Introducing Repla: A Live-Coding Tool

How to make innovating programming tools that are compatible with existing workflows.

Roben Kleene
@robenkleene
robenkleene.com
github.com/robenkleene/repla-live-coding

Who Am I

— Developer for Apple platforms (iOS & Mac)
— Currently making a programming tool called Repla
— Left job managing WSJ iOS app team to make it
— Not a researcher, trying to create a business, not a boundary pusher

Simple Poll APP 4:46 AM

The thing I envision for the future of coding is for:

1 Programmers 16

@Miles Sabin, @David Piepgrass, @yairchu, @Wouter, @robenkleene, @Will, @Kartik Agaram, @John Austin, @Brian Hempel, @missingfaktor, @Tony Cheal, @wtaysom, @Stathis, @Vladimir Gordeev, @gman, @piggy1

2 Non Programmers 15

@Philipp Krüger, @Stefan, @ahmedr, @Niluka Satharasinghe, @yairchu, @Duncan Cragg, @Jacob Sandlund, @Peter Abrahamsen, @Scott Anderson, @jonathoda, @missingfaktor, @Tony Cheal, @Robin, @Nick Smith, @Tim Swast

1 2 Delete Poll

Simple Poll Edit Settings

19 replies Last reply 26 days ago

Traits of Successful Programming Tools (For Existing Programmers) 1/2

— **Browser UI**: A lingua franca for graphics.
— **Language Agnostic**: Work with existing programming languages.
— **Plain Text**: An open data format.
— **Packages Written in Scripting Languages**: Make customizations easy to share and modify.

Traits of Successful Programming Tools (For Existing Programmers) 2/2

— **Text Editors**: The concentration of programming hours.

— **Unix Processes**: Use child processes to extend capabilities.

— **Version Control**: State of the art collaboration.

What do users have to give up to use your tool?

— Text editors?
— Version control?

Is your solution more useful than these things?

Case Study: Visual Studio Code

Visual Studio Code is a **language agnostic text editor** used to edit **plain text** files that can be stored in **version control**. It's user interface is displayed using a **browser rendering engine**. It's **packages are written in scripting languages** that run **Unix processes**.

