

# Refactotum JavaScript

Stuart Halloway  
[thinkrelevance.com](http://thinkrelevance.com)

# How To Be a Refactotum

- 1. Set aside four hours**
- 2. Choose an open source project**
- 3. Get edge**
- 4. Get metrics**
- 5. Analyze**
- 6. Contribute**
- 7. Submit**



“If this is your first night at refactorotum,  
you have to refactor.”

--Brad Pitt as Tyler Durden, *Fight Club*

*(almost)*

# The Old App

- Codecite 2.x
- Used for Relevance Training Courses
- Client-only: TiddlyWiki + Prototype + Custom

# Ruby on Rails Labs and Resources Copyright 2006-7, Relevance, LLC

[Welcome](#)  
[Setup](#)  
[Ruby/Rails Track](#)  
[Ruby Track](#)  
[Rails Track](#)  
[Enterprise Track](#)  
[Relevance](#)  
[Pragmatic Studio](#)  
[Prag Progs](#)  
[Streamlined](#)  
[LICENSE](#)

[close](#) [close others](#) [edit](#) [permalink](#) [references](#) [back](#) [jump](#)

## DRb Lab Step 1: Creating The Server

*YourName*, 10 September 2006 (created 8 September 2006)

### Creating the Sever

no tags

1. Create a file named `drb_server.rb`. In this file, you will need to `require` both `'drb'` and `'drb/observable'`.
2. Create a class named `DRbClassroomServer` that includes `DRb::DRbObservable` [Hide the Code](#)

```
require 'drb'
require 'drb/observer'
Socket::do_not_reverse_lookup = true

class DRbClassroomServer
  include DRb::DRbObservable
```

[ruby\\_samples/lab\\_solutions/dr\\_b\\_server.rb](#)

# The New App

- Codecite 3.x
- GPL
- Client: Browser + YUI
- Server: Ruby on Rails

# Bad 2.x Author



# YUIDomCollapse

## YUIDOMCOLLAPSE - TURN ANY ELEMENT INTO A SHOW AND HIDE TRIGGER



YUIDomCollapse is a script built upon the [Yahoo User Interface library](#). It does the following:

- It scans the document for elements with a CSS class name of `trigger`.
- If the element is not a link it will
  - hide the next element in the document
  - create a new link inside the current element and
  - alternately show and hide the next element when you activate the element (either by clicking with a mouse, or hitting enter on the keyboard after tabbing to it).
- If the element is a link it will show and hide the element the link's href points to. If that element does not exist it will show and hide the next element in the node tree instead.

<http://onlinetools.org/tools/yuidomcollapse/>

# Crosscheck Setup

```
crosscheck.require("../..//public/javascripts/yahoo.js");  
crosscheck.require("../..//public/javascripts/dom.js");  
crosscheck.require("../..//public/javascripts/event.js");  
crosscheck.require("../..//public/javascripts/yuidomcollapse.js");
```

# Redirect Yahoo Log

```
//redirect Yahoo log to crosscheck log during unit tests  
YAHOO.log = function(msg, cat, src) {  
    crosscheck.print(cat + ": " + msg + "[" + src + "]" );  
}
```

# Work Around “Browser” Detection

```
crosscheck.onSetup( function( ) {  
    YAHOO_config = {injecting: true};  
    crosscheck.require( "test_helper.js" );  
});
```

# Pick Something Easy to Test First

```
// tool method to get the next sibling that is not a text node  
getNext:function(o){  
  var t = o.nextSibling;  
  if(t){  
    while(t.nodeType !== 1 && t.nextSibling){  
      t = t.nextSibling;  
    }  
  }  
  return t;  
},
```

# Passing Test

```
testGetNext: function() {  
    var first = document.createElement('div');  
    document.body.appendChild(first);  
    var text = document.createTextNode("foo");  
    document.body.appendChild(text);  
    var second = document.createElement('div');  
    document.body.appendChild(second);  
    assertEquals(text, first.nextSibling);  
    assertEquals(second, YAHOO.otorg.DomCollapse.getNext(first));  
}
```

# No Need For getNext!

```
testGetNext: function() {
  var first = document.createElement('div');
  document.body.appendChild(first);
  var text = document.createTextNode("foo");
  document.body.appendChild(text);
  var second = document.createElement('div');
  document.body.appendChild(second);
  assertEquals(text, first.nextSibling);
  assertEquals(second, YAHOO.util.Dom.getNextSibling(first));
}
```

