Intertwingling the TiddlyWiki

Joe Armstrong
Jeremy Ruston

Plan

- Intertwingledness
- Avoiding ground hog day
- Making complex things and the joy of all-in-oneness
- The TiddlyWiki - the correct level of granularity - but deeply intertwingled
- Reusing the TiddlyWiki - CST (Communicating Sequential TiddlyWikis)
- The experiments

"EVERYTHING IS DEEPLY INTERTWINGLED. In an important sense there are no "subjects" at all; there is only all knowledge, since the cross-connections among the myriad topics of this world simply cannot be divided up neatly.”

Ted Nelson -
Computer Lib/Dream Machines 1974

Hierarchical and sequential structures, especially popular since Gutenberg, are usually forced and artificial. Intertwingularity is not generally acknowledged—people keep pretending they can make things hierarchical, categorizable and sequential when they can't.

Ted Nelson -
Computer Lib/Dream Machines 1974

In Buddhism, saṃsāra is the "suffering-laden cycle of life, death, and rebirth, without beginning or end

The Groundhog cycle

1. Invent something simple
2. Add features
3. Add more features
4. Add features to the features
5. …
6. It’s so complicated nobody very few people can change it
7. It’s very powerful and very useful
8. Add more features
9. … we’ll add one more feature … what could possibly go wrong?

…

N. Goto 1

What do we do when
things get very complicated
but are very useful?

- Throw away and start again
(Groundhog) - you miss the good things

- Stick it in a container and only use a defined interface - (one day you'll get containers in containers - but never mind)

- Add a messaging interface
(Biological model)

CSCT

- Communication Sequential Complex Things (CSP ++)

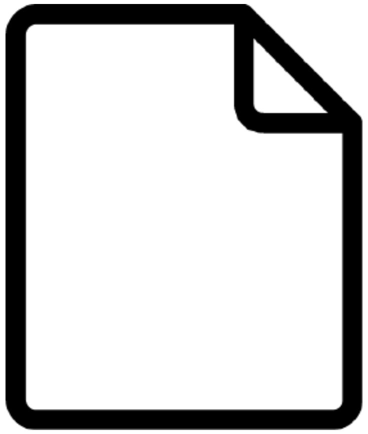
1. Add a mailbox to the complex thing
2. Define a messaging protocol

All in-one’ness

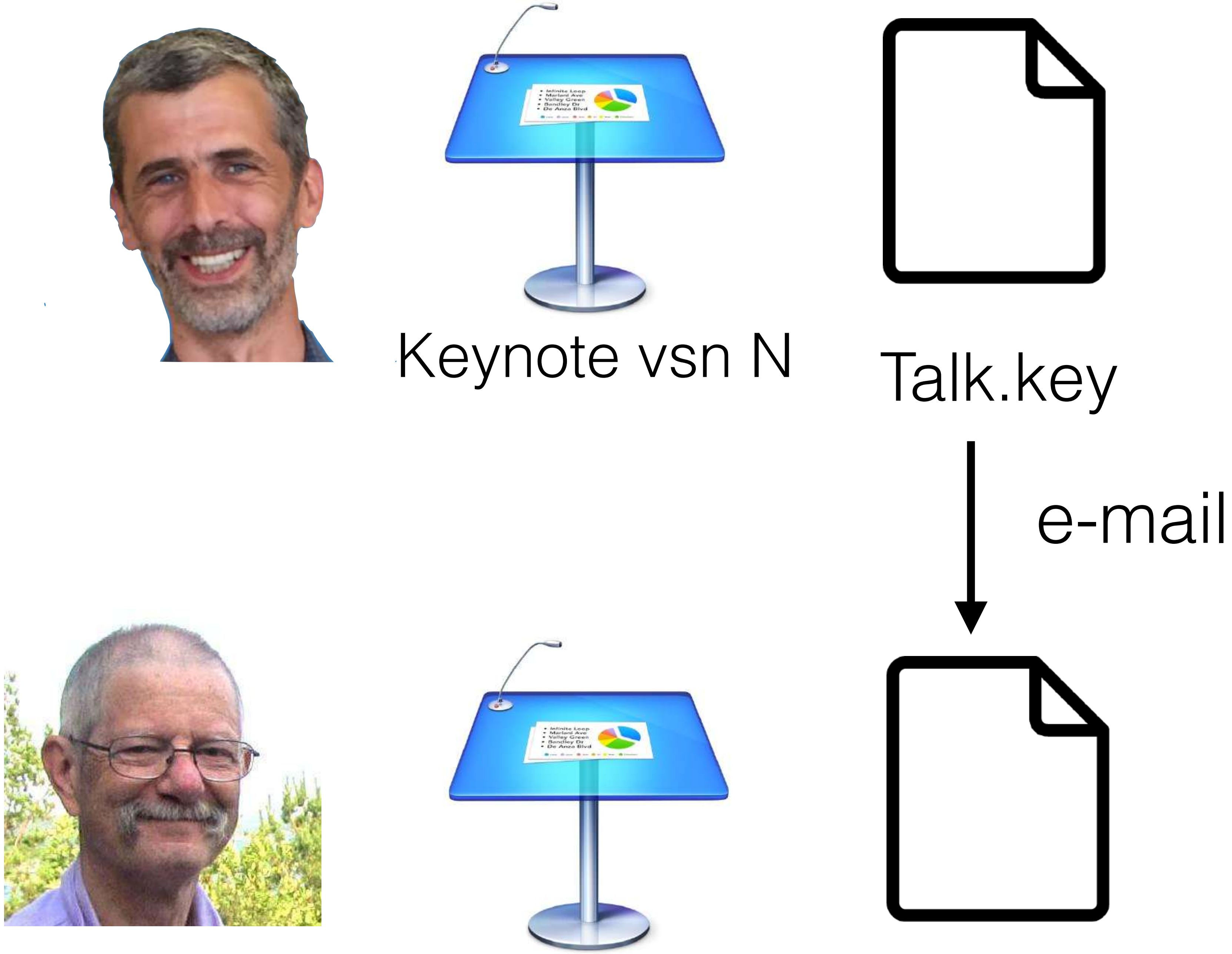
An “all-in-one” application has the code and all the data in a container

