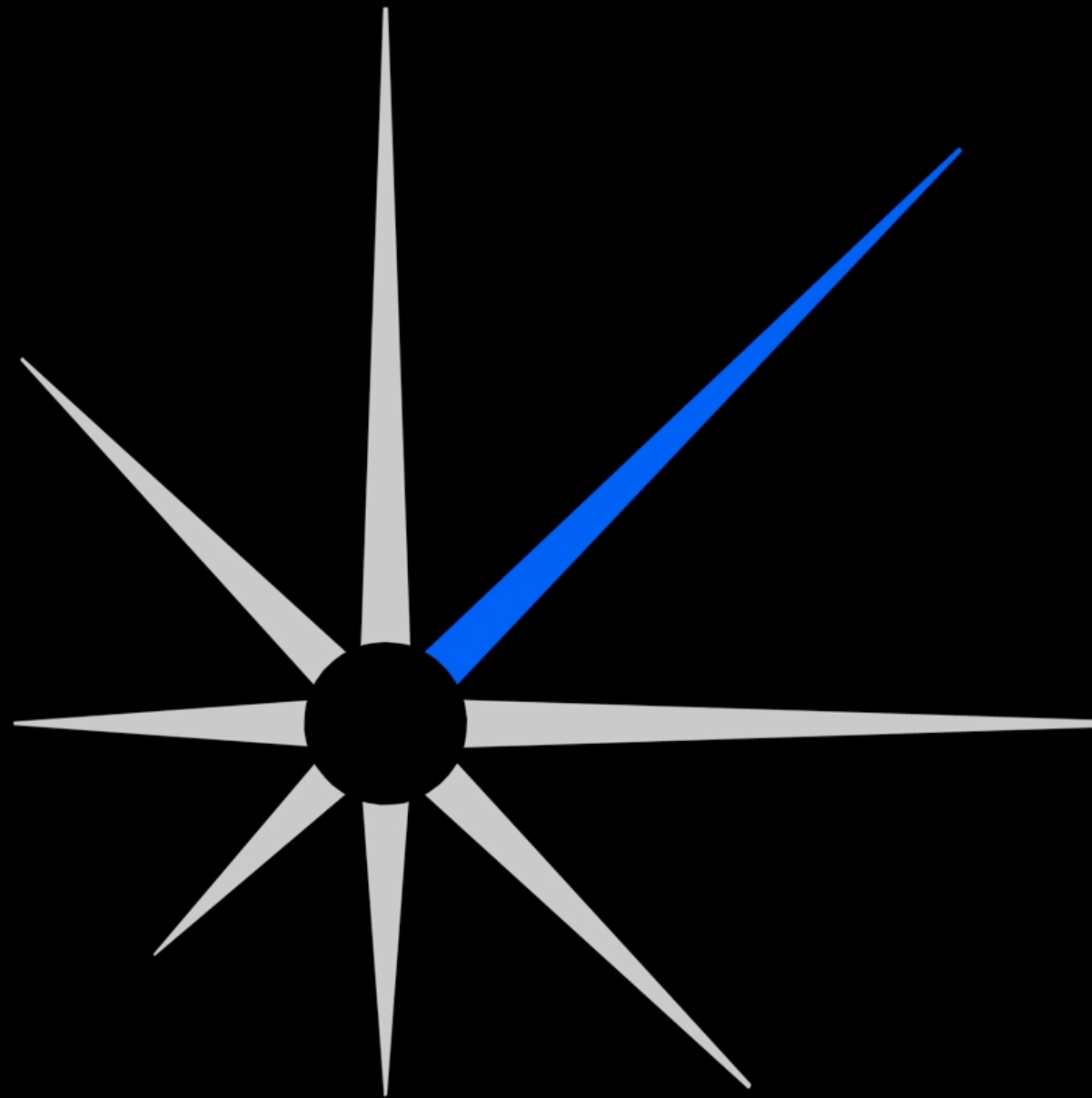


TDD and JavaScript



8th light

Jim Suchy

Software Craftsman

jsuchy@8thlight.com || @jsuchy

Manifesto for Software Craftsmanship

Raising the bar.

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software,
but also **well-crafted software**

Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

How do I ensure I'm
creating well-crafted
software?

