Beautiful Development in One Window

Coding with Coda

O'REILLY®

Eric J. Gruber

Coding with Coda

Like to build websites in the wild with your MacBook? This concise hands-on guide introduces you to the ideal editor: Coda 2. Rather than clutter your screen with shell access, a separate CSS editor, and a version control app, you'll discover how Coda's "one-window web development" bundles everything into one neat application. Take Coda on a trial run, then learn step-by-step how to configure each feature to fit your working style. You'll find out firsthand how Coda will save you time and effort on your next project.

- Get to know Coda's workflow by building a sample site
- Delve into features such as the tab bar, path bar, sidebar, and Sites view
- Set up your own development environment—and dig deeper into the editor's options
- Get tips for taking full advantage of the text and MySQL editors
- Create a Git or Subversion repository for source control management
- Learn the finer points of sharing project documents across a network
- Discover the built-in reference books, and learn how to extend Coda

Purchase the ebook edition of this O'Reilly title at oreilly.com and get free updates for the life of the edition. Our ebooks are optimized for several electronic formats, including PDF, EPUB, Mobi, and DAISY—all DRM-free.

Twitter: @oreillymedia
facebook.com/oreilly

US $12.99 CAN $13.99
ISBN: 978-1-449-35609-5

51299

9 781449 356095

O'REILLY®
oreilly.com

Coding with Coda

Eric J. Gruber

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Coding with Coda

by Eric J. Gruber

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editors: Simon St. Laurent and Meghan Blanchette
Production Editor: Rachel Steely

Technical Editors: Wade Cosgrove and David Eldridge
Proofreader: Rachel Steely
Cover Designer: Karen Montgomery
Interior Designer: David Futato
Illustrator: Robert Romano

January 2013: First Edition

Revision History for the First Edition:

2013-01-24 First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449356095* for release details.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-35609-5

[LSI]

Table of Contents

Preface

One of the first questions asked when someone learns to code is, “What editor should I use?”

It is not a question easily answered. There are purists who believe you should only code with SimpleText or TextWrangler, but there are also those who prefer the many features of Dreamweaver. For those who also need to use the command line, the debate about vi and Emacs rages on. In addition, robust GUI editors such as TextMate and Sublime Text have taken the development world by storm.

Your personal preferences, your workflow, your code, your designs, your wallet…it’s all about you, baby. And that is why the question isn’t easily answered.

As someone who considers himself a designer first with a splash of developer mixed in, I’ve found it’s good to play around with different types of tools to see what you like best. I like Dreamweaver for some things and Sublime Text for others. I’ve even been known to dabble in a plain-text editor from time to time.

Ultimately, my primary tool is Panic’s Coda.

Panic has earned a reputation for making fantastic software that works great. I first heard about Panic when I was shopping around for an FTP client, and purchased Transmit. That became the gateway to Coda, and I’ve been a fan ever since.

What I like most about Coda is its focus on one-window web development. You don’t need to have an editor window open with Terminal off to the side, a separate CSS editor, and another application or Terminal tab for version control. Coda has everything for you—it’s a Swiss Army Knife of web development bundled into one beautiful package.

If you've been to a conference or coffee shop lately, you've likely seen the proliferation of coders out in the wild with their favorite laptops. Coda's environment makes excellent use of a laptop's reduced screen real estate. No more tweaking and adjusting every little window just right so that you can get in your groove. Just fire up Coda and start coding.

There is no perfect editor, but there are some pretty great ones. For me, Coda is one of the best. Perhaps most importantly, it gives me an answer to the question, "Which editor should I use?"

Who This Book Is For

Coding with Coda is for designers and developers with some experience under their belt. If you're new to web development, this book isn't for you.

It's preferable for you to have some experience with the command line, but if not, that's OK. In the same vein, experience with CSS is also welcome. If not, don't worry about it. The book will show you how to use the tools, and you can use them when you're ready.

Coda is a Mac-only editor made by Panic, a company that makes Mac- and iOS-only applications. More importantly, this book is for Macintosh users only. Coda isn't free; as of this writing, the app is on sale for $75, normally $99. That said, the download offers a demo so you can give it a solid spin around the block before making a purchasing decision.

What This Book Will Do for You

Coding with Coda is organized in a start-to-finish fashion. To begin, you'll run through the basics of Coda 2 and then get started with the initial phases of building a project, or *Site*.

A chapter is spent on the source control management, how it integrates with Coda, and how you can use the two in your development process. You'll learn to tinker to your heart's delight with the many available preferences, while also getting into the more intricate aspects of using Coda for editing and document collaboration.

You'll wrap up with an overview of constructing a page (no web app building in this book!) and then learn about some extra features to make Coda even more enjoyable to use.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`

Shows commands or other text that should be typed literally by the user.

`Constant width italic`

Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, if this book includes code examples, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Coding with Coda* by Eric J. Gruber (O'Reilly). Copyright 2013 Eric J. Gruber, 978-1-449-35609-5."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://oreil.ly/Coding_Coda*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

Thanks to the community of designers and developers in the Lawrence, Kansas, area who have helped me grow as a developer.

Thank you to tech reviewers David “my dad always told me to learn your keyboard shortcuts” Eldridge, and Wade Cosgrove from Panic for your fantastic work. Meghan Blanchette, your gentle editor’s touch is fantastic; thank you. Thanks to Brandon Edling and Ben Henick for the encouragement and to Simon St.Laurent for taking a chance on me.

Thank you to my lovely wife for putting up with me and my parents and sister for always encouraging me to find my own way.

And to my daughters: Ember, your reading ability at such a young age makes me work hard to write better. And Remi, when you would sit on my lap and demand I hover over “The leaf! The leaf!” of Coda’s beautiful green icon in my dock to magnify it, I knew I was on the right path.

CHAPTER 1

Getting Started

Since the journey of 1,000 lines of code begins with the first keystroke, it's probably best to learn a little bit about the award-winning company responsible for the inspiration of this book.

About Panic

Coda was created by Panic, a small company that hails from Portland, Oregon. Its founders are Cabel Sasser and Steven Frank, who started the company in 1997.

Panic makes a handful of apps in addition to Coda, all of which run exclusively on the Mac:

Transmit
: A top-notch file transfer application that works with FTP, SFTP, and Amazon's S3 service

Unison
: A client for Usenet, the oldest internet messaging board

The company also has gotten into developing for iOS in the last few years:

Prompt
: An SSH client

Diet Coda
: A stripped down but powerful little brother to Coda 2

There are also "retired apps," which include Stattoo, a kind of dashboard on the desktop, Desktastic, a fun little app for drawing on your desktop, and Candy Bar, a utility for changing and organizing icons across your system. In addition, there is Transmit 1.7 for Mac OS 9 and Audion 3, one of the first MP3 players. For a cool story about Audion and its history with iTunes, read *The True Story of Audion*. You can download older versions of Panic's apps here.

Installation

A fully-functional, seven-day trial version of Coda is available from Panic's website. To clarify, if you launched Coda on a Monday, quit the app, then launched it on Wednesday, that would only count as two days of the trial.

