Grimoire — *Smooth CoffeeScript*

□

> This literate program is *interactive* in its HTML$_5$ form. Edit a CoffeeScript segment to try it. You can see the generated JavaScript as you modify a CoffeeScript function by typing 'show name' after its definition.

Grimoire

A Grimoire is a book of magical incantations. This program got the name because it is full of magic. Not the superstitious kind — but magical constants, assumptions and implicit definitions — follow its style and conventions at your own peril. As it is with magic, this program does not actually work, it is all sleight of hand.

There is a little touch of self-referential magic in here: This literate program documents the internals of its own kind of literate program documents including itself.

If you have some CoffeeScript that would be more spiffy packaged in an interactive HTML$_5$ document, then be warned: As in magical potions, the ingredients are exotic and intoxicating. They include `ssam` (streaming sam rc script) from `plan9port`, `pandoc` (the Haskell universal format converter) and the enchanted `CodeMirror` — the embedded editor that shows your code as it *really* is.

The Crystal Ball of Hindsight One night, under the full moon, I might make this more approachable. There isn't anything going on in the `sam` structural regular expressions or the `ssam` pipeline that could not be done in, say, a CoffeeScript program. It is this way because the `plan9` toolset is eminent for prototyping and experiments and `pandoc` does most of the work. Creating these documents is a little something I am doing for fun. Building a professional quality publishing pipeline — while interesting — would be an order of magnitude more work. This is not like any commercial application I have ever seen or written.

The program attempts to change as little as possible in the `markdown` format, but meddles — quite unwisely — in the internals of the HTML$_5$ created by `pandoc`. This is to create one document that is simultaneously a `markdown` document, a literate CoffeeScript program, and an interactive HTML$_5$ application.

Editor Commandline Commands

I use `acme` to stir the ingredients, but you can use any editor/command line you like — don't blame me for the result. Speaking of don't blame me: If this program does not work for you, then it is not a bug, it is because you don't believe in magic — because *seriously* you don't, do you?

Below are the commands I use to extract CoffeeScript code, execute it and to format these Grimoire documents. To execute the commands; middle-button select them in the `acme` environment (if you are reading the markdown source code). As you can see, the magical constants start right here. To use the commands with another document, snarf them and `s/grimoire/new-document-name/g`.

Simple Extraction This command extracts CoffeeScript to an editor buffer — only bird tracks followed by the capitalization that you see below are extracted. This is used to avoid extracting code blocks that are spelled in all capitals i.e ~~~~ `{.COFFEESCRIPT}`. In the online environment those blocks become read-only and are not executed.

```
Edit ,x/^~~+[   ]*{\.[cC]offee[sS]cript.*}$/+,/^~~+$/-p
```

Extract and Run Standalone The next command extracts the CoffeeScript and prepend definitions that are provided by the interactive environment but do not exist in the standard standalone environment. Creates a CoffeeScript file. Compiles to a JavaScript file. Runs the CoffeeScript to produce an HTML and an output file. You can edit this bit if the document is not producing HTML. Opens the HTML in your web browser and plumbs it back into `acme`.

```
Edit ,>ssam -n 'x/^~~+[   ]*{\.[cC]offee[sS]cript.*}$/+,/^~~+$/-' |cat embed-standalone.coffee - |
```

Create PDF with TeX The following command produces a PDF version of the document. It uses XeTeX which in this case must be present on your system together with too many TeX packages to list (try a full 1.6Gb install if you have the bandwidth or spend hours as I do: start from a basic TeX and add individual packages one-by-one in the Tex Live Utility). It is not required unless you want a PDF version. By the way XeTeX has some trouble with Unicode characters, sometimes. Anyway it assumes that the command above has been run, the `VerbatimInput` lines near the end of this document (you can only see them in the markdown not in the HTML because they are TeX commands) uses the created files to insert program output and JavaScript translation into the PDF. You can delete those lines if you don't want that.

```
Edit ,>markdown2pdf --listings --xetex '--template=pandoc-template.tex' -o grimoire.pdf; open grim
```

Create an HTML$_5$ Application The last command produces an interactive HTML$_5$ document. It does assume that the 2nd commands above has been run sometime earlier (to bootstrap itself). The command below works as a straight pipeline. First running the markdown through `pandoc` to produce HTML$_5$ with `section` and `mathml` elements[1]. It creates a standalone document based on the template in `pandoc-template.html` and includes different CSS files for styling itself and its embedded editor. It also includes itself with the `-B` (`include-before-body`) option, not the interactive HTML that you might be reading, but the output-HTML that the interactive HTML produces.

The next four stages are all `ssam` substitutions in the HTML that `pandoc` produced. Talk about being dependent on internal implementation details — the `pandoc` used is version is 1.8.2.

(1) The relevant <`code`> elements (based on their class and spelling) are made contenteditable. That is only used to allow them to get focus and could possibly be replaced by pseudo onclick handlers. They are also set to *not* be `spellcheck`ed, which should be obvious for <`code`> elements. But even with that flag, Firefox still spellchecks the CoffeeScript in the elements displaying a lot of ugly squiggles.

(2) Sections with an id that starts with `view-solution` are hidden and given a `reveal` event handler. The hidden part goes on to the next section, which in the markdown can be marked with a header ##### without a text. That bit is now depending on `pandoc` not optimizing those superfluous headers away. But it makes it possible for solutions to have multiple code blocks and images in them.

(3) This part replaces the 1st and only the 1st image reference in a document with a canvas element. This is something that should be changed: a canvas element isn't usable without scripting and this could be done in scripting.

(4) The last defaults images to be centered instead of left aligned. This should be done in CSS instead.

```
Edit ,>pandoc -f markdown -t html -S -5 --mathml --section-divs --css pandoc-template.css --css co
```

Interactive Environment

```
webfragment = ->
```

```
  div id: ”feature-detect”, ->
    div id: ”feature-javascript”, style: ”color:#FF0000; display: block”, ->
      text ”No JavaScript &rarr; no output and no interactivity.”
    div id: ”feature-mathml”, style: ”color:#FF0000; display: none”, ->
      text ”No MathML 1998 &rarr; math is not readable.”
    div id: ”feature-canvas”, style: ”color:#FF0000; display: none”, ->
      text ”No Canvas &rarr; graphical output is not rendered.”
    div id: ”feature-contenteditable”, style: ”color:#FF0000; display: none”, ->
      text ”No ContentEditable &rarr; CoffeeScript sections can not be changed.”
```

[1] MathML is at this time not rendered in the two consumer browsers, Microsoft Internet Explorer and Google Chrome. I have tried MathJax but that resulted in Opera displaying garbage instead of equations. I found that it was unacceptable that Opera which does show W3C standard MathML get degraded, just because Google and Microsoft have chosen not to implement the standard. Of course in a commercial project the decision would likely be the other way round because of Opera's tiny market share compared with IE and Chrome. MathML renders fine in Firefox, Opera, and Safari on OS X & iOS.

```
input class: 'field', type: 'button', value: 'Adjust layout', onclick: ->
  @value = if toggleLayout() then 'Layout: fixed' else 'Layout: freeflow'
input class: 'field', type: 'button', value: 'Select editor', onclick: ->
  @value = if switchEditor() then 'Editor: CodeMirror' else 'Editor: plain text'
input class: 'field', type: 'button', value: 'Evaluation', onclick: ->
  @value = if switchEvaluation() then ↑'    Evaluate' else 'Auto Evaluate'
```

External scripts These scripts should be the same as those listed in the application manifest, `pandoc-template.appcache` that is mentioned in the `pandoc-template.html` file.

```
script src: 'node_modules/coffee-script.js'
script src: 'node_modules/coffeekup.js'
script src: 'node_modules/underscore.js'
script src: 'node_modules/qc.js'
script src: 'codemirror/codemirror.js'
script src: 'codemirror/coffeescript.js'
```