When the code and data are separated disasters will happen

Keynote



Talk.key



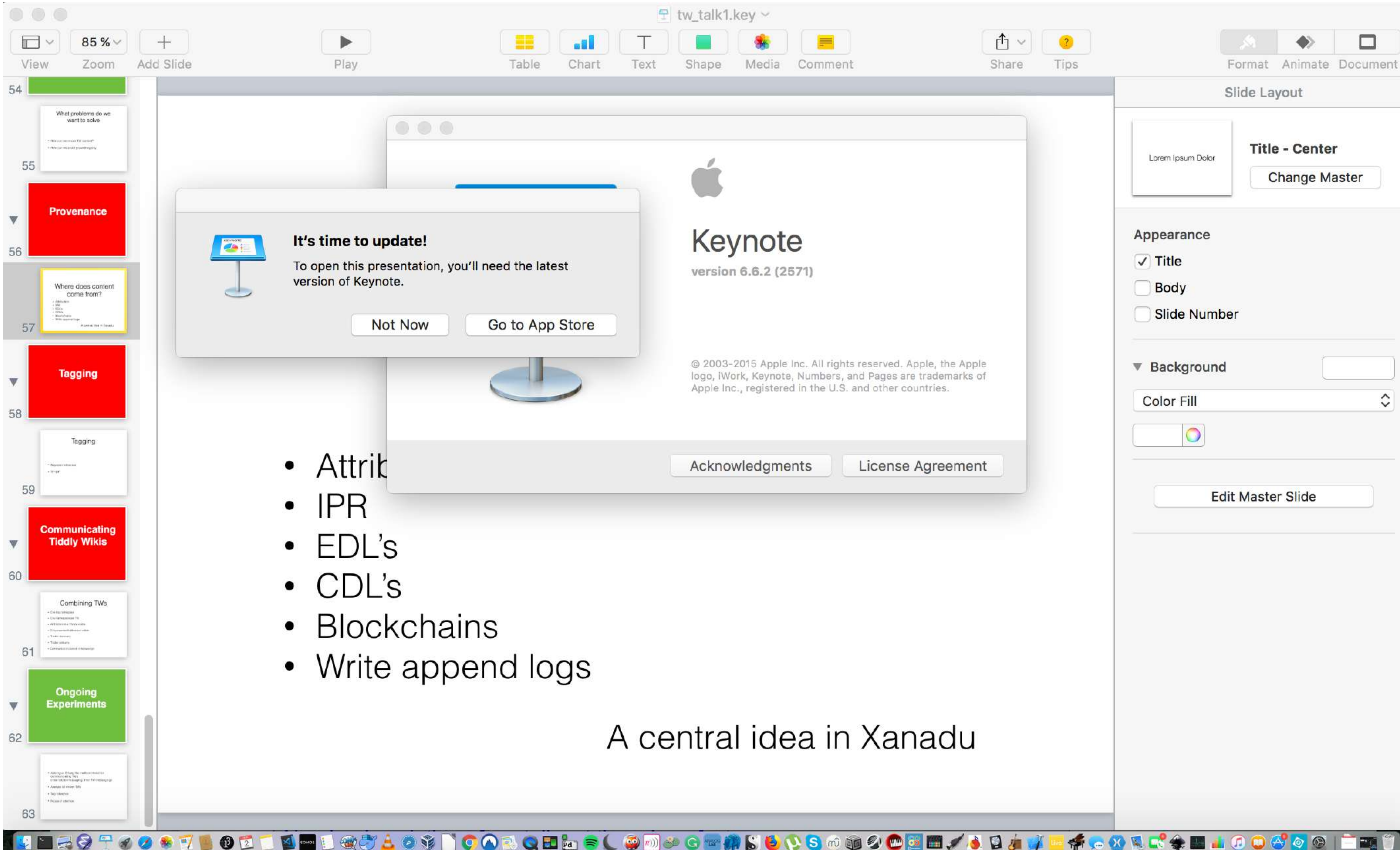
Keynote vsn N

Talk.key

e-mail

Keynote vsn N-1

Talk.key



It's time to update!

To open this presentation, you'll need the latest version of Keynote.

Not Now | Go to App Store

Keynote

version 6.6.2 (2571)

© 2003-2015 Apple Inc. All rights reserved. Apple, the Apple logo, iWork, Keynote, Numbers, and Pages are trademarks of Apple Inc., registered in the U.S. and other countries.

Acknowledgments | License Agreement

- Attrib
- IPR
- EDL's
- CDL's
- Blockchains
- Write append logs

A central idea in Xanadu

Slide Layout

Title - Center

Change Master

Appearance

☑ Title

□ Body

□ Slide Number

Background

Color Fill

Edit Master Slide

All in one ness

- Opposite of “reusing dependencies”
- - Slow and difficult to write
- + Increased understanding
- +++++++ works “forever”

5 steps
to
all-in-one ness

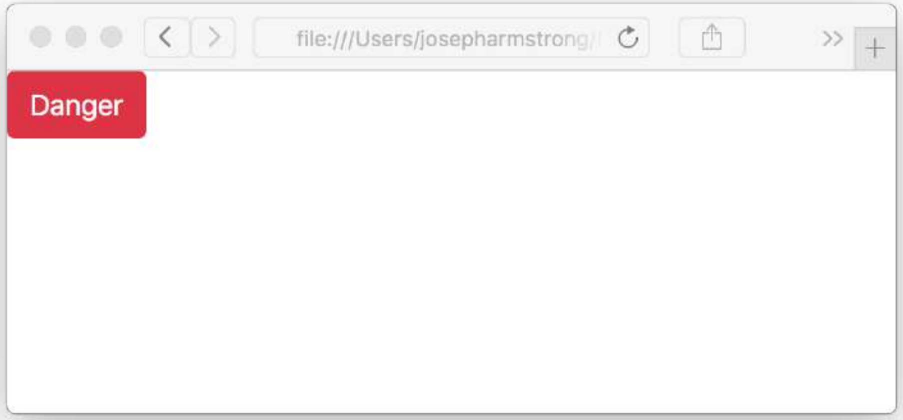
Write a program to display a nice button in the browser

Step 1

Use a remote CDN

```
<link rel="stylesheet"
     href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
     integrity="sha384-MCw98/SFnGE8fJT3…ERdknLPMO"
     crossorigin="anonymous">

<button class="btn btn-danger">Danger</button>
```



- Not future proof
  CDN will break
- 141 KB of css
- can’t work off-line

+ fast time to market

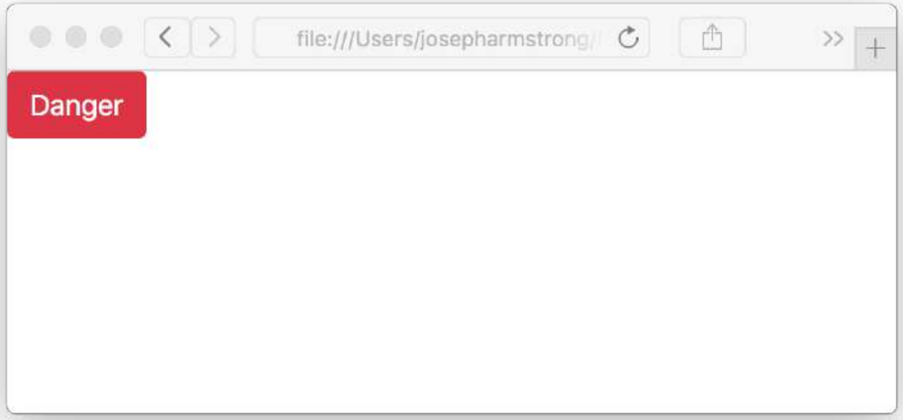
“I’d never do that!”
“It’s a security nightmare”
“141KBytes to get 10 lines of css”
“Why is my web page 4 MBytes”
“I bet the CDN will be dead in 3 years time”

Step 2

Make a local copy of the CDN

```
<link rel="stylesheet"
     href="./bootstrap.min.css">

<button class="btn btn-danger">Danger</button>
```



- Not future proof
  move the files and you'll
  break things

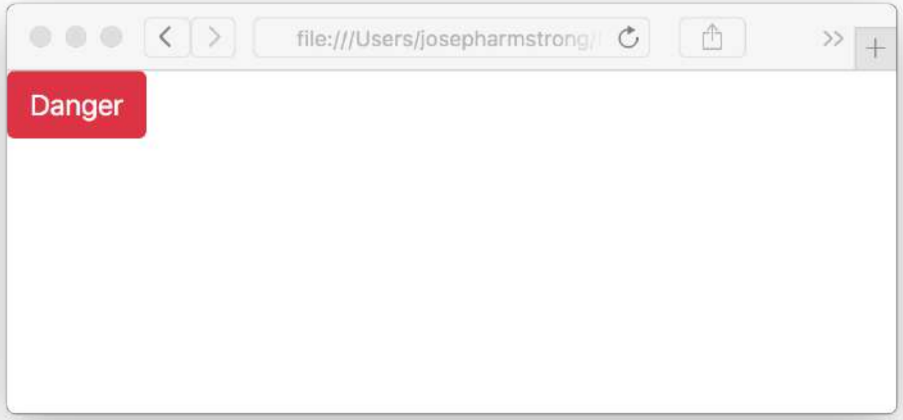
+ can work off-line

Step 3

Move the css into to the HTML file

```
<style>
 ...
 173 Kbytes of Css
  ...
</style>

<button class="btn btn-danger">Danger</button>
```