The download instructions are pretty straightforward if you're familiar with a Mac, but we'll review them here for good measure.Visit the Panic website and go to the Coda page. From here, you have two options: purchase it through Apple's Mac App Store, or purchase the app directly from Panic.

Mac App Store or Buy Direct?

As of Mac OS X v10.6.6, Apple introduced the Mac App Store for developers to sell and host their applications in a one stop shop for consumers. Much like a record label or book publisher, the store provides a way for consumers to easily find new apps they might not otherwise have discovered on their own. For this, Apple takes a 30 percent cut of each sale.

Some developers choose to sell their apps only through the Mac App Store, while others, like Panic, also offer the apps for purchase on its website.

So should you purchase from the Mac App Store or buy directly from the developer?

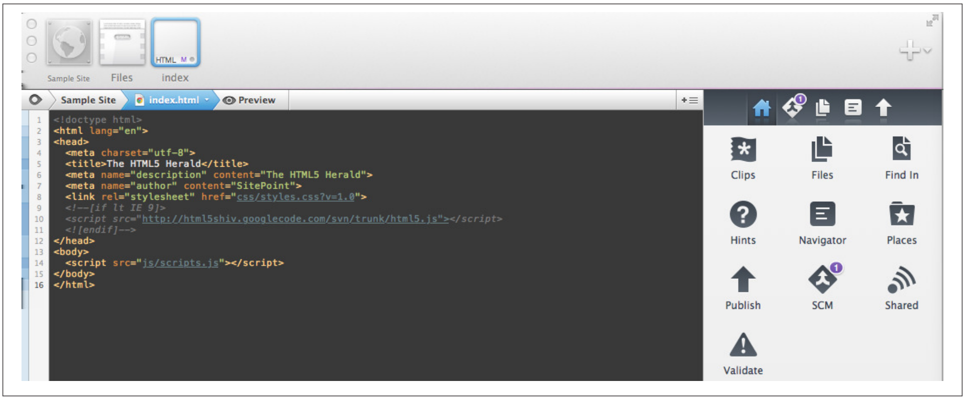
For the consumer, the store is an easy way to manage your applications and their updates. The Mac App Store will alert you when any of your apps has an update, and you can choose to update them all at once or individually. Also, iCloud Site data syncing across computers is a feature only available to apps sold through the App Store.

But, there's a catch—Apple has its requirements for what is "approved" for sale through the store, and since apps must go through Apple's review process, it might take additional time before you get an update to an app that's already been updated through the developer's website. Perhaps more importantly, if you care about rewarding developers for their hard work, purchasing directly through them gives them a greater cut of the purchase.

Panic obviously doesn't mind if you choose to go the Mac App Store route since it provides its apps through there, as well. My preference is to buy directly, but ultimately, the choice is yours.

After downloading Coda, simply drag the Coda 2 application to your Applications folder to install it. To start Coda, just navigate to the Applications folder and double-click on the Coda 2 icon (but don't do this yet).

Now that we've got Coda installed, let's get familiar with its interface (Figure 1-1).



```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>The HTML5 Herald</title>
  <meta name="description" content="The HTML5 Herald">
  <meta name="author" content="SitePoint">
  <link rel="stylesheet" href="css/styles.css?v=1.0">
  <!--[if lt IE 9]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
  <script src="js/scripts.js"></script>
</body>
</html>
```

Figure 1-1. An overview of Coda's development window

A Quick Tour

Before we get too deep into this, we'll take a look around some of the basics of Coda so we're on the same page with our terminology.

Across the top of the app is the *tab bar*. This lets the coder know which tabs are open, and a blue border highlights which tab is currently active. Tabs can include a document, a terminal shell, a book (Coda ships with built-in references for many languages), or a MySQL editor.

Tabs can be easily closed by hovering over them until an X appears and is clicked. Also, the tab bar can be resized by dragging the bottom of the tab bar up. Drag up all the way, and the tabs disappear, leaving only a text reference of your open tabs.

The first two tabs will always be the Site you are working from and the file browser tab, respectively.

Just underneath the tab bar lies the *path bar*, which shows the drilled-down path to the file you are editing. If you're working on an *index.html* file in a directory called *awesomesauce* in your Mac user's Sites folder, the path to that file would be *[Username]/Sites/awesomesauce/index.html*. The path bar serves as a visual reminder of where you are in relation to your project.

But more than that, if you click on any of the individual path items (such as “awesomesauce”) at the moment, a box will pop up and list all the files in a miniature file browser. This makes for a very easy way to open files from all sorts of locations within your project while taking advantage of the editing window.

The *sidebar* is a bit of a Swiss Army knife of different views and actions for your projects. For example, there’s a *files* view that has a file browser available, then you can switch to the Source Control Management view for managing your code in a version control system, such as Git. We’ll get into the different types of views as we go along.

By default the sidebar sits to the right, but can be changed in Coda’s preferences. And the top of the sidebar is a dock, where you can drag and drop your most-used views for easy access while you work.

One of the final views to discuss is *Sites*, the core of your project, which we’ll go deeper into in the next chapter.

CHAPTER 2

Creating a Site

While you could certainly open any single file and edit it using Coda as an editor, bigger projects require us to be wise and build upon a strong foundation. Coda enforces that premise with *Sites*.

A Site could have been called practically anything: pocket, container, jar, etc., but since Coda is typically geared toward the web designer and developer, Site seems as good a choice as any.

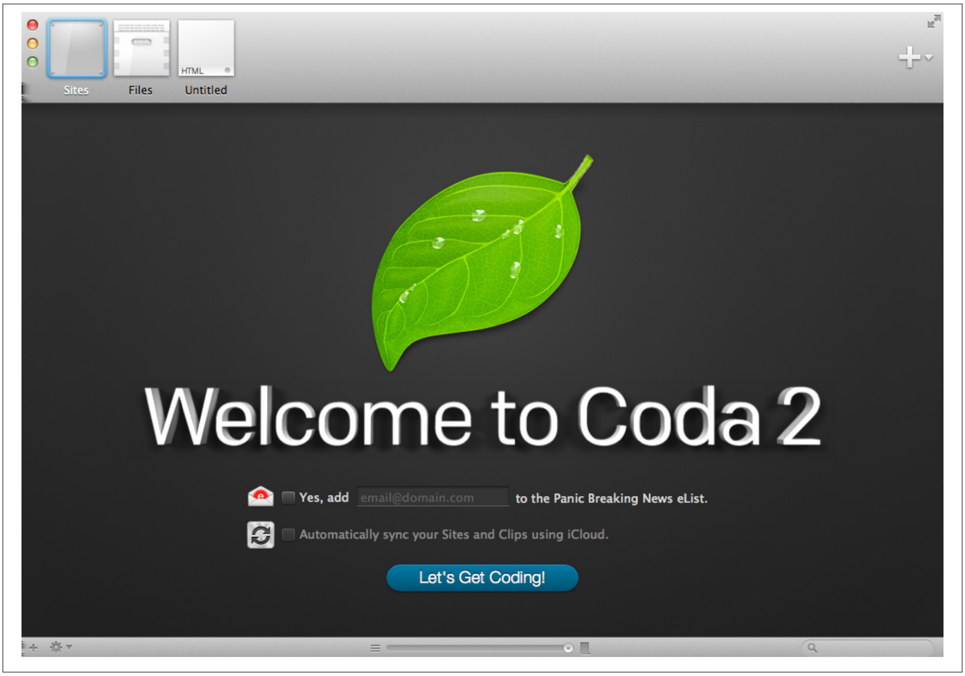
So what is a Site? It is simply a container for the project you're working on that has related files and all the pertinent information that goes along to build and deploy it. A typical website will have HTML, CSS, and JavaScript files, as well as directories with things like images and graphic art. A more dynamic website will have files with additional functionality, such as a PHP script or page, and even connectivity to a database like MySQL.