The Interactive Environment

```
coffeescript ->
```

```
  @reveal = (instance) -> # Very pandoc specific: requires --section-divs
    instance.getElementsByTagName('section')[0]?.style.display = 'block'
    instance.getElementsByTagName('h5')[0]?.innerHTML = 'Solution'
    instance.onclick = undefined
```

```
  @switchEditor = ->
    localStorage?.editor = if @useCodeMirror then 'TextArea' else 'CodeMirror'
    @useCodeMirror = not @useCodeMirror
  @switchEvaluation = ->
    localStorage?.evaluation =
      if localStorage?.evaluation is 'manual'
        'automatic'
      else
        'manual'
    localStorage?.evaluation is 'manual'
  @toggleLayout = ->
    fixedLayout = document.getElementById('page').style.maxWidth is ''
    localStorage?.fixedLayout = fixedLayout
    switchLayout fixedLayout
  switchLayout = (fixedLayout) -> # Should be in CSS
    document.getElementById('page').style.maxWidth =
      if fixedLayout then '600px' else ''
    s = document.getElementById('page').style
    if fixedLayout
      s.webkitHyphens = 'auto'
      s.mozHyphens = 'auto'
      s.msHyphens = 'auto'
      s.hyphens = 'auto'
      s.textAlign = 'justify'
    else
      s.webkitHyphens = ''
      s.mozHyphens = ''
      s.msHyphens = ''
      s.hyphens = ''
      s.textAlign = ''
    fixedLayout
```

```
featureDetect = ->
  adjustCoverPosition = (idFeature) -> # Should be in CSS
    canvasCover = document.getElementById(’drawCanvas’)
    topPos = parseInt canvasCover?.style.top
    lineHeight = document.getElementById(idFeature)?.scrollHeight
    canvasCover?.style.top = (topPos + lineHeight) + ’px’

  mathmlDetect = ->
    (e = document.createElement ’div’).innerHTML = ’<math></math>’
    passed = e.firstChild and ’namespaceURI’ of e.firstChild and
      e.firstChild.namespaceURI is ’http://www.w3.org/1998/Math/MathML’
    # TODO Needs a better test. Testing on Chrome directly because
    # it reports it supports MathML even though that is not the case.
    passed and not /Chrome/.test navigator.userAgent

  document.getElementById(’feature-javascript’)?.style.display = ”none”
  used = (document.getElementsByTagName(’math’)?.length > 0)
  if used and mathmlDetect() is false
    document.getElementById(’feature-mathml’)?.style.display = ”block”
    adjustCoverPosition ’feature-mathml’
  unless document.createElement(’canvas’).getContext?
    document.getElementById(’feature-canvas’)?.style.display = ”block”
    adjustCoverPosition ’feature-canvas’
  unless ’isContentEditable’ of document.documentElement
    document.getElementById(’feature-contenteditable’)?.style.display = ”block”
    adjustCoverPosition ’feature-contenteditable’
```

```
window.onload = ->
  featureDetect()
  switchLayout(on) if localStorage?.fixedLayout isnt ’false’
  @useCodeMirror = localStorage?.editor is ’CodeMirror’
```

Runtime canvas support As mentioned in the command used to create an HTML5 application, item #3 this should not be done automatically.

```
  canvas = document.getElementById(’drawCanvas’)
  window.ctx = canvas.getContext ’2d’ if canvas?

  clearCanvas = ->
    if window.ctx?
      window.ctx.clearRect 0, 0,
        window.ctx.canvas.width, window.ctx.canvas.height
  drawCanvas = (draw) ->
    clearCanvas()
    draw window.ctx
```

Runtime output support The `dataurl` in `addFrame` is not displayed in IE9 nor is its `innerHTML`.

```
  getParent = (child) -> child?.parentElement ? child?.parentNode

  addElement = (parent, text) ->
    newelem = document.createElement ’code’
    newelem.setAttribute ’class’, ’sourceCode output’
    newelem.innerHTML = text
    parent.appendChild newelem

  separator = (parent) ->
    if parent.getElementsByClassName(’output’).length is 0
      addElement parent, ’<hr><br>’
```

```
addFrame = (parent, width, height, content) ->
  newelem = document.createElement ’iframe’
  newelem.setAttribute ’class’, ’output’
  newelem.setAttribute ’width’,  width
  newelem.setAttribute ’height’, height
  newelem.setAttribute ’src’,
    ”data:text/html;charset=utf-8,#{encodeURIComponent content}”
  newelem.innerHTML =
    ’’’<div id=”feature-dataurl” style=”color:#FF0000; display: block”>
       No or insufficient Data URL support &rarr;
       web page output is not rendered in this browser.</div>’’’
  parent.appendChild newelem

showHere = (atTag, obj, shallow = false, symbol = ’&rarr;’) ->
  displayShallow = (o) ->
    ”””{#{”\n  #{k}: #{v}” for own k,v of o}\n}”””
  msg = switch typeof obj
    when ’undefined’, ’string’, ’function’
      obj
    when ’object’
      if shallow
        displayShallow obj
      else
        try
          JSON.stringify obj
        catch err
          displayShallow obj
    else obj?.toString()
  parentTag = getParent atTag
  separator parentTag
  addElement parentTag, ”#{symbol} #{msg}<br>”
  obj

showDocumentHere = (atTag, content, width = 300, height = 300) ->
  parentTag = getParent atTag
  separator parentTag
  addFrame parentTag, width, height, content
  return
```

CoffeeScript Evaluation This is called during startup for a full evaluation. It is also triggered by changes in one of the editor types, where it can be full evaluation with code stitching for short articles or incremental i.e. one code block for a whole book.

```
evaluateSource = (field = null) ->
```

```
  show = (obj, shallow = false, symbol = ’&rarr;’) ->
    showHere currentTag, obj, shallow, symbol
    return # Suppress display of the return value
  view = (obj, shallow = false, symbol = ’&rarr;’) ->
    showHere currentTag, obj, shallow, symbol
  showDocument = (content, width = 300, height = 300) ->
    showDocumentHere currentTag, content, width, height
  addErrorElement = (text) ->
    show ”””<span class=”er”>#{text}</span>”””
```

```
  # Naming convention: ’button-’ <type> ’-’ <codeTagId>
  getCurrentTagId = (id) -> id.match(/button-\w+-(.*)/)[1]
```

```
runOnDemand = (func) ->
  show ”<button id='button-run-#{currentTag.id}'> Run </button>”
  document.getElementById(”button-run-#{currentTag.id}”).onclick = ->
    currentTag = document.getElementById getCurrentTagId @id
    view = show = (obj, shallow = false, symbol = '&rArr;') ->
      showHere currentTag, obj, shallow, symbol
    showDocument = (content, width = 300, height = 300) ->
      showDocumentHere currentTag, content, width, height
    func()
  return

confirm = (message, func) ->
  show ”#{message}” +
    ”  <button id='button-yes-#{currentTag.id}'> Yes </button>” +
    ”  <button id='button-no-#{currentTag.id}'> No </button>”
  document.getElementById(”button-yes-#{currentTag.id}”).onclick = ->
    currentTag = document.getElementById getCurrentTagId @id
    view = show = (obj, shallow = false, symbol = '&rArr;') ->
      showHere currentTag, obj, shallow, symbol
    showDocument = (content, width = 300, height = 300) ->
      showDocumentHere currentTag, content, width, height
    func true
  document.getElementById(”button-no-#{currentTag.id}”).onclick = ->
    currentTag = document.getElementById getCurrentTagId @id
    view = show = (obj, shallow = false, symbol = '&rArr;') ->
      showHere currentTag, obj, shallow, symbol
    showDocument = (content, width = 300, height = 300) ->
      showDocumentHere currentTag, content, width, height
    func false
  return

prompt = (message, defaultValue, func) ->
  show ”#{message}” +
    ”  <input id='input-prompt-#{currentTag.id}' value='#{defaultValue}' />” +
    ”  <button id='button-prompt-#{currentTag.id}'> Go </button>”
  document.getElementById(”button-prompt-#{currentTag.id}”).onclick = ->
    currentTag = document.getElementById getCurrentTagId @id
    view = show = (obj, shallow = false, symbol = '&rArr;') ->
      showHere currentTag, obj, shallow, symbol
    showDocument = (content, width = 300, height = 300) ->
      showDocumentHere currentTag, content, width, height
    func document.getElementById(”input-prompt-#{getCurrentTagId @id}”).value
  return
```

QuickCheck You may ask: “What are these completely unrelated QuickCheck helpers doing here?” Well, it could be that I am simply trying to distract you while I pull something out of my sleeve — or it could be that I didn't get round to moving them into `qc.js`, which as its name implies is written in JavaScript (and it is not a CommonJS module either).