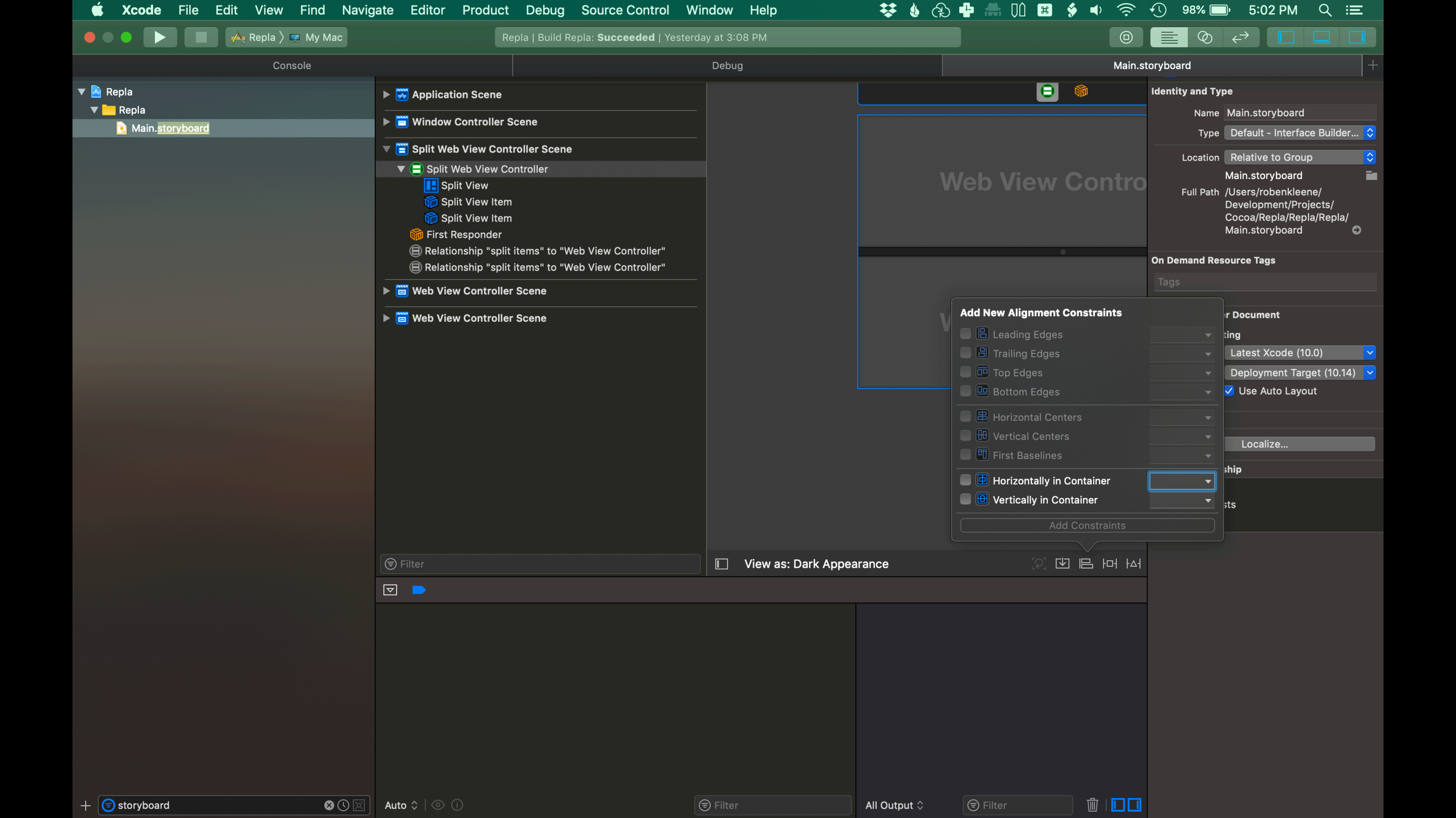
Visual Studio Code Market Share

— **2016:** 7.2%
— **2017:** 22.3%
— **2018:** 34.9%
— **2019:** 50.7%

Stack Overflow Insights.

Competes with text editors: 20-25 employees.

Why Not Integrate With an IDE?


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
98% 5:02 PM
Repla 〉 My Mac
Repla | Build Repla: Succeeded | Yesterday at 3:08 PM
Console
Debug
Main.storyboard
Repla
Repla
Main.storyboard
Application Scene
Window Controller Scene
Split Web View Controller Scene
Split Web View Controller
Split View
Split View Item
Split View Item
First Responder
Relationship "split items" to "Web View Controller"
Relationship "split items" to "Web View Controller"
Web View Controller Scene
Web View Controller Scene
Filter
Web View Contro
Add New Alignment Constraints
Leading Edges
Trailing Edges
Top Edges
Bottom Edges
Horizontal Centers
Vertical Centers
First Baselines
Horizontally in Container
Vertically in Container
Add Constraints
View as: Dark Appearance
Identity and Type
Name Main.storyboard
Type Default - Interface Builder...
Location Relative to Group
Main.storyboard
Full Path /Users/robenkleene/Development/Projects/Cocoa/Repla/Repla/Repla/Main.storyboard
On Demand Resource Tags
Tags
Latest Xcode (10.0)
Deployment Target (10.14)
Use Auto Layout
Localize...
storyboard
Auto
Filter
All Output
Filter

When to integrate with an IDE?

If your feature fits into existing user interface features and requires no interaction.

— 👍 Linters

— 👎 Splits & Folds

Repla

What is Repla?

