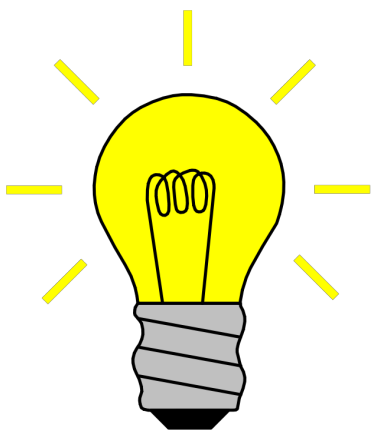
HTML5 Is the Future of Book Authorship

Digital Book World
January 14, 2014

Sanders Kleinfeld
O'Reilly Media, Inc.

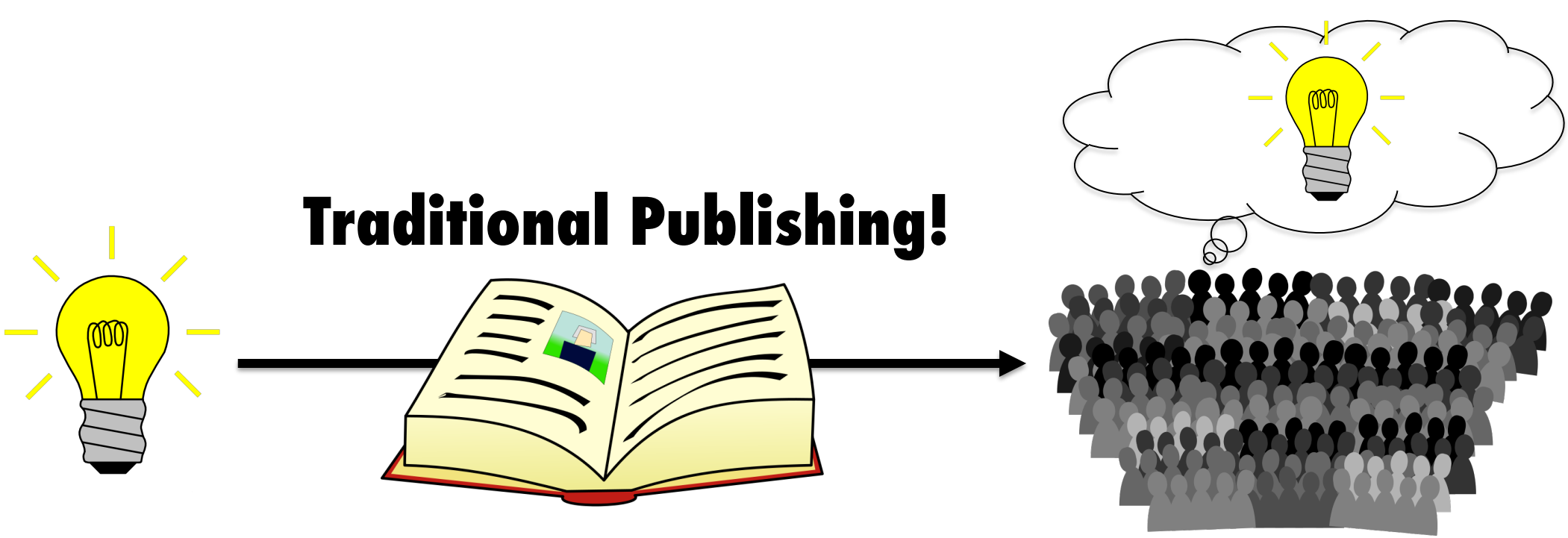
The Goal of Publishing:

Packaging and Distribution of Ideas

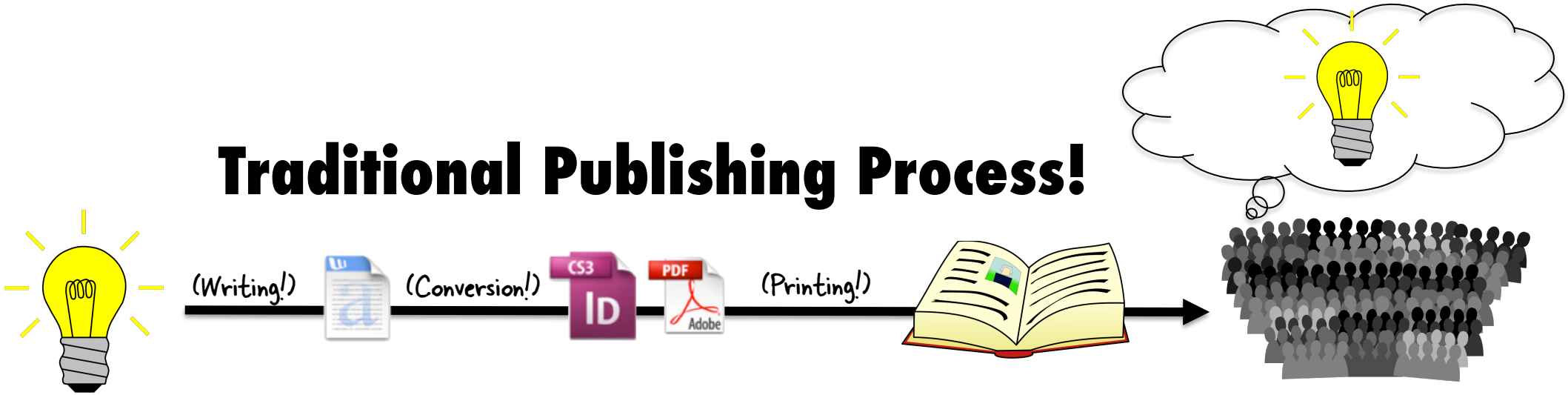


Publishing!

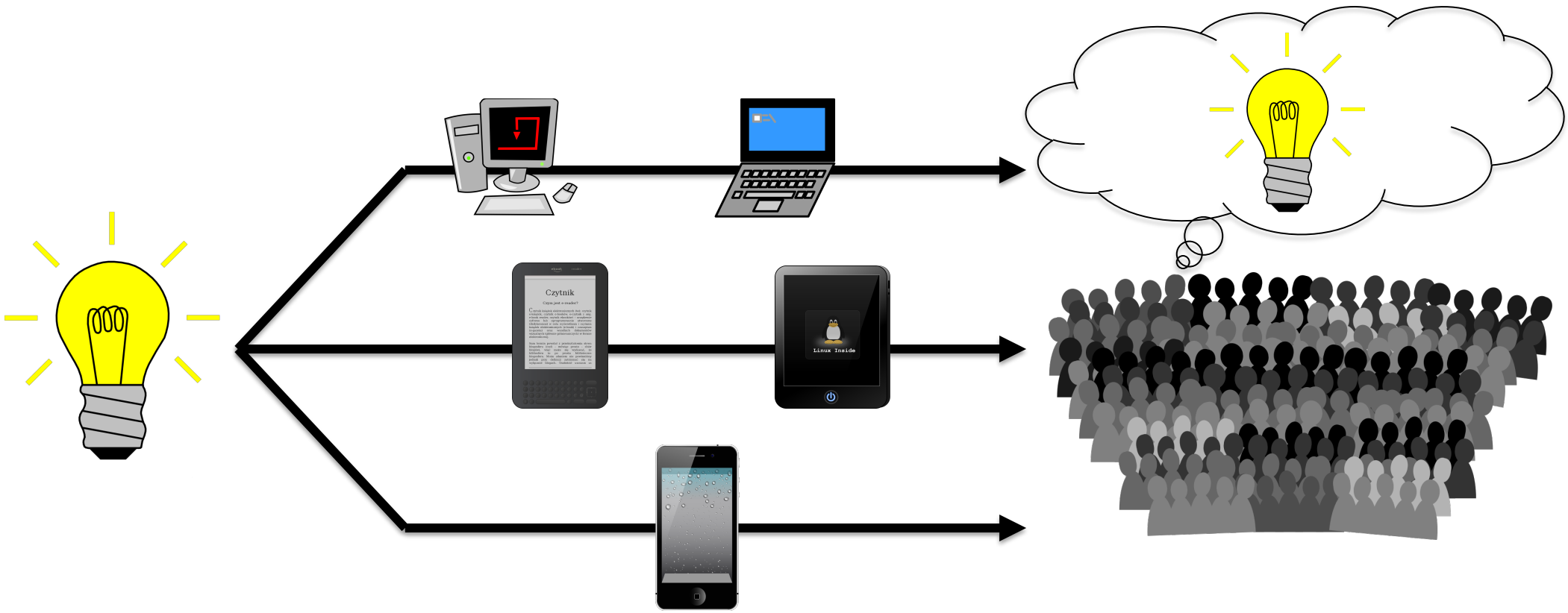
Traditional Publishing!



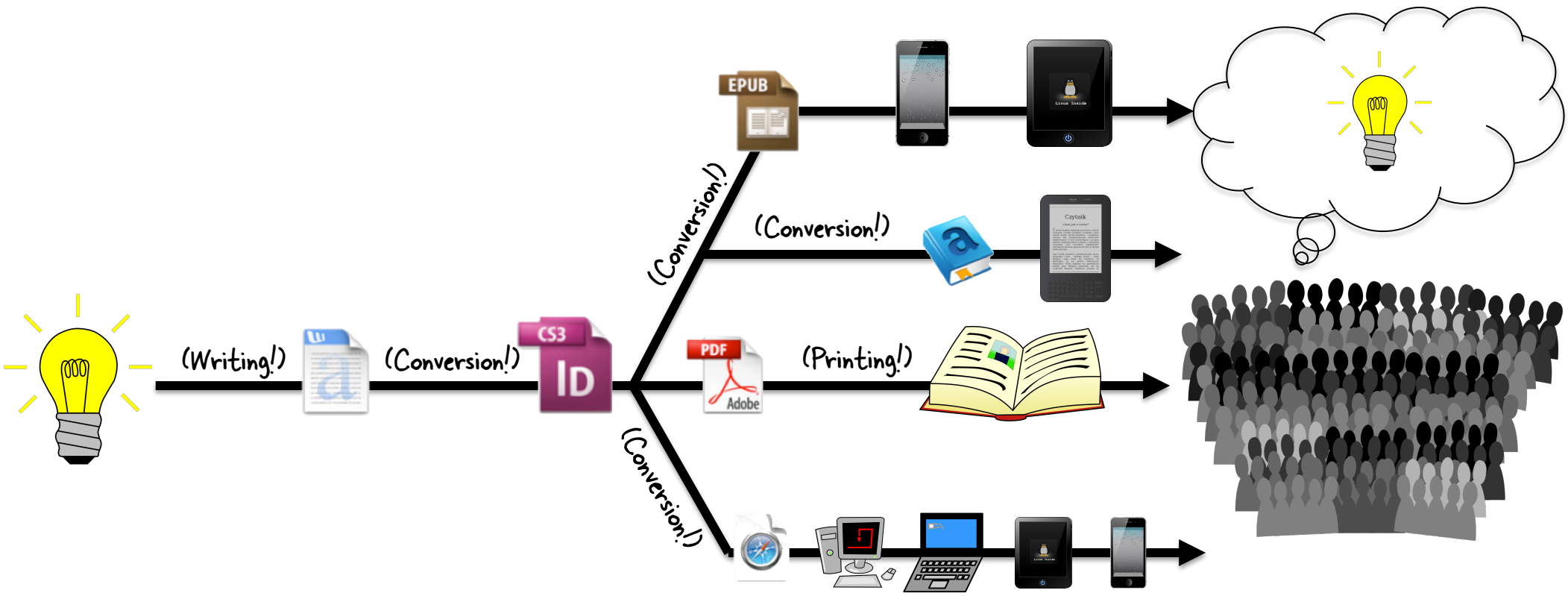
Traditional Publishing Process!