- Future proof
  move the files and you
  won’t break things
- Huge and horrible

Step 4

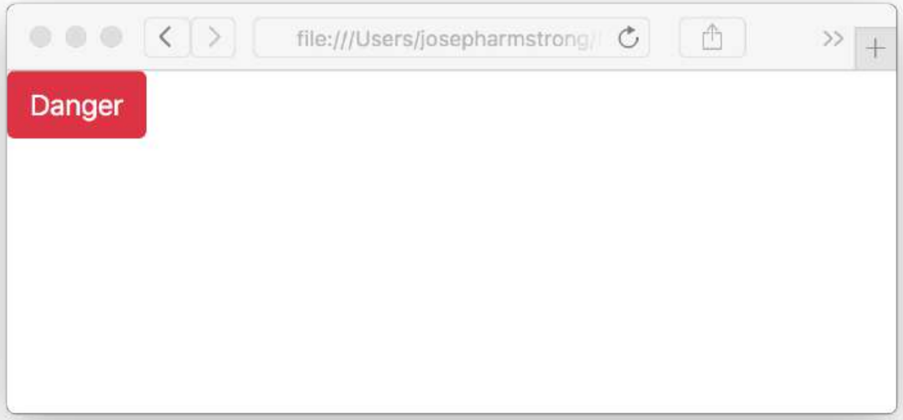
Manual garbage collection

```
<style>

.btn {
  display: inline-block;
  font-weight: 400;
  text-align: center;
  white-space: nowrap;
  vertical-align: middle;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  border: 1px solid transparent;
  padding: 0.375rem 0.75rem;
  font-size: 1rem;
  line-height: 1.5;
  border-radius: 0.25rem;
  transition: color 0.15s ease-in-out, background-color 0.15s ease-in-out,
   border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
}

.btn-danger {
  color: #fff;
  background-color: #dc3545;
  border-color: #dc3545;
}
</style>

<button class="btn btn-danger">Danger</button>
```



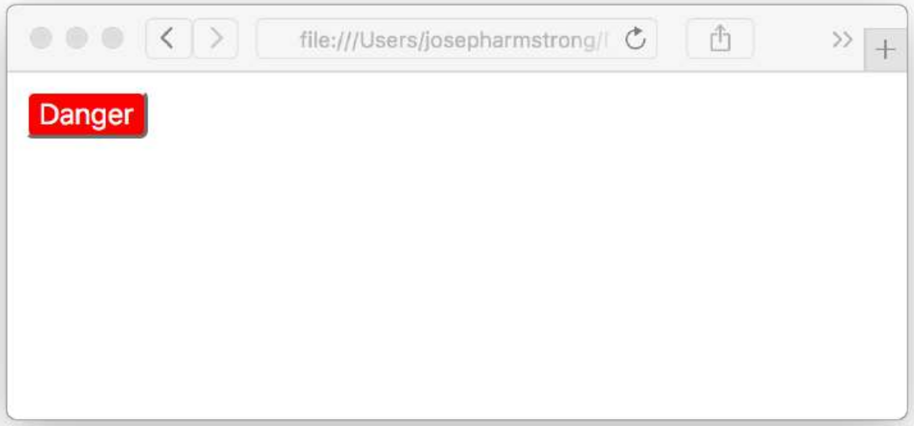
- Takes time
- Copyright???
+ Small
+ Increased comprehension
+ Future proof

Step 5

Manual refactoring

```
<style>
.danger {
  font-size: 1rem;
  border-radius: 0.3rem;
  color: white;
  background-color: red;
}
</style>


<button class="danger">Danger</button>
```



+ Small
+ Increased comprehension
+ Future proof
+ Maintainable
- Bad time to market

Do you really want to
add <N> Mbytes of
incomprehensible
stuff to your application in
order to use the 0.01% of
features you actually use?

- No build system
- No webpack/grunt/bower/

Two outstanding all-in-one apps

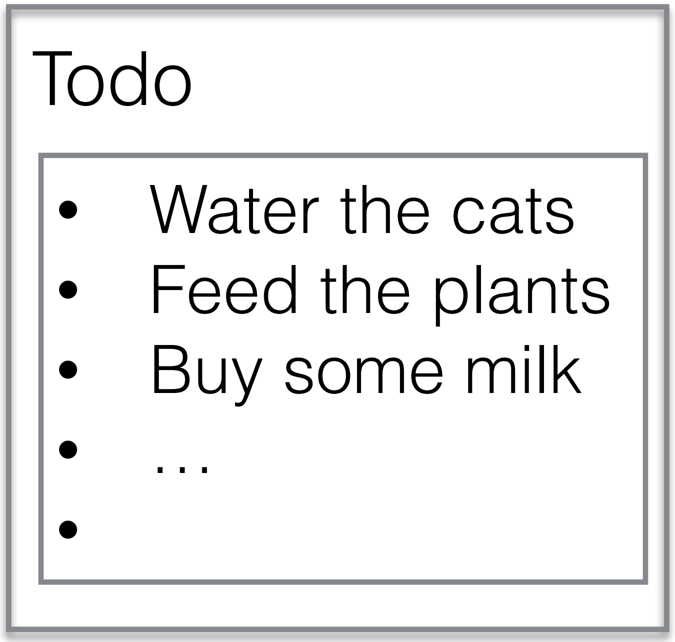
- Smalltalk
- TiddlyWiki

The TiddlyWiki

- Small-grain knowledge
- Highly intertwingled
- Proper translusions
- Logic queries “findall tags such that …”

A todo list
in the TiddlyWiki

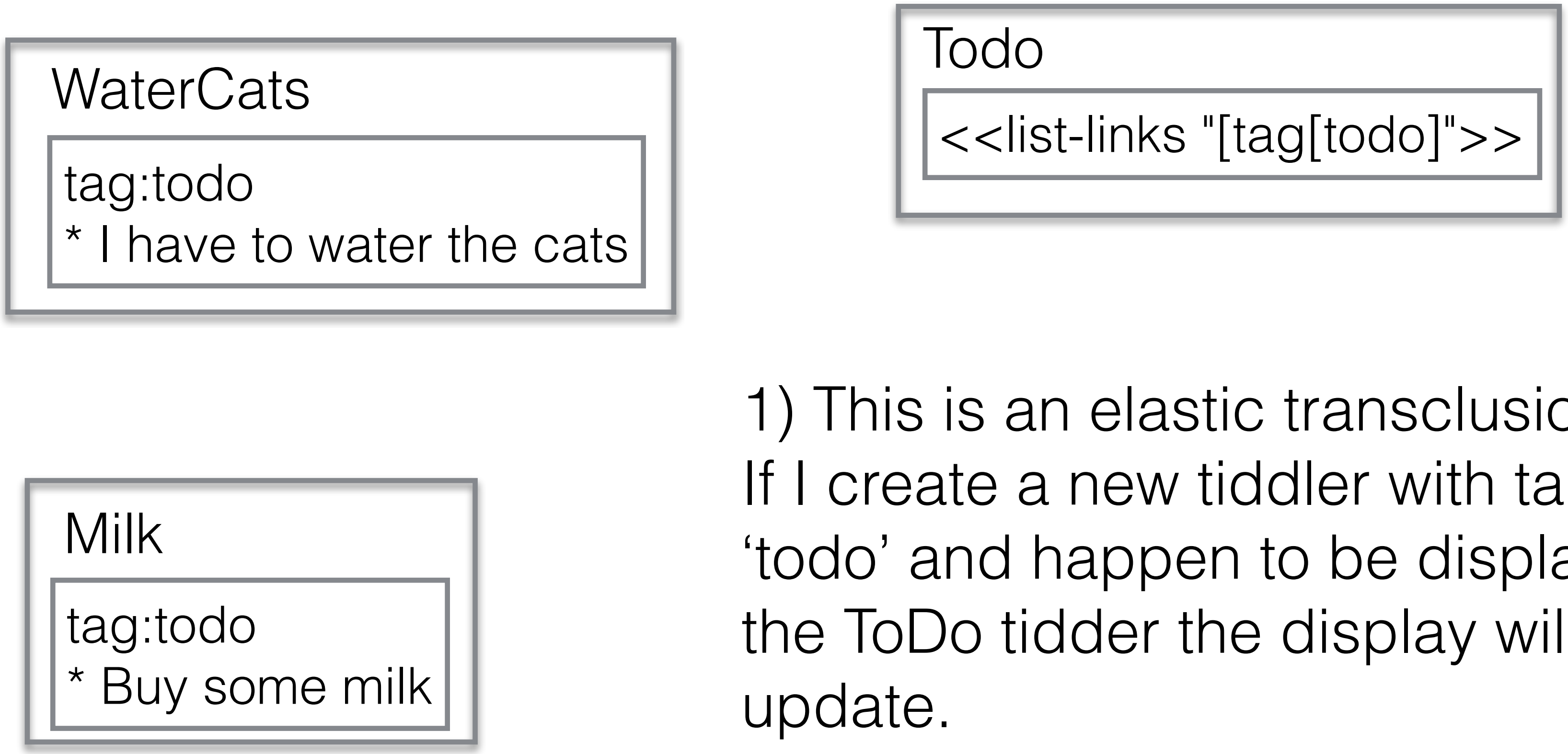
I create a single tiddler called ToDo and start typing