— A web rendering engine
— Unix process management
— A packaging system
— Not an editor, works alongside existing tools
— A platform

Well maybe try to replace one thing… the browser

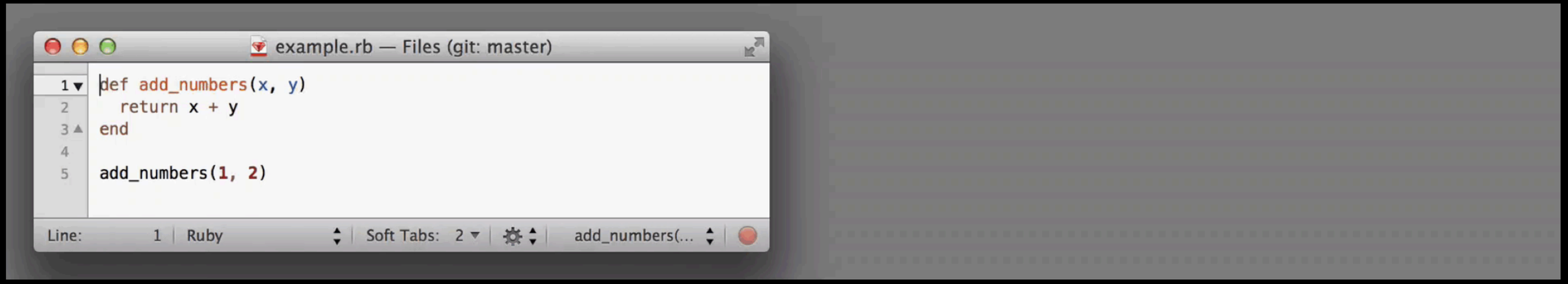
— Sounds crazy?

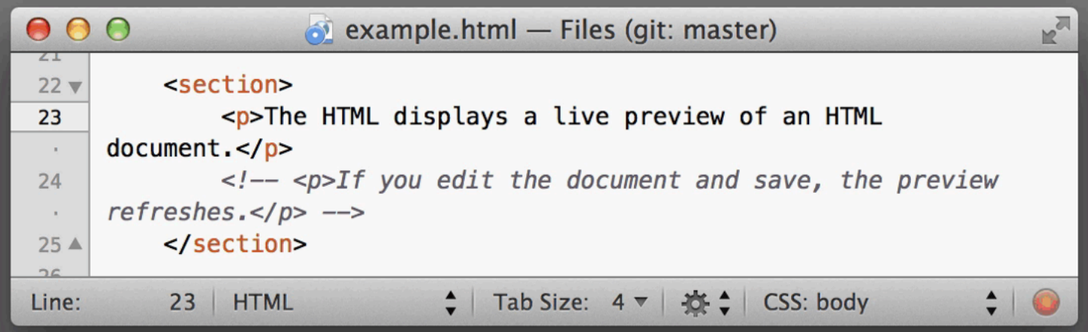
— The developer trio: a **browser**, a **text editor**, and a **terminal**.

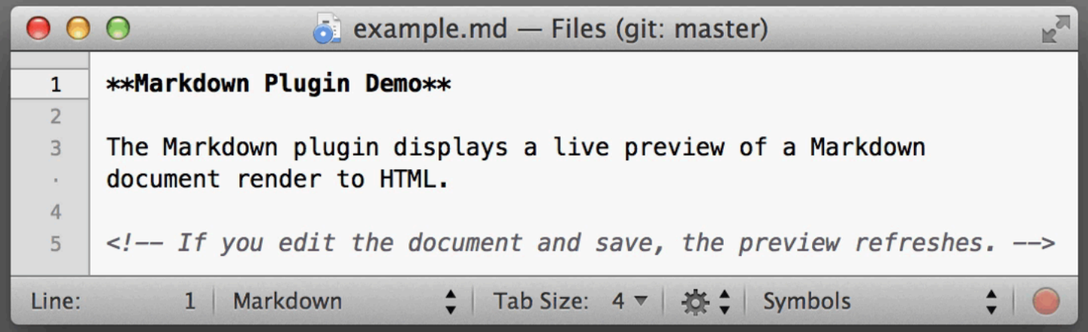
— A text editor and a terminal both use a **packaging system** to extend functionality by running **Unix processes**. The browser?