# Eating The Elephant

```
init:function(){
  // shortcut for CSS properties
  var css = YAHOO.otorg.DomCollapse.css;
  if(typeof(css) != 'undefined'){
    // get possible bookmark
    var bookmark = window.location.hash.replace('#','');
    // get all elements with the correct class
    var elms = YAHOO.util.Dom.getElementsByClassName(css.triggerClass);
    // loop over all the elements
    for(var i=0,j=elms.length;i<j;i++){
      // if the trigger is not a link
      if(elms[i].nodeName.toLowerCase() != 'a'){
        // get the next element
        var t = YAHOO.util.Dom.getNextSibling(elms[i]);
        if(t){
          // get the element's ID or create a new one
          var newID = t.id || YAHOO.util.Dom.generateId();
          t.setAttribute('id',newID);
          // create a new target and replace the element's
          // content with this new element
          var a = document.createElement('a');
          a.setAttribute('href','#'+newID);
          var c = elms[i].innerHTML;
          a.innerHTML = elms[i].innerHTML;
          elms[i].innerHTML = '';
          elms[i].appendChild(a);
          // if the ID is not the bookmark add the parent class
          // to the trigger element and hide the element by
          // adding the hide class
          if(newID != bookmark){
            YAHOO.util.Dom.addClass(elms[i],css.parentClass);
            YAHOO.util.Dom.addClass(t,css.hideClass);
            // otherwise remove the hide class and add the
            // open class to the trigger
          } else {
            YAHOO.util.Dom.addClass(elms[i],css.openClass);
            YAHOO.util.Dom.removeClass(t,css.hideClass);
          }
          // add a click handler to the link pointing to toggle()
          YAHOO.util.Event.on(a, 'click', YAHOO.otorg.DomCollapse.toggle);
        }
      }
      // if the trigger is a link
    } else {
      // get the ID from the href attribute
      var newID = elms[i].href.replace(/.*#/, '');
      // grab the connected element, or the next sibling in case
      // it doesn't exist
      var t = document.getElementById(newID) || YAHOO.util.Dom.getNextSibling(elms[i]);
      if(t != null){
        // re-set the href attribute to this element
        if(t.id != newID){
          newID = t.id;
          elms[i].setAttribute('href','#'+newID);
        }
        // if the ID is not the bookmark, hide the element
        // and add the parent class
        if(newID != bookmark){
          YAHOO.util.Dom.addClass(elms[i],css.parentClass);
          YAHOO.util.Dom.addClass(t,css.hideClass);
          // otherwise add the open class to the trigger
        } else {
          YAHOO.util.Dom.addClass(elms[i],css.openClass);
        }
        // add a click handler to the link pointing to toggle
        YAHOO.util.Event.on(elms[i], 'click', YAHOO.otorg.DomCollapse.toggle);
      }
    }
  }
}
```

# A Risky Strategy

- **Break the big method into pieces**
- **Sniff test the changes in the UI**
- **Write tests once the pieces are smaller**

# -60 Lines and -1 Indent Later...

```
init:function
  // shortcut
  var css = 'collapse.css';
  if(typeof(YAHOO.util.Dom)!='undefined'){
    return;
  }
  // get possible bookmark
  var bookmark = window.location.hash.replace('#', '');
  // get all elements with the correct class
  var elms = YAHOO.util.Dom.getElementsByClassName('collapse', triggerClass);
  // loop over all the elements
  for(var i=0, j=elms.length; i<j; i++){
    // if the trigger is a link
    if(elms[i].nodeName.toLowerCase()!='a'){
      YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);
    } else {
      YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);
    }
  }
},
```

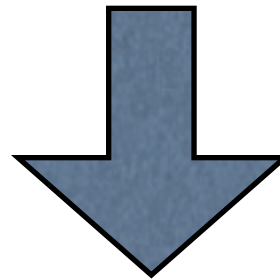
fallout  
return

helper

helper

# Replace Comments With Functions

```
// if the trigger is not a link  
if(elms[i].nodeName.toLowerCase() !== 'a') {  
    YAHOO.otorg.DomCollapse.initNonLink(elms[i], bookmark, css);  
// if the trigger is a link  
} else {  
    YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);  
}
```



```
if(YAHOO.otorg.DomCollapse.isLink(elms[i])) {  
    YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);  
} else {  
    YAHOO.otorg.DomCollapse.initNonLink(elms[i], bookmark, css);  
}
```

# Testing Former Comments

```
testIsLink: function() {  
    var link = document.createElement("A");  
    var notLink = document.createElement("div");  
    assertTrue(YAHOO.otorg.DomCollapse.isLink(link));  
    assertFalse(YAHOO.otorg.DomCollapse.isLink(notLink));  
}
```

