through naming convention
  -> Create a spec file with the same name that matches
  the source file you are unit testing,
  calculator.js -> calculator.spec.js

- Keep the folder structure

- By nesting suites!

```javascript
describe('calculator.js', function() {
   describe('Calculator', function() {
      it('should initialize the total', function() {
         const calculator = new Calculator();
         expect(calculator.total).toBe(0);   // pass
         expect(calculator.total).toBe("0");  // fail
      });
   })
});
```

```javascript
describe('calculator.js', function() {
   describe('Calculator', function() {
      describe('add()', function() {
         it('should add numbers to total', function() {
            const calculator = new Calculator();
            calculator.add(5);
            expect(calculator.total).toBe(5);
         });
      })
   })
});
```

```javascript
describe('updateResult()', () => {
   let element;

  beforeAll(() => {
    // Executed ONCE before all specs are executed
    // element = document.createElement('div');
    element = document.createElement('div');
```

```javascript
describe('updateResult()', () => {
  let element;
  beforeAll(() => {
    // Executed ONCE before all specs are executed
    // element = document.createElement('div');
    element = document.createElement('div');
    element.setAttribute('id', 'result');
    document.body.appendChild(element);
  });
  afterAll(() => {
    // Executed ONCE after all specs are executed
    // const element = document.getElementById('result');
    // document.body.removeChild(element);
    document.body.removeChild(element);
  });
  it('adds result to DOM element', function() {
    updateResult('5');
    // expect(element.innerText).toBe('5');
    expect(element.innerText).toBe('5');
  });
});
```