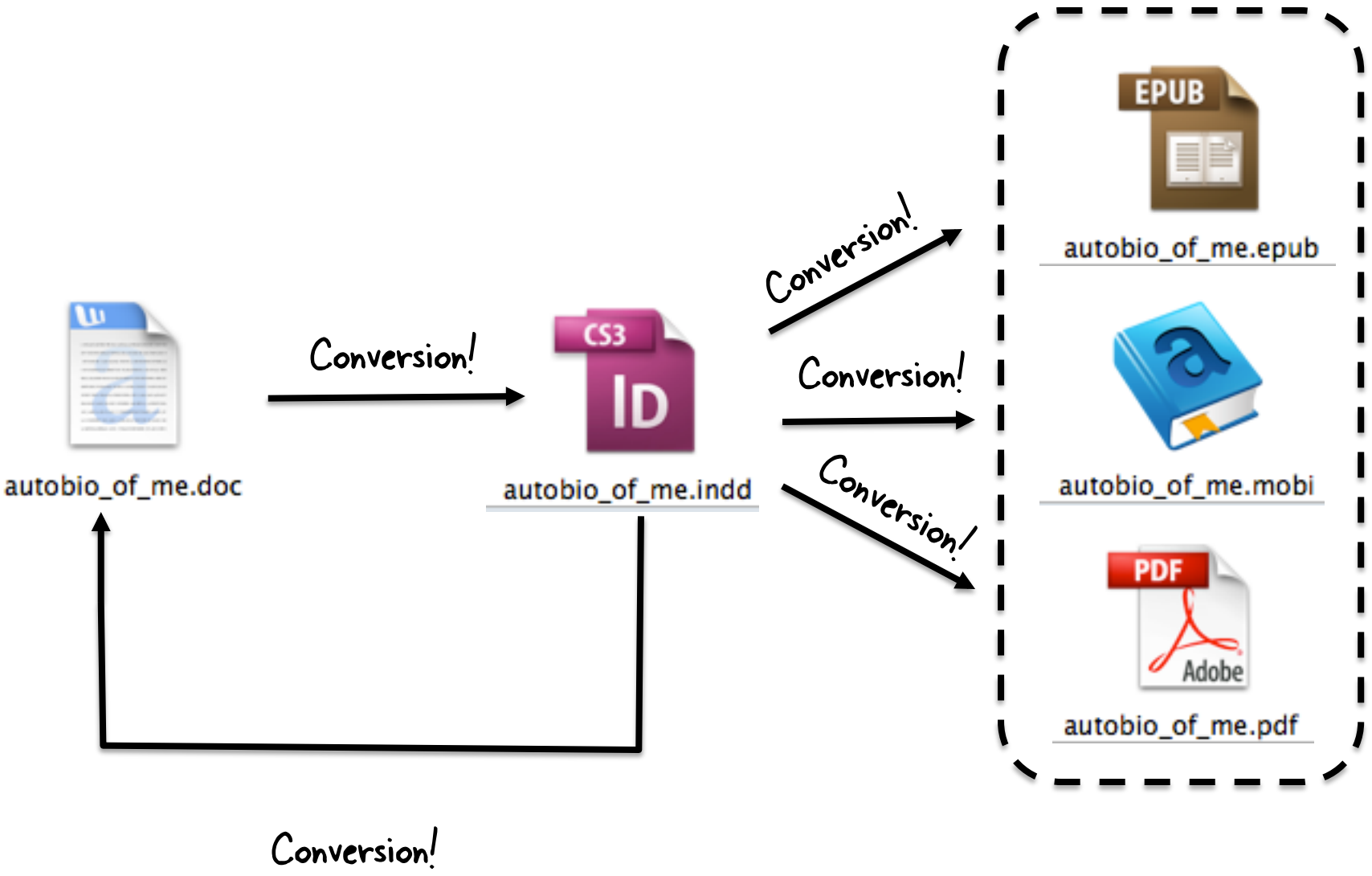
Digital Publishing!



(Post)Modern Publishing Process! (Both Print and Digital)



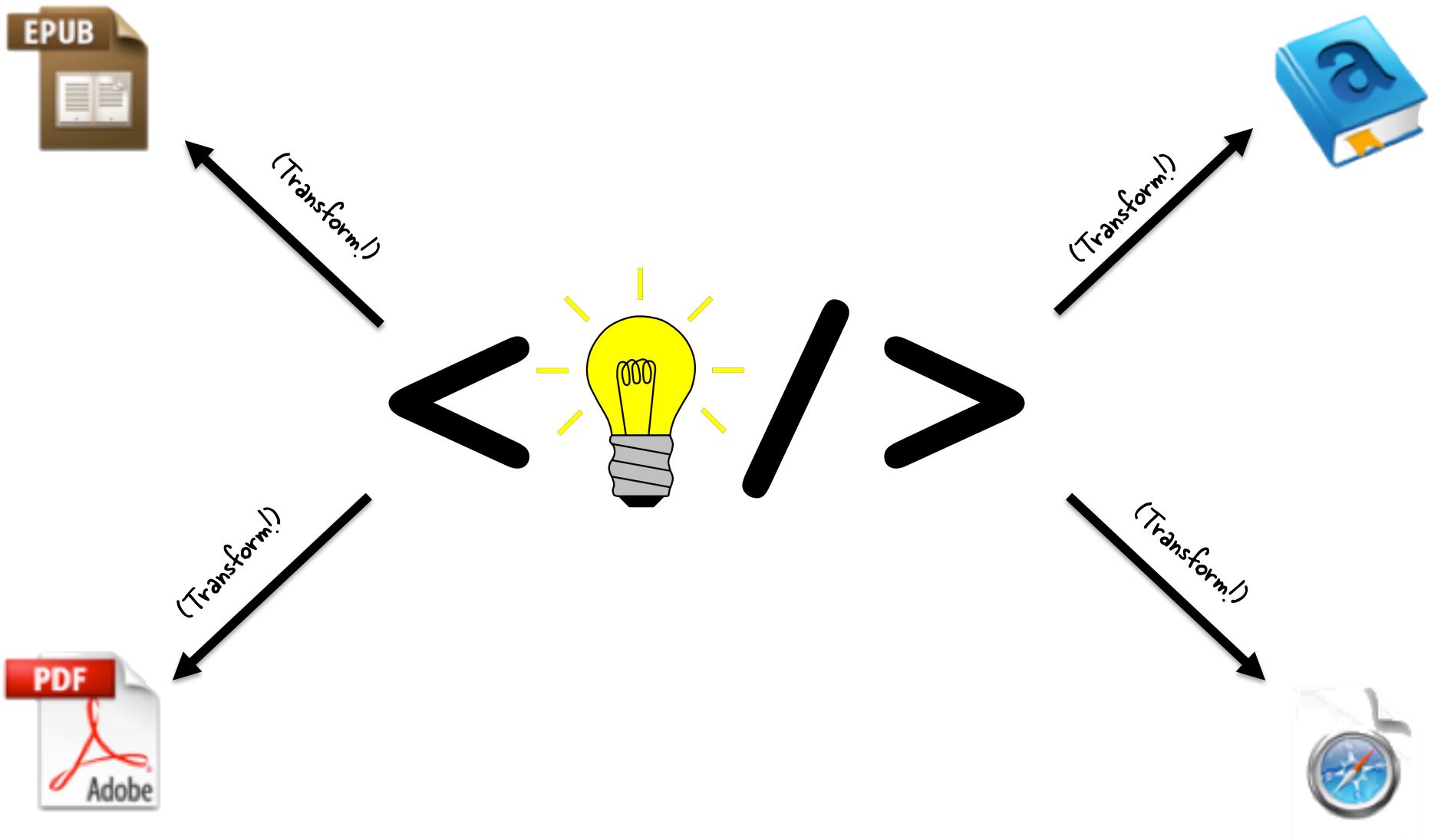
Welcome to Conversion City



Enjoy Your Stay ☺

The Single-Source Solution:

Replace conversions with semantic markup and automated transforms



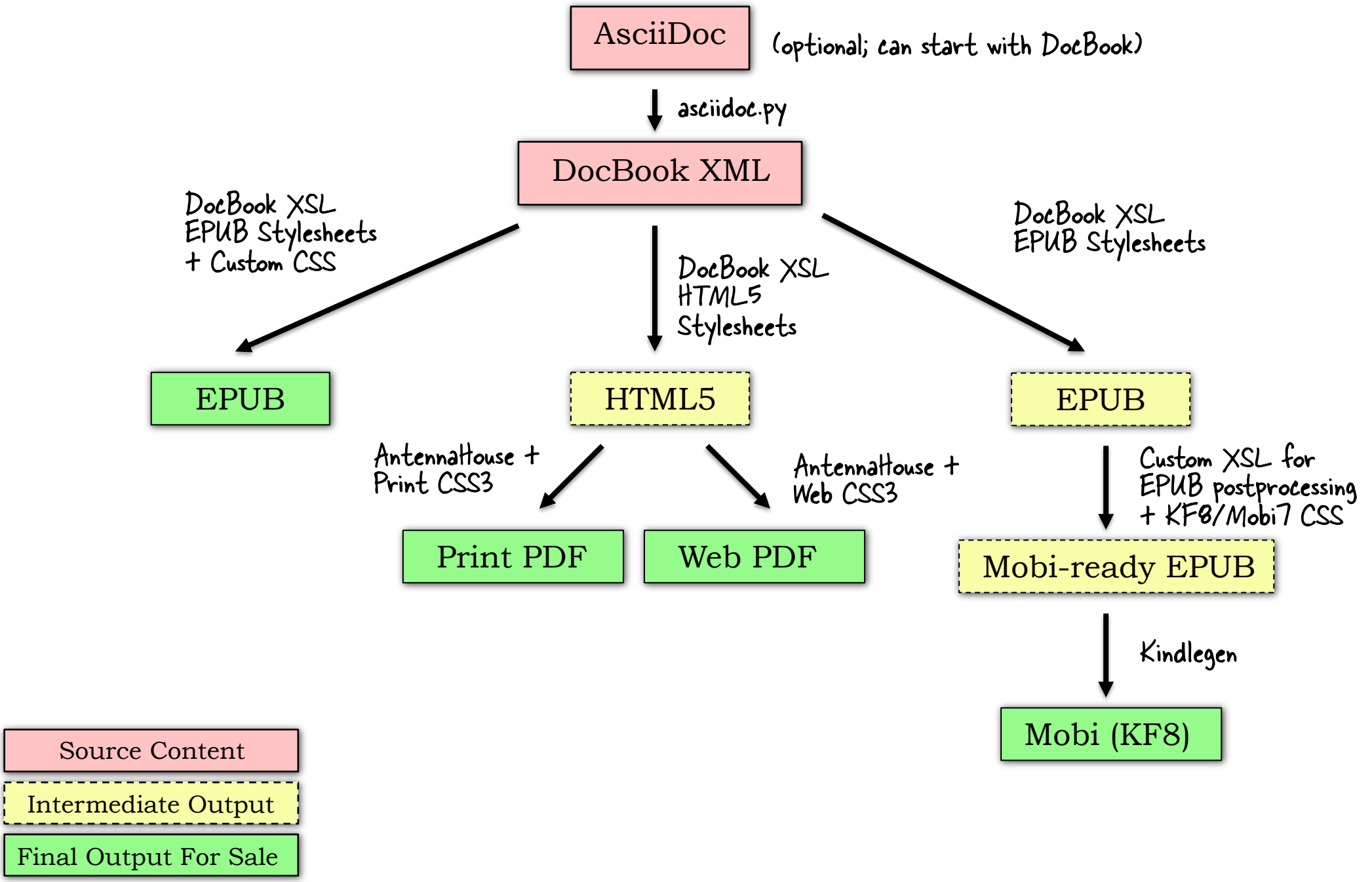
How We Did It:

1. Encourage authors to write in DocBook (heavyweight, *semantic* XML markup) or AsciiDoc (lightweight, *semantic* wiki-like "markdown")

2. If authors prefer to write in Microsoft Word (☹), let them (☹☹☹), but convert to DocBook when book goes into Production

3. Maintain a customized version of the DocBook project stylesheets for automatically generating print/ebook outputs

O'Reilly's Single-Source Workflow (2006-2013):



Three Slowly Dawning Realizations About Our Workflow

Realization #1: Our toolchain is rather heavyweight, complex

PDF* Toolchain Stats

The DocBook project XHTML5 stylesheets contain:**

- **33,707 lines of HTML-generation code...**
- **...which rely on 8,346 lines of common dependencies**

Or, in terms of functions, they contain:

- **1,857 <xsl:template>s...**
- **...which rely on 272 common dependency <xsl:template>s**

* Separate code base for EPUB/Mobi
** docbook-epub3-addon-b3

When doing transforms, this complexity is a necessary evil, emphasis on *evil*

Peril of a Transform-Heavy Workflow:

Troubleshooting is a real $!%#@&*

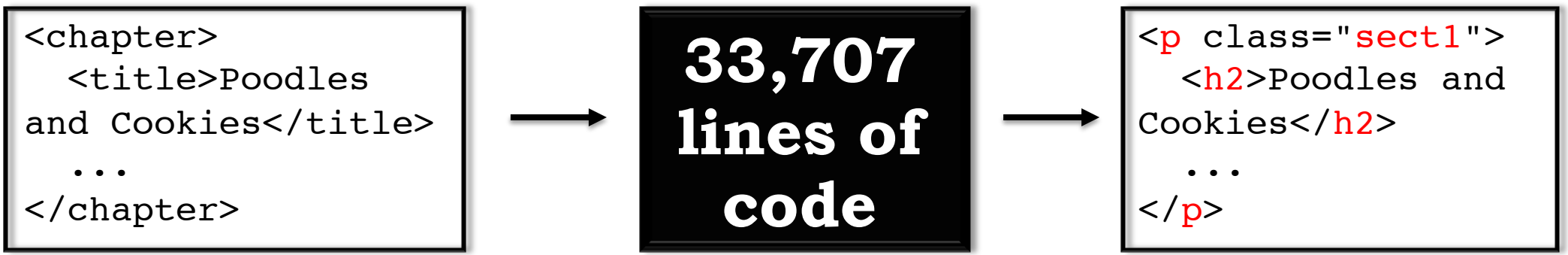
An example!
Let's say your DocBook source is:

```
<chapter>
  <title>Poodles and Cookies</title>
  ...
</chapter>
```

And your desired/expected HTML output is:

```
<div class="chapter">
  <h1>Poodles and Cookies</h1>
  ...
</div>
```

But then you run your transform...



...And you say, "What the $!%#@&*?"

What about implementing validation to preemptively catch errors before running the transforms?

Validation is still troubleshooting:

```
book.xml:2192: element chapter: validity error : Element chapter content
does not follow the DTD, expecting (beginpage? , chapterinfo? , (title ,
subtitle? , titleabbrev?) , (toc | lot | index | glossary |
bibliography)* , tocchap? , (((calloutlist | glosslist | bibliolist |
itemizedlist | orderedlist | segmentedlist | simplelist | variablelist |
caution | important | note | tip | warning | literallayout |
programlisting | programlistingco | screen | screenco | screenshot |
synopsis | cmdsynopsis | funcsynopsis | classsynopsis | fieldsynopsis |
constructorsynopsis | destructorsynopsis | methodsynopsis | formalpara |
para | simpara | address | blockquote | graphic | graphicco |
mediaobject | mediaobjectco | informalequation | informalexample |
informalfigure | informaltable | equation | example | figure | table |
msgset | procedure | sidebar | qandaset | task | anchor | bridgehead |
remark | highlights | abstract | authorblurb | epigraph | indexterm |
beginpage)+ , (sect1* | refentry* | simplesect* | section*)) | sect1+ |
refentry+ | simplesect+ | section+) , (toc | lot | index | glossary |
bibliography)*), got (para title para para para sect1 sect1 sect1 sect1
sect1 sect1 sect1 sect1 sect1 sect1 sect1 sect1 )
  </chapter>
```

Again you say, “What the $!%#@&*?”

Streamlining Production Workflows isn't just about *automating conversions* whenever possible…

…It's about *eliminating conversions* whenever possible!

Rather than build a toolchain around:

Or:

Or:

Why not seriously consider:

Realization #2: HTML5 Is Ideal for Digital-First Content

Digital-First Content Development

When doing digital-first (ebook/web) content development, these are the key output formats:

- **EPUB (2.0 and 3.0)** 

- **Amazon Kindle Mobi (Mobi7/KF8)** 
- **PDF** 

- **HTML** 

* docbook-epub3-addon-b3

The Common Thread: HTML + CSS

= HTML + CSS + open source packaging

= HTML + CSS + proprietary packaging

= HTML + CSS + PDF processor*

= HTML + CSS (duh!)

