Front End Workshops

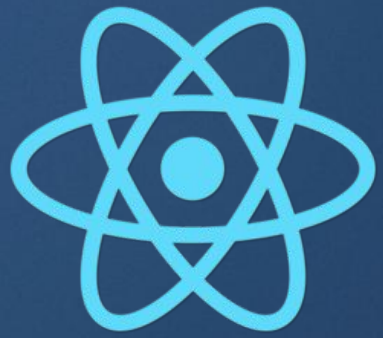
React Testing

VISUAL
ENGINEERING
ADVANCED MOBILE
TECHNOLOGY

www.visual-engin.com

Cristina Hernández García
chernandez@visual-engin.com

Mario García Martín
mgarcia@visual-engin.com



JavaScript Testing

Remember, remember...

Testing basics

- describe
- suite hooks
- it
- expect

```
describe("Use describe to group similar tests", function() {
  beforeEach(function() {});
  afterEach(function() {});
  it("use it to test an attribute of a target", function() {
    // use expect to make an assertion about a target
    expect(foo).to.be.a('string');
    expect(foo).to.equal('bar');
  });
});
```

Tools we'll be using

Test framework

Assertions library

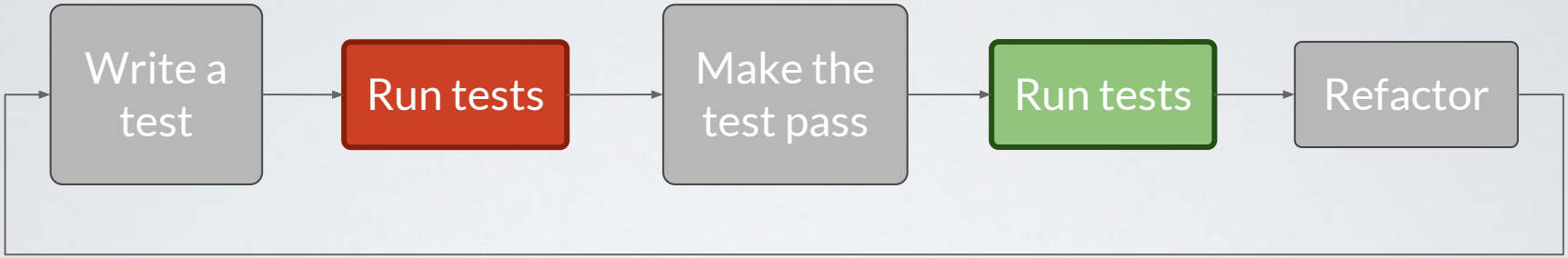
Sinon.JS

Test spies, stubs and mocks

*More info at http://mochajs.org/, http://chaijs.com/, and http://sinonjs.org/

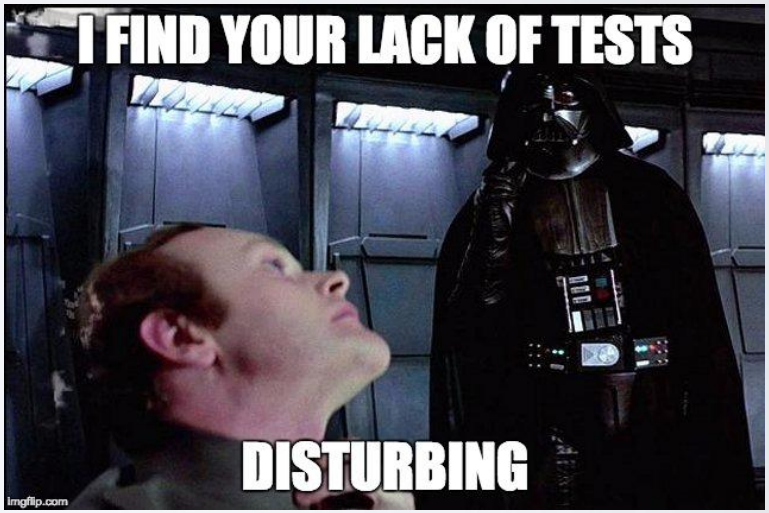
Test Driven Development (TDD)