Screenshots


example.js — Files (git: master)
1 function addNumbers(x, y) {
2 return x + y;
3 };
4
5 addNumbers(1, 2);
Line: 1 JavaScript Soft Tabs: 2 Symbols


example.rb — Files (git: master)
1 def add_numbers(x, y)
2 return x + y
3 end
4
5 add_numbers(1, 2)
Line: 1 Ruby Soft Tabs: 2 add_numbers(...


example.html — Files (git: master)
22 <section>
23 The HTML displays a live preview of an HTML document.
24 <!-- <p>If you edit the document and save, the preview refreshes.</p> -->
25 </section>
Line: 23 HTML Tab Size: 4 CSS: body


example.md — Files (git: master)
1 **Markdown Plugin Demo**
2
3 The Markdown plugin displays a live preview of a Markdown document render to HTML.
4
5 <!-- If you edit the document and save, the preview refreshes. -->
Line: 1 Markdown Tab Size: 4 Symbols

Your Text Editor

```
1  function addNumbers(a, b) {
2      return a + b;
3  }
4
5  var left = 1;
6
7  var right = 2;
8
9  addNumbers(left, right);
10
```

Repla

```
   function addNumbers(a, b) {
       return a + b;
   }
=> undefined
   var left = 1;
=> undefined
   var right = 2;
=> undefined
   addNumbers(left, right);
=> 3
```

Ruby on Rails

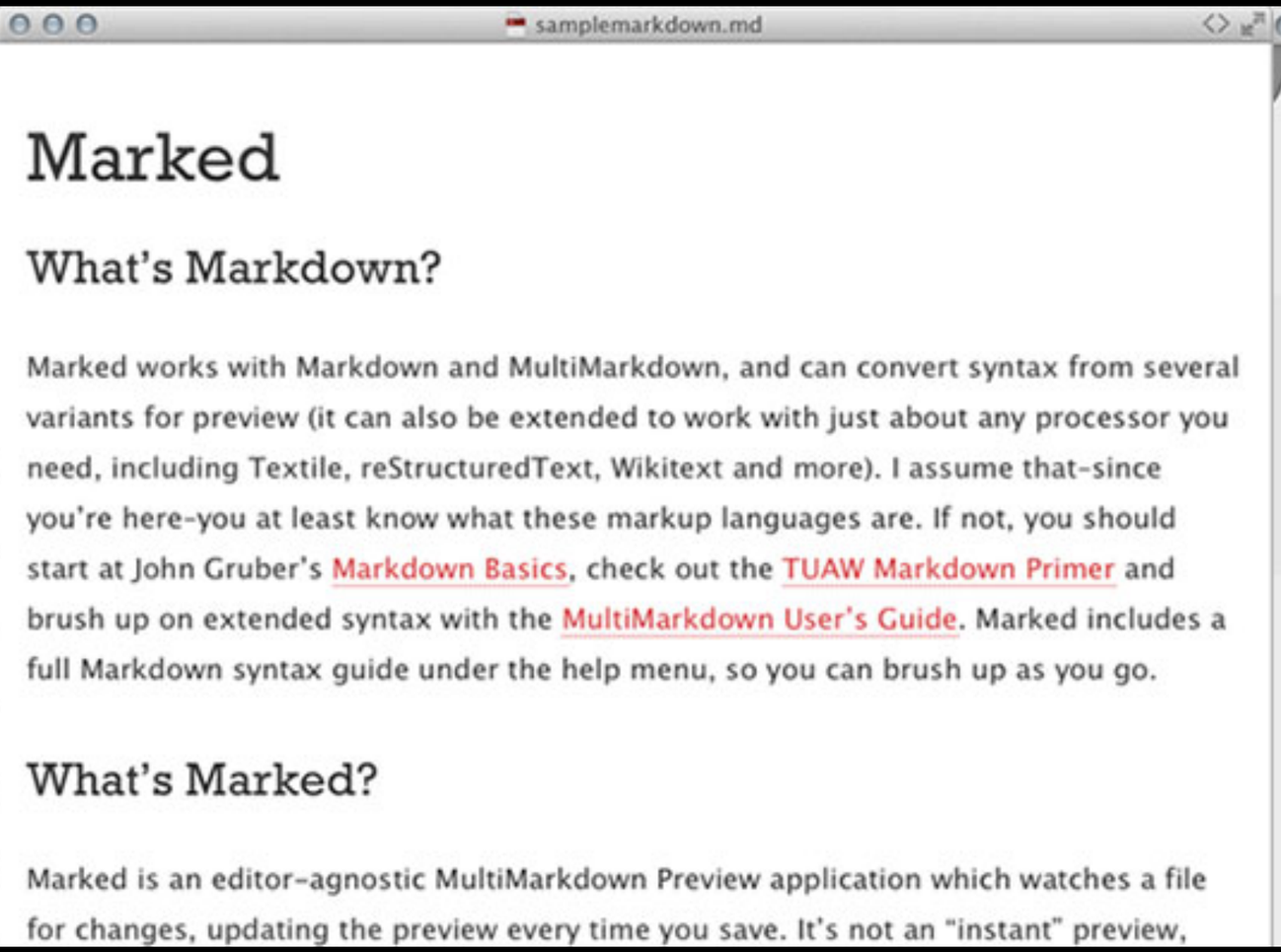
Yay! You're on Rails!