Todo

- Water the cats
- Feed the plants
- Buy some milk
- …
-

Eureka moment



WaterCats

tag:todo
* I have to water the cats

Todo

<<list-links "[tag[todo]">>

Milk

tag:todo
* Buy some milk

1) This is an elastic transclusion If I create a new tiddler with tag ‘todo’ and happen to be displaying the ToDo tidder the display will update.

2) I now have small grain objects that can be transcluded elsewhere

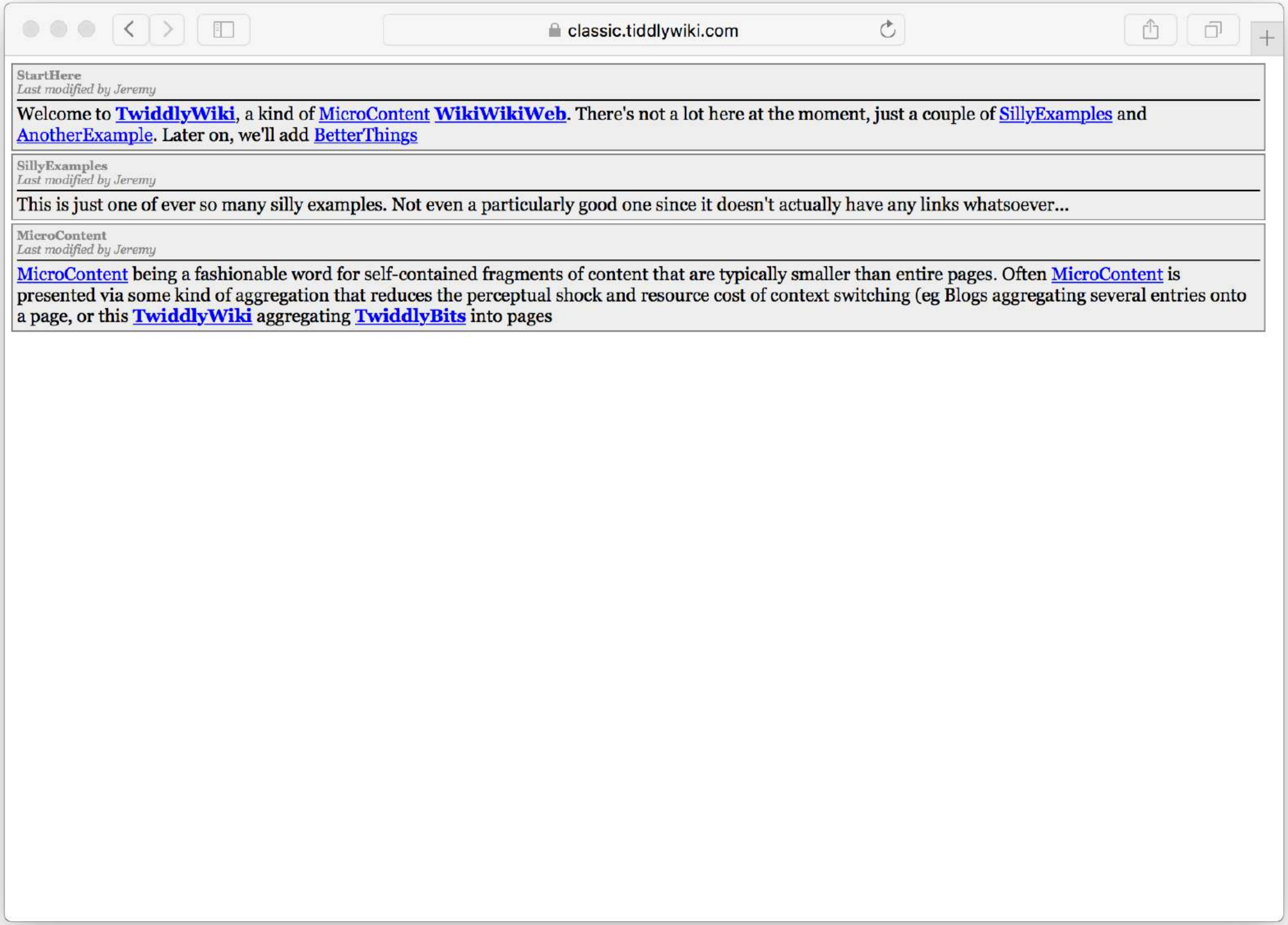
Jeremy Ruston

- The Human Garbage collector
- https://classic.tiddlywiki.com/archive/secondversion.html
- https://classic.tiddlywiki.com/archive/

Early TW’s
what did we learn

“TwiddlyWiki”

- First unreleased version, August 2004
- A wiki for microcontent
- Shows multiple pages at once (cribbed from GMail)
- Explores visualising navigation between them
- 200 lines of HTML/CSS/JS



StartHere
Last modified by Jeremy

Welcome to **TwiddlyWiki**, a kind of MicroContent **WikiWikiWeb**. There's not a lot here at the moment, just a couple of SillyExamples and AnotherExample. Later on, we'll add BetterThings

SillyExamples
Last modified by Jeremy

This is just one of ever so many silly examples. Not even a particularly good one since it doesn't actually have any links whatsoever...

MicroContent
Last modified by Jeremy

MicroContent being a fashionable word for self-contained fragments of content that are typically smaller than entire pages. Often MicroContent is presented via some kind of aggregation that reduces the perceptual shock and resource cost of context switching (eg Blogs aggregating several entries onto a page, or this **TwiddlyWiki** aggregating **TwiddlyBits** into pages