* e.g., AntennaHouse Formatter or Prince

Interactivity/Multimedia Is Ultimately All About HTML5

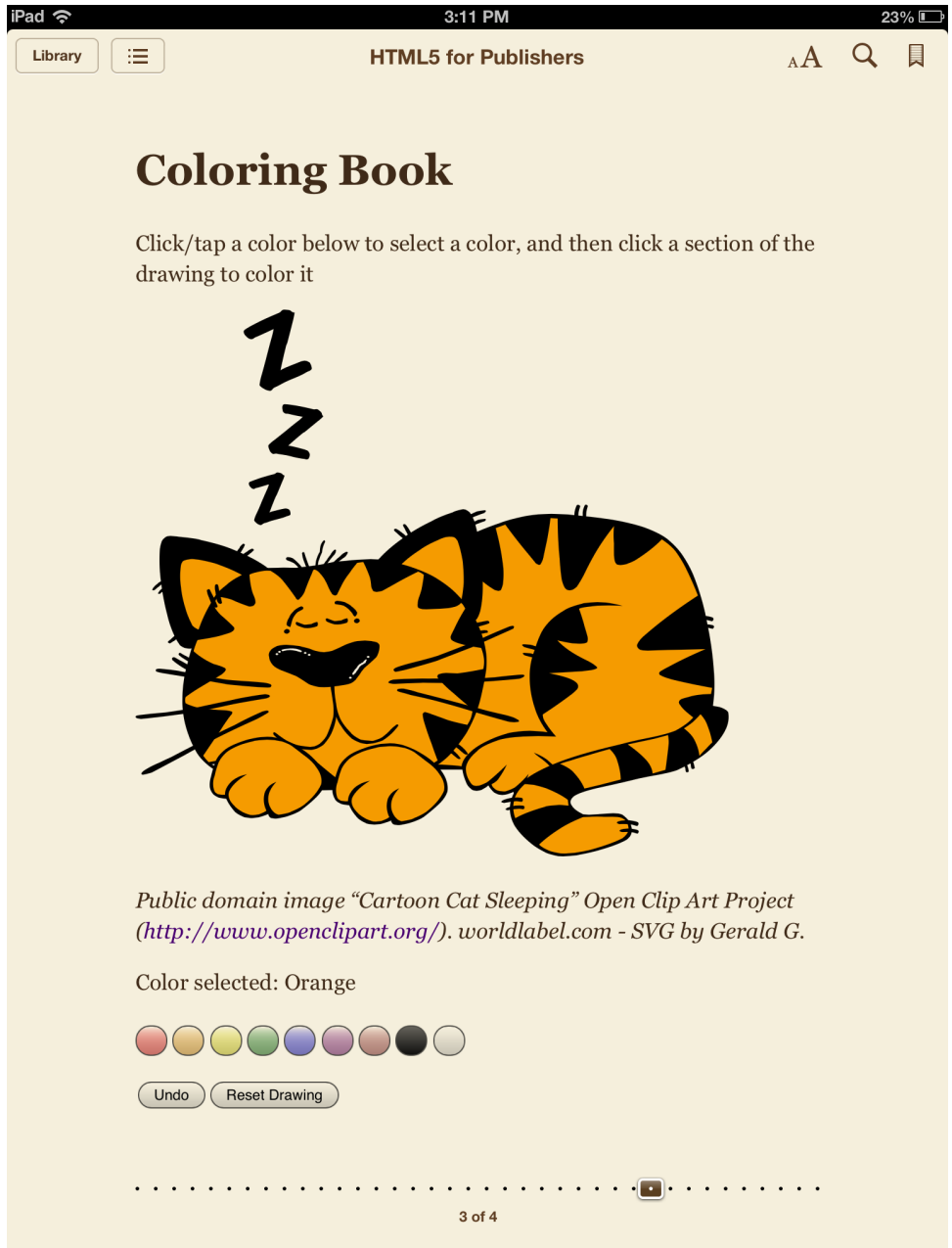
Animation/Games → `<canvas> or <svg>`

Music/Narration → `<audio>`

Video Clips → `<video>`

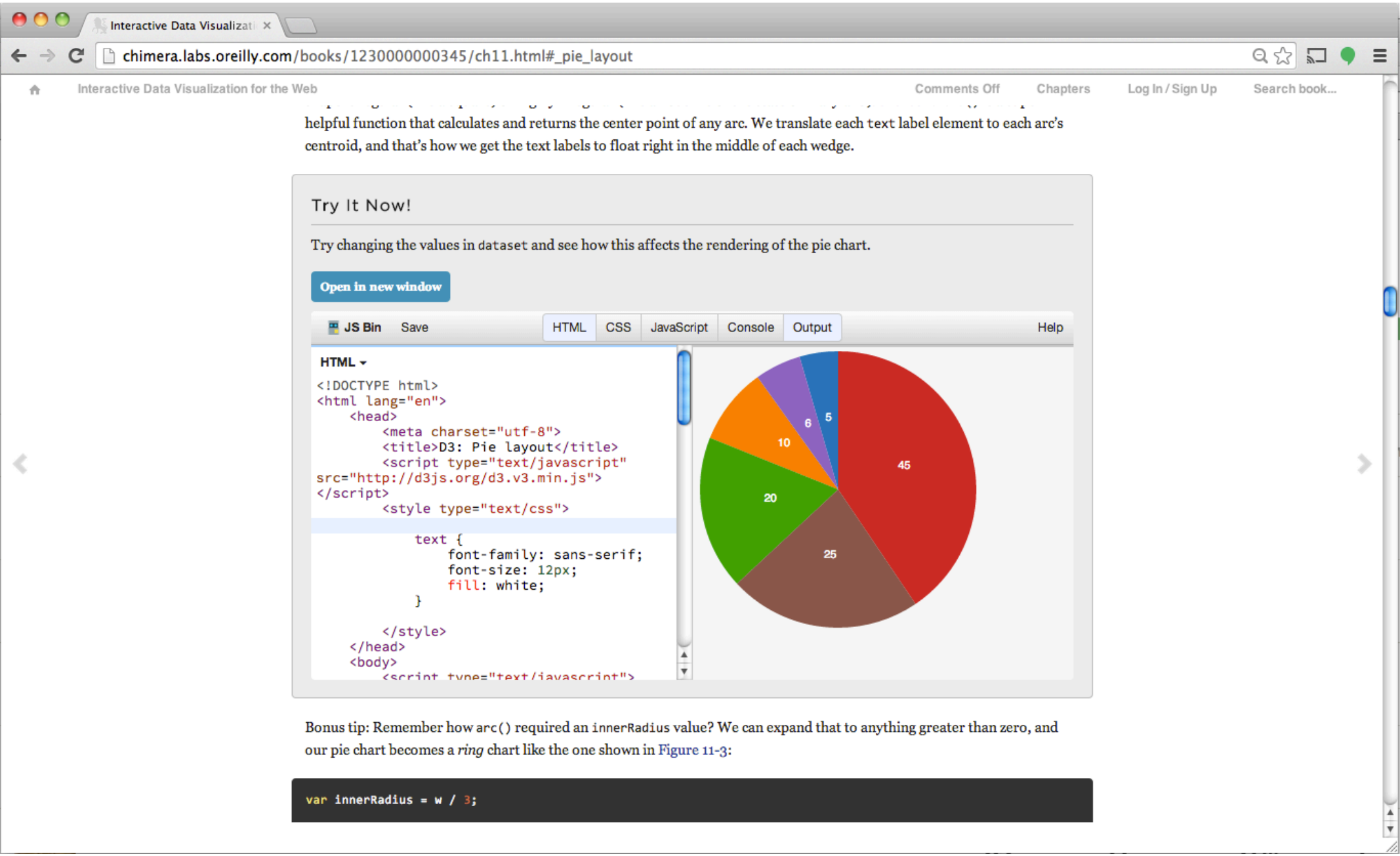
Math Equations → `<math>`

It’s way easier to do stuff like this:



If you start with HTML5

It's way easier to do stuff like this:



If you start with HTML5

Realization #3: Authors Generally Don't Want to Deal with Markup

Authors prefer *visual* authoring platforms because...

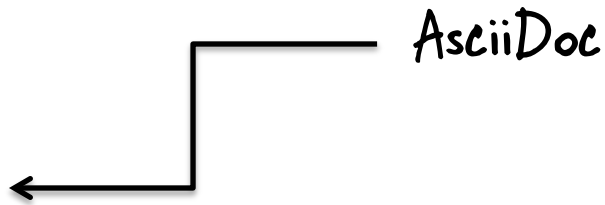
“Nobody’s going to learn your markup language”

Books in Browsers 2012: Liza Daly & Keith Fahlgren,
“The self-publishing book”
http://www.youtube.com/watch?v=UWftLHopWQ0#t=5m25s

Non-Technical Authors Don't Like This...


```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd">
<chapter>
  <title>Autobiography of Me</title>
  <para>I was born in 1980, I love chocolate ice cream, and I
am a <emphasis>wicked awesome</emphasis> writer, yo!</para>
</chapter>
```

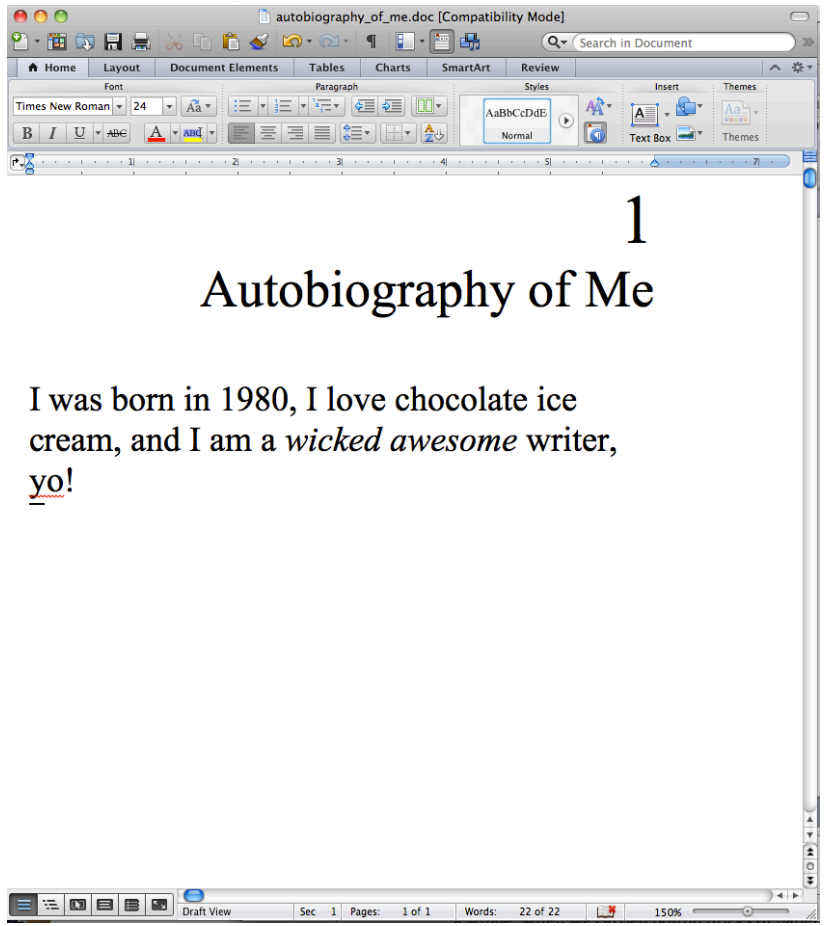
Non-Technical Authors Will Sometimes Tolerate This...




```
== Autobiography of Me

I was born in 1980, I love chocolate ice cream,
and I am a _wicked awesome_ writer, yo!
```

Non-Technical Authors Really Want This...

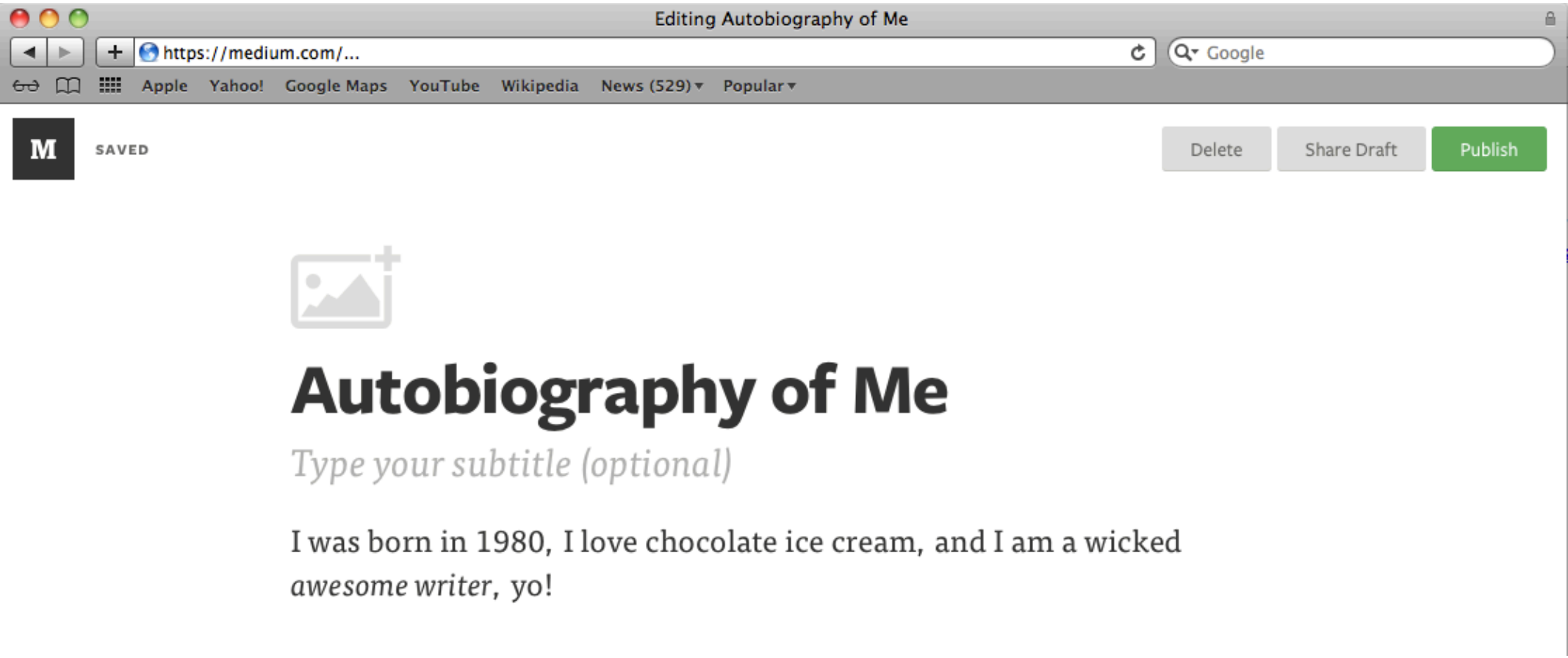


Microsoft Word

But This is the Future of Digital Content Creation:

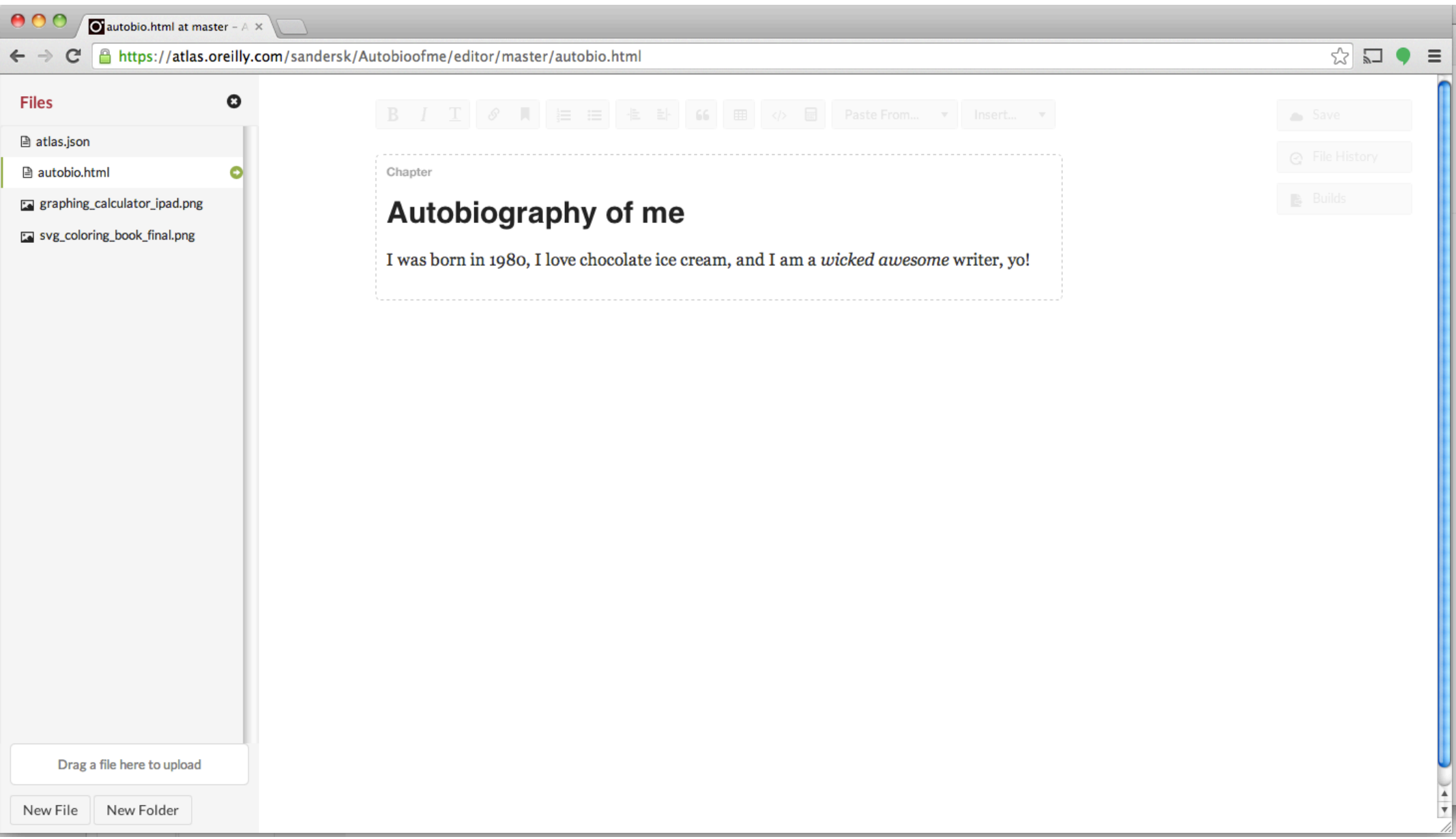
Medium

(Short-Form Web Publishing)



O'Reilly Atlas

(Short and Long-Form Print, Digital, and Web Publishing)



Next-Generation Content Authoring =

- Visual Editing
- Web-Based (Responsive Design)
- Version-Controlled
- Seamless

Visual Editing

Connecting_an_LED.html

https://atlas.oreilly.com/oreilly-media-content/razzpisampler/editor/master/Connecting_an_LED.html

Back

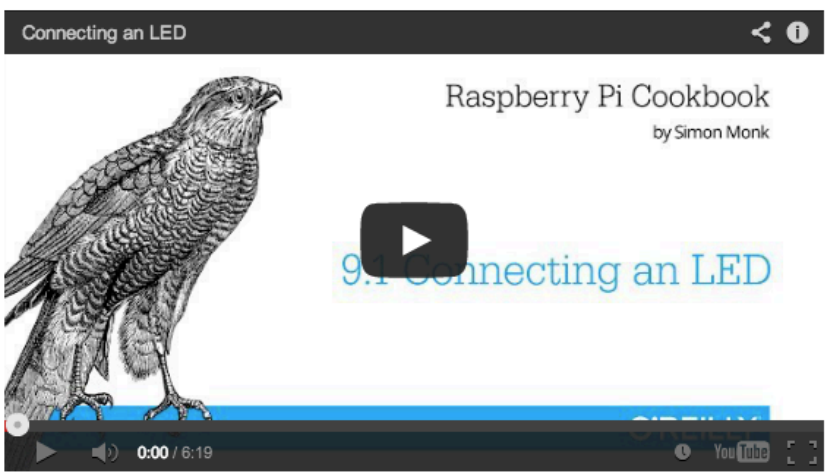
Code Editor

Files

B I T Paste From... Insert...

Chapter

Connecting an LED



Section 1

Problem

You want to know how to connect an LED to the Raspberry Pi.

Section 1

Solution

Connect an LED (see "Opto-Electronics") to one of the GPIO pins using a 470Ω or 1kΩ series resistor (see "Resistors and Capacitors") to limit the current. To make this

Builds

PDF EPUB
MOBI HTML

Build

10 minutes ago
HTML
Build Log

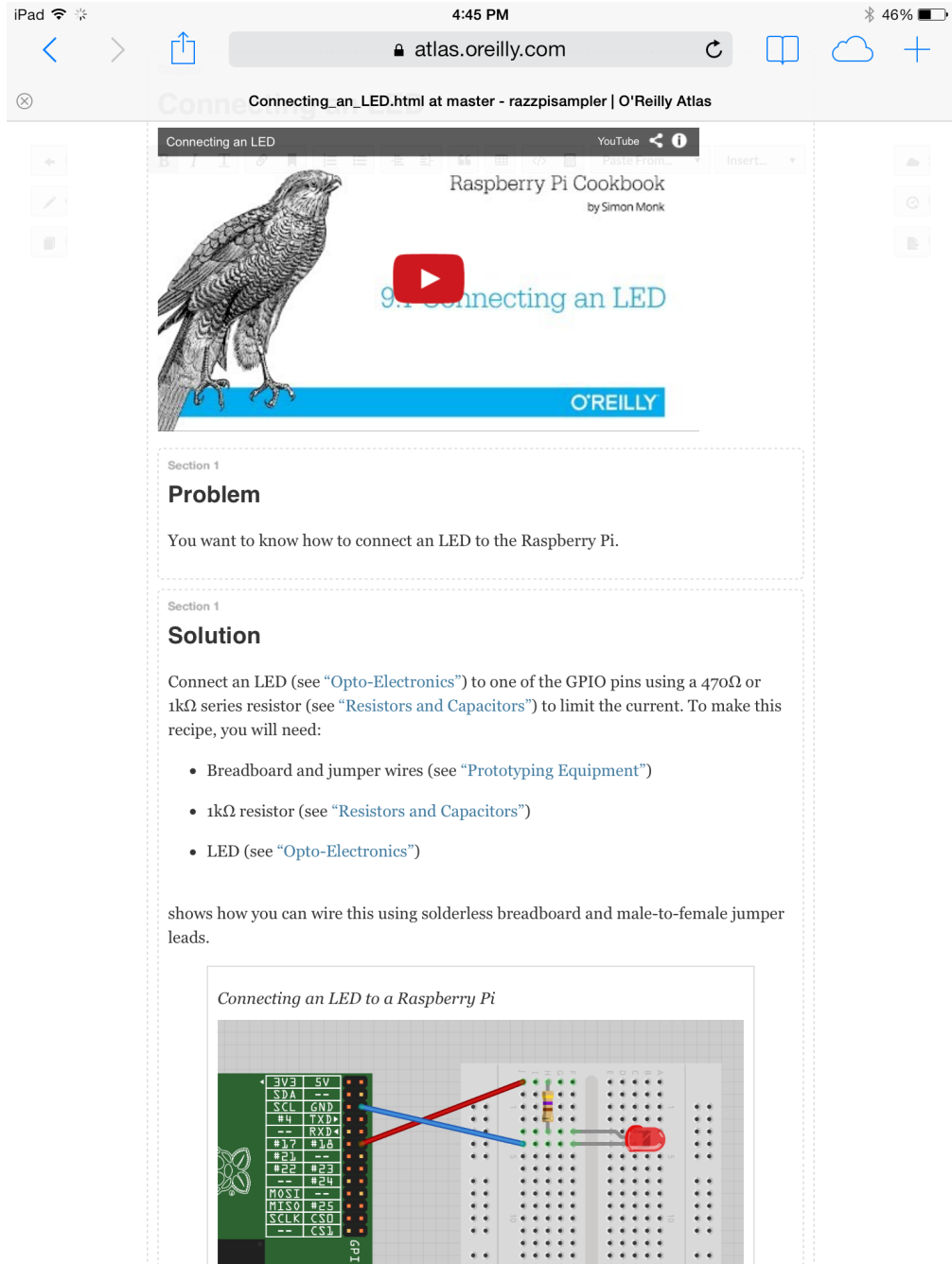
11 minutes ago
PDF
Build Log

17 days ago
HTML Download
Build Log

17 days ago
HTML Download
Build Log

20 days ago

Responsive Design



On Version Control...