Test Driven Development

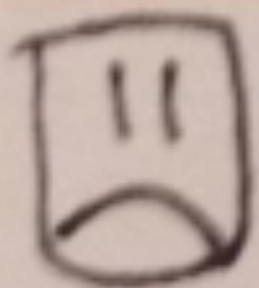
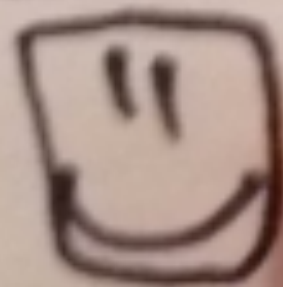



Uncle Bob's Three Laws ①

1. Write NO production code except to pass a failing test.
2. Write only enough of a test to demonstrate a failure
3. Write only enough production code to pass the test.

Core of TDD

③

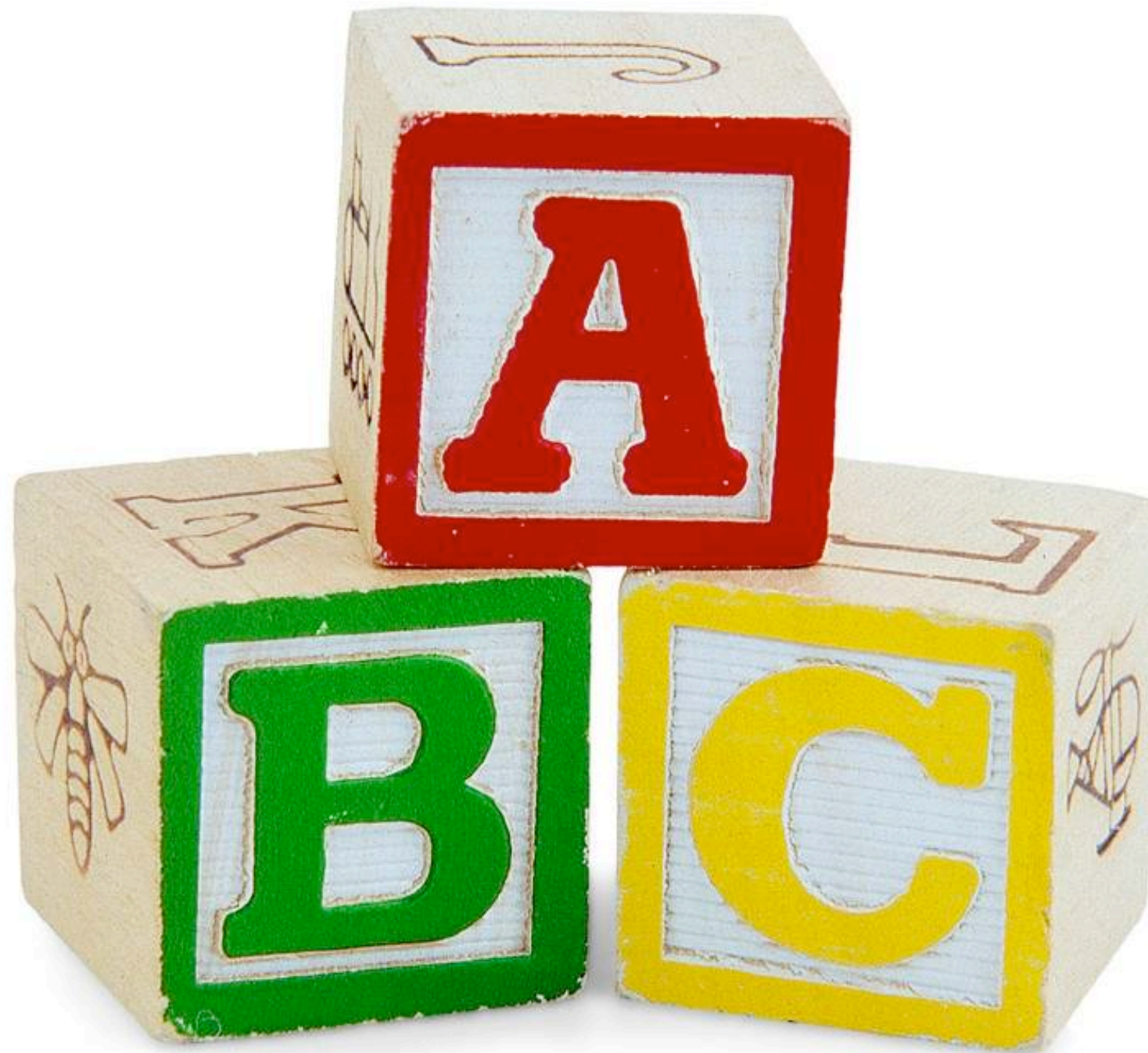
- **RED**: test fails 
- **GREEN**: test passes 
- **REFACTOR**
↳ CLEAN Code + Tests 

JavaScript === Evil?