And so, a Site is a placeholder that quickly allows you to access all this information so you can get working on your project from Coda's launch screen. We'll set up our first Site now and see how all of that works together.

First Run

Let's go ahead and start the application by navigating to the Applications folder and double-clicking the Coda 2 icon (Figure 2-1).

> I prefer to use Mac OS X's Spotlight app by typing Command and then the spacebar, then typing Coda 2. Spotlight will bring up the application, and then by pressing the Enter key, it will launch it. You should find what works best for your workflow, but since I love keyboard shortcuts, I thought it was worth a mention.



□ Yes, add email@domain.com to the Panic Breaking News eList.

□ Automatically sync your Sites and Clips using iCloud.

Figure 2-1. The opening screen for your inaugural launch of Coda

On first run, you'll be asked if you want to join the Panic Breaking News eList. Of course you do! Be sure to sign up for the email list if you want to stay informed about product information and updates. The folks at Panic won't fill up your inbox.

If you purchased Coda through the Mac App Store, you're also given the option of automatically syncing your Sites and Clips using iCloud. This handy feature serves as a backup of your files and Clips (which we'll get into later). If you purchased Coda directly from Panic, don't fret: there are still options to have your important files backed up outside of Apple's ecosystem for little or no cost. For the purpose of this book, we'll proceed without using iCloud as our backup system.

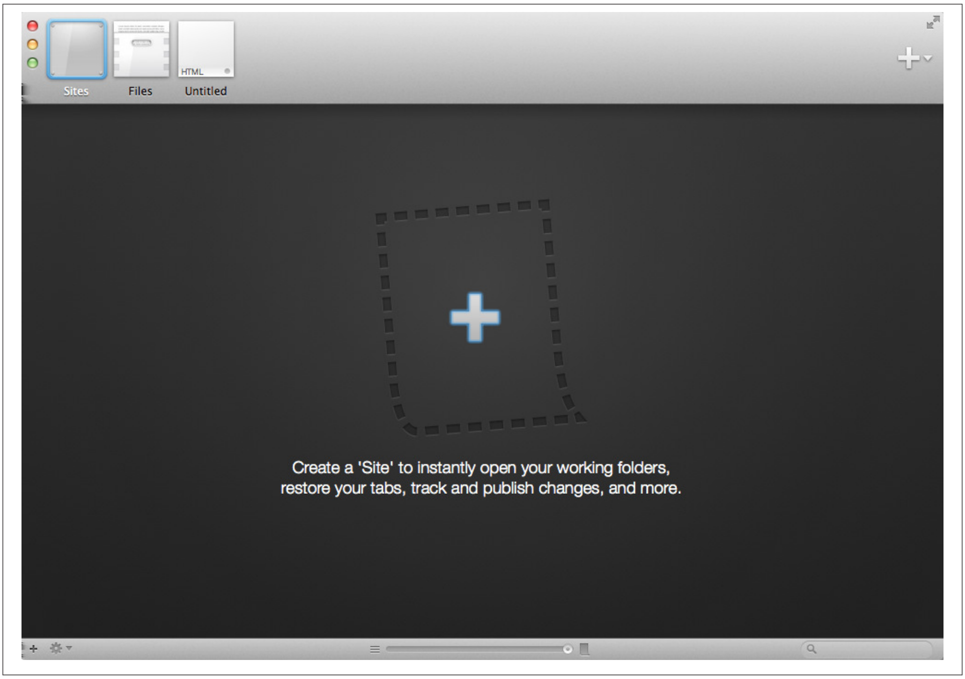
Should you choose to skip this now but later decide you want to take advantage of iCloud sync, go to Coda → Preferences → General and select "Sync Sites and Clips" in Coda's preferences.

For iCloud sync to work, you must have Documents & Data selected in iCloud preferences. Find iCloud's preference pane by going to the Apple Icon → System Preferences → iCloud and then checking the box next to Documents & Data.

When you're ready to proceed, click the Let's Get Coding! button and we'll set up a Site.

The Setup

Now we're ready to create our first Site. Clicking the plus sign in the center of the app (or the plus sign in the lower lefthand corner after this first time) brings up a dialog box for moving forward (Figure 2-2).



Create a 'Site' to instantly open your working folders, restore your tabs, track and publish changes, and more.

Figure 2-2. It's practically begging you to get started!

In the dialog box, there are several fields you can edit to start building your Site.

Nickname
: The name of your project. This is rather informal; just keep it something you'll easily remember. An example might include Blog or Joe's Car Wash.

Protocol
: This is how you'll connect to a server so you can upload your files when you're ready to deploy. While FTP and SFTP are the most well used, there are also options for FTP with Implicit SSL, FTP with TLS/SSL, WebDAV, WebDAV HTTPS, and Amazon S3. You'll need to know what your needs are for your server, but we'll mostly talk about FTP and SFTP here.

Server

The name of the server you're connecting to, which might be already set up with a domain name. You could also use an IP address instead. It also includes the options of specifying which port to use; typically websites use port 21 on FTP for file uploads and port 80 for reading files.

Username

Every web server requires credentials to gain access (provided by you webhost); put yours here.

Password

Enter your password for the corresponding user name. Checking the "Ask each time" selection adds an extra layer of security. Any time you try and connect to the server, you'll still be prompted to enter your password. If you feel as though you have a reasonable expectation of security on your machine, you could probably keep this unchecked.

Root URL

This is the website domain. Coda will create a thumbnail of the website for future reference. Also, Coda will use this URL to grab images off the web server for previewing (yes, you can preview images in Coda, too).

Local URL

Fairly self-explanatory, this is the base URL for previewing local pages and images. Locally, this is your Site root.

Remote Root

When connecting to your remote server, this is the directory you'll log into and whose contents you will see. This path will be the base used for generating previews.

Local Root

The same as your remote root connection, but only on your local machine. By clicking the Set button, you can browse your system for the place you want your Site's files to reside. In subsequent uses, Coda restores the last used location when a Site is reopened.