The cycle of TDD



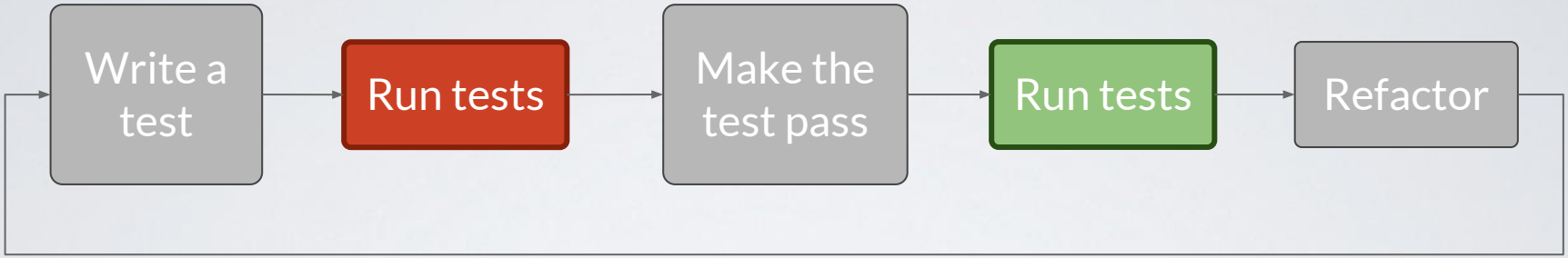
Benefits of TDD

- Produces code that works
- Honors the Single Responsibility Principle
- Forces conscious development
- Productivity boost



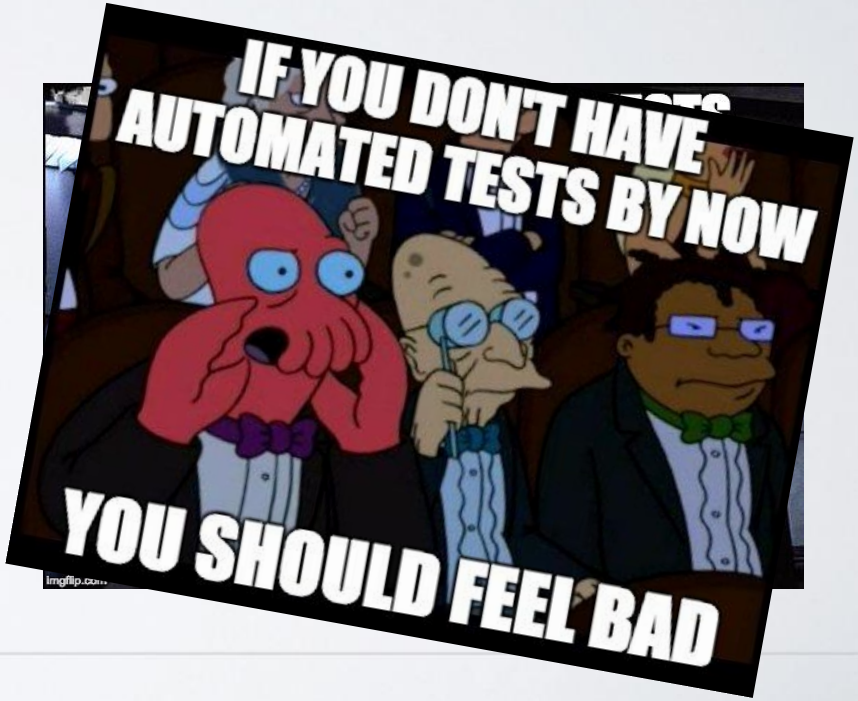
Test Driven Development (TDD)

The cycle of TDD



Benefits of TDD

- Produces code that works
- Honors the Single Responsibility Principle
- Forces conscious development
- Productivity boost



Testing React applications

What's different?

React testing - Particularities (1 of 2)

Components are rendered to a VDOM...

No need to fully render our components while testing!

Although not always required, sometimes it's necessary to have a full DOM API.