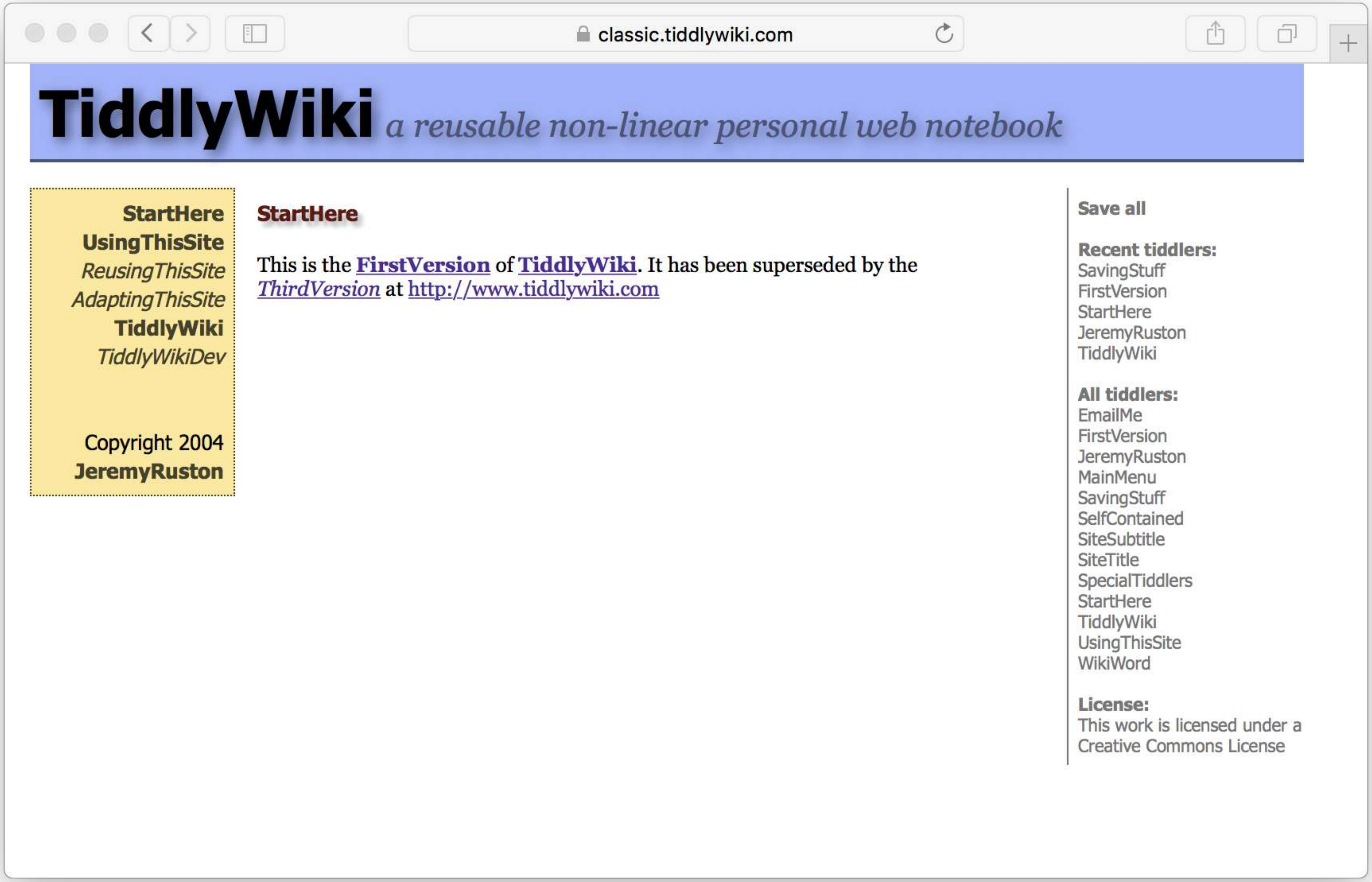
What’s So Wonderful About Wikis?

- A rudimentary implementation of hypertext that gets one thing right: making linking be part of the punctuation of writing

- Not so much the anybody-can-edit ethos

September 2004

- First released version
- Coins the ludicrous names “TiddlyWiki” and “tiddler”
- Tries to look like a blog
- Adds support for editing tiddlers
- Went viral, 2004-style



TiddlyWiki *a reusable non-linear personal web notebook*

StartHere
UsingThisSite
ReusingThisSite
AdaptingThisSite
TiddlyWiki
TiddlyWikiDev

Copyright 2004
JeremyRuston

StartHere

This is the **FirstVersion** of **TiddlyWiki**. It has been superseded by the *ThirdVersion* at http://www.tiddlywiki.com

Save all

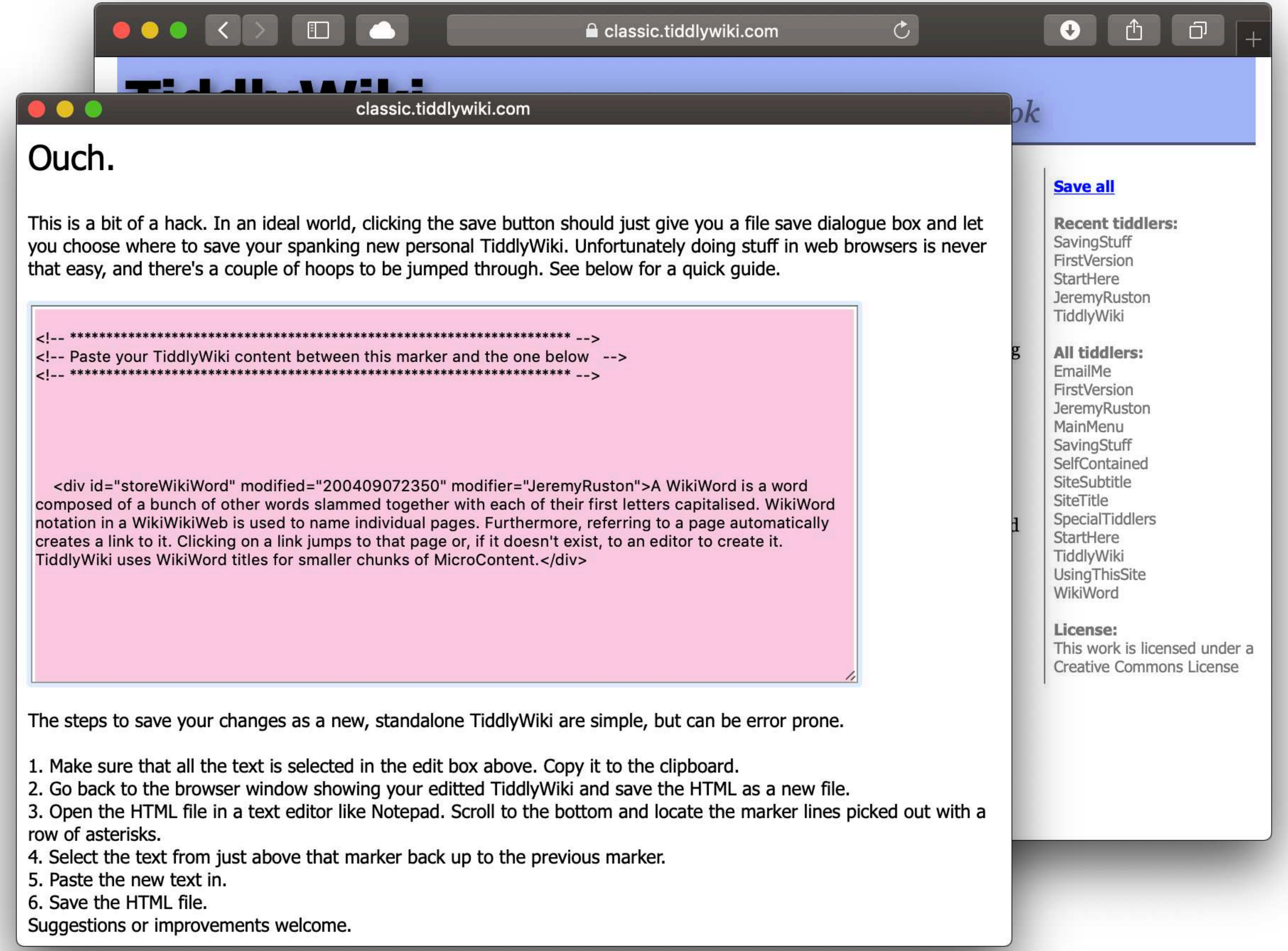
Recent tiddlers:
SavingStuff
FirstVersion
StartHere
JeremyRuston
TiddlyWiki

All tiddlers:
EmailMe
FirstVersion
JeremyRuston
MainMenu
SavingStuff
SelfContained
SiteSubtitle
SiteTitle
SpecialTiddlers
StartHere
TiddlyWiki
UsingThisSite
WikiWord

License:
This work is licensed under a Creative Commons License

“Just a demo”

Implemented as a single HTML file, so impossible to save changes anywhere



Ouch.

This is a bit of a hack. In an ideal world, clicking the save button should just give you a file save dialogue box and let you choose where to save your spanking new personal TiddlyWiki. Unfortunately doing stuff in web browsers is never that easy, and there's a couple of hoops to be jumped through. See below for a quick guide.

```
<!-- ************************************************************************ -->
<!-- Paste your TiddlyWiki content between this marker and the one below   -->
<!-- ************************************************************************ -->




   <div id="storeWikiWord" modified="200409072350" modifier="JeremyRuston">A WikiWord is a word composed of a bunch of other words slammed together with each of their first letters capitalised. WikiWord notation in a WikiWikiWeb is used to name individual pages. Furthermore, referring to a page automatically creates a link to it. Clicking on a link jumps to that page or, if it doesn't exist, to an editor to create it. TiddlyWiki uses WikiWord titles for smaller chunks of MicroContent.</div>
```

The steps to save your changes as a new, standalone TiddlyWiki are simple, but can be error prone.