In Another Cloud

If you're looking for an alternative to iCloud sync for your files, using a service like Dropbox might be what you're looking for. Dropbox establishes a folder on your machine that syncs with its servers on regular intervals, and includes sharing, viewing deleted files, and opportunities for getting additional server space at little financial cost. Should you use something like Dropbox for backing up files, you can use the Local Root setting mentioned earlier and choose a Dropbox folder, giving you much of the same cloud-based backups that iCloud provides.

Terminal Settings

Coda lets you connect locally, or via SSH to your server, just like using Mac OS X's built-in Terminal application. Clicking on the Terminal icon in the dialog box brings up fields so you can edit your terminal settings.

SSH Server
: To connect to your remote server via the shell, you'll need the address. It's usually the address of your website. Typically, you connect via port 22, but if not, make the necessary adjustments.

Username
: The username for your SSH server.

Password
: The password for the corresponding username of your SSH server.

When you open a Site, the Terminal information for the Site will populate using the information you put in these settings. All you'll need to do is go to File → New Tab → New Terminal Tab and then connect with either your local or remote connection (with SSH) by pressing the Connect button.

> You can also connect to a new Terminal tab by clicking the plus button in the upper righthand of the Coda tab bar.

There are two other icons in the dialog box we have left to discuss: Source and Database. We'll discuss the Database preferences later in "Terminal Preferences" (page 23), but now we'll move on and discuss Source Control Management.

CHAPTER 3

Source Control Management

My personal journey of how I came to source control management, also known as *version control*, is likely a familiar and tragic story to many.

Many of my attempts at controlling revisions to my code in my early years of learning web development went a little like this:

1. Zip up a directory that contained all the code I wanted to save.
2. Name the file something like *backup10102012.zip*.
3. Fool myself into thinking that was all I needed if things went wrong.
4. Back up that zip file to an external drive.

Just four steps. Rock solid, right?

You probably know how this story ends. At some point, I needed to revert back to a previously saved version of a site I was working on and my four-step zipping process failed me. Looking at it now, it seems obvious:

- There were no notes as to what "version" of the zip file was different from the other, similar-looking zip files.
- If I did have documentation, it was likely zipped up and I wouldn't see it without unzipping the directory.
- Incremental changes received less-than-useful names like *backup10102012-a.zip* and *backup10102012-b.zip*.
- The ability to collaborate with others was virtually nonexistent.

Fortunately, there is a better way to manage code.

Like Wikipedia for Code

In case you haven't been introduced, source control management software can be described as "Wikipedia for your code." When you edit a page in Wikipedia, you and others can see the revisions, roll back to previous edits, make comments, and track the history of changes to a given page over time.

Likewise, modern SCM software acts similarly. As you make changes, you can "commit" your code, make a comment about that commit, and keep going. If you need to share your code, others can see the before and after changes in your code, and (provided you gave good, descriptive comments), can also see the "what" and "why" of your edits.

Another great feature of SCM is *branching*. When you want to work on a new feature—say, creating a new footer—then you create a branch. Like moving from the trunk of a tree to a branch as you climb it (or even branch to branch), you can switch to a new branch while coding and work on new features while safely keeping your original code intact. If it works fine, you can *merge* the code into the master code. If not, keep making changes, discard it, or make a new branch and work on other code.

It's much more sane than obscurely-named zip files.

Coda and Git

There are several types of source control management software available for developers, including Subversion, Mercurial, and CVS. But one of the hottest trends of late in source control management is the adoption of the version control system, *Git*.

Git was created by Linus Torvalds, who you might know as also being the driving force behind the Linux kernel. Git is a *distributed revision control system*, meaning that each user has his own copy of the code, acting as its own separate code repository, rather than everyone pushing and pulling from a centralized repository. Then, when the changes are pushed back together, Git is smart enough to mesh everything as it should be (most of the time) without many major hiccups.

To get started with Git, you need to have it installed on your machine. Download Git at its official website and follow the download instructions.

Git has become very popular, with much of the credit going to the social coding site Github, an online code repository hosting service. It has quickly become the *de facto* way to manage code within the development community, so it should come as no surprise that Panic built Coda 2 with support for Git.

Git and the SCM Sidebar

For Coda's built-in SCM features to work, we must first have an established Site open. We'll create our sample Site now and begin the process of going through a Coda workflow. Open Coda, and in the lower left corner of the window, click the plus button, or click the gear icon and select Add Site from the pop-up menu. This will create a new Site (Figure 3-1).

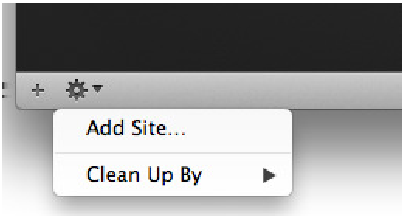


Figure 3-1. Getting a new Site started

In the dialog box that opens, give your Site the nickname of "Sample Site." Also set your Local Root to a directory on your computer where you'll save your Site's files. For simplicity's sake, create a new directory and call it Sample Site, then select it for use.

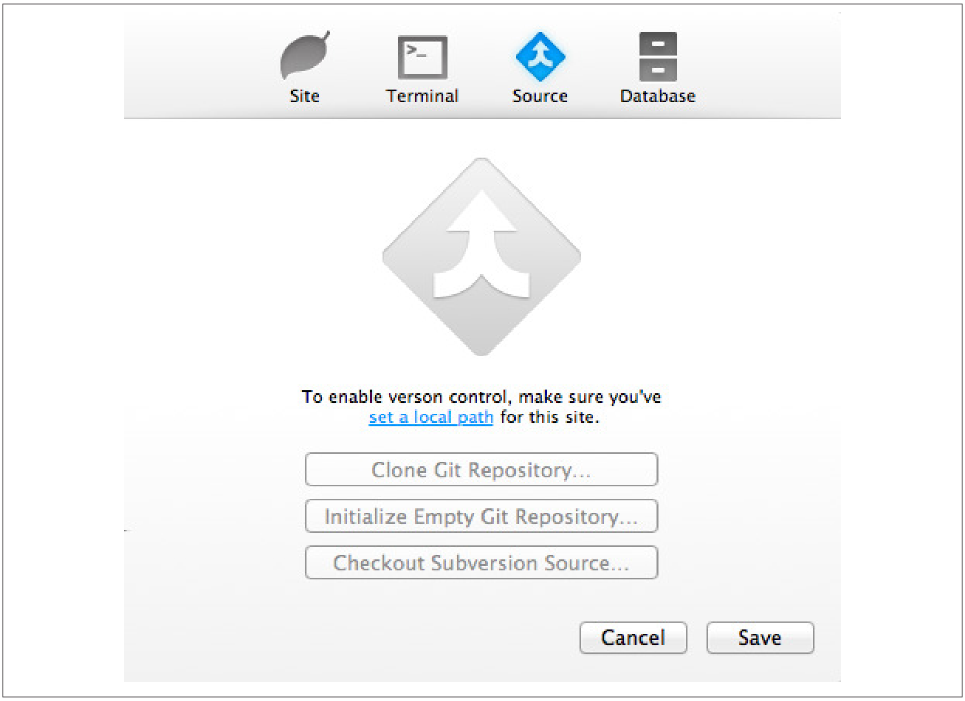
Next, click the Source preference tab across the top of the new Site dialog box. We'll need to make a new Git repository to get started, so click the middle button to initialize an empty Git repository (Figure 3-2).