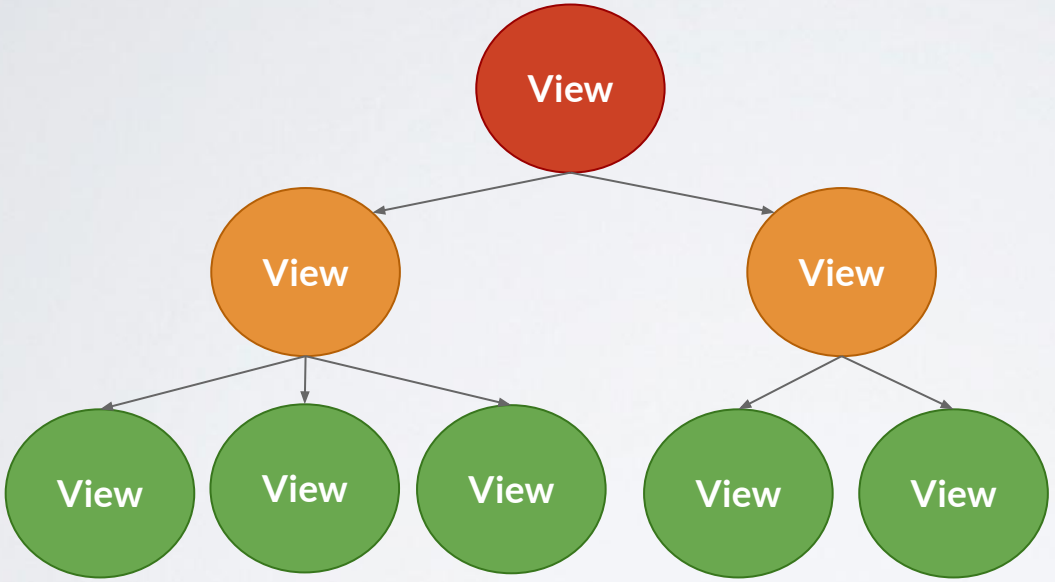
For console-based testing environments you can use the **jsdom** library to mock the DOM document.

```
global.document = jsdom.jsdom('<!doctype html><html><body></body></html>');
global.window = global.document.defaultView;
global.navigator = {
    userAgent: 'node.js'
};
```

React testing - Particularities

Components are rendered to a VDOM...

Higher level components can be tested in isolation. **Shallow rendering**.



Lets you render a component “one level deep”.

You don’t have to worry about the behavior of child components.

Events simulation is needed

React test utilities

Makes it easy to test React components in the testing framework of your choice.

React test utilities (1 of 3) (react-addons-test-utils)

Rendering components

Render a component into a detached DOM node in the document.

```
ReactComponent renderIntoDocument(ReactElement instance)
```

Requires a full DOM API available at the global scope.

Shallow rendering

It does not require a DOM API.

```
ReactShallowRenderer
createRenderer()
shallowRenderer.render(ReactElement element)
ReactElement shallowRenderer.getRenderOutput()
```

React test utilities (2 of 3) (react-addons-test-utils)

Shallow rendering example

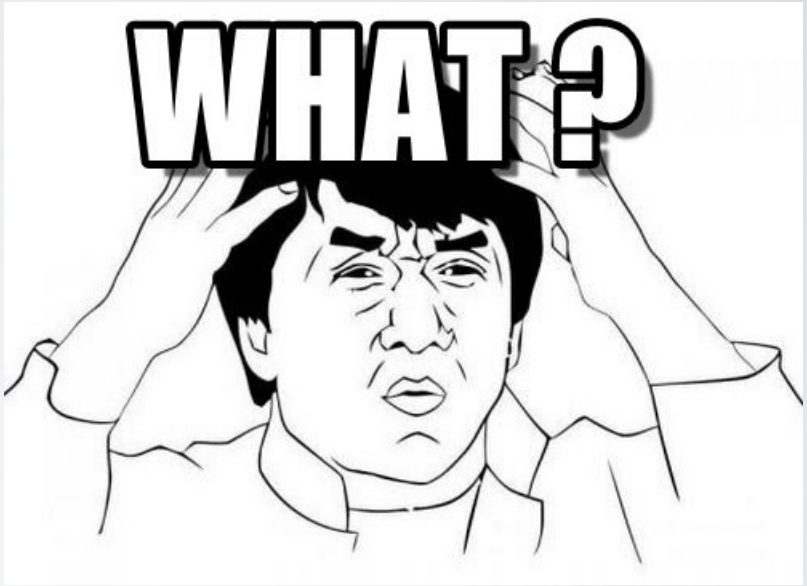
```
// MyComponent.js
import React, { Component } from
'react';
import Subcomponent from
'./Subcomponent';

class MyComponent extends Component {
  render() {
    return (
      <div>
        <span className="heading">
          Title
        </span>
        <Subcomponent foo="bar" />
      </div>
    );
  }
}
export default MyComponent;
```

```
// MyComponent.spec.js
import React from 'react';
import TestUtils from
'react-addons-test-utils';
import MyComponent from 'mycomponent';

const renderer =
TestUtils.createRenderer();
renderer.render(<MyComponent />);
const result =
renderer.getRenderOutput();

expect(result.type).to.equal('div');
expect(result.props.children).to.eql([
  <span className="heading">Title</span>,
  <Subcomponent foo="bar" />
]);
```