Global Variables



Lack of Block Scope

Semicolon Insertion

```
return  
{  
  foo: "bar"  
}
```

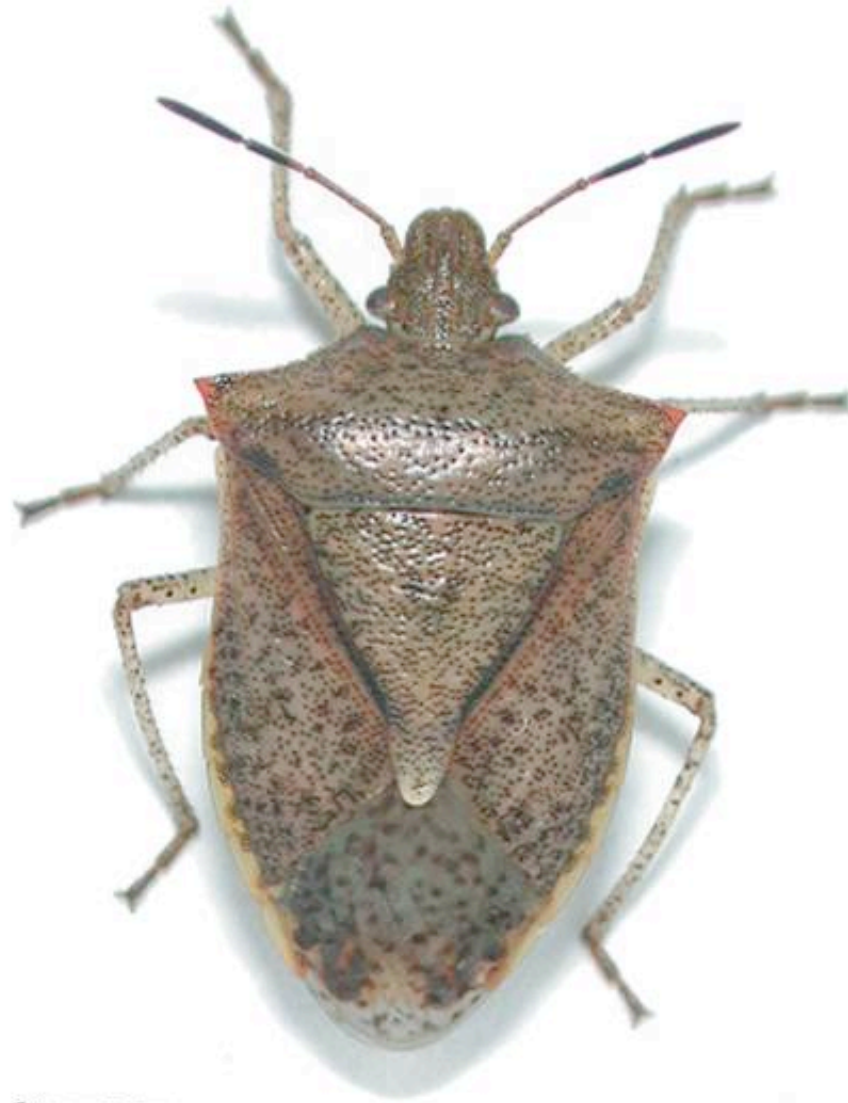
== VS ==



EQUALITY

Demons. Nazis. Ninjas. Zombies.
One size fits all.

Buggy JavaScript Code!