1. Make sure that all the text is selected in the edit box above. Copy it to the clipboard.
2. Go back to the browser window showing your editted TiddlyWiki and save the HTML as a new file.
3. Open the HTML file in a text editor like Notepad. Scroll to the bottom and locate the marker lines picked out with a row of asterisks.
4. Select the text from just above that marker back up to the previous marker.
5. Paste the new text in.
6. Save the HTML file.

Suggestions or improvements welcome.

Save all

Recent tiddlers:
SavingStuff
FirstVersion
StartHere
JeremyRuston
TiddlyWiki

All tiddlers:
EmailMe
FirstVersion
JeremyRuston
MainMenu
SavingStuff
SelfContained
SiteSubtitle
SiteTitle
SpecialTiddlers
StartHere
TiddlyWiki
UsingThisSite
WikiWord

License:
This work is licensed under a Creative Commons License

We’re Not in Kansas Anymore

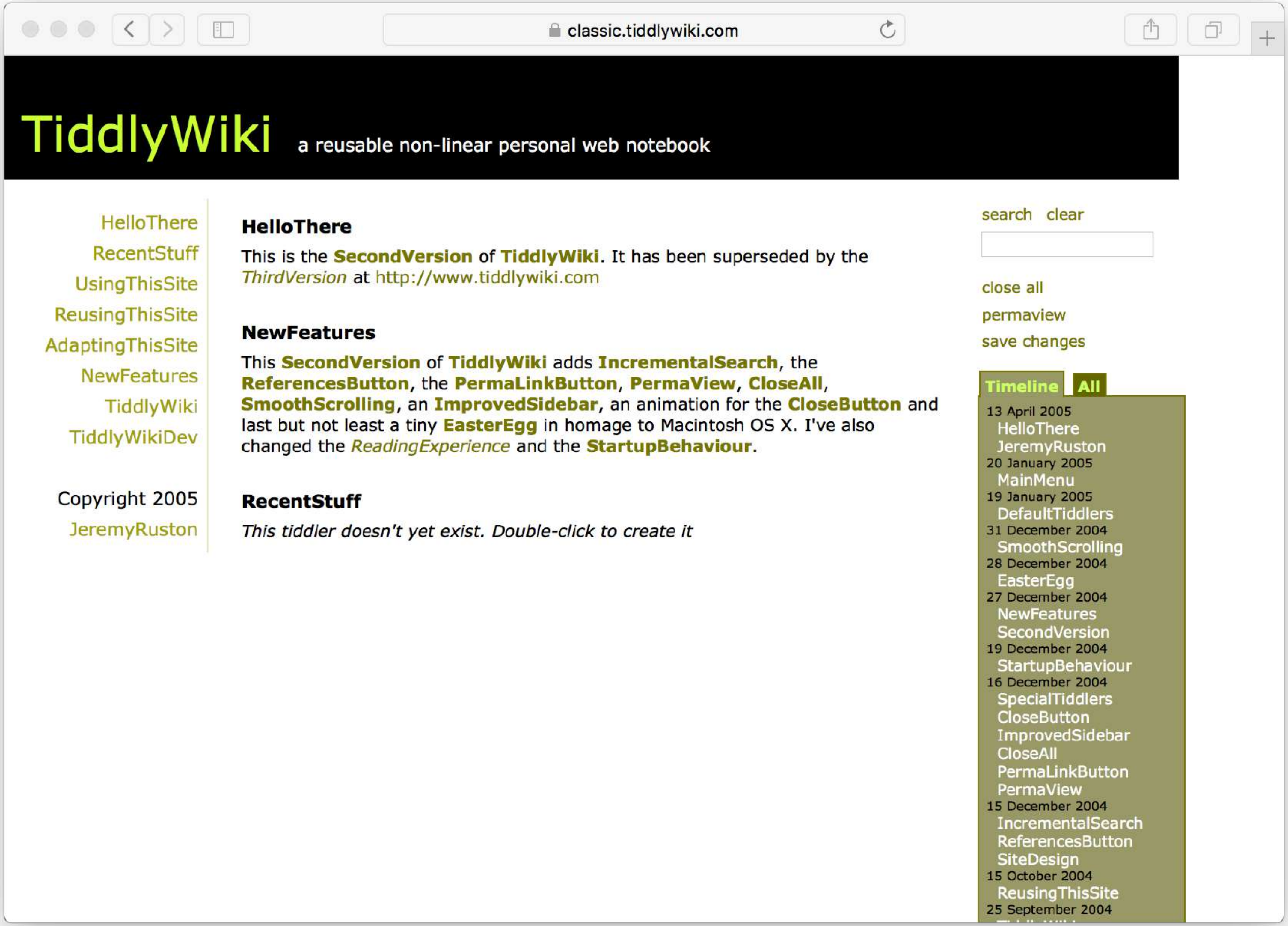
- Several people wrote server-side code to enable TiddlyWiki to save changes; result was that tech people could find their favourite flavour: PHP, Zope, Python

- Somebody made a Firefox extension that used XUL file system APIs to save locally

- Discovered those same APIs would work in a regular HTML page (but pop up a permission dialogue)

- Then discovered similar APIs in Internet Explorer

- Wow… a cross-platform self-contained app



HelloThere

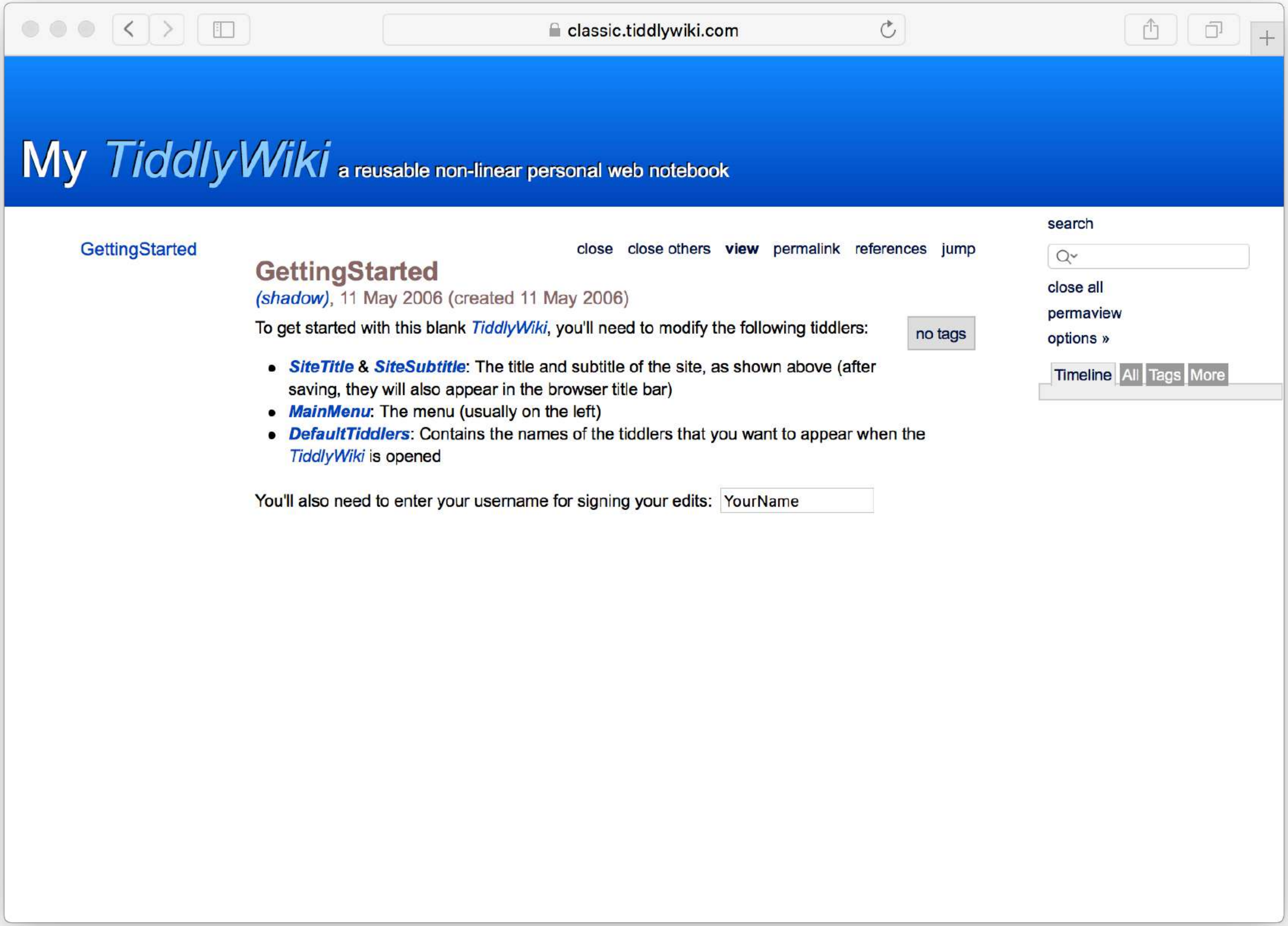
This is the **SecondVersion** of **TiddlyWiki**. It has been superseded by the *ThirdVersion* at http://www.tiddlywiki.com