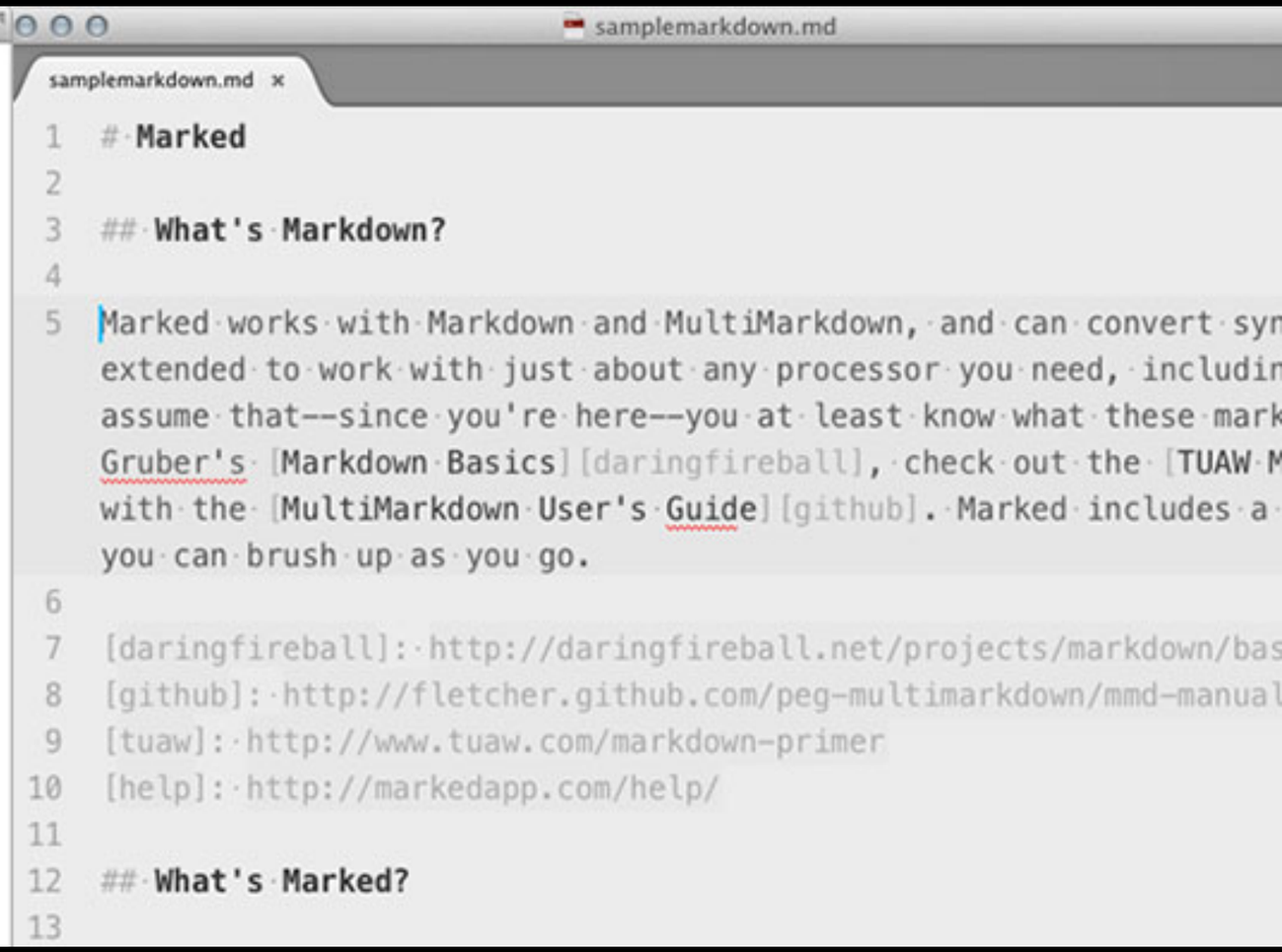

```
=> Booting Puma
=> Rails 5.2.3 application starting in development
=> Run `rails server -h` for more startup options
Puma starting in single mode...
* Version 3.12.1 (ruby 2.3.7-p456), codename: Llamas in Pajamas
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://localhost:3000
Use Ctrl-C to stop
Started GET "/" for ::1 at 2019-05-01 15:33:11 -0400
Processing by Rails::WelcomeController#index as HTML
Rendering vendor/bundle/ruby/2.3.0/gems/railties-5.2.3/lib/rails/templates/rails/welcome/index.html.erb
Rendered vendor/bundle/ruby/2.3.0/gems/railties-5.2.3/lib/rails/templates/rails/welcome/index.html.erb (11.2ms)
Completed 200 OK in 41ms (Views: 17.5ms | ActiveRecord: 0.0ms)
```