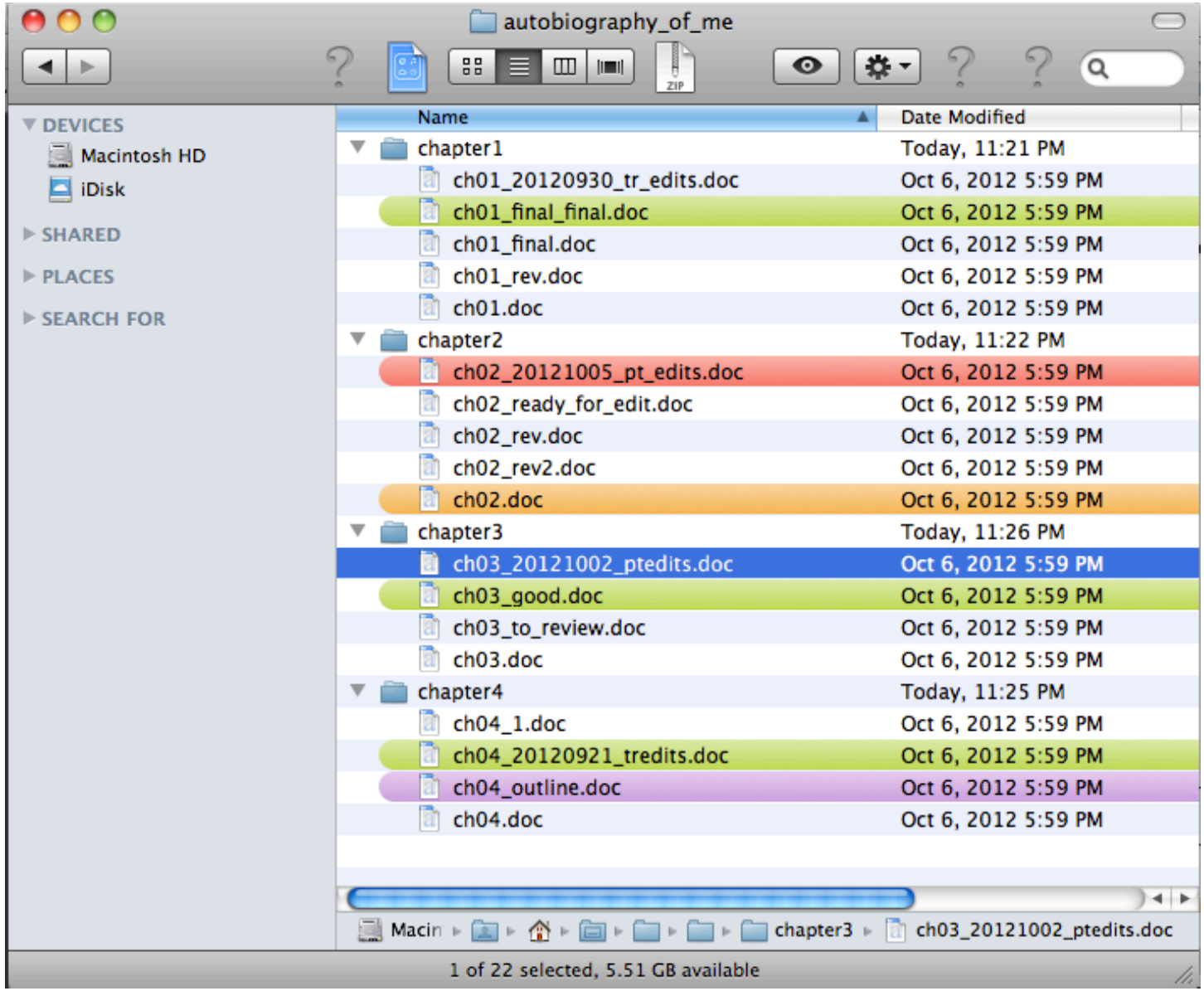
Two Questions About Your (e)Book's Editorial Lifecycle

1. Will more than one person be working on the manuscript files?

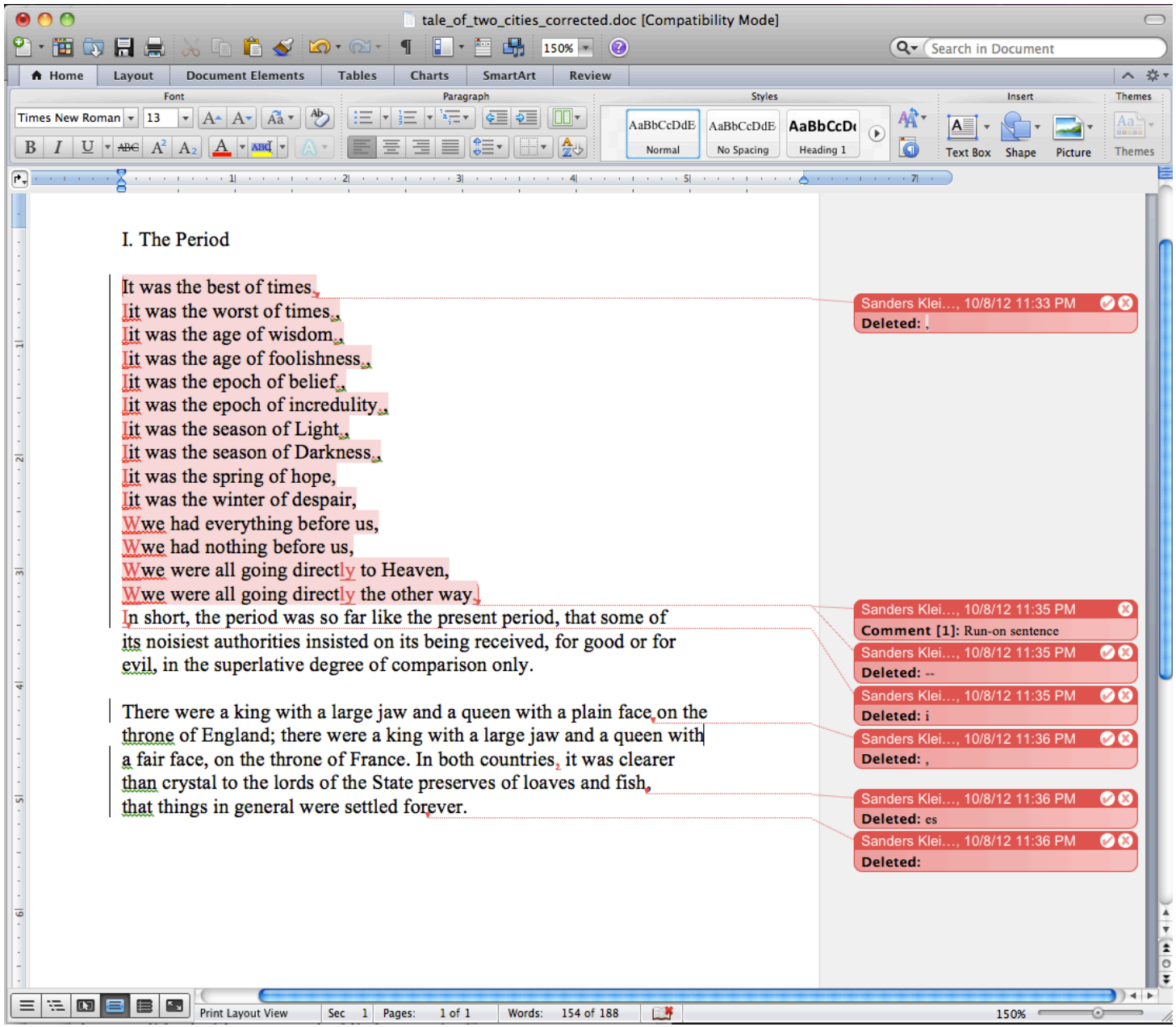
2. Will there be more than one draft of the manuscript?

If you answered *yes* to either question, you *need* a version-control system.

Key Feature #1 of Version Control: Revision Snapshots

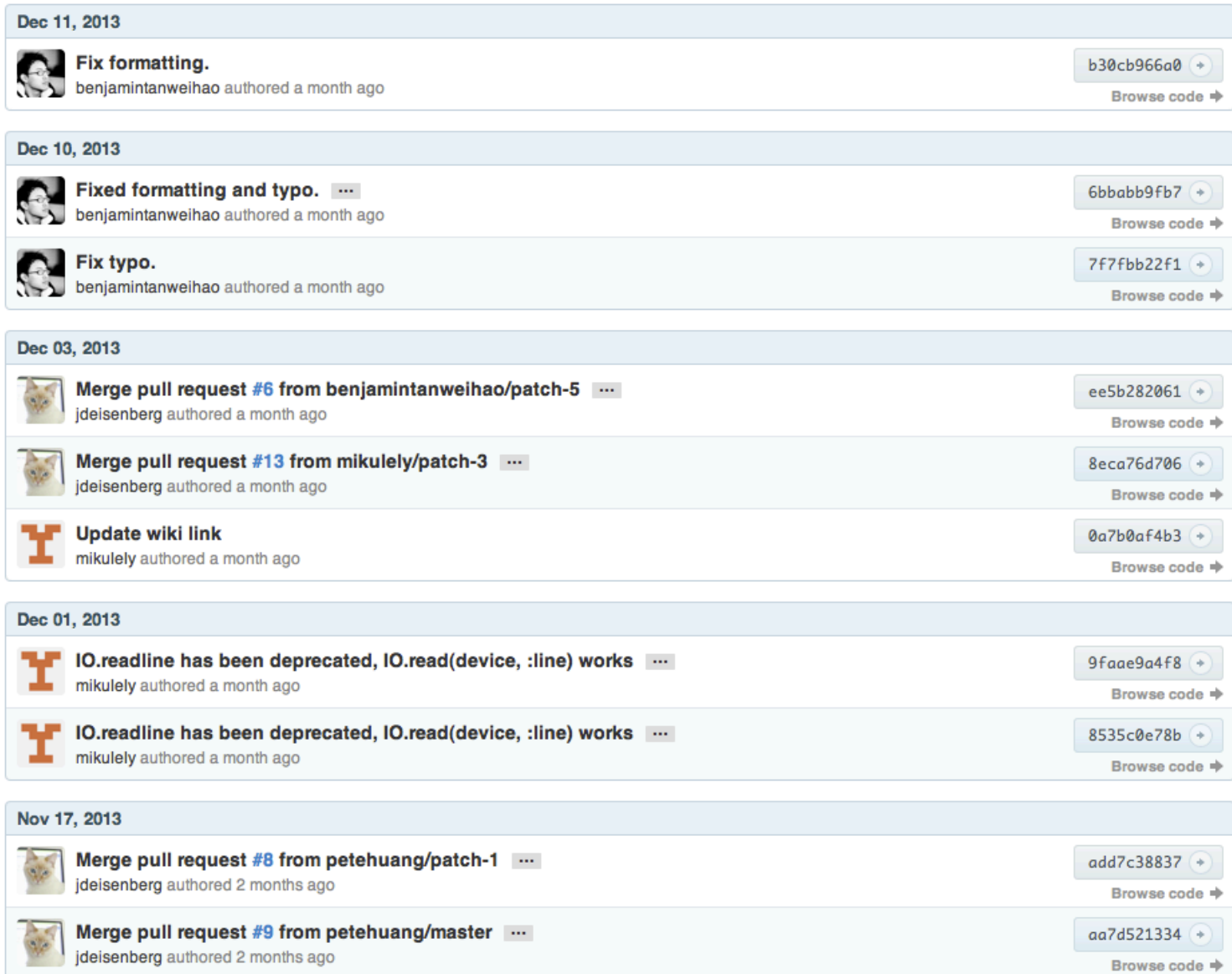


Key Feature #2 of Version Control: *Diffing*



What if we versioned manuscripts like software developers version code?