Incidentally JavaScript could learn a thing or two from CoffeeScript. For example: it could use a backtick feature for embedding CoffeeScript into it. Until the day when these helpers are moved to a better resting place, at least they are not doing much harm here … unless you summon them. QuickCheck is a minor imp, using randomness to cause havoc in your code. Several examples are shown in 'Smooth CoffeeScript' which is why the helpers are here. Let me know if you should have converted `qc.js` to CoffeeScript or have written something better.

```
# HTML colored output for QuickCheck.
class HtmlListener extends ConsoleListener
  constructor: (@maxCollected = 10) ->
  log: (str) -> show str
  passed: (str) -> # print message as OK
```

```
    show ”””<span class=”kw”>#{str}</span>”””
  invalid: (str) -> # print message as warning
    show ”””<span class=”dt”>#{str}</span>”””
  failure: (str) -> # print message as error
    show ”””<span class=”al”>#{str}</span>”””
  done: ->
    show ’Completed test.’
    resetProps() # Chain here if needed

# Enhanced noteArg returning its argument so it can be used inline.
Case::note = (a) -> @noteArg a; a
# Same as Case::note but also logs the noted args.
Case::noteVerbose = (a) -> @noteArg a; show @args; a

# Helper to declare a named test property for
# a function func taking types as arguments.
# Property is passed the testcase, the arguments
# and the result of calling func, it must return
# a boolean indicating success or failure.
testPure = (func, types, name, property) ->
  declare name, types, (c, a...) ->
    c.assert property c, a..., c.note func a...
  return

# Default qc configuration with 100 pass and 1000 invalid tests
qcConfig = new Config 100, 1000

# Test all known properties
test = (msg, func) ->
  _.each [msg, func, runAllProps qcConfig, new HtmlListener],
    (o) -> unless _.isUndefined o then show o
```

Traversing and unescaping The code for the field or the document is obtained with tags that relate the code back to its HTML element. The tag is needed to direct output to its place of origin. When an editor is instantiated for a <code> element it changes to a <textarea> so matching is on the class of a <pre>’s firstChild.

```
getCode = (codeTag) ->
  if codeTag.value?
    segment = codeTag.value
  else
    segment = codeTag.innerHTML
  #console.debug ”From: #{segment}”
  code = segment.replace /<br>/g, ’\n’
  code = code.replace /<[\/]?span[^>]*>/g, ’’
  code = code.replace /[&]gt;/g, ’>’
  code = code.replace /[&]lt;/g, ’<’
  code = code.replace /[&]nbsp;/g, ’ ’
  #console.debug ”To: #{code}”
  codeTag:codeTag, code:code

if field? and @incrementalEvaluation
  segments = [getCode field]
else
  # Obtain each code segment with an ownership tag
  segments =
    for codeSegment in window.document.getElementsByTagName ’pre’
      if codeSegment.className is ’sourceCode’
        codeTag = codeSegment.firstChild
        if codeTag.className is ’sourceCode coffeescript’ \
            or codeTag.className is ’sourceCode CoffeeScript’
```

```
          getCode codeTag
```

Stitch together continuing code segments. This is done based on a test of whether they begin with whitespace. For example a class with methods in different code blocks or a program that is split up in sections as in this document. This only works during full evaluation.

```
  segments = (segment for segment in segments when segment?)
  for segment, i in segments
    if i > 0 and /^[\s]/.test segment.code
      segment.code = segments[i-1].code + '\n' + segment.code
      segments[i-1] = undefined
  segments = (segment for segment in segments when segment?)
```

Clear output A performance test showed that the vast majority of time was consumed by this little loop: deleting previously shown output from the DOM. It made full code reevaluation an impossibility for anything but short articles. It was tested in three different browsers and `removeChild` was unbeliably slow in all of them. I don't know of alternatives for quickly removing elements, so the evaluation model for large documents was changed to incremental.

```
for segment in segments
  outList = getParent(segment.codeTag).getElementsByClassName 'output'
  outLength = outList?.length
  while outList and outLength > 0
    elem = outList[--outLength]
    getParent(elem).removeChild elem
```

Evaluation No real sandbox here, so the names and loop construct was changed to minimize the chance of conflict with the code that is being executed. Since clearing output made full reevaluation impossible, a small system where variables that are mentioned in `globalNames` are automatically promoted from the execution environment to the global environment was set up. This makes the named previous definitions available in subsequent code blocks during incremental evaluation (where there is no code stitching).

```
# Evaluate code segments
incredibleLength = segments.length
for incredibleIndex in [0...incredibleLength] by 1
  incredibleStart = new Date()
  currentSegment = segments[incredibleIndex]
  currentTag = currentSegment.codeTag
  if currentTag.id is ''
    currentTag.id = "codeTag#{incredibleIndex}"
  do (currentTag) ->
    try
      draw = undefined
      incredibleResult = eval CoffeeScript.compile \
        currentSegment.code, bare:on
      drawCanvas draw if draw?
      if incredibleResult isnt undefined and
         typeof incredibleResult isnt 'function'
        show incredibleResult, on, '&crarr;'
      if globalNames?
        promoteToGlobal = (magic) ->
          try
            magicValue = eval magic
          catch error
            return # doesn't exist
          this[magic] = magicValue
        for magic in globalNames.split ' '
          promoteToGlobal magic
    catch error
      console.log error.message
```

```
        addErrorElement error
      return
    if showTiming? and showTiming is on
      incredibleTime = Number(0.001 *
        (new Date() - incredibleStart)).toFixed 3
      show ”#{incredibleTime}s”, on,  ’’
  return
```

```
# Textarea editor code evaluation
keyDownEvaluation = (evt) ->
  return unless evt?
  if localStorage?.evaluation is ’manual’
    if evt.keyCode is 13 and evt.shiftKey # ↑ Shift /   Enter
      evt.preventDefault()
      evaluateSource evt.currentTarget
  if evt.keyCode is 13 and not evt.shiftKey
    evt.target?.rows++
  return
keyUpEvaluation = (evt) ->
  return unless evt?
  unless localStorage?.evaluation is ’manual’
    evaluateSource evt.currentTarget unless evt.keyCode in
      [16,37,38,39,40] # Ignore shift and arrow keys

# CodeMirror editor code evaluation
cmEvaluation = (editor) ->
  editor.save()
  evaluateSource editor.getTextArea()
cmAutoEvaluation = (editor) ->
  unless localStorage?.evaluation is ’manual’
    cmEvaluation editor
cmManualEvaluation = (editor) ->
  if localStorage?.evaluation is ’manual’
    cmEvaluation editor
```

Create an editor by converting a <`code`> element to a <`textarea`>. If CodeMirror is enabled then the <`textarea`> is used as a basis for that editor.

```
createEditor = (codeElement, text) ->
  newelem = document.createElement ’textarea’
  newelem.setAttribute ’id’, codeElement.id
  newelem.setAttribute ’class’, codeElement.getAttribute(’class’)
  countLines = (c for c in text when c is ’\n’).length + 1
  newelem.setAttribute ’rows’, countLines
  newelem.setAttribute ’style’, ’width: 98%;’
  newelem.setAttribute ’autofocus’, ’true’
  newelem.setAttribute ’spellcheck’, ’false’
  newelem.innerHTML = text
  newelem.addEventListener ’keydown’, keyDownEvaluation, false
  newelem.addEventListener ’keyup’, keyUpEvaluation, false
  getParent(codeElement).replaceChild newelem, codeElement
  if @useCodeMirror
    elemCodeMirror = CodeMirror.fromTextArea newelem,
      onChange: cmAutoEvaluation
      extraKeys: {’Shift-Enter’: cmManualEvaluation}
      lineNumbers: true
      theme: ’pantheme’
```

Setup editor activation

```
activateEditor = ->
  @removeEventListener ’focus’, activateEditor, false
```

```
        sourcecode = @innerHTML.toString().replace /<\/?span[^>]*>/g, ''
        sourcecode = sourcecode.replace /<br[ ]*[^>]*>/g, '\n'
        createEditor this, sourcecode

      # Activate text editor when code gets focus
      for segment in document.getElementsByTagName 'pre'
        segment.firstChild.addEventListener 'focus', activateEditor, false
```

Complete evaluation works for smaller articles, but not for a whole book. This is primarily because getting rid of old output elements with the DOM `removeChild` is very slow.

So set `@incrementalEvaluation` to `on` in larger texts.

A suitable place is the same — likely hidden — section were `@globalNames` is initialized. Code stitching is only in effect when `@incrementalEvaluation` is `off`.

```
      @incrementalEvaluation = off
      evaluateSource()
```

Bootstrap Create the embeddable HTML/JavaScript fragment as output. This does not actually bootstrap Grimoire. If it did then as soon as an error happened during editing, the whole thing would come tumbling down. To bootstrap use the Editor Commandline Commands or copy the code from the output field.