Roadmap

1. **Browser & Processes:** Web Development (integrate the server and browser into one window, automatically refresh)
2. **Packaging System:** Distributing Web Apps (one click install, one click run, e.g., Jupyter Notebooks)
3. **Packages 1:** Live Coding View
4. **Packages 2:** Framer Classic & Processing

Use Cases

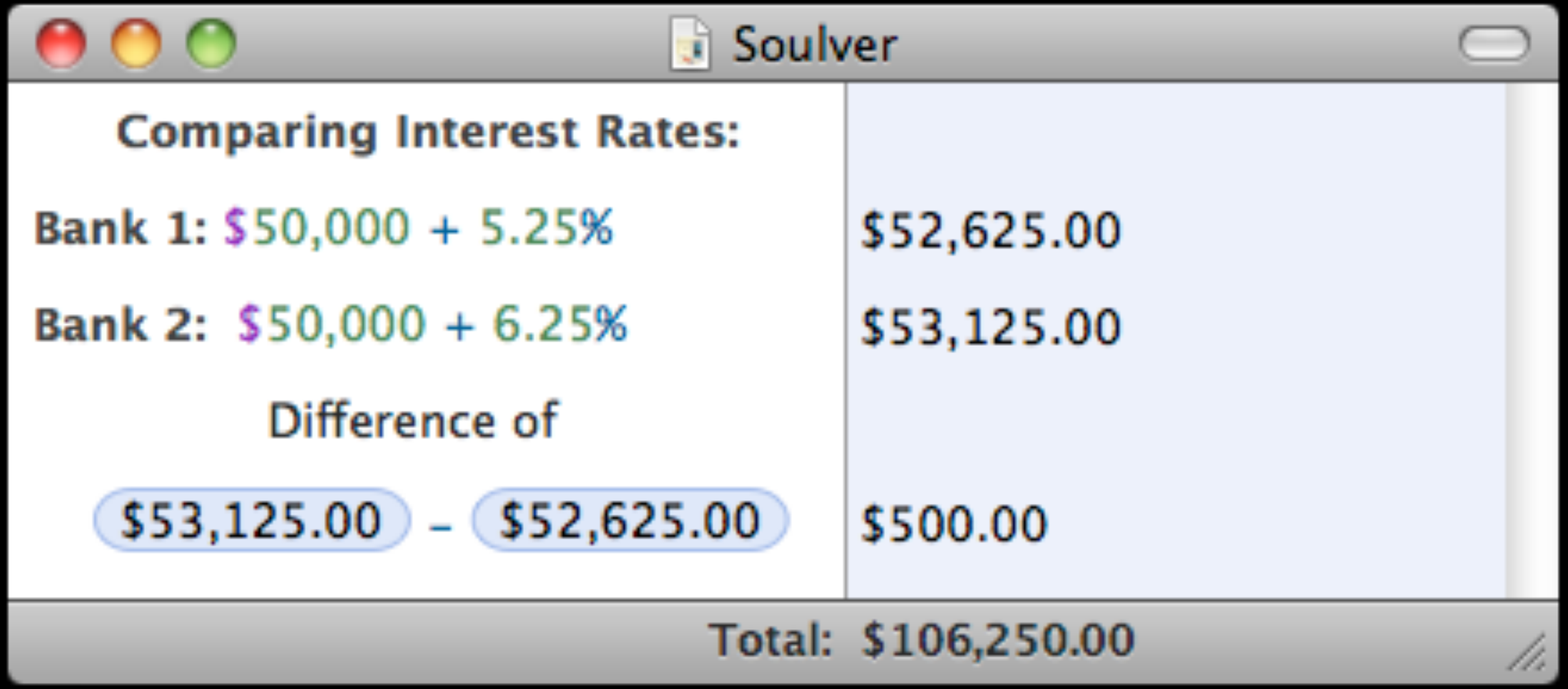

samplemarkdown.md
Marked
What's Markdown?
Marked works with Markdown and MultiMarkdown, and can convert syntax from several variants for preview (it can also be extended to work with just about any processor you need, including Textile, reStructuredText, Wikitext and more). I assume that–since you're here–you at least know what these markup languages are. If not, you should start at John Gruber's Markdown Basics, check out the TUAW Markdown Primer and brush up on extended syntax with the MultiMarkdown User's Guide. Marked includes a full Markdown syntax guide under the help menu, so you can brush up as you go.
What's Marked?
Marked is an editor–agnostic MultiMarkdown Preview application which watches a file for changes, updating the preview every time you save. It's not an "instant" preview,


samplemarkdown.md
samplemarkdown.md
Marked
What's Markdown?
Marked works with Markdown and MultiMarkdown, and can convert syn
extended to work with just about any processor you need, includin
assume that--since you're here--you at least know what these mark
Gruber's [Markdown Basics][daringfireball], check out the [TUAW M
with the [MultiMarkdown User's Guide][github]. Marked includes a