React test utilities (3 of 3) (react-addons-test-utils)

The rendered components interface...

```
findAllInRenderedTree

scryRenderedDOMComponentsWithClass
findRenderedDOMComponentWithClass

scryRenderedDOMComponentsWithTag
findRenderedDOMComponentWithTag

scryRenderedDOMComponentsWithType
findRenderedDOMComponentWithType
```



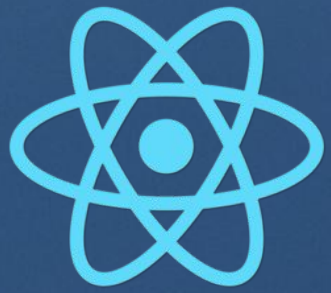
Simulating React synthetic events

Simulate an event dispatch on a DOM node with optional event data.

```
Simulate.{eventName}(DOMElement element, [object eventData])
```

Enzyme

JavaScript Testing utility for React.



Enzyme - Introduction

Now, with 150% neater interface!

Mimicks jQuery's API DOM manipulation and traversal.

shallow **mount** **render**

```
.find(selector)
.findWhere(predicate)

.state([key])
.setState(nextState)

.prop([key])
.setProps(nextProps)
```

```
.parent()
.children()

.simulate(event[,
data])
.update()
.debug()
```

*The `render` function may not have all the methods advertised

Enzyme - Shallow rendering

```
shallow(node[, options]) => ShallowWrapper
```

```
const row = shallow(<TableRow columns={5} />)

// Using prop to retrieve the columns
property
expect(row.prop('columns')).to.eql(5);

// Using 'at' to retrieve the forth column's
content
expect(row.find(TableColumn).at(3).prop('content')).to.exist;

// Using first and text to retrieve the columns text
content
expect(row.find(TableColumn).first().text()).to.eql('First column');

// Simulating events
const button = shallow(<MyButton
/>);
button.simulate('click');
expect(button.state('myActionWasPerformed')).to.be.true;
```

*More info at http://airbnb.io/enzyme/docs/api/shallow.html

Enzyme - Full DOM rendering

```
mount(node[, options]) => ReactWrapper
```

Use it when interacting with DOM or testing full lifecycle (**componentDidMount**).

Requires a full DOM API available at the global scope.

```
it('calls componentDidMount', function() {
  spy(Foo.prototype, 'componentDidMount');
  const wrapper = mount(<Foo />);
  expect(Foo.prototype.componentDidMount.calledOnce).to.equal(true);
});

it('allows us to set props', function() {
  const wrapper = mount(<Foo bar='baz' />);
  expect(wrapper.prop('bar')).to.equal('baz');
  wrapper.setProps({ bar: 'foo' });
  expect(wrapper.props('bar')).to.equal('foo');
});
```

*More info at http://airbnb.io/enzyme/docs/api/mount.html

Enzyme - Static rendered markup

```
render(node[, options]) => CheerioWrapper
```

Use it to render react components into static HTML (Uses Cheerio library).

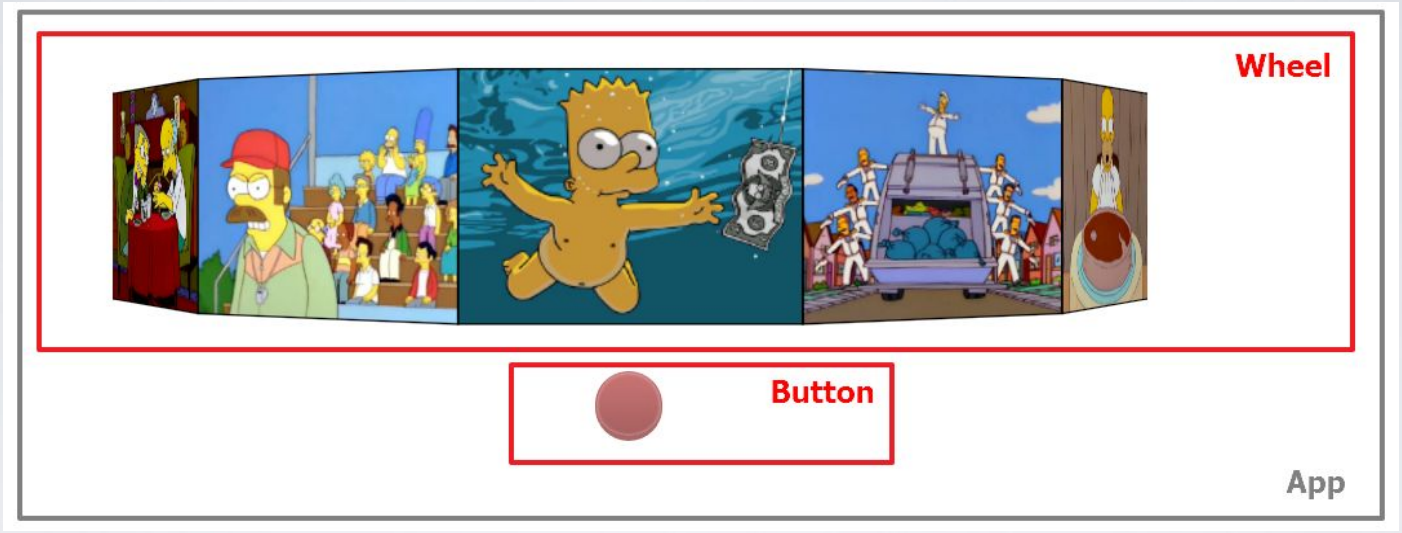
```
it('renders three .foo-bar', function() {
  const wrapper = render(<Foo />);
  expect(wrapper.find('.foo-bar')).to.have.length(3);
});

it('rendered the title', function() {
  const wrapper = render(<Foo title="unique" />);
  expect(wrapper.text()).to.contain("unique");
});
```