©MARLIN E. RICE

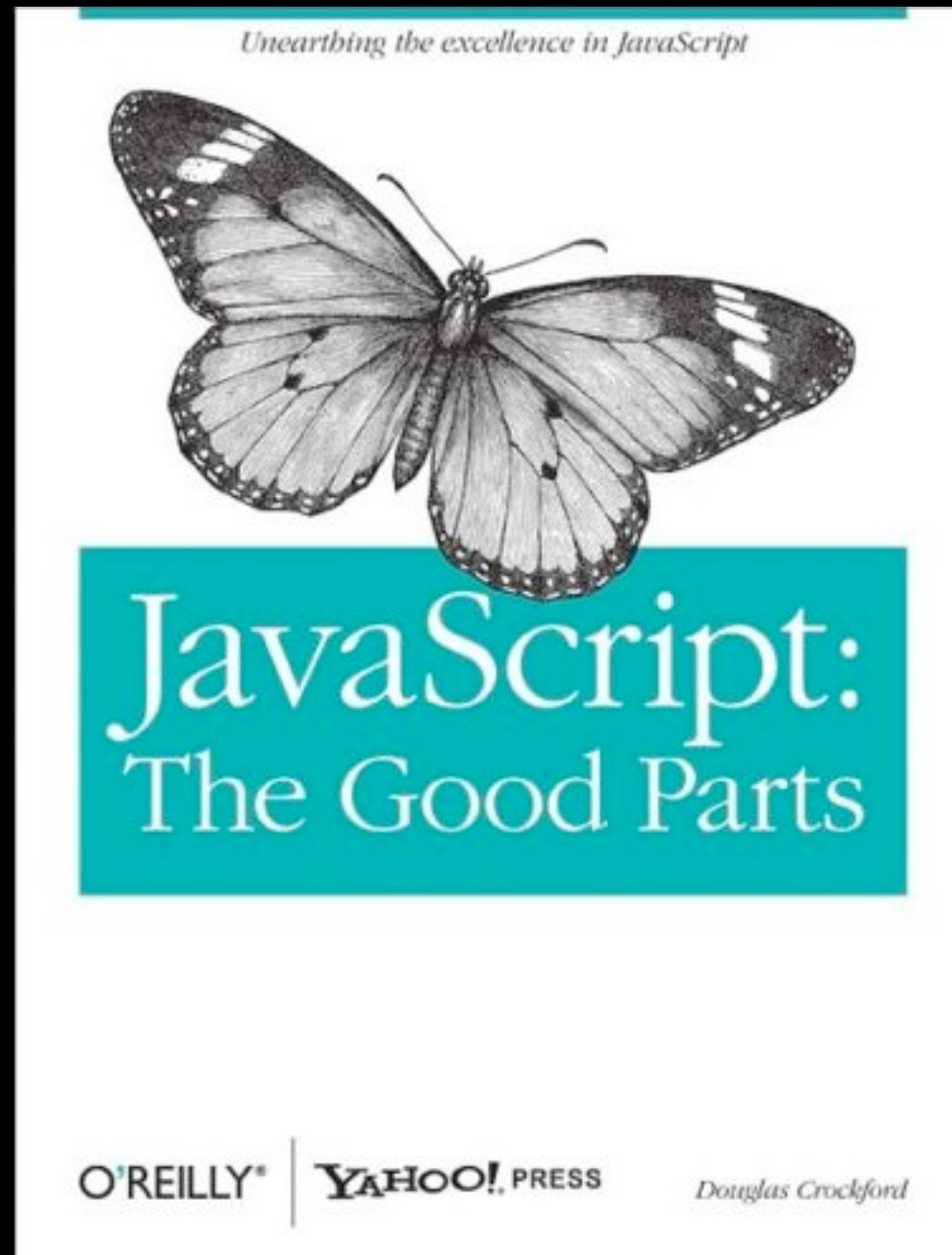
Why do we really hate
JavaScript?

Working with the
DOM sucks!

It's (perceived to be)
difficult to test...

Forces us to leave the
comfort of our
language of choice.

JavaScript has many good parts!