you can brush up as you go.
[daringfireball]: http://daringfireball.net/projects/markdown/bas
[github]: http://fletcher.github.com/peg-multimarkdown/mmd-manual
[tuaw]: http://www.tuaw.com/markdown-primer
[help]: http://markedapp.com/help/
What's Marked?

Marked

— 👍 Plain text
— 👍 Works with text editors
— 👍 Works with version control
— 👎 Not language agnostic

See also: Deckset


Soulver
Comparing Interest Rates:
Bank 1: $50,000 + 5.25%
$52,625.00
Bank 2: $50,000 + 6.25%
$53,125.00
Difference of
$53,125.00 - $52,625.00
$500.00
Total: $106,250.00

Soulver

— 👎 Not plain text
— 👎 Not language agnostic
— 👎 Doesn't work with text editors
— 👎 Doesn't work with version control

See also: Calca

observablehq.com

Observable Search Demo ••• Fork Sign in

👋 Welcome. This is live code! Click the left margin to view or edit.

Observable · Dec 4, 2017
The magic notebook for visualization.
By Mike Bostock

90

Featured in Introduction 35 forks

Five-Minute Introduction

Welcome to Observable! This notebook gives a quick but technical overview of Observable's features. For a slightly longer introduction, see Introduction to Notebooks and introductory tutorial series. If you want to watch rather than read, we've also posted a short video introduction!

An Observable notebook is a sequence of cells. Each cell is a snippet of JavaScript. You can see (and edit!) the code for any cell by clicking the menu ⋮ in the left margin.

```
42
```

```
2 * 3 * 7
```

```
5050
```

```
{
  let sum = 0;
  for (let i = 0; i <= 100; ++i) {
    sum += i;
  }
  return sum;
}
```

?