NewFeatures

This **SecondVersion** of **TiddlyWiki** adds **IncrementalSearch**, the **ReferencesButton**, the **PermaLinkButton**, **PermaView**, **CloseAll**, **SmoothScrolling**, an **ImprovedSidebar**, an animation for the **CloseButton** and last but not least a tiny **EasterEgg** in homage to Macintosh OS X. I've also changed the *ReadingExperience* and the **StartupBehaviour**.

RecentStuff

This tiddler doesn't yet exist. Double-click to create it



My *TiddlyWiki* a reusable non-linear personal web notebook

GettingStarted

(shadow), 11 May 2006 (created 11 May 2006)

To get started with this blank *TiddlyWiki*, you'll need to modify the following tiddlers:

- ***SiteTitle*** & ***SiteSubtitle***: The title and subtitle of the site, as shown above (after saving, they will also appear in the browser title bar)
- ***MainMenu***: The menu (usually on the left)
- ***DefaultTiddlers***: Contains the names of the tiddlers that you want to appear when the *TiddlyWiki* is opened

You'll also need to enter your username for signing your edits: YourName

Insights

- The purpose of recording information is to reuse it.
- To optimise for reuse we need to cut information up into the smallest semantically meaningful chunks.
- We must also assign sufficient metadata for the fragments to be woven back together into a plurality of representations.
- Not everyone wants to run a server; the browser is the VM/container for everyone – provision a new VM with ctrl-T

Satisfying Things About TiddlyWiki

- Self contained, single file
- Entire user interface is constructed from a handful of wiki text primitives
- A practical Quine (a program that prints its own source code)

TiddlyWiki Philosophy

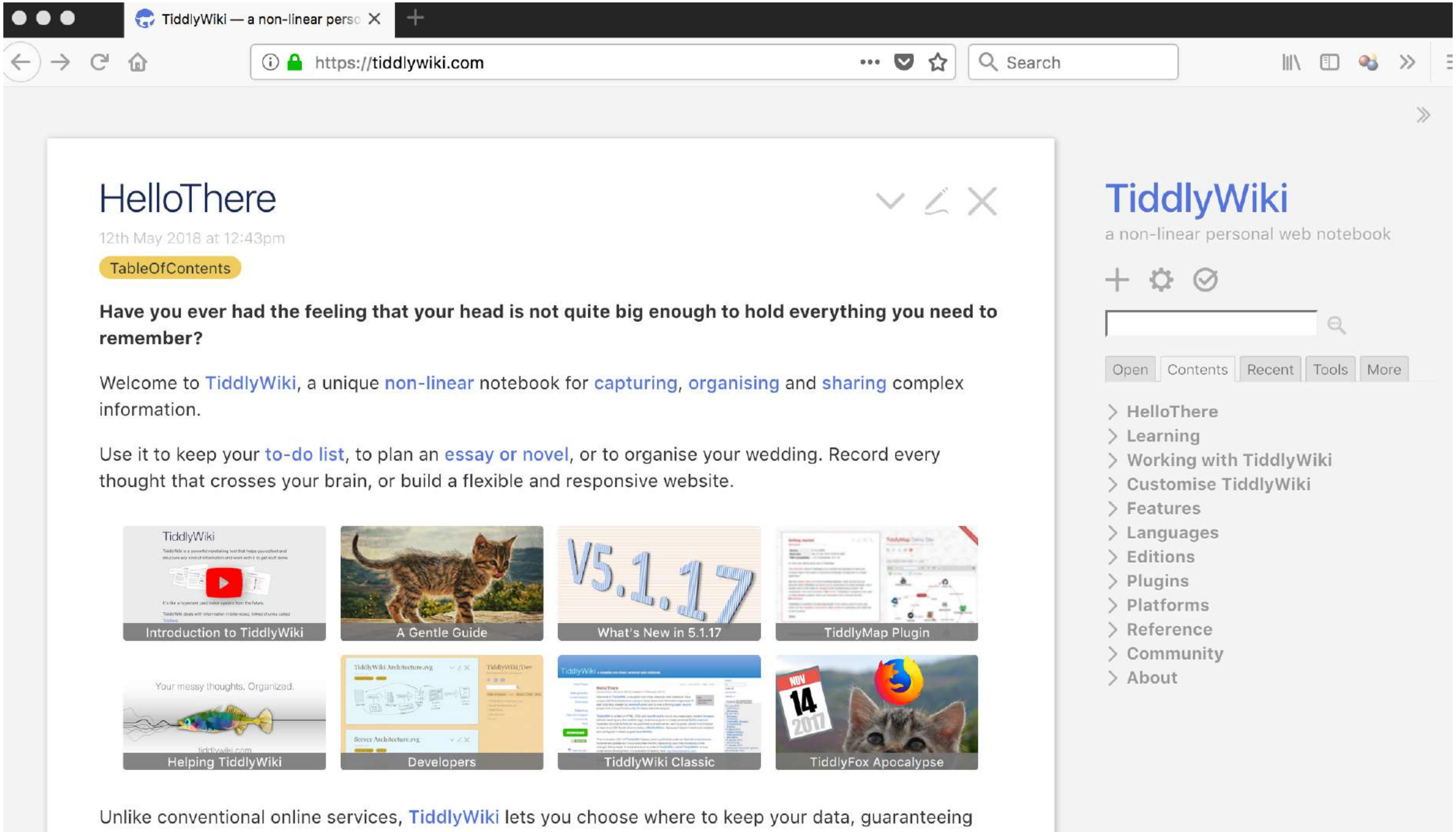
- Tiddlers are the smallest semantic units of information; everything is a tiddler
- The only process is wikification
- Solve the meta problem: don't build a notetaking application, build an application that can be used to create notetaking applications

TiddlyWiki 5 = Groundhog Day

- “A reboot of TiddlyWiki for the next 25 years” (2011 - will actually last much longer!)

- Still JavaScript, but runs under Node.js as well as the browser

- Completely redesigned internals: a small number of primitives recombined to make everything you see



HelloThere

12th May 2018 at 12:43pm

TableOfContents

Have you ever had the feeling that your head is not quite big enough to hold everything you need to remember?