Revision snapshots



***Études for Elixir*: https://github.com/oreillymedia/etudes-for-elixir**

Diffing

PUBLIC oreillymedia / etudes-for-elixir

Unwatch 20 | Star 66 | Fork 18

Changed grammar error in comments for solution ch07-06.

Browse code

master

jdeisenberg authored a day ago 1 parent 47a70c1 commit 5933a23555ad9c371c5679b60f545901d7613771

Showing **1 changed file** with **1 addition** and **1 deletion**.

Show Diff Stats

2 code/ch07-06/cards.ex View file @ 5933a23

```
@@ -35,7 +35,7 @@ defmodule Cards do
   # That first card goes onto a new pile ([h | acc]).
   # Now put together the part above the split and the
   # part below the split (leading ++ t) and go through
-  # the process with the deck (which is now has one less card).
+  # the process with the deck (which now has one less card).
   # This keeps going until you run out of cards to shuffle;
   # at that point, all the cards will have gotten to the
   # new pile, and that's your shuffled deck.
```

0 notes on commit 5933a23

Show line notes below

***Études for Elixir*: https://github.com/oreillymedia/etudes-for-elixir**

Seamless Authoring & Production...

Step #1: Author!

Connecting_an_LED.html

https://atlas.oreilly.com/oreilly-media-content/razzpisampler/editor/master/Connecting_an_LED.html

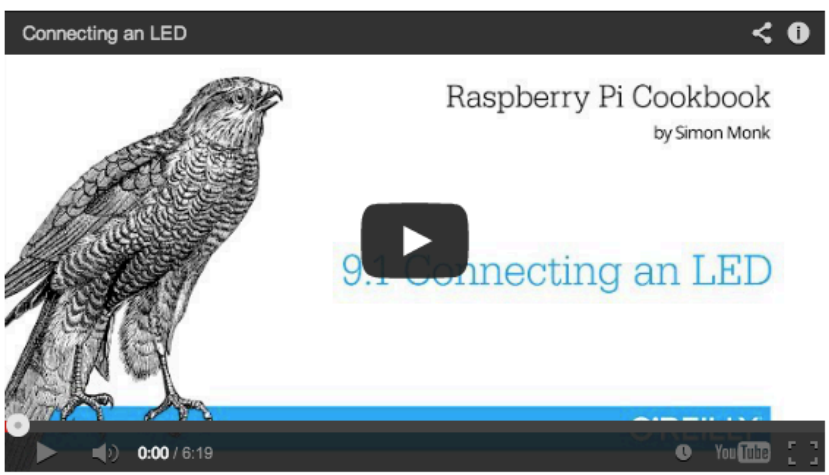
Back | Code Editor | Files

B I T | Paste From... | Insert...

Save | File History | Builds

Chapter

Connecting an LED



Section 1

Problem

You want to know how to connect an LED to the Raspberry Pi.

Section 1

Solution

Connect an LED (see "Opto-Electronics") to one of the GPIO pins using a 470Ω or 1kΩ series resistor (see "Resistors and Capacitors") to limit the current. To make this

Step #2: Build!

Connecting_an_LED.html

https://atlas.oreilly.com/oreilly-media-content/razzpisampler/editor/master/Connecting_an_LED.html

Back

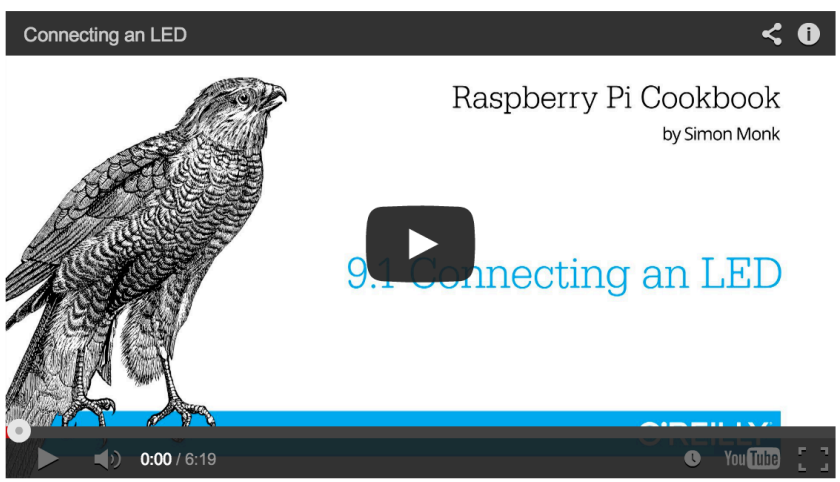
Code Editor

Files

Paste From... Insert...

Chapter

Connecting an LED



Section 1

Problem

You want to know how to connect an LED to the Raspberry Pi.

Section 1

Solution

Connect an LED (see "Opto-Electronics") to one of the GPIO pins using a 470Ω or 1kΩ

Builds

PDF EPUB
MOBI HTML

Build

a few seconds ago
EPUB Download
MOBI Download
HTML Download
PDF Download
Build Log

a few seconds ago
EPUB Download
MOBI Download
PDF Download
HTML Download
Build Log

a day ago
HTML Download

Step #3: Review Results!

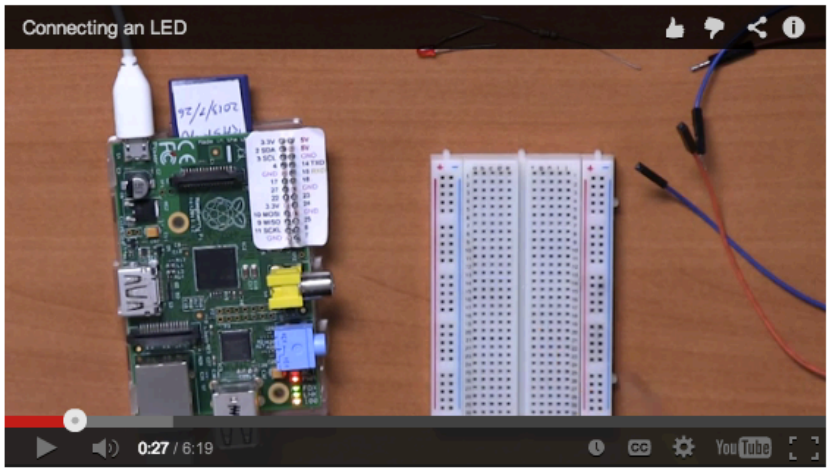
razzpisampler

razzpisampler.oreilly.com/ch03.html

O'REILLY Raspberry Pi Cookbook | Previous | Next

Discuss on GitHub

Connecting an LED



Problem

You want to know how to connect an LED to the Raspberry Pi.

Solution

Connect an LED (see "Opto-Electronics") to one of the GPIO pins using a 470Ω or 1kΩ series resistor (see "Resistors and Capacitors") to limit the current. To make this recipe, you will need:

- Breadboard and jumper wires (see "Prototyping Equipment")
- 1kΩ resistor (see "Resistors and Capacitors")
- LED (see "Opto-Electronics")

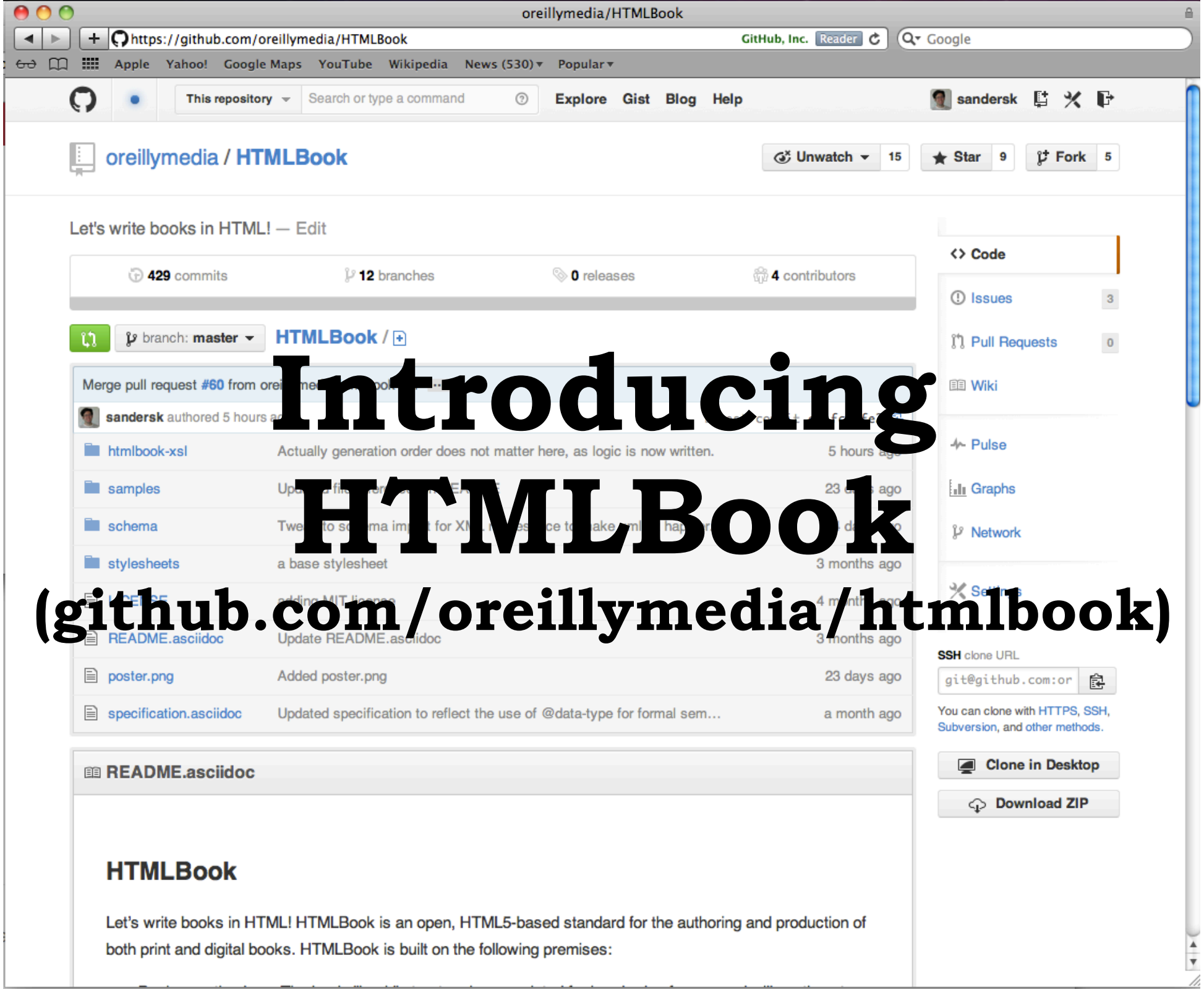
Figure 3-1 shows how you can wire this using solderless breadboard and male-to-female jumper

Sampler Contents:

Buy the book on oreilly.com.