Functions



Dynamic Language with Loose Typing



Expressive object literal notation

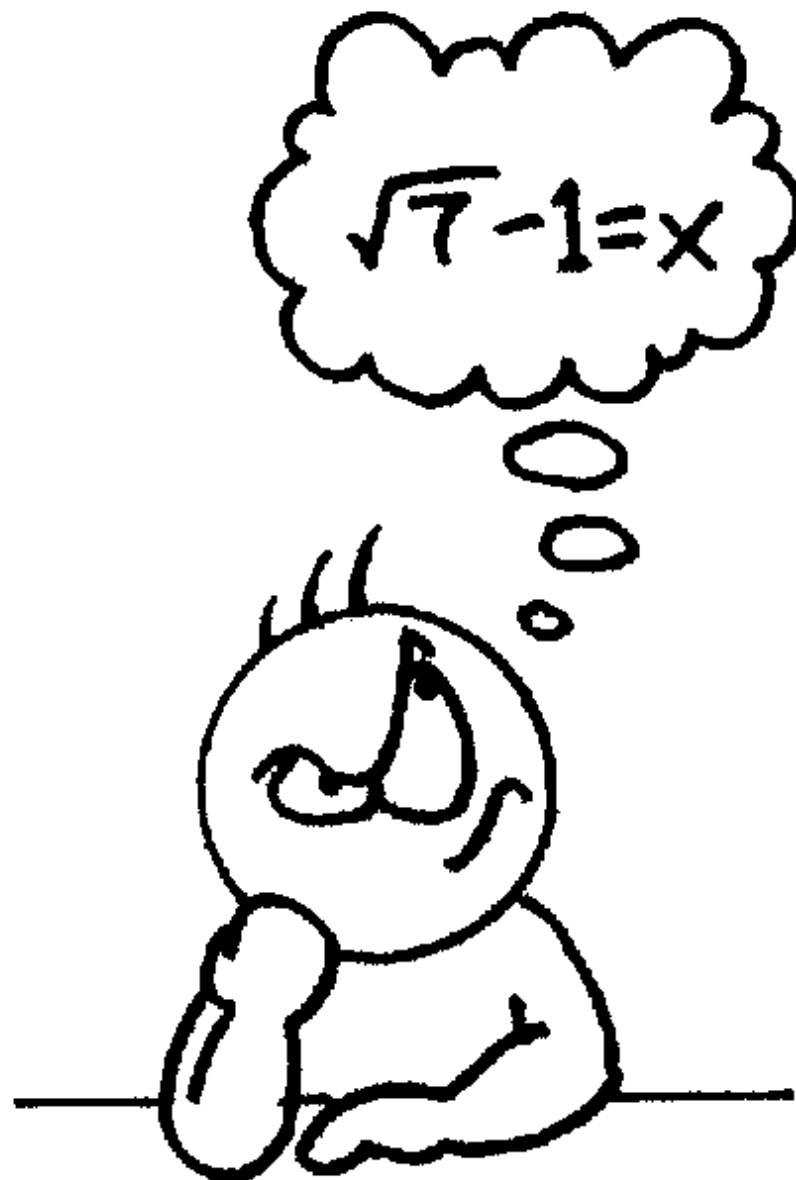
```
var javascriptGuru = {  
  firstName: "Douglas",  
  lastName: "Crockford"  
}
```

Portable



Every computer with a browser can run it!

Are you testing your JavaScript?



There are a ton of
JavaScript testing
frameworks available

RhinoUnit



YUI Test



ScrewUnit

```
describe("a nested describe", function() {  
  var invocations = [];  
  
  before(function() {  
    invocations.push("before");  
  });  
  
  describe("a doubly nested describe", function() {  
    before(function() {  
      invocations.push('inner before');  
    });  
  
    it("runs before in all ancestors prior to an it", function() {  
      expect(invocations).toEqual(["before", "inner before"]);  
    });  
  });  
});
```