Is it possible to copyright magic? Let's say that the HTML/JavaScript code that is produced by this document is public domain or MIT licensed. Then you can use that any way you want to at your own risk. This document itself is licensed as CC-BYSA. To get the all the files this depend on go to the `interactive` directory in the repository branch `gh-pages`.

```

kup = if exports? then require 'coffeekup' else window.CoffeeKup

display = if exports?
  console.log
else
  (html) -> show _.escape html

display kup.render webfragment, format:on
```

Output

```
<div id="feature-detect">
  <div id="feature-javascript" style="color:#FF0000; display: block">
No JavaScript &rarr; no output and no interactivity.  </div>
  <div id="feature-mathml" style="color:#FF0000; display: none">
No MathML 1998 &rarr; math is not readable.  </div>
  <div id="feature-canvas" style="color:#FF0000; display: none">
No Canvas &rarr; graphical output is not rendered.  </div>
  <div id="feature-contenteditable" style="color:#FF0000; display: none">
No ContentEditable &rarr; CoffeeScript sections can not be changed.  </div>
</div>
<input class="field" type="button" value="Adjust layout" onclick="(function () {
        return this.value = toggleLayout() ? 'Layout: fixed' : 'Layout: freeflow';
      }).call(this);" />
<input class="field" type="button" value="Select editor" onclick="(function () {
        return this.value = switchEditor() ? 'Editor: CodeMirror' : 'Editor: plain text';
      }).call(this);" />
<input class="field" type="button" value="Evaluation" onclick="(function () {
        return this.value = switchEvaluation() ? '↑    Evaluate' : 'Auto Evaluate';
      }).call(this);" />
<script src="node_modules/coffee-script.js"></script>
```

```
<script src=”node_modules/coffeekup.js”></script>
<script src=”node_modules/underscore.js”></script>
<script src=”node_modules/qc.js”></script>
<script src=”codemirror/codemirror.js”></script>
<script src=”codemirror/coffeescript.js”></script>
<script>var __slice = Array.prototype.slice;var __hasProp = Object.prototype.hasOwnProperty;var __bind = function(fn, me){ return fu
      var featureDetect, switchLayout;
      this.reveal = function(instance) {
        var _ref, _ref2;
        if ((_ref = instance.getElementsByTagName(’section’)[0]) != null) {
          _ref.style.display = ’block’;
        }
        if ((_ref2 = instance.getElementsByTagName(’h5’)[0]) != null) {
          _ref2.innerHTML = ’Solution’;
        }
        return instance.onclick = void 0;
      };
      this.switchEditor = function() {
        if (typeof localStorage !== ”undefined” && localStorage !== null) {
          localStorage.editor = this.useCodeMirror ? ’TextArea’ : ’CodeMirror’;
        }
        return this.useCodeMirror = !this.useCodeMirror;
      };
      this.switchEvaluation = function() {
        if (typeof localStorage !== ”undefined” && localStorage !== null) {
          localStorage.evaluation = (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0
        }
        return (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’;
      };
      this.toggleLayout = function() {
        var fixedLayout;
        fixedLayout = document.getElementById(’page’).style.maxWidth === ’’;
        if (typeof localStorage !== ”undefined” && localStorage !== null) {
          localStorage.fixedLayout = fixedLayout;
        }
        return switchLayout(fixedLayout);
      };
      switchLayout = function(fixedLayout) {
        var s;
        document.getElementById(’page’).style.maxWidth = fixedLayout ? ’600px’ : ’’;
        s = document.getElementById(’page’).style;
        if (fixedLayout) {
          s.webkitHyphens = ’auto’;
          s.mozHyphens = ’auto’;
          s.msHyphens = ’auto’;
          s.hyphens = ’auto’;
          s.textAlign = ’justify’;
        } else {
          s.webkitHyphens = ’’;
          s.mozHyphens = ’’;
          s.msHyphens = ’’;
          s.hyphens = ’’;
          s.textAlign = ’’;
        }
        return fixedLayout;
      };
      featureDetect = function() {
        var adjustCoverPosition, mathmlDetect, used, _ref, _ref2, _ref3, _ref4, _ref5;
        adjustCoverPosition = function(idFeature) {
          var canvasCover, lineHeight, topPos, _ref;
          canvasCover = document.getElementById(’drawCanvas’);
          topPos = parseInt(canvasCover != null ? canvasCover.style.top : void 0);
          lineHeight = (_ref = document.getElementById(idFeature)) != null ? _ref.scrollHeight : void 0;
          return canvasCover != null ? canvasCover.style.top = (topPos + lineHeight) + ’px’ : void 0;
        };
        mathmlDetect = function() {
          var e, passed;
          (e = document.createElement(’div’)).innerHTML = ’<math></math>’;
          passed = e.firstChild && ’namespaceURI’ in e.firstChild && e.firstChild.namespaceURI === ’http://www.w3.org/1998/Math/Math
          return passed && !/Chrome/.test(navigator.userAgent);
        };
        if ((_ref = document.getElementById(’feature-javascript’)) != null) {
```

```
        _ref.style.display = ”none”;
      }
      used = ((_ref2 = document.getElementsByTagName(’math’)) != null ? _ref2.length : void 0) > 0;
      if (used && mathmlDetect() === false) {
        if ((_ref3 = document.getElementById(’feature-mathml’)) != null) {
          _ref3.style.display = ”block”;
        }
        adjustCoverPosition(’feature-mathml’);
      }
      if (document.createElement(’canvas’).getContext == null) {
        if ((_ref4 = document.getElementById(’feature-canvas’)) != null) {
          _ref4.style.display = ”block”;
        }
        adjustCoverPosition(’feature-canvas’);
      }
      if (!(’isContentEditable’ in document.documentElement)) {
        if ((_ref5 = document.getElementById(’feature-contenteditable’)) != null) {
          _ref5.style.display = ”block”;
        }
        return adjustCoverPosition(’feature-contenteditable’);
      }
    };
    return window.onload = function() {
      var activateEditor, addElement, addFrame, canvas, clearCanvas, cmAutoEvaluation, cmEvaluation, cmManualEvaluation, createEdi
      featureDetect();
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.fixedLayout : void 0) !== ’false’) {
        switchLayout(true);
      }
      this.useCodeMirror = (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.editor : void 0) === ’CodeM
      canvas = document.getElementById(’drawCanvas’);
      if (canvas != null) window.ctx = canvas.getContext(’2d’);
      clearCanvas = function() {
        if (window.ctx != null) {
          return window.ctx.clearRect(0, 0, window.ctx.canvas.width, window.ctx.canvas.height);
        }
      };
      drawCanvas = function(draw) {
        clearCanvas();
        return draw(window.ctx);
      };
      getParent = function(child) {
        var _ref;
        return (_ref = child != null ? child.parentElement : void 0) != null ? _ref : child != null ? child.parentNode : void 0;
      };
      addElement = function(parent, text) {
        var newelem;
        newelem = document.createElement(’code’);
        newelem.setAttribute(’class’, ’sourceCode output’);
        newelem.innerHTML = text;
        return parent.appendChild(newelem);
      };
      separator = function(parent) {
        if (parent.getElementsByClassName(’output’).length === 0) {
          return addElement(parent, ’<hr><br>’);
        }
      };
      addFrame = function(parent, width, height, content) {
        var newelem;
        newelem = document.createElement(’iframe’);
        newelem.setAttribute(’class’, ’output’);
        newelem.setAttribute(’width’, width);
        newelem.setAttribute(’height’, height);
        newelem.setAttribute(’src’, ”data:text/html;charset=utf-8,” + (encodeURIComponent(content)));
        newelem.innerHTML = ’<div id=”feature-dataurl” style=”color:#FF0000; display: block”>\nNo or insufficient Data URL support
        return parent.appendChild(newelem);
      };
      showHere = function(atTag, obj, shallow, symbol) {
        var displayShallow, msg, parentTag;
        if (shallow == null) shallow = false;
        if (symbol == null) symbol = ’&rarr;’;
        displayShallow = function(o) {
          var k, v;
          return ”{” + ((function() {
            var _results;
            _results = [];
            for (k in o) {
              if (!__hasProp.call(o, k)) continue;
              v = o[k];
              _results.push(”\n  ” + k + ”: ” + v);
            }
            return _results;
          })()) + ”\n}”;
        };
        msg = (function() {
          switch (typeof obj) {
            case ’undefined’:
            case ’string’:
            case ’function’:
              return obj;
            case ’object’:
              if (shallow) {
                return displayShallow(obj);
              } else {
                try {
                  return JSON.stringify(obj);
                } catch (err) {
                  return displayShallow(obj);
                }
              }
              break;
            default:
              return obj != null ? obj.toString() : void 0;
          }
        })();
        parentTag = getParent(atTag);
        separator(parentTag);
        addElement(parentTag, ”” + symbol + ” ” + msg + ”<br>”);
        return obj;
      };
      showDocumentHere = function(atTag, content, width, height) {
        var parentTag;
        if (width == null) width = 300;
        if (height == null) height = 300;
        parentTag = getParent(atTag);
        separator(parentTag);
        addFrame(parentTag, width, height, content);
      };
      evaluateSource = function(field) {
        var HtmlListener, addErrorElement, codeSegment, codeTag, confirm, currentSegment, currentTag, elem, getCode, getCurrentTag
        if (field == null) field = null;
        show = function(obj, shallow, symbol) {
          if (shallow == null) shallow = false;
          if (symbol == null) symbol = ’&rarr;’;
          showHere(currentTag, obj, shallow, symbol);
        };
        view = function(obj, shallow, symbol) {
          if (shallow == null) shallow = false;
          if (symbol == null) symbol = ’&rarr;’;
          return showHere(currentTag, obj, shallow, symbol);
        };
        showDocument = function(content, width, height) {
          if (width == null) width = 300;
          if (height == null) height = 300;
          return showDocumentHere(currentTag, content, width, height);
        };
        addErrorElement = function(text) {
          return show(”<span class=\”er\”>” + text + ”</span>”);
        };
        getCurrentTagId = function(id) {
          return id.match(/button-\w+-(.*)/)[1];
        };
        runOnDemand = function(func) {
          show(”<button id=’button-run-” + currentTag.id + ”’> Run </button>”);
          document.getElementById(”button-run-” + currentTag.id).onclick = function() {
```

```
    var currentTag;
    currentTag = document.getElementById(getCurrentTagId(this.id));
    view = show = function(obj, shallow, symbol) {
      if (shallow == null) shallow = false;
      if (symbol == null) symbol = ’&rArr;’;
      return showHere(currentTag, obj, shallow, symbol);
    };
    showDocument = function(content, width, height) {
      if (width == null) width = 300;
      if (height == null) height = 300;
      return showDocumentHere(currentTag, content, width, height);
    };
    return func();
  };
};