In the next window that opens, make sure the slider is set to "on," and select Save. The window will close, your Site will be stored with the name you gave it, and now you can double-click your way to working on your new Site.

Open your Site, and instantly you are given a blank canvas to start with called *Untitled.html*. Instead, let's save the file using Command-S or going to File → Save (or Save as…, depending on what you're doing) and rename the file *index.html*. Make sure the *index.html* file is saved in the Sample Site directory you made when you created your Site.

Now we have a new alert in the sidebar. The SCM icon is showing an indicator of how many files have been changed. Clicking on the SCM icon brings us to a GUI for interacting with Git and committing our changes (Figure 3-3).

In the sidebar, next to the *index.html* filename, is a button to Add the file to the repository. Click it, and it will add the file to the repository. Next, click the new Commit button and give a description of what your change was in the box that appears. For this first commit, type (without quotes) "Initial commit." Then press the Commit button (Figure 3-4).



To enable verson control, make sure you've set a local path for this site.

Site
Terminal
Source
Database
Clone Git Repository...
Initialize Empty Git Repository...
Checkout Subversion Source...
Cancel
Save

Figure 3-2. Setting up your Git repository in a Site

And there you have it—you created a Git repository, added a file to it, and gave a little message to detail your change. Your Site is now under source control management.

Unfortunately, this book is not a comprehensive look at all Git has to offer. Rather, take a look at the O'Reilly book *Version Control with Git* for a step-by-step walkthrough of this powerful source control management system.

When you're more comfortable using Git, you'll find that Coda has many features built in to assist with incorporating it into your workflow. At the bottom of the SCM sidebar, there are several options available for using Git at an advanced level. From here you can refresh the sidebar, pull and push changes, create new branches, switch branches, merge branches, and check out revisions (Figure 3-5).

Source control management is a powerful tool; using Git within Coda makes it easy to stick to your one window for development as you keep making changes to your Site in a very manageable way.

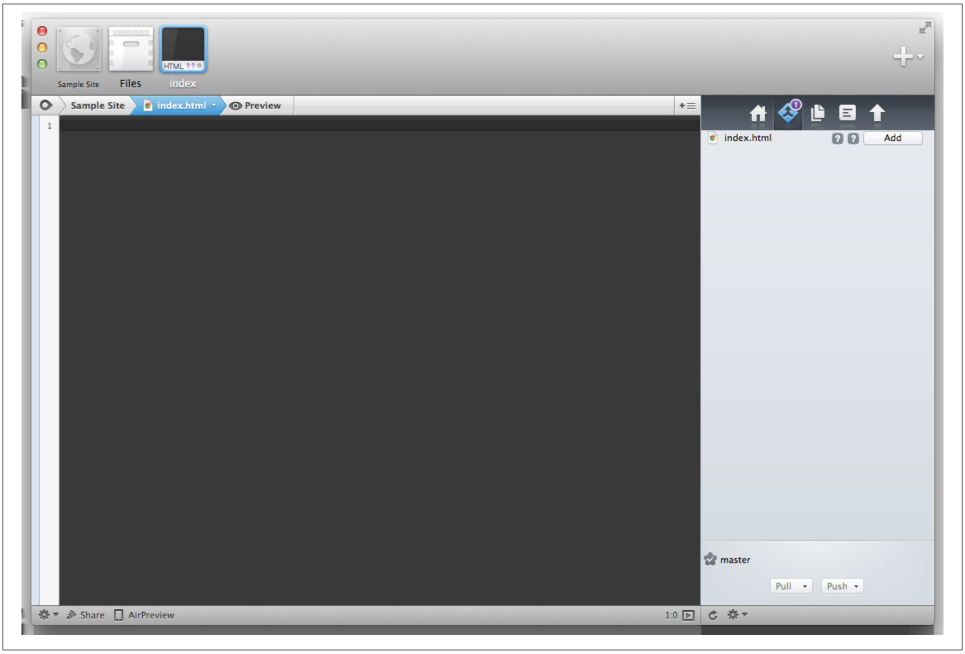


Figure 3-3. An alert in the SCM sidebar lets us know changes can be added to Git

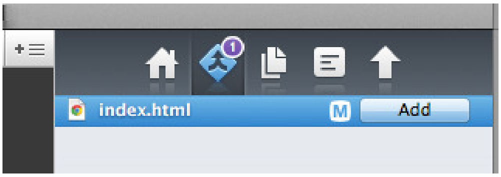


Figure 3-4. The SCM sidebar notification shows that a change has been made

About Subversion

Subversion is another open source system for managing your files with source control.

Subversion is different than Git in that it is a centralized version control system. Rather than Git's distributed system, a Subversion repository is typically on a central server, and you pull and push code revisions back to that server.

When we created our Site and went to the Source preference pane, there was an option to check out a Subversion source. This is how it integrates with Coda and into your local Site.

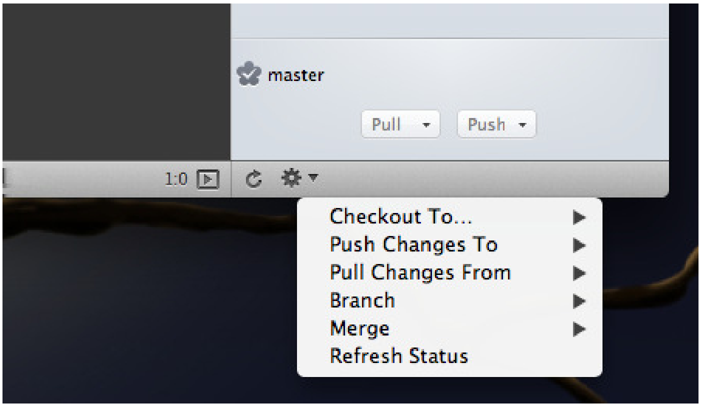


Figure 3-5. More verison control options are at the bottom of the SCM sidebar

Instructions for configuring and using source control are outside the scope of this book. If you're interested in learning more about Subversion, check out *Version Control with Subversion* from O'Reilly.

CHAPTER 4

Tinkering with Preferences

When I was but a teenager, a neighbor of mine once bought a brand-new Harley-Davidson motorcycle and had it torn apart, sitting on stands in his front yard two days later. My dad and I walked over to see what he was doing, as we were completely shocked by what he had done.

"What's wrong with your bike?" my dad asked him.

"Nothing," he said. "I just wanted to tinker a little bit with it."

It's a good thing Coda doesn't cost as much as a Harley-Davidson motorcycle, because one of the first things I want to do with any new program is to "tinker a little bit with it" and set it up to my preferences.

Selecting Coda → Preferences from the menu brings up a window where you can tinker to your heart's content. There are nine tabs to choose from, so let's set up Coda the way we like it.