JSUnit



(another) JSUnit



JsUnitTest

Blue Ridge

Cross Check

Test Case

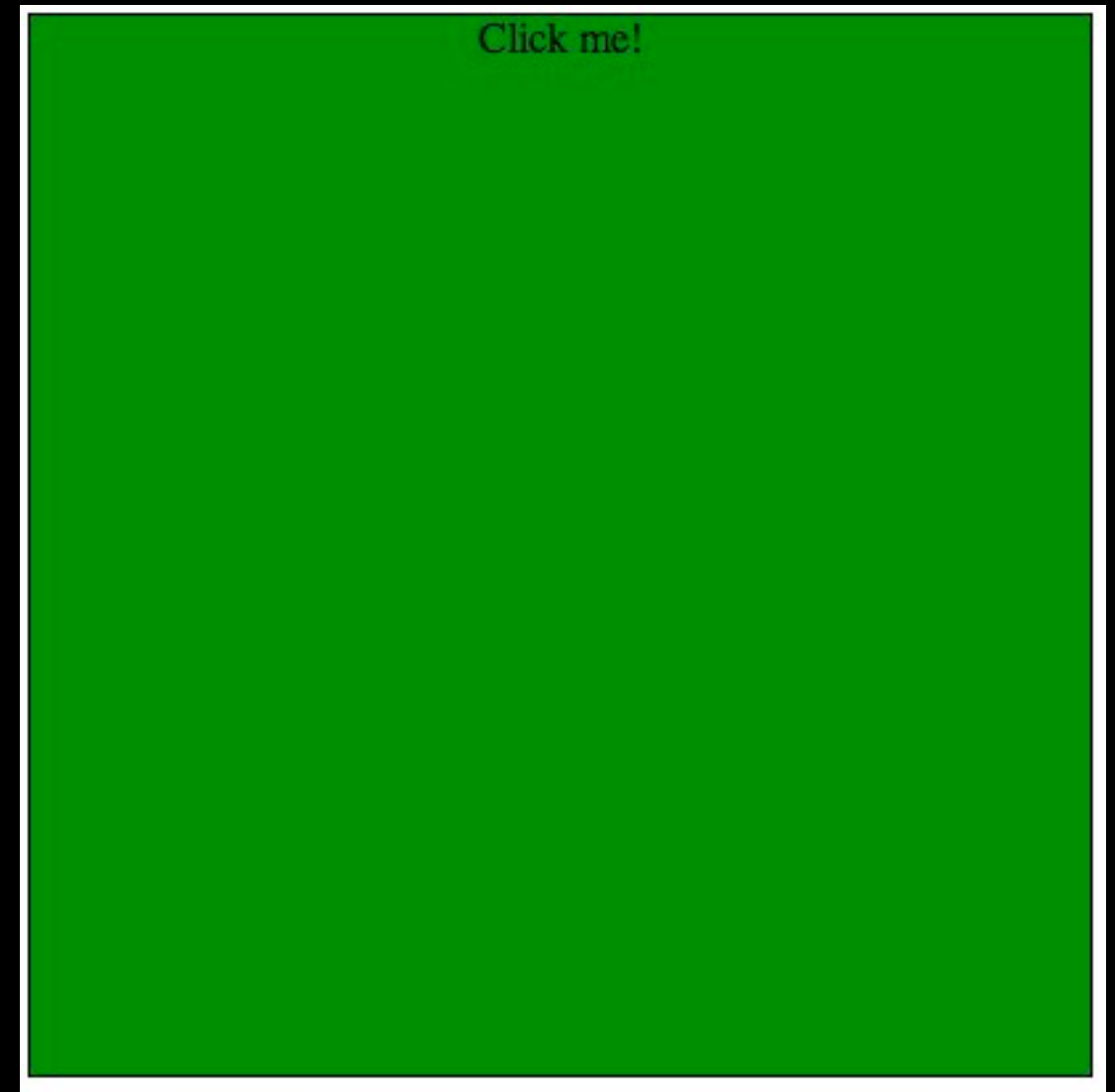
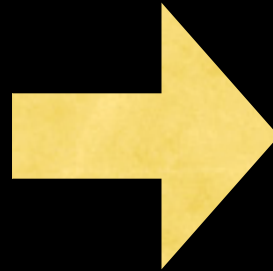
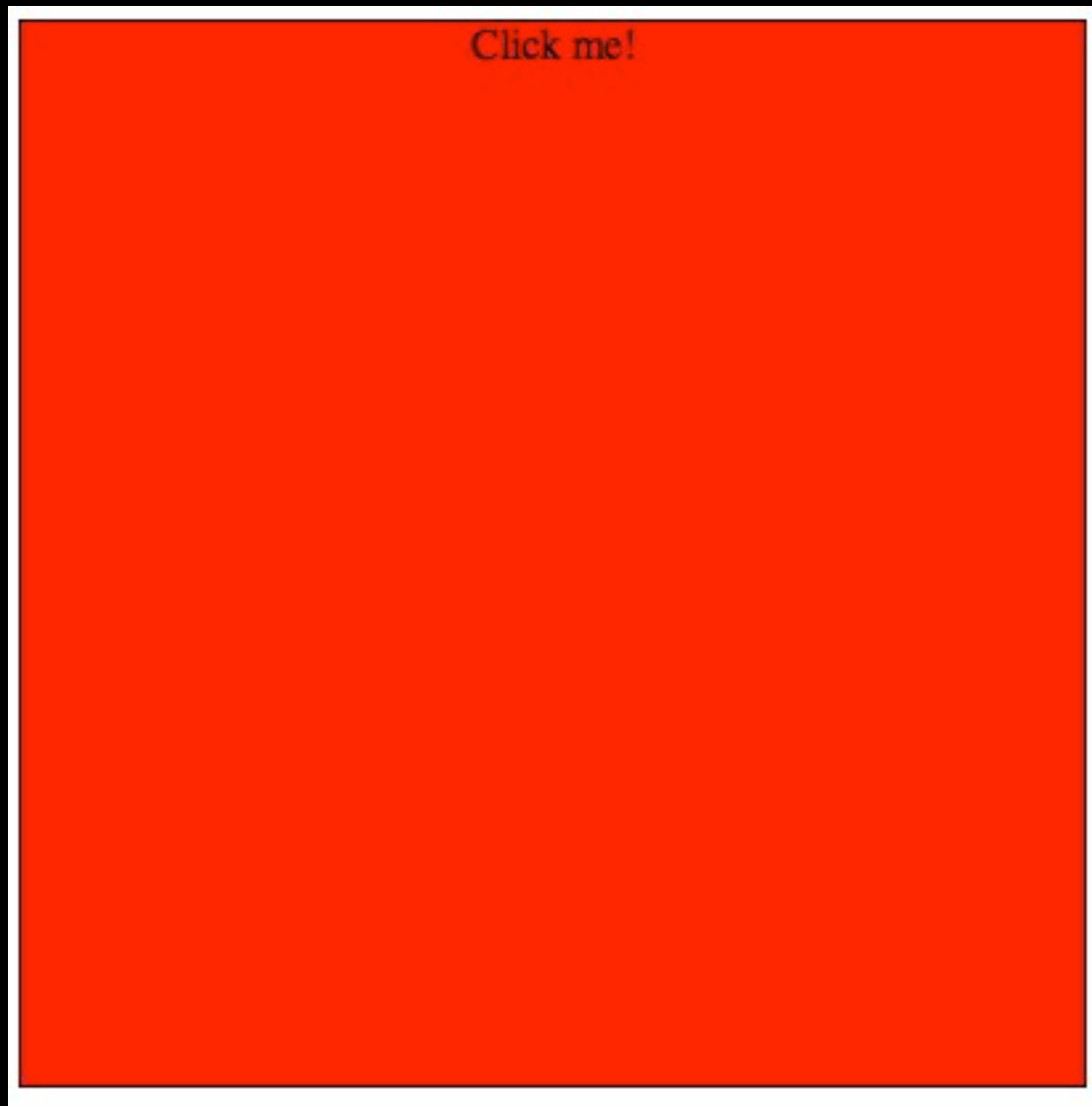
The list goes on...