confirm = function(message, func) {
  show((”” + message) + (”  <button id=’button-yes-” + currentTag.id + ”’> Yes </button>”) + (”  <button id=’button-no-” +
  document.getElementById(”button-yes-” + currentTag.id).onclick = function() {
    var currentTag;
    currentTag = document.getElementById(getCurrentTagId(this.id));
    view = show = function(obj, shallow, symbol) {
      if (shallow == null) shallow = false;
      if (symbol == null) symbol = ’&rArr;’;
      return showHere(currentTag, obj, shallow, symbol);
    };
    showDocument = function(content, width, height) {
      if (width == null) width = 300;
      if (height == null) height = 300;
      return showDocumentHere(currentTag, content, width, height);
    };
    return func(true);
  };
  document.getElementById(”button-no-” + currentTag.id).onclick = function() {
    var currentTag;
    currentTag = document.getElementById(getCurrentTagId(this.id));
    view = show = function(obj, shallow, symbol) {
      if (shallow == null) shallow = false;
      if (symbol == null) symbol = ’&rArr;’;
      return showHere(currentTag, obj, shallow, symbol);
    };
    showDocument = function(content, width, height) {
      if (width == null) width = 300;
      if (height == null) height = 300;
      return showDocumentHere(currentTag, content, width, height);
    };
    return func(false);
  };
};
prompt = function(message, defaultValue, func) {
  show((”” + message) + (”  <input id=’input-prompt-” + currentTag.id + ”’ value=’” + defaultValue + ”’ />”) + (”  <button
  document.getElementById(”button-prompt-” + currentTag.id).onclick = function() {
    var currentTag;
    currentTag = document.getElementById(getCurrentTagId(this.id));
    view = show = function(obj, shallow, symbol) {
      if (shallow == null) shallow = false;
      if (symbol == null) symbol = ’&rArr;’;
      return showHere(currentTag, obj, shallow, symbol);
    };
    showDocument = function(content, width, height) {
      if (width == null) width = 300;
      if (height == null) height = 300;
      return showDocumentHere(currentTag, content, width, height);
    };
    return func(document.getElementById(”input-prompt-” + (getCurrentTagId(this.id))).value);
  };
};
HtmlListener = (function(_super) {

  __extends(HtmlListener, _super);

  function HtmlListener(maxCollected) {
    this.maxCollected = maxCollected != null ? maxCollected : 10;
```

```
      }

      HtmlListener.prototype.log = function(str) {
        return show(str);
      };

      HtmlListener.prototype.passed = function(str) {
        return show(”<span class=\”kw\”>” + str + ”</span>”);
      };

      HtmlListener.prototype.invalid = function(str) {
        return show(”<span class=\”dt\”>” + str + ”</span>”);
      };

      HtmlListener.prototype.failure = function(str) {
        return show(”<span class=\”al\”>” + str + ”</span>”);
      };

      HtmlListener.prototype.done = function() {
        show(’Completed test.’);
        return resetProps();
      };

      return HtmlListener;

    })(ConsoleListener);
    Case.prototype.note = function(a) {
      this.noteArg(a);
      return a;
    };
    Case.prototype.noteVerbose = function(a) {
      this.noteArg(a);
      show(this.args);
      return a;
    };
    testPure = function(func, types, name, property) {
      declare(name, types, function() {
        var a, c;
        c = arguments[0], a = 2 <= arguments.length ? __slice.call(arguments, 1) : [];
        return c.assert(property.apply(null, [c].concat(__slice.call(a), [c.note(func.apply(null, a))])));
      });
    };
    qcConfig = new Config(100, 1000);
    test = function(msg, func) {
      return _.each([msg, func, runAllProps(qcConfig, new HtmlListener)], function(o) {
        if (!_.isUndefined(o)) return show(o);
      });
    };
    getCode = function(codeTag) {
      var code, segment;
      if (codeTag.value != null) {
        segment = codeTag.value;
      } else {
        segment = codeTag.innerHTML;
      }
      code = segment.replace(/<br>/g, ’\n’);
      code = code.replace(/<[\/]?span[^>]*>/g, ’’);
      code = code.replace(/[&]gt;/g, ’>’);
      code = code.replace(/[&]lt;/g, ’<’);
      code = code.replace(/[&]nbsp;/g, ’ ’);
      return {
        codeTag: codeTag,
        code: code
      };
    };
    if ((field != null) && this.incrementalEvaluation) {
      segments = [getCode(field)];
    } else {
      segments = (function() {
        var _i, _len, _ref, _results;
        _ref = window.document.getElementsByTagName(’pre’);
        _results = [];
```

```
          for (_i = 0, _len = _ref.length; _i < _len; _i++) {
            codeSegment = _ref[_i];
            if (codeSegment.className === ’sourceCode’) {
              codeTag = codeSegment.firstChild;
              if (codeTag.className === ’sourceCode coffeescript’ || codeTag.className === ’sourceCode CoffeeScript’) {
                _results.push(getCode(codeTag));
              } else {
                _results.push(void 0);
              }
            } else {
              _results.push(void 0);
            }
          }
          return _results;
        })();
        segments = (function() {
          var _i, _len, _results;
          _results = [];
          for (_i = 0, _len = segments.length; _i < _len; _i++) {
            segment = segments[_i];
            if (segment != null) _results.push(segment);
          }
          return _results;
        })();
        for (i = 0, _len = segments.length; i < _len; i++) {
          segment = segments[i];
          if (i > 0 && /^[\s]/.test(segment.code)) {
            segment.code = segments[i - 1].code + ’\n’ + segment.code;
            segments[i - 1] = void 0;
          }
        }
        segments = (function() {
          var _i, _len2, _results;
          _results = [];
          for (_i = 0, _len2 = segments.length; _i < _len2; _i++) {
            segment = segments[_i];
            if (segment != null) _results.push(segment);
          }
          return _results;
        })();
      }
      for (_i = 0, _len2 = segments.length; _i < _len2; _i++) {
        segment = segments[_i];
        outList = getParent(segment.codeTag).getElementsByClassName(’output’);
        outLength = outList != null ? outList.length : void 0;
        while (outList && outLength > 0) {
          elem = outList[--outLength];
          getParent(elem).removeChild(elem);
        }
      }
      incredibleLength = segments.length;
      _fn = function(currentTag) {
        var draw, incredibleResult, magic, promoteToGlobal, _j, _len3, _ref;
        try {
          draw = void 0;
          incredibleResult = eval(CoffeeScript.compile(currentSegment.code, {
            bare: true
          }));
          if (draw != null) drawCanvas(draw);
          if (incredibleResult !== void 0 && typeof incredibleResult !== ’function’) {
            show(incredibleResult, true, ’&crarr;’);
          }
          if (typeof globalNames !== ”undefined” && globalNames !== null) {
            promoteToGlobal = function(magic) {
              var magicValue;
              try {
                magicValue = eval(magic);
              } catch (error) {
                return;
              }
              return this[magic] = magicValue;
            };
            _ref = globalNames.split(’ ’);
            for (_j = 0, _len3 = _ref.length; _j < _len3; _j++) {
              magic = _ref[_j];
              promoteToGlobal(magic);
            }
          }
        } catch (error) {
          console.log(error.message);
          addErrorElement(error);
        }
      };
      for (incredibleIndex = 0; incredibleIndex < incredibleLength; incredibleIndex += 1) {
        incredibleStart = new Date();
        currentSegment = segments[incredibleIndex];
        currentTag = currentSegment.codeTag;
        if (currentTag.id === ’’) currentTag.id = ”codeTag” + incredibleIndex;
        _fn(currentTag);
        if ((typeof showTiming !== ”undefined” && showTiming !== null) && showTiming === true) {
          incredibleTime = Number(0.001 * (new Date() - incredibleStart)).toFixed(3);
          show(”” + incredibleTime + ”s”, true, ’ ’);
        }
      }
    };
    keyDownEvaluation = function(evt) {
      var _ref;
      if (evt == null) return;
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’) {
        if (evt.keyCode === 13 && evt.shiftKey) {
          evt.preventDefault();
          evaluateSource(evt.currentTarget);
        }
      }
      if (evt.keyCode === 13 && !evt.shiftKey) {
        if ((_ref = evt.target) != null) _ref.rows++;
      }
    };
    keyUpEvaluation = function(evt) {
      var _ref;
      if (evt == null) return;
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) !== ’manual’) {
        if ((_ref = evt.keyCode) !== 16 && _ref !== 37 && _ref !== 38 && _ref !== 39 && _ref !== 40) {
          return evaluateSource(evt.currentTarget);
        }
      }
    };
    cmEvaluation = function(editor) {
      editor.save();
      return evaluateSource(editor.getTextArea());
    };
    cmAutoEvaluation = function(editor) {
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) !== ’manual’) {
        return cmEvaluation(editor);
      }
    };
    cmManualEvaluation = function(editor) {
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’) {
        return cmEvaluation(editor);
      }
    };
    createEditor = function(codeElement, text) {
      var c, countLines, elemCodeMirror, newelem;
      newelem = document.createElement(’textarea’);
      newelem.setAttribute(’id’, codeElement.id);
      newelem.setAttribute(’class’, codeElement.getAttribute(’class’));
      countLines = ((function() {
        var _i, _len, _results;
        _results = [];
        for (_i = 0, _len = text.length; _i < _len; _i++) {
          c = text[_i];
          if (c === ’\n’) _results.push(c);
        }
        return _results;
```

```
          })()).length + 1;
          newelem.setAttribute(’rows’, countLines);
          newelem.setAttribute(’style’, ’width: 98%;’);
          newelem.setAttribute(’autofocus’, ’true’);
          newelem.setAttribute(’spellcheck’, ’false’);
          newelem.innerHTML = text;
          newelem.addEventListener(’keydown’, keyDownEvaluation, false);
          newelem.addEventListener(’keyup’, keyUpEvaluation, false);
          getParent(codeElement).replaceChild(newelem, codeElement);
          if (this.useCodeMirror) {
            return elemCodeMirror = CodeMirror.fromTextArea(newelem, {
              onChange: cmAutoEvaluation,
              extraKeys: {
                ’Shift-Enter’: cmManualEvaluation
              },
              lineNumbers: true,
              theme: ’pantheme’
            });
          }
        };
        activateEditor = function() {
          var sourcecode;
          this.removeEventListener(’focus’, activateEditor, false);
          sourcecode = this.innerHTML.toString().replace(/<\/?span[^>]*>/g, ’’);
          sourcecode = sourcecode.replace(/<br[ ]*[^>]*>/g, ’\n’);
          return createEditor(this, sourcecode);
        };
        _ref = document.getElementsByTagName(’pre’);
        for (_i = 0, _len = _ref.length; _i < _len; _i++) {
          segment = _ref[_i];
          segment.firstChild.addEventListener(’focus’, activateEditor, false);
        }
        this.incrementalEvaluation = false;
        return evaluateSource();
      };
    }).call(this);</script>