General Preferences

Split Direction

When editing files that can be previewed within Coda, such as an HTML file, you can split the viewing area. For example, your code could be in one split pane, and a preview of what you're editing could be in another. This preference sets the default for whether the split is horizontal or vertical.

An added bonus is that you can quickly switch to the opposite of your default setting by holding down the Option key when splitting a direction in editor, followed by clicking the Split button in the path bar.

Sidebar Position

This sets the default positioning of the sidebar in the editing area.

iCloud

This option is only available if you purchased Coda through the Mac App Store. If you did, you can sync your Sites and Clips (reusable snippets of code) through Apple's iCloud service.

AppleScript

AppleScript fan? Click this option to show the Scripts icon in the menu bar.

MySQL rows per page

Coda 2 has a built-in MySQL editor. This sets how many rows are visible at a time.

Updates

For versions of Coda purchased outside of the Mac App Store, you can choose to automatically download and install updates, or just notify if there are updates. You can also not check for updates at all, although that doesn't seem like a very good idea.

Editor Preferences

Default File Type

Select a file type (the default is HTML) from the dropdown. When a new file is created, it will use the type you selected here.

Default File Encoding

Don't you just love it when you paste something from a word processing document only to have it to look all goofy when you load the page in a browser? We've all seen something like "â€" before. You can blame file encoding for that, causing things like your quotation marks and apostrophes to end up as "confused characters".

Let's take a file with ASCII text encoding. To display a letter in the English (Latin) alphabet, it might use a certain set of instructions that, when opened in a different text encoding, displays differently. This can occur when dealing with files across national boundaries, and if you're ever in the position where you work with someone from another country, you should be aware of it. It would be a shame if all your A's ended up looking like å or something.

You can select a default file encoding from this selection or customize the encodings list to have only what you prefer.

Default Line Endings

At some point, we'll no doubt end up exchanging files with another developer who might be using a different system than a Mac. It's a cruel world, but we have to live in it. Because of this, we'll need to know some subtle nuances of other operating systems and how those systems use different ways for showing ends of a line in text files.

Choose from one of the three common line endings available for their respective operating systems:

- Mac OS X/Unix like systems, which includes Linux, uses LF or line feed
- Classic Mac OS (pre Mac OS X) uses CR or *carriage return*
- Windows uses CR + LF or *carriage return* plus *line feed* endings

Other Options

Below the Default Line Endings selection are some additional options for your editor preferences:

- *Suggest completions while typing.* This will attempt to complete your code for common design and development functions.
- *Automatically insert closing brackets.* For example, this will give you a final "{" after you type the initial "{" (and so on).
- *Show invisible characters.* An option that gives symbolic nomenclature for spaces, carriages returns, etc.
- *Select prefix when double clicking a variable.* For example, in PHP, when you select the $ sign at the beginning of a variable rather than selecting a variable from $variable, you can pick the prefix associated with it.
- *Indent new lines.* When checked, if you have spaces at the beginning of a line, those spaces are repeated when you move to a new line by hitting return. If left unchecked, you'll just go to the beginning of the next line without the spaces.
- *Typing delimiters wraps selected text.* Selected, you can type the world "hello," highlight it, and then press Shift followed by the < key. The result will produce <hello>.
- *Wrap lines.* Causes the text to flow within the viewable window area provided. Otherwise, you'd be doing a lot of right scrolling.
- *Indent wrapped lines by (number of) spaces.* This is a visual tool so that if you have a line that wraps past the end without going to a new line, Coda will indent to give a visual effect that it is wrapped by the number of spaces specified.

Tab Width

Set how far the cursor moves when you press the Tab button by setting up its default behavior. Rubyists like to use two spaces, so this is where you'd set your default to only two and select "Use spaces instead of tabs."

Editor Font

Change the fonts for the documents you work with. Click the Select button to find a new font and font size from the Font panel.

Custom Syntax Modes

In most cases, Coda will automatically match the correct syntax mode with the file extension, such as HTML for *.html* files or Ruby for *.rb* ones. However, it is possible to create a new file with no extension right off the bat, so be mindful of how to change it.

Coda is pretty smart, but there might be times where it doesn't recognize a file extension and therefore doesn't know which syntax mode to assign to the file. When this happens, you'll need to do that here. Click the plus button at the bottom of the Custom Syntax Mode window. Then create the extension name and assign a syntax mode from the list of available options.

Additional modes can be downloaded as plug-ins from Panic's website. After downloading a third-party plug-in, follow the instructions included. You can manually copy the *.mode* file using a Terminal window into *~/Library/Application\ Support/Coda\ 2/Modes/*. Using the Finder, navigate to your user account's Home folder, and then drop the file into Library → Application Support → Coda 2 → Modes.

Starting with Mac OS 10.7, the Home folder's Library is hidden. To make it viewable in the finder, hold the Option key and select *Go* from the Finder menu. This will temporarily allow for easy access to the Library. For the daring, you can also open a Terminal window and type **`chflags nohidden ~/Library/`** to make it appear, and **`chflags hidden ~/Library/`** to hide it again if you wish.

Sharing Preferences

Coda includes a collaborative editing mode that lets multiple people simultaneously edit the same file in real time over a network. We cover this more extensively in Chapter 5.

The info in the Sharing tab includes the personal information you'll display to a collaborator, as well as the color for your edits.

Colors Preferences

Stylesheet

Coda ships with four stylesheets. Birds of Paradise and Specials Board have darker backgrounds, while Coda Bright and Coda Classic are lighter. These act as a kind of *base* for customizing the colors of the text editor. Other customizations include highlighting the current line, syntax coloring, and the ability to change the document background, document text, line highlight, and invisible character colors.

Also included are text and background color options, as well as bold, italic, underline, and strikethrough settings.

Importing and Exporting

You won't be limited to these options, however. To the right of the stylesheet dropdown is a setting menu for importing and exporting the *.sss* file associated with the color scheme chosen. The *.sss* file extension is a Coda stylesheet, and you can use this file to create your own style or to back up your preferred colors.

Fortunately, if you really screw up, there's a Revert All button to help you get back to the style sheet defaults.

Files Preferences

View Options

Choose to use relative dates (today, yesterday, etc.) and to group folders together above files when viewing files.

Show Columns

Coda shows only the size and date of a file by default, but you also can pick kind, owner, group, and permissions visibility for both local and remote views.

Custom Editors

Use the plus, minus, and pencil editing buttons to edit the External Editors table with filename extensions and their respective applications. This makes it easy to right-click the file in the file browser and open it for quick editing in the proper application (like a *.png* image in Photoshop).

Subversion Tool Path, Git Tool Path

Default location of your source control management tool. This likely won't need changing unless you've got your machine set up different than a typical install.

Source control changes

Have Coda choose from the following while your code is under version control, particularily when working with a repository that others are also making changes to:

- *Prompt me to update my repository.* When changes have been pushed from a remote repository, this option will prompt you to decide whether you want to update your local version or not.
- *Automatically update my repository.* This option tells Coda to go ahead and allow changes made to a remote repository on your local machine without being prompted to do so.
- *Do not update my repository.* No changes will be made; your local repository will not be updated.