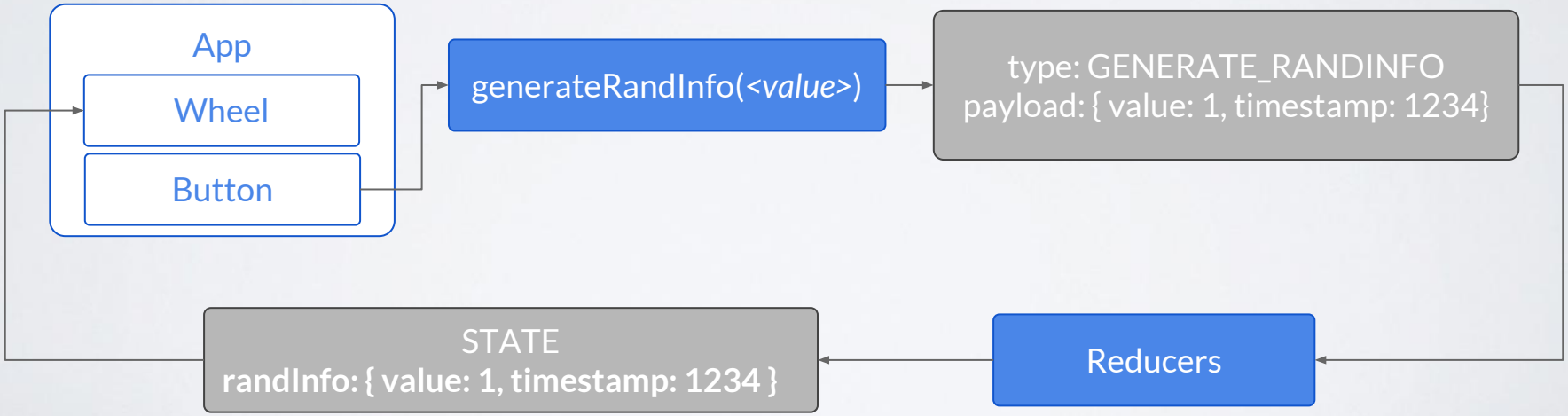
*More info at http://airbnb.io/enzyme/docs/api/render.html

Hands on code

Simpsons Wheel

Simpsons Wheel - General overview





Simpsons Wheel - Component details (1 of 2)

App component

Renders Wheel component, passing items prop

Renders Button component, passing max prop

Button component

Receives a max prop

When clicked, computes a random value between 1 and max

Calls action creator with the random number

Simpsons Wheel - Component details (2 of 2)

Wheel component

- Renders the images
- Stores the carousel rotation in its state
- Listens to Redux state changes
- Updates the rotation when receives new props

Thanks for your time!

Do you have any questions?

VISUAL
ENGINEERING
ADVANCED MOBILE
TECHNOLOGY

www.visual-engin.com
info@visual-engin.com

Passeig de Gràcia, 67 1º 2ª
(08008) Barcelona. España
T. : (+34) 93 215 77 35

Linked in
www.linkedin.com/companies/visual-engineering

twitter
www.twitter.com/visualengin