```

JavaScript

```
(function() {
  var display, kup, show, showDocument, webfragment,
    __hasProp = Object.prototype.hasOwnProperty,
    __extends = function(child, parent) { for (var key in parent) { if (__hasProp.call(parent, key)) child[key] = parent[key]; } fun
    __slice = Array.prototype.slice;

  show = console.log;

  showDocument = function(doc, width, height) {
    return show(doc);
  };

  webfragment = function() {
    div({
      id: ”feature-detect”
    }, function() {
      div({
        id: ”feature-javascript”,
        style: ”color:#FF0000; display: block”
      }, function() {
        return text(”No JavaScript &rarr; no output and no interactivity.”);
      });
      div({
        id: ”feature-mathml”,
        style: ”color:#FF0000; display: none”
      }, function() {
        return text(”No MathML 1998 &rarr; math is not readable.”);
      });
      div({
        id: ”feature-canvas”,
        style: ”color:#FF0000; display: none”
      }, function() {
        return text(”No Canvas &rarr; graphical output is not rendered.”);
      });
      return div({
        id: ”feature-contenteditable”,
        style: ”color:#FF0000; display: none”
      }, function() {
        return text(”No ContentEditable &rarr; CoffeeScript sections can not be changed.”);
      });
    });
    input({
      ”class”: ’field’,
      type: ’button’,
      value: ’Adjust layout’,
      onclick: function() {
        return this.value = toggleLayout() ? ’Layout: fixed’ : ’Layout: freeflow’;
      }
    });
    input({
      ”class”: ’field’,
      type: ’button’,
      value: ’Select editor’,
      onclick: function() {
        return this.value = switchEditor() ? ’Editor: CodeMirror’ : ’Editor: plain text’;
      }
    });
    input({
      ”class”: ’field’,
      type: ’button’,
      value: ’Evaluation’,
      onclick: function() {
        return this.value = switchEvaluation() ? ’↑    Evaluate’ : ’Auto Evaluate’;
      }
    });
    script({
      src: ’node_modules/coffee-script.js’
    });
    script({
      src: ’node_modules/coffeekup.js’
    });
    script({
      src: ’node_modules/underscore.js’
    });
    script({
      src: ’node_modules/qc.js’
    });
    script({
      src: ’codemirror/codemirror.js’
    });
    script({
      src: ’codemirror/coffeescript.js’
    });
    return coffeescript(function() {
      var featureDetect, switchLayout;
      this.reveal = function(instance) {
        var _ref, _ref2;
        if ((_ref = instance.getElementsByTagName(’section’)[0]) != null) {
          _ref.style.display = ’block’;
        }
        if ((_ref2 = instance.getElementsByTagName(’h5’)[0]) != null) {
          _ref2.innerHTML = ’Solution’;
        }
        return instance.onclick = void 0;
      };
      this.switchEditor = function() {
        if (typeof localStorage !== ”undefined” && localStorage !== null) {
          localStorage.editor = this.useCodeMirror ? ’TextArea’ : ’CodeMirror’;
        }
        return this.useCodeMirror = !this.useCodeMirror;
      };
      this.switchEvaluation = function() {
```

```
      if (typeof localStorage !== ”undefined” && localStorage !== null) {
        localStorage.evaluation = (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0
      }
      return (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’;
    };
    this.toggleLayout = function() {
      var fixedLayout;
      fixedLayout = document.getElementById(’page’).style.maxWidth === ’’;
      if (typeof localStorage !== ”undefined” && localStorage !== null) {
        localStorage.fixedLayout = fixedLayout;
      }
      return switchLayout(fixedLayout);
    };
    switchLayout = function(fixedLayout) {
      var s;
      document.getElementById(’page’).style.maxWidth = fixedLayout ? ’600px’ : ’’;
      s = document.getElementById(’page’).style;
      if (fixedLayout) {
        s.webkitHyphens = ’auto’;
        s.mozHyphens = ’auto’;
        s.msHyphens = ’auto’;
        s.hyphens = ’auto’;
        s.textAlign = ’justify’;
      } else {
        s.webkitHyphens = ’’;
        s.mozHyphens = ’’;
        s.msHyphens = ’’;
        s.hyphens = ’’;
        s.textAlign = ’’;
      }
      return fixedLayout;
    };
    featureDetect = function() {
      var adjustCoverPosition, mathmlDetect, used, _ref, _ref2, _ref3, _ref4, _ref5;
      adjustCoverPosition = function(idFeature) {
        var canvasCover, lineHeight, topPos, _ref;
        canvasCover = document.getElementById(’drawCanvas’);
        topPos = parseInt(canvasCover != null ? canvasCover.style.top : void 0);
        lineHeight = (_ref = document.getElementById(idFeature)) != null ? _ref.scrollHeight : void 0;
        return canvasCover != null ? canvasCover.style.top = (topPos + lineHeight) + ’px’ : void 0;
      };
      mathmlDetect = function() {
        var e, passed;
        (e = document.createElement(’div’)).innerHTML = ’<math></math>’;
        passed = e.firstChild && ’namespaceURI’ in e.firstChild && e.firstChild.namespaceURI === ’http://www.w3.org/1998/Math/MathM
        return passed && !/Chrome/.test(navigator.userAgent);
      };
      if ((_ref = document.getElementById(’feature-javascript’)) != null) {
        _ref.style.display = ”none”;
      }
      used = ((_ref2 = document.getElementsByTagName(’math’)) != null ? _ref2.length : void 0) > 0;
      if (used && mathmlDetect() === false) {
        if ((_ref3 = document.getElementById(’feature-mathml’)) != null) {
          _ref3.style.display = ”block”;
        }
        adjustCoverPosition(’feature-mathml’);
      }
      if (document.createElement(’canvas’).getContext == null) {
        if ((_ref4 = document.getElementById(’feature-canvas’)) != null) {
          _ref4.style.display = ”block”;
        }
        adjustCoverPosition(’feature-canvas’);
      }
      if (!(’isContentEditable’ in document.documentElement)) {
        if ((_ref5 = document.getElementById(’feature-contenteditable’)) != null) {
          _ref5.style.display = ”block”;
        }
        return adjustCoverPosition(’feature-contenteditable’);
      }
    };
    return window.onload = function() {
      var activateEditor, addElement, addFrame, canvas, clearCanvas, cmAutoEvaluation, cmEvaluation, cmManualEvaluation, createEdi
```

```
featureDetect();
if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.fixedLayout : void 0) !== ’false’) {
  switchLayout(true);
}
this.useCodeMirror = (typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.editor : void 0) === ’CodeM
canvas = document.getElementById(’drawCanvas’);
if (canvas != null) window.ctx = canvas.getContext(’2d’);
clearCanvas = function() {
  if (window.ctx != null) {
    return window.ctx.clearRect(0, 0, window.ctx.canvas.width, window.ctx.canvas.height);
  }
};
drawCanvas = function(draw) {
  clearCanvas();
  return draw(window.ctx);
};
getParent = function(child) {
  var _ref;
  return (_ref = child != null ? child.parentElement : void 0) != null ? _ref : child != null ? child.parentNode : void 0;
};
addElement = function(parent, text) {
  var newelem;
  newelem = document.createElement(’code’);
  newelem.setAttribute(’class’, ’sourceCode output’);
  newelem.innerHTML = text;
  return parent.appendChild(newelem);
};
separator = function(parent) {
  if (parent.getElementsByClassName(’output’).length === 0) {
    return addElement(parent, ’<hr><br>’);
  }
};
addFrame = function(parent, width, height, content) {
  var newelem;
  newelem = document.createElement(’iframe’);
  newelem.setAttribute(’class’, ’output’);