Transfers Preferences

When you're ready to deploy your website or application, Coda gives you four options to set as default for downloading and uploading files and folders if an item already exists:

- Ask me what to do
- Replace the existing file (or folder)
- Try to resume (or merge) the transfer
- Skip the file (or folder)

Bandwidth

You can limit uploads and downloads to a set number of kilobits per second, and set the maximum number of files for simultaneous transfer.

Options

Here you can use passive mode for data transfers, although according to Coda's Transfers tab, Sites "have their own passive setting which overrides this one." Still, it is possible to transfer files outside of a Site in Coda, using its FTP. If you need this option, it's available.

Also, you can choose to set an audible alert to let you know when a transfer has completed.

Lastly, there are ASCII File Extension options pre-loaded, or you can add (or delete) to your own preferences.

Rules Preferences

Coda's Rules preferences will give you the ability to really fine tune your workflow. It allows you to hide certain files (overriding operating system defaults). Conversely, it allows you to hide certain things contrary to OS defaults. Lastly, it allows you to filter certain files from uploading (e.g., PSDs, BMPs, and so on). With File Rules and Permissions, Coda is at your disposal to play nicely with your preferred workflow.

Terminal Preferences

Coding with a plain, white background just doesn't feel right. I prefer instead to set up my environment with a more colorful aesthetic.

Terminal Font
Click the Select button to change the font family, style and size.

Options
Click the checkboxes if you want each option to be "yes":

- *Smooth terminal font* (for type fanatics)
- *Use visual beep* (which will give a "flash" when there is a user error)

Text colors
Choose the foreground and background colors for your Terminal window.

Prompt Before Closing
If you close Coda while a process is running (unless you specify an exception), Coda will ask you if you want to close the tab/window.

Advanced Preferences

Working with a proxy server? Coda has advanced options for integrating it into your work environment. Choose from Proxy Server Type, Proxy Server name, Proxy Port, Proxy User Name, Proxy Password, and an option to "Try to keep idle connections alive." Advanced setting options are also available.

CHAPTER 5
The Finer Points

We've worked on centralizing our workflow in a Site, recognized the need for Source Control Management, and checked out our preferences. Now, time to dig deeper into the finer points of Coda's editing environment.

The Views

With Sample Site open in Coda, we can see three distinct columns (provided you haven't changed your preferences too much) in the Files tab: Local View, Remote View, and the Sidebar.

Local View

At the top we have the path bar. One thing to note is the loupe at the far left of the path bar. Clicking on it opens a search window where you can find a file by name if you know it. This isn't incredibly useful with the one file we have in our Site, but it will be when you have 200 files in a Site. Also, you can narrow your search to just a folder by clicking the arrow next to the magnifying glass in the search box and selecting Choose Folder.

Below the path bar is a file list. You can Control-click at the top where it lists Name, Size, and Date to bring up additional information display options.

Remote View

If you have configured your Site to connect to a remote server, this is where you'll find those files once you're connected, just like just using a typical FTP application. When you open your Site from Coda's launch screen, the remote view will automatically connect. You can disconnect from the remote connection by clicking the eject icon at the top-right side of the Remote View section.

If you don't have your Site ready to connect to a remote connection, or if you don't have a Site open at all, the remote view will have a panel ready for you to enter in your connection settings. Then you can go about your merry file-transferring way (Figure 5-1).

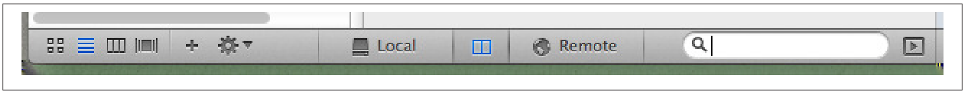


Figure 5-1. At the bottom of Coda's window is a bar that has other options for local and remote views

When a Site is open, a bar at the bottom of Coda's window provides other options for interacting with the files and folders of a Site, in addition to the Sidebar:

- Like the Mac's Finder, files can be viewed as icons, a list, in columns, or in Cover Flow. The Action button (shaped like a gear) gives other common Mac-centric operations, such as New Folder, New File, Move to Trash, Get Info…all the things you've likely used time and time again in Mac OS X.
- Next in the bar are the view options, which let you pick from Local View, both views open, or Remote view.
- A search box provides a quick way to search within wherever your location is in the view you are in. A quick visual glance at the path bar will clue you in as to what folder or system location you are in.
- A Show/Hide button collapses the Sidebar.

The Sidebar

As discussed in Chapter 2, the Sidebar manages different views and actions for your projects.

The sidebar options include: Clips, Files, Hints, Navigation, Places, Publish, SCM, Find In, Shared, and Validate.

Clips

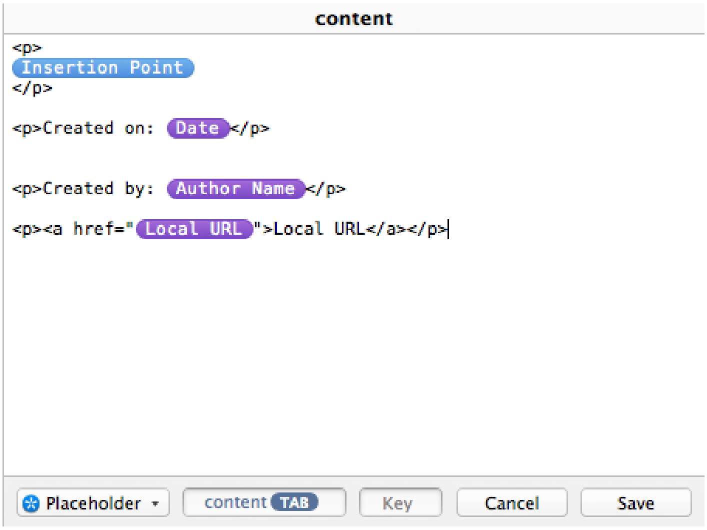
At the home of the sidebar (as indicated by the icon that looks like a house), the first of the available options you'll find is *Clips*.

A Clip is simply a bit of code or text you want to reuse later. You can create *Global clips* which are available in any document opened in Coda, such as the Lorem Ipsum clip that ships with Coda. Create *Site-specific clips* to work only with a designated Site, or _mode-specific_—limiting the clips there to HTML, PHP, Ruby, etc.

The bar at the bottom of Clips has buttons for adding individual clippings or groups of clips. The Action button will provide you with an *Export Clips* option once you add your own to Coda, which save with a *.clips* extension. Back your clips up, as those clips can also be imported through the same Action button.

Also in the bar is a place for us to edit a “tab trigger,” where we designate a shortcut to easily use a clip. By specifying a trigger, when the trigger is typed and followed by the Tab button, Coda will place the entire clip in the document. For example, in the tab trigger area, if we assign the number 1, we can use that clip in our document by typing 1 and then the tab key. Tab triggers offer huge time-saving potential for quickly inserting clips in your document. Take advantage of them.