Since everyone else is
doing it...

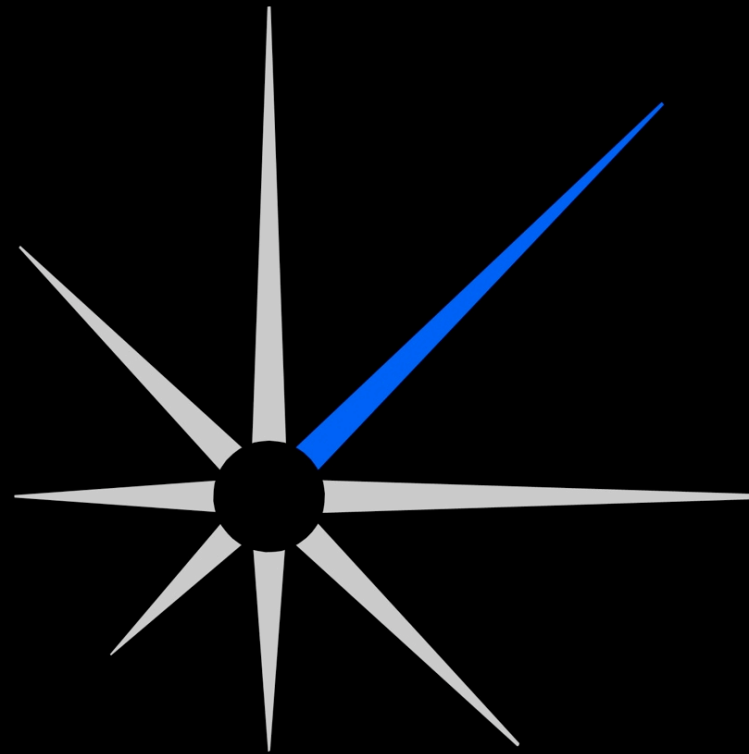
Let's write our own!

<http://github.com/jsuchy/JsSimpleUnit/tree/master>

Simple Use Case



Thank you!



8th light

Jim Suchy

jsuchy@8thlight.com || @jsuchy