  newelem.setAttribute(’width’, width);
  newelem.setAttribute(’height’, height);
  newelem.setAttribute(’src’, ”data:text/html;charset=utf-8,” + (encodeURIComponent(content)));
  newelem.innerHTML = ’<div id=”feature-dataurl” style=”color:#FF0000; display: block”>\nNo or insufficient Data URL support
  return parent.appendChild(newelem);
};
showHere = function(atTag, obj, shallow, symbol) {
  var displayShallow, msg, parentTag;
  if (shallow == null) shallow = false;
  if (symbol == null) symbol = ’&rarr;’;
  displayShallow = function(o) {
    var k, v;
    return ”{” + ((function() {
      var _results;
      _results = [];
      for (k in o) {
        if (!__hasProp.call(o, k)) continue;
        v = o[k];
        _results.push(”\n  ” + k + ”: ” + v);
      }
      return _results;
    })()) + ”\n}”;
  };
  msg = (function() {
    switch (typeof obj) {
      case ’undefined’:
      case ’string’:
      case ’function’:
        return obj;
      case ’object’:
        if (shallow) {
          return displayShallow(obj);
        } else {
          try {
            return JSON.stringify(obj);
          } catch (err) {
            return displayShallow(obj);
          }
        }
        break;
      default:
        return obj != null ? obj.toString() : void 0;
    }
  })();
  parentTag = getParent(atTag);
  separator(parentTag);
  addElement(parentTag, ”” + symbol + ” ” + msg + ”<br>”);
  return obj;
};
showDocumentHere = function(atTag, content, width, height) {
  var parentTag;
  if (width == null) width = 300;
  if (height == null) height = 300;
  parentTag = getParent(atTag);
  separator(parentTag);
  addFrame(parentTag, width, height, content);
};
evaluateSource = function(field) {
  var HtmlListener, addErrorElement, codeSegment, codeTag, confirm, currentSegment, currentTag, elem, getCode, getCurrentTag
  if (field == null) field = null;
  show = function(obj, shallow, symbol) {
    if (shallow == null) shallow = false;
    if (symbol == null) symbol = ’&rarr;’;
    showHere(currentTag, obj, shallow, symbol);
  };
  view = function(obj, shallow, symbol) {
    if (shallow == null) shallow = false;
    if (symbol == null) symbol = ’&rarr;’;
    return showHere(currentTag, obj, shallow, symbol);
  };
  showDocument = function(content, width, height) {
    if (width == null) width = 300;
    if (height == null) height = 300;
    return showDocumentHere(currentTag, content, width, height);
  };
  addErrorElement = function(text) {
    return show(”<span class=\”er\”>” + text + ”</span>”);
  };
  getCurrentTagId = function(id) {
    return id.match(/button-\w+-(.*)/)[1];
  };
  runOnDemand = function(func) {
    show(”<button id=’button-run-” + currentTag.id + ”’> Run </button>”);
    document.getElementById(”button-run-” + currentTag.id).onclick = function() {
      var currentTag;
      currentTag = document.getElementById(getCurrentTagId(this.id));
      view = show = function(obj, shallow, symbol) {
        if (shallow == null) shallow = false;
        if (symbol == null) symbol = ’&rArr;’;
        return showHere(currentTag, obj, shallow, symbol);
      };
      showDocument = function(content, width, height) {
        if (width == null) width = 300;
        if (height == null) height = 300;
        return showDocumentHere(currentTag, content, width, height);
      };
      return func();
    };
  };
  confirm = function(message, func) {
    show((”” + message) + (”  <button id=’button-yes-” + currentTag.id + ”’> Yes </button>”) + (”  <button id=’button-no-” +
    document.getElementById(”button-yes-” + currentTag.id).onclick = function() {
      var currentTag;
      currentTag = document.getElementById(getCurrentTagId(this.id));
      view = show = function(obj, shallow, symbol) {
        if (shallow == null) shallow = false;
        if (symbol == null) symbol = ’&rArr;’;
        return showHere(currentTag, obj, shallow, symbol);
      };
      showDocument = function(content, width, height) {
        if (width == null) width = 300;
        if (height == null) height = 300;
        return showDocumentHere(currentTag, content, width, height);
      };
      return func(true);
    };
    document.getElementById(”button-no-” + currentTag.id).onclick = function() {
      var currentTag;
      currentTag = document.getElementById(getCurrentTagId(this.id));
      view = show = function(obj, shallow, symbol) {
        if (shallow == null) shallow = false;
        if (symbol == null) symbol = ’&rArr;’;
        return showHere(currentTag, obj, shallow, symbol);
      };
      showDocument = function(content, width, height) {
        if (width == null) width = 300;
        if (height == null) height = 300;
        return showDocumentHere(currentTag, content, width, height);
      };
      return func(false);
    };
  };
  prompt = function(message, defaultValue, func) {
    show((”” + message) + (”  <input id=’input-prompt-” + currentTag.id + ”’ value=’” + defaultValue + ”’ />”) + (”  <button
    document.getElementById(”button-prompt-” + currentTag.id).onclick = function() {
      var currentTag;
      currentTag = document.getElementById(getCurrentTagId(this.id));
      view = show = function(obj, shallow, symbol) {
        if (shallow == null) shallow = false;
        if (symbol == null) symbol = ’&rArr;’;
        return showHere(currentTag, obj, shallow, symbol);
      };
      showDocument = function(content, width, height) {
        if (width == null) width = 300;
        if (height == null) height = 300;
        return showDocumentHere(currentTag, content, width, height);
      };
      return func(document.getElementById(”input-prompt-” + (getCurrentTagId(this.id))).value);
    };
  };
  HtmlListener = (function(_super) {

    __extends(HtmlListener, _super);

    function HtmlListener(maxCollected) {
      this.maxCollected = maxCollected != null ? maxCollected : 10;
    }

    HtmlListener.prototype.log = function(str) {
      return show(str);
    };

    HtmlListener.prototype.passed = function(str) {
      return show(”<span class=\”kw\”>” + str + ”</span>”);
    };

    HtmlListener.prototype.invalid = function(str) {
      return show(”<span class=\”dt\”>” + str + ”</span>”);
    };

    HtmlListener.prototype.failure = function(str) {
      return show(”<span class=\”al\”>” + str + ”</span>”);
    };

    HtmlListener.prototype.done = function() {
      show(’Completed test.’);
      return resetProps();
    };

    return HtmlListener;

  })(ConsoleListener);
  Case.prototype.note = function(a) {
    this.noteArg(a);
    return a;
  };
  Case.prototype.noteVerbose = function(a) {
    this.noteArg(a);
    show(this.args);
    return a;
  };
  testPure = function(func, types, name, property) {
    declare(name, types, function() {
      var a, c;
      c = arguments[0], a = 2 <= arguments.length ? __slice.call(arguments, 1) : [];
      return c.assert(property.apply(null, [c].concat(__slice.call(a), [c.note(func.apply(null, a))])));
    });
  };
  qcConfig = new Config(100, 1000);
  test = function(msg, func) {
    return _.each([msg, func, runAllProps(qcConfig, new HtmlListener)], function(o) {
      if (!_.isUndefined(o)) return show(o);
    });
  };
  getCode = function(codeTag) {
    var code, segment;
    if (codeTag.value != null) {
      segment = codeTag.value;
    } else {
      segment = codeTag.innerHTML;
    }
    code = segment.replace(/<br>/g, ’\n’);
    code = code.replace(/<[\/]?span[^>]*>/g, ’’);
    code = code.replace(/[&]gt;/g, ’>’);