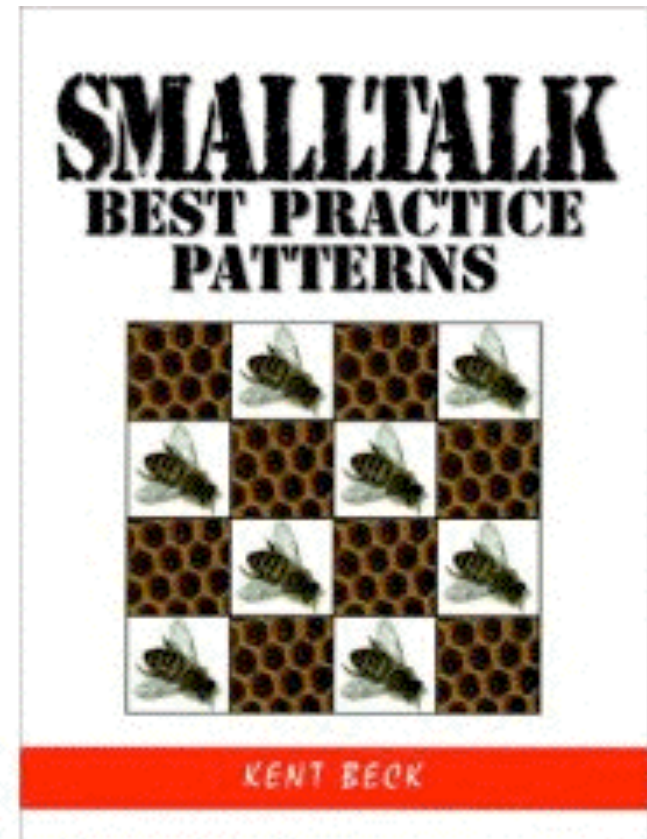
# Getting There...

```
init:function(){
  // shortcut for CSS properties
  var css = YAHOO.otorg.DomCollapse.css;
  if(typeof(css) == 'undefined'){
    return;
  }
  // get possible bookmark
  var bookmark = window.location.hash.replace('#', '');
  // get all elements with the correct class
  var elms = YAHOO.util.Dom.getElementsByClassName(css.triggerClass);

  for(var i=0, j=elms.length; i<j; i++){
    if(YAHOO.otorg.DomCollapse.isLink(elms[i])) {
      YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);
    } else {
      YAHOO.otorg.DomCollapse.initNonLink(elms[i], bookmark, css);
    }
  }
},
```

# Composed Method

Divide your program into methods that perform one identifiable task. *Keep all the operations in a method at the same level of abstraction...* (p.22, emphasis added)



# Composition Issues?

```
init:function(){
  // shortcut for CSS properties
  var css = YAHOO.otorg.DomCollapse.css;
  if(typeof(css) == 'undefined'){
    return;
  }
  // get possible bookmark
  var bookmark = window.location.hash.replace('#', '');
  // get all elements with the correct class
  var elms = YAHOO.util.Dom.getElementsByClassName(css.triggerClass);

  for(var i=0, j=elms.length; i<j; i++){
    if(YAHOO.otorg.DomCollapse.isLink(elms[i])) {
      YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);
    } else {
      YAHOO.otorg.DomCollapse.initNonLink(elms[i], bookmark, css);
    }
  }
},
```

Browser  
too low level

# Extracting Bookmark

```
bookmark: function() {  
    return window.location.hash.replace('#', '');  
},
```

```
testBookmark: function() {  
    assertEquals("", YAHOO.otorg.DomCollapse.bookmark());  
    crosscheck.Host.setLocation("file:///before_hash#after_hash");  
    assertEquals("after_hash", YAHOO.otorg.DomCollapse.bookmark());  
}
```

# Eliminate Namespaces...

```
init:function(){
  // shortcut for CSS properties
  var css = YAHOO.otorg.DomCollapse.css;
  if(typeof(css) == 'undefined'){
    return;
  }
  var bookmark = YAHOO.otorg.DomCollapse.bookmark();
  // get all elements with the correct class
  var elms = YAHOO.util.Dom.getElementsByClassName(css.triggerClass);

  for(var i=0,j=elms.length;i<j;i++){
    if(YAHOO.otorg.DomCollapse.isLink(elms[i])) {
      YAHOO.otorg.DomCollapse.initLink(elms[i], bookmark, css);
    } else {
      YAHOO.otorg.DomCollapse.initNonLink(elms[i], bookmark, css);
    }
  }
},
```

# ...With Object Scope

```
YAHOO.otorg.DomCollapse = function() {  
    // shortcut for CSS properties  
    var css = YAHOO.otorg.DomCollapse.css;  
    if(typeof(css) == 'undefined'){  
        return;  
    }  
    var bookmark = this.bookmark();  
    // get all elements with the correct class  
    var elms = this.dom.getElementsByClassName(css.triggerClass);  
  
    for(var i=0, j=elms.length; i<j; i++){  
        if(this.isLink(elms[i])) {  
            this.initLink(elms[i], bookmark, css);  
        } else {  
            this.initNonLink(elms[i], bookmark, css);  
        }  
    }  
};
```

**How helpful is it to write tests  
for trivial code that probably  
works?**

# The Tragedy of The Commons

The Tragedy of the Commons is a type of social trap, often economic, that involves a conflict over resources between individual interests and the common good.

[http://en.wikipedia.org/wiki/Tragedy\\_of\\_the\\_commons](http://en.wikipedia.org/wiki/Tragedy_of_the_commons)



---

# Questions?

---

**Get the Slides: [thinkrelevance.com/blog](http://thinkrelevance.com/blog)**

**Contact me: [stu@thinkrelevance.com](mailto:stu@thinkrelevance.com)**