All Roads Lead to HTML5...

...But Is It Semantic Enough for Book Publishing?



Introducing HTMLBook

(github.com/oreillymedia/htmlbook)

HTMLBook =

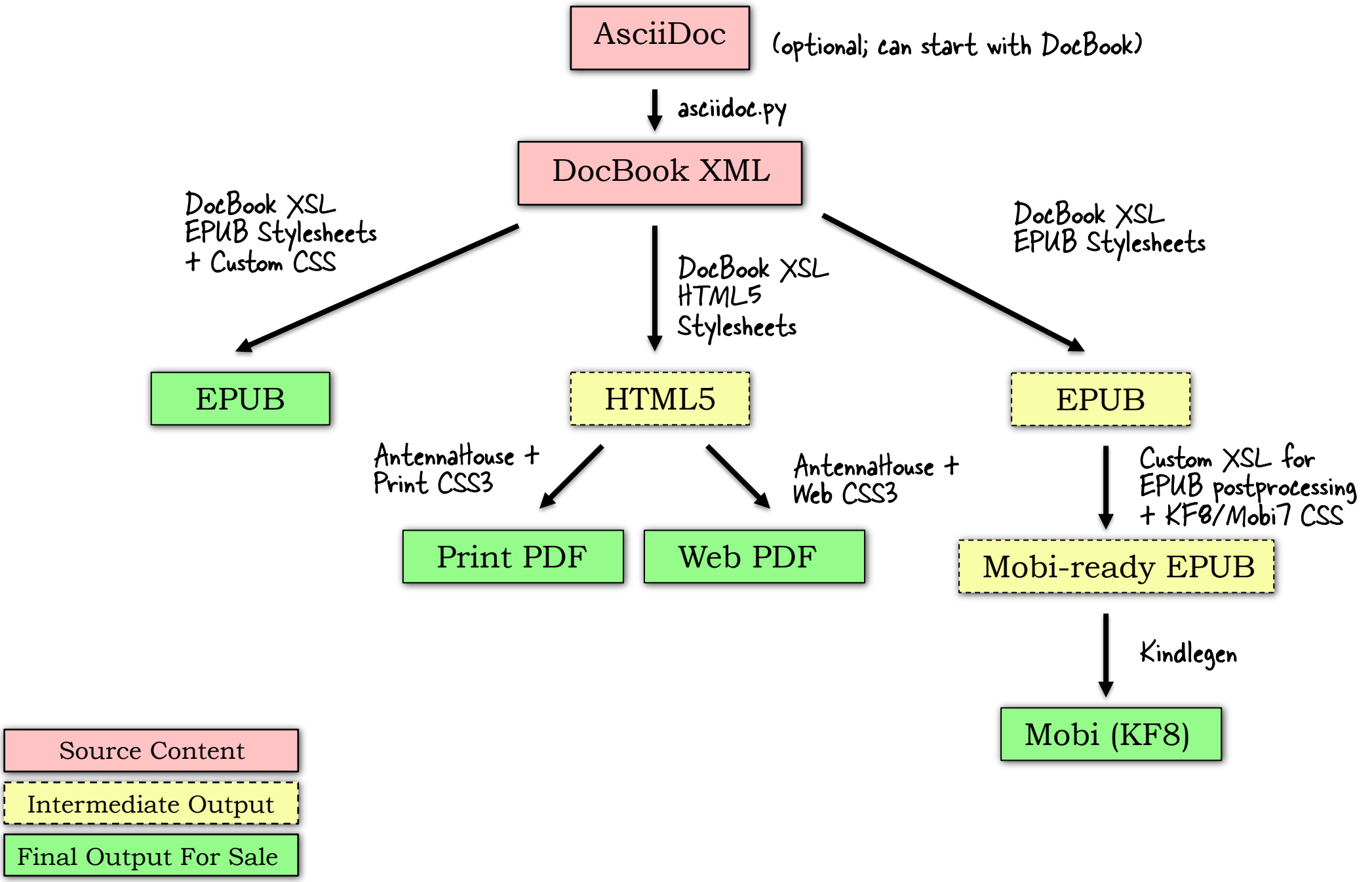
- **Open Spec for Book Authoring**
- **Subsets XHTML5 Vocabulary and Content Model**
- **Adds Book-Specific Semantics (e.g., `<section data-type="chapter”>`)**
- **Open Source Tooling for Producing Ebook Outputs**

HTMLBook Sample

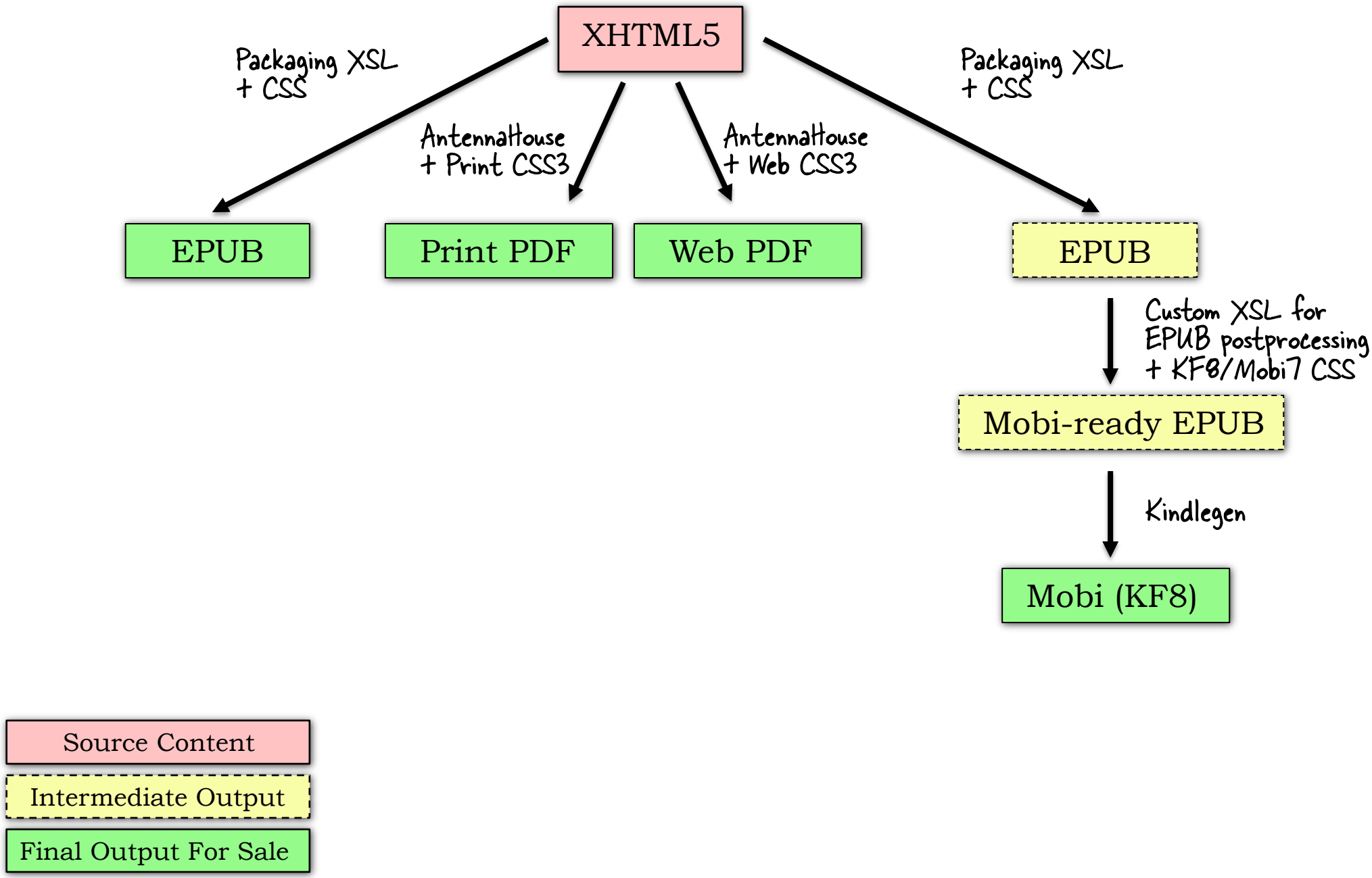
```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>HTMLBook Sample</title>
  </head>
  <body data-type="book" id="htmlbook">
      <section data-type="chapter" id="chapter01">
          <h1>Chapter 1. HTMLBook Markup</h1>
           <p>This chapter describes and demonstrates the
types of markup<a data-type="indexterm"  data-
primary="markup" data-secondary="types of"></a> that might
appear in a chapter. See <em>mappings.asciidoc</em> for
more information. HTMLBook borrows much of its
semantics from the EPUB 3 specification, as applied via the
<a href="http://idpf.org/accessibility/guidelines/content/
semantics/epub-type.php"><code>epub:type</code></a>
attribute.</p>
      </section>
   </body>
</html>
```

(github.com/oreillymedia/HTMLBook/blob/master/samples/htmlbook.html

O'Reilly's Single-Source Workflow (2006-2013):



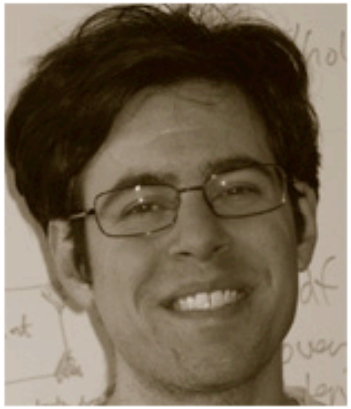
O'Reilly's Single-Source Workflow (2014!):



Contact Me!

Email: sanders@oreilly.com

Twitter: @sandersk



Sanders Kleinfeld

sanders@oreilly.com

http://twitter.com/sandersk

Cambridge, Massachusetts

Ebook Alchemist

Areas of Expertise:

- DocBook XML
- XSLT
- XPath
- XQuery
- RDF
- SPARQL
- EPUB
- Mobi
- HTML5
- speaking
- writing

Biography | Books | Blog | Multimedia

Sanders Kleinfeld has been employed at O'Reilly Media since 2004 and has held a variety of positions, including roles on O'Reilly's Production, Editorial, and Tools teams. Currently, he works as a Publishing Technologies Specialist, maintaining O'Reilly's XML-based toolchain for generating EPUB and Mobi formats of both frontlist and backlist titles. He also helps coordinate O'Reilly's digital distribution efforts to electronic sales channels, and is currently assisting in R&D efforts surrounding HTML5

Hire Sanders Kleinfeld

For Inquiries Contact
talent@oreilly.com

Press Inquiries
press@oreilly.com

Find Other Authors
Consultants

Also find Sanders on:

Recent Twitter posts:

- @nbousquet Most EPUBs out now use XHTML 1.1, so EPUB is already pretty Web-compliant. Maybe HTML5 ebooks will be more attractive to pirates?
  2 days ago

Author Events

Webcast: HTML5 for Publishers