    code = code.replace(/[&]lt;/g, ’<’);
    code = code.replace(/[&]nbsp;/g, ’ ’);
    return {
      codeTag: codeTag,
      code: code
    };
  };
  if ((field != null) && this.incrementalEvaluation) {
    segments = [getCode(field)];
  } else {
    segments = (function() {
      var _i, _len, _ref, _results;
      _ref = window.document.getElementsByTagName(’pre’);
      _results = [];
      for (_i = 0, _len = _ref.length; _i < _len; _i++) {
        codeSegment = _ref[_i];
        if (codeSegment.className === ’sourceCode’) {
          codeTag = codeSegment.firstChild;
          if (codeTag.className === ’sourceCode coffeescript’ || codeTag.className === ’sourceCode CoffeeScript’) {
            _results.push(getCode(codeTag));
          } else {
            _results.push(void 0);
          }
        } else {
          _results.push(void 0);
        }
      }
      return _results;
    })();
    segments = (function() {
      var _i, _len, _results;
      _results = [];
      for (_i = 0, _len = segments.length; _i < _len; _i++) {
        segment = segments[_i];
        if (segment != null) _results.push(segment);
      }
      return _results;
    })();
```

```
        for (i = 0, _len = segments.length; i < _len; i++) {
          segment = segments[i];
          if (i > 0 && /^[\s]/.test(segment.code)) {
            segment.code = segments[i - 1].code + ’\n’ + segment.code;
            segments[i - 1] = void 0;
          }
        }
        segments = (function() {
          var _i, _len2, _results;
          _results = [];
          for (_i = 0, _len2 = segments.length; _i < _len2; _i++) {
            segment = segments[_i];
            if (segment != null) _results.push(segment);
          }
          return _results;
        })();
      }
      for (_i = 0, _len2 = segments.length; _i < _len2; _i++) {
        segment = segments[_i];
        outList = getParent(segment.codeTag).getElementsByClassName(’output’);
        outLength = outList != null ? outList.length : void 0;
        while (outList && outLength > 0) {
          elem = outList[--outLength];
          getParent(elem).removeChild(elem);
        }
      }
      incredibleLength = segments.length;
      _fn = function(currentTag) {
        var draw, incredibleResult, magic, promoteToGlobal, _j, _len3, _ref;
        try {
          draw = void 0;
          incredibleResult = eval(CoffeeScript.compile(currentSegment.code, {
            bare: true
          }));
          if (draw != null) drawCanvas(draw);
          if (incredibleResult !== void 0 && typeof incredibleResult !== ’function’) {
            show(incredibleResult, true, ’&crarr;’);
          }
          if (typeof globalNames !== ”undefined” && globalNames !== null) {
            promoteToGlobal = function(magic) {
              var magicValue;
              try {
                magicValue = eval(magic);
              } catch (error) {
                return;
              }
              return this[magic] = magicValue;
            };
            _ref = globalNames.split(’ ’);
            for (_j = 0, _len3 = _ref.length; _j < _len3; _j++) {
              magic = _ref[_j];
              promoteToGlobal(magic);
            }
          }
        } catch (error) {
          console.log(error.message);
          addErrorElement(error);
        }
      };
      for (incredibleIndex = 0; incredibleIndex < incredibleLength; incredibleIndex += 1) {
        incredibleStart = new Date();
        currentSegment = segments[incredibleIndex];
        currentTag = currentSegment.codeTag;
        if (currentTag.id === ’’) currentTag.id = ”codeTag” + incredibleIndex;
        _fn(currentTag);
        if ((typeof showTiming !== ”undefined” && showTiming !== null) && showTiming === true) {
          incredibleTime = Number(0.001 * (new Date() - incredibleStart)).toFixed(3);
          show(”” + incredibleTime + ”s”, true, ’ ’);
        }
      }
    };
    keyDownEvaluation = function(evt) {
      var _ref;
      if (evt == null) return;
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’) {
        if (evt.keyCode === 13 && evt.shiftKey) {
          evt.preventDefault();
          evaluateSource(evt.currentTarget);
        }
      }
      if (evt.keyCode === 13 && !evt.shiftKey) {
        if ((_ref = evt.target) != null) _ref.rows++;
      }
    };
    keyUpEvaluation = function(evt) {
      var _ref;
      if (evt == null) return;
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) !== ’manual’) {
        if ((_ref = evt.keyCode) !== 16 && _ref !== 37 && _ref !== 38 && _ref !== 39 && _ref !== 40) {
          return evaluateSource(evt.currentTarget);
        }
      }
    };
    cmEvaluation = function(editor) {
      editor.save();
      return evaluateSource(editor.getTextArea());
    };
    cmAutoEvaluation = function(editor) {
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) !== ’manual’) {
        return cmEvaluation(editor);
      }
    };
    cmManualEvaluation = function(editor) {
      if ((typeof localStorage !== ”undefined” && localStorage !== null ? localStorage.evaluation : void 0) === ’manual’) {
        return cmEvaluation(editor);
      }
    };
    createEditor = function(codeElement, text) {
      var c, countLines, elemCodeMirror, newelem;
      newelem = document.createElement(’textarea’);
      newelem.setAttribute(’id’, codeElement.id);
      newelem.setAttribute(’class’, codeElement.getAttribute(’class’));
      countLines = ((function() {
        var _i, _len, _results;
        _results = [];
        for (_i = 0, _len = text.length; _i < _len; _i++) {
          c = text[_i];
          if (c === ’\n’) _results.push(c);
        }
        return _results;
      })()).length + 1;
      newelem.setAttribute(’rows’, countLines);
      newelem.setAttribute(’style’, ’width: 98%;’);
      newelem.setAttribute(’autofocus’, ’true’);
      newelem.setAttribute(’spellcheck’, ’false’);
      newelem.innerHTML = text;
      newelem.addEventListener(’keydown’, keyDownEvaluation, false);
      newelem.addEventListener(’keyup’, keyUpEvaluation, false);
      getParent(codeElement).replaceChild(newelem, codeElement);
      if (this.useCodeMirror) {
        return elemCodeMirror = CodeMirror.fromTextArea(newelem, {
          onChange: cmAutoEvaluation,
          extraKeys: {
            ’Shift-Enter’: cmManualEvaluation
          },
          lineNumbers: true,
          theme: ’pantheme’
        });
      }
    };
    activateEditor = function() {
      var sourcecode;
      this.removeEventListener(’focus’, activateEditor, false);
      sourcecode = this.innerHTML.toString().replace(/<\/?span[^>]*>/g, ’’);
```

```
          sourcecode = sourcecode.replace(/<br[ ]*[^>]*>/g, ’\n’);
          return createEditor(this, sourcecode);
        };
        _ref = document.getElementsByTagName(’pre’);
        for (_i = 0, _len = _ref.length; _i < _len; _i++) {
          segment = _ref[_i];
          segment.firstChild.addEventListener(’focus’, activateEditor, false);
        }
        this.incrementalEvaluation = false;
        return evaluateSource();
      };
    });
  };

  kup = typeof exports !== ”undefined” && exports !== null ? require(’coffeekup’) : window.CoffeeKup;

  display = typeof exports !== ”undefined” && exports !== null ? console.log : function(html) {
    return show(_.escape(html));
  };

  display(kup.render(webfragment, {
    format: true
  }));

}).call(this);
```

Formats Standalone CoffeeScript Markdown PDF HTML