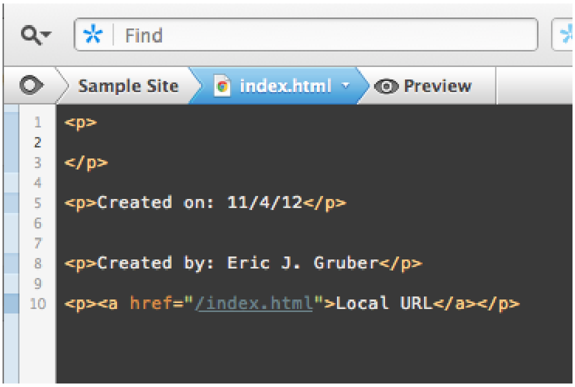
When creating a Clip, Coda has built-in options for placeholders, available at the bottom left of the Clip creation box. Placeholder options include things like Insertion Point, which places your cursor at that placeholder if used in your clip; Date, which automatically inserts the current date in that placeholder; Author Name—in this case, Eric J. Gruber—and more (Figure 5-2).



```
<p>
Insertion Point
</p>

<p>Created on: Date</p>


<p>Created by: Author Name</p>

<p><a href="Local URL">Local URL</a></p>
```

Figure 5-2. Creating a content tab trigger

As you can see from the above example, the content clip I’ve created has several placeholders in it. By using the content tab trigger, the text is inserted into my file, giving the result you see in Figure 5-3.



```
<p>

</p>

<p>Created on: 11/4/12</p>


<p>Created by: Eric J. Gruber</p>

<p><a href="/index.html">Local URL</a></p>
```

Figure 5-3. Inserting the content clip into the page

After the clip is inserted into the file, you can then press the Tab key to move to the next placeholder for easy, quick editing.

Files

This is a simple implementation of the Finder with local and remote view options available at the bottom of the Sidebar.

Find In

Using the Find In sidebar, find and replace can be executed within a Site or in multiple open files.

To refine your search, clicking the arrow by the magnifying glass allows for additional options. And if you're a fan of regular expressions, you can even use those, too.

Hints

This useful feature has to be seen to be appreciated. With the hints tab open, a treasure trove of reference information is made available as you type for HTML, CSS, and JavaScript code. You can see the information based on where your cursor is (such as the beginning of a <p> tag), or by navigation through other hints at the top of the Hints sidebar.

Navigator

Have a name in your document? How about a `<div>` with an ID associated with it? The navigator element names help you bounce around the document by clicking on them in the Navigation sidebar. You can also create your own bookmark by putting an exclamation point at the start inside a comment. For example, an HTML comment with an interior exclamation point would look like this: `<!-- !Start of the Page -->`. Just use the correct syntax for whatever language your using to create a comment, insert the exclamation point, and you're good to go.

You can rearrange them in the order you like, although they will show up in the order they appear in whatever document you're in by default. A search box is also available at the bottom of the Navigation sidebar, should you need to hunt within your document that way.

Places

Yet another of the many ways to navigate your system with Coda, the Places sidebar lets you bookmark frequently visited folders. Simply use the plus or minus sign at the bottom of the Places sidebar to browse your system and find what local or remote folder you wish to add.

Publish

For the Publish sidebar to work, you'll need to have local and remote paths configured in your Site setup.

With your Site open, you'll make changes locally and save them. As changes are made, the Publish sidebar will give you a list of what files are changes and where they will be published on your remote location. When you're ready to publish, simply press the Publish All button. Should you decide not to upload a file, you also can manually uncheck files to remove them from the uploading process. Depending on what changes are made, you can also select to Clear Checked or Publish Checked in the process.

Lastly, a Clear All button is available to remove all the files from the Publish sidebar to prohibit everything from being uploaded.

SCM

Covered extensively in Chapter 3, this is the Source Control Management sidebar.

Shared

Other Coda users can share a document across a network to be edited or viewed by multiple coders. In this sidebar, available shared documents are listed. You can double-click a shared document to connect to it, and you can view others (and they can view you) making changes to that document.

If you've ever used an online document collaboration tool such as Google Docs, this will be quite familiar to you.

Shared documents don't have to be on your network. If they're available by public IP address or hostname, simply click the Connect To button at the bottom of the Shared sidebar and enter in the needed information.

Validate

In my early days of learning the web, I would often visit *http://validator.w3.org* to validate my HTML code. Panic has this feature built into the Validate sidebar, complete with feedback on what needs fixing. (Not that you would ever have code that needs fixing, of course.)

Using the Text Editor

When editing a file, there are some basic aspects of the text editor that we'll need to be familiar with to take full advantage of Coda's power.

Code Folding

New in Coda is the ability to "fold" code that you don't need to see; this helps make your workspace easier to work in. For example, if you have a block of text, like a `<div>`, that you want to collapse, you can put your cursor inside the `<div>` and go to View → Cold Folding → Fold Current Block. The folded code will be indicated in your file by a blue ellipsis. You can unfold the code simply by clicking the ellipsis.

You can also fold the code using the vertical gutter to the left of the line numbers in the editing window. There are shades of blue representing the depth of nesting in the code. Darker shades represent deeper nesting. There are several code folding options available. View them at View → Code Folding and then review the available options in the submenu.

Indenting and Outdenting Lines

I use this feature quite a bit to clean up my code and align large blocks of text for a cleaner display. Select a block of text, then press Command-[to shift the code to the left, or Command-] to shift it right. You can also do this from the menu by choosing Text → Shift Left or Text → Shift Right, respectively, but learning the Command-Bracket shortcut will save you a lot more time in the end.

Quickly Select Text Within Brackets

This trick works with parentheses (), square brackets [], and curly brackets {}; double-clicking either bracket will highlight all the text between two matched brackets.

Quickly Comment Out Selected Text

By highlighting text and pressing Command-+, Command-/, or by using Text → Comment Section from the menu, you can comment out text with ease.

Close Current Tag

Got an open tag lying around? You can manually type it, or you can place your cursor at the end of the text that needs fixing and go to Text → Close Current Tag Pair. The Command-Option-. key combination will give the same result.

Find and Replace Text in a Single File

We discussed find and replace in multiple files earlier in this chapter. A quick Command-F will bring up the find and replace bar above your editing area.

Like the Find In sidebar, there are many options available to suit your needs. Going to Edit → Find and reviewing the submenus there will help you figure out what's best for your task, including using regular expressions and placeholders. Click on the magnifying glass for additional options as well.

Multiline Editing

By pressing the Option key and clicking on similar lines of code, any change you make to one line will propagate to the rest of your selection. Say you have a series of `<h2>` tags you want to add a class to: `class="headline"`. Using multiline editing, you can add this change to the multiple lines you selected once, and the rest will follow. Hitting the Esc key will end your multiline edit. You can also use Text → Blockedit Selection to start or end the multiline editing.

Open a File with Another Application

Got an image that needs editing? Coda can't do that, but with the Files tab, you can right-click the file and select the appropriate application from the Open With menu.

Changing Syntax Modes

As discussed in Chapter 4, your