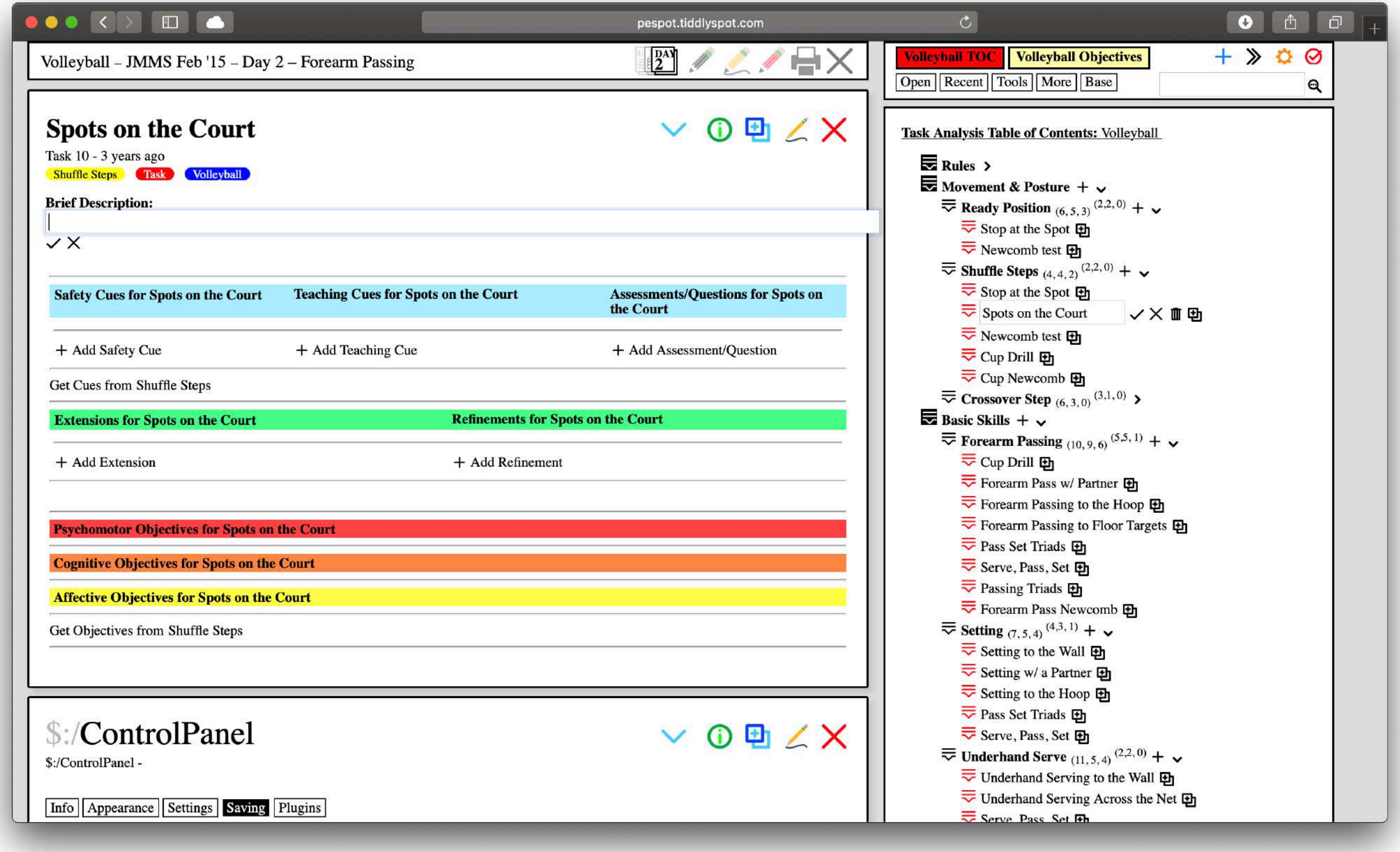
Welcome to TiddlyWiki, a unique non-linear notebook for capturing, organising and sharing complex information.

Use it to keep your to-do list, to plan an essay or novel, or to organise your wedding. Record every thought that crosses your brain, or build a flexible and responsive website.

Unlike conventional online services, TiddlyWiki lets you choose where to keep your data, guaranteeing

Unexpected Consequences of TiddlyWiki

- Empowers users by not needing a server
    - Setting up a server is hard
    - Using somebody else's server involves compromises to privacy and/or security
- Absurdly scalable: 100MB TW's work on a 2013 laptop



pespot.tiddlyspot.com

Volleyball – JMMS Feb '15 – Day 2 – Forearm Passing

DAY 2

Spots on the Court

Task 10 - 3 years ago

Shuffle Steps Task Volleyball

Brief Description:

✓ ✕

Safety Cues for Spots on the Court	Teaching Cues for Spots on the Court	Assessments/Questions for Spots on the Court
+ Add Safety Cue	+ Add Teaching Cue	+ Add Assessment/Question

Get Cues from Shuffle Steps

Extensions for Spots on the Court	Refinements for Spots on the Court
+ Add Extension	+ Add Refinement

Psychomotor Objectives for Spots on the Court

Cognitive Objectives for Spots on the Court

Affective Objectives for Spots on the Court

Get Objectives from Shuffle Steps

$:/ControlPanel

$:/ControlPanel -

Info Appearance Settings Saving Plugins

Volleyball TOC Volleyball Objectives

Open Recent Tools More Base

Task Analysis Table of Contents: Volleyball

- **Rules** ›
- **Movement & Posture** +
  - **Ready Position** (6, 5, 3) $^{(2,2,0)}$ +
    - Stop at the Spot
    - Newcomb test
  - **Shuffle Steps** (4, 4, 2) $^{(2,2,0)}$ +
    - Stop at the Spot
    - Spots on the Court ✓ ✕
    - Newcomb test
    - Cup Drill
    - Cup Newcomb
  - **Crossover Step** (6, 3, 0) $^{(3,1,0)}$ ›
- **Basic Skills** +
  - **Forearm Passing** (10, 9, 6) $^{(5,5,1)}$ +
    - Cup Drill
    - Forearm Pass w/ Partner
    - Forearm Passing to the Hoop
    - Forearm Passing to Floor Targets
    - Pass Set Triads
    - Serve, Pass, Set
    - Passing Triads
    - Forearm Pass Newcomb
  - **Setting** (7, 5, 4) $^{(4,3,1)}$ +
    - Setting to the Wall
    - Setting w/ a Partner
    - Setting to the Hoop
    - Pass Set Triads
    - Serve, Pass, Set
  - **Underhand Serve** (11, 5, 4) $^{(2,2,0)}$ +
    - Underhand Serving to the Wall
    - Underhand Serving Across the Net
    - Serve, Pass, Set

Disrupting High School Volleyball Teaching

- Made through trial and error by someone who isn’t a software developer…

- …without them having to explain what they wanted to somebody else (hard!)

- There’s no business model for building tools that are so specific to a particular niche. It’s hard to imagine raising VC funding for a software company specialising in high school volleyball

Experiments

What problems do we want to solve

- How can we re-use TW content?
- How can we avoid groundhog day

Provenance

Provenance

Noun

1. The place of origin or earliest known history of something.

Oxford English Dictionary

- We don’t know where the stuff in a file comes from

- The fault lies in editors “cut-and-paste” does not retain information about where the data came from (there should be an invisible tag recording this)

Where does content come from?

- Attribution
- IPR
- EDL’s
- CDL’s
- Blockchains
- Write append logs

A central idea in Xanadu

Tagging

Tagging

- Bayesian inference
- TF*IDF

Communicating Tiddly Wikis

Combining TWs

- One big namespace
- One namespace per TW
- All Tiddlers in a TW are visible
- Only exported tiddlers are visible
- Tiddler discovery
- Tiddler similarity
- Communication islands of knowledge

Ongoing Experiments

- Adding an Erlang like mailbox model for communicating TW's
(Inter-tiddler messaging, Inter TW-messaging)
- Analyse all known TWs
- Tag inference
- Focus of attention

Resources

https://jermolene.com/intertwingling

Thank You

Joe

https://www.sics.se/SSW2016/speakers/joe-armstrong

Jeremy

Personal: https://jermolene.com

Work: https://federatial.com