- Visual testing for observables
- “-” equals 10sec of time, “a” = emit, “|” complete, “#” is an error.
- ex: — —– = observable.never
- ex: —(ab)–| = 30ms, emit a&b, 20ms, complete.
- flush() = runs the sequence. → always flush at the end of the test.
- Setup: jasmine-marbles

- Create component fixtures quickly
- Magically mock out dependencies
- TestBed.configureTestingModule({imports: [NoopAnimationsModule ← disables animations], …}) createComponentFixture ← similar to the test bed.
  - Rather than inject, use TestBed.get() - I’m doing this today.
- Examples: Ngrx Sample Book App in GitHub

NgRx

- ngrx-data: no longer have to write actions/reducers/selectors again…
- Reducers
  - Should always be pure. Input = Output
- Selectors
  - Should always be pure.
  - Can use Snapshot testing.
- Actions
- Side Effects
  - Can use Jasmine Marbles to verify input actions
- Components
  - Given set of inputs = same outputs
  - Has no global observable dependencies: (modifying global vars, global state, rendering other side effect components)
    1. create component fixture
    2. beforeEach: async() + await fixture.compile({book});
    3. Should Compile: expect(fixture).toMatchSnapshot();
- Container Tests
  - Store aware components
  - Verify they work correctly
  - Verify they dispatch the right actions

Jasmine Marbles

- Visual testing for observables
- “-” equals 10sec of time, “a” = emit, “|” complete, “#” is an error.
- ex: — —– = observable.never
- ex: —(ab)–| = 30ms, emit a&b, 20ms, complete.
- flush() = runs the sequence. → always flush at the end of the test.
- Setup: jasmine-marbles

Angular Testing Library

- Create component fixtures quickly
- Magically mock out dependencies
- TestBed.configureTestingModule({imports: [NoopAnimationsModule ← disables animations], …}) createComponentFixture ← similar to the test bed.
  - Rather than inject, use TestBed.get() - I’m doing this today.
- Examples: Ngrx Sample Book App in GitHub