Observable

— 👎 Not plain text

— 👎 Not language agnostic

— 👎 Doesn't work with text editors

— 👎 Doesn't work with version control

See also: Jupyter Notebooks, Swift Playgrounds, R Markdown

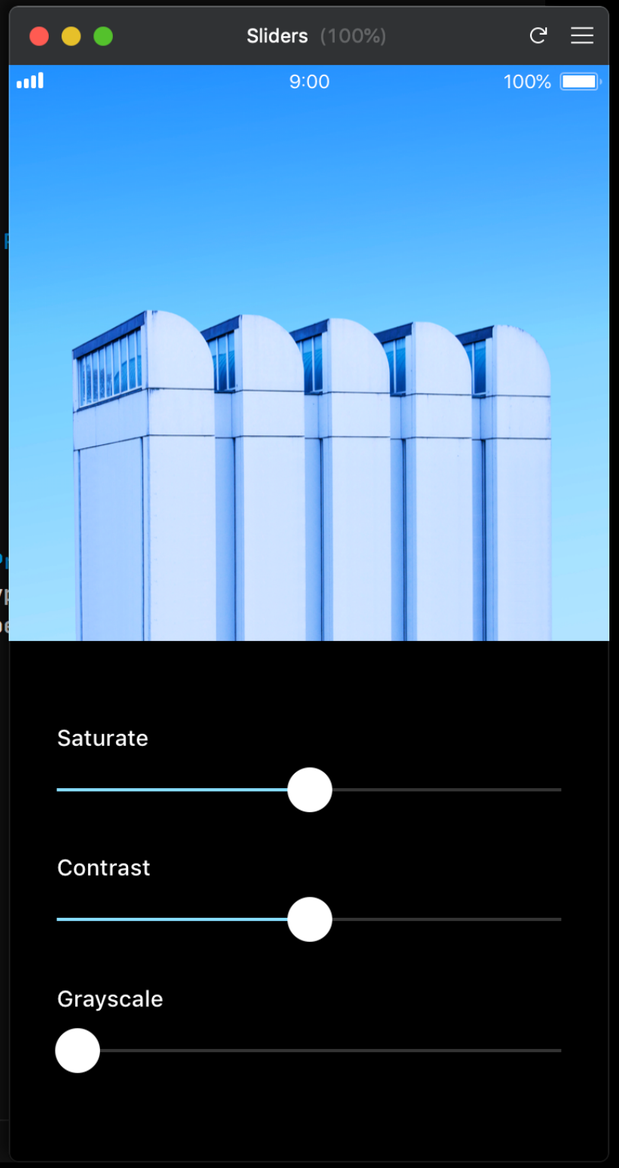
Slider.tsx

```tsx
import * as React from "react"
import { Frame, PropertyControls, ControlType } from "framer"

interface Props {
  value: number
  knob: string
  track: string
  tint: string
  width: number
  height: number
}

export class Slider extends
  static defaultProps = {
    width: 120,
    height: 44,
    value: 50,
    knob: "#FFF",
    track: "#333",
    tint: "#8DF",
  }

  static propertyControls: P
    value: { type: ControlTy
    tint: { type: ControlType
  }
```



Framer Classic

— 👍 Plain text
— 👍 Works with version control
— 👎 Doesn't work with text editors
— 👎 Not language agnostic

See also: Processing

Live Coding With Repla

Advantages

— Language packages: `IRB.replaplugin`, `Python.replaplugin`, `Node.replaplugin`

— Use your existing code editor, with regular file extensions: `.rb`, `.py`, `.js`

— Check them into version control

Live Coding Implementation

```
def initialize(command)
  PTY.spawn(command) do |output, input, _pid|
    Thread.new do
      output.each do |line|
        output_controller.parse_output(line)
      end
    end
    @input = input
  end
end

def parse_input(input)
  input_controller.parse_input(input)
  @input.write(input)
end
```

Other Implementation Details

— The Ruby process watches the file system (`gem 'listen'`)

Introducing Repla: A Live-Coding Tool

How to make innovating programming tools that are compatible with existing workflows.

That's it, thanks!

Roben Kleene
@robenkleene
robenkleene.com
github.com/robenkleene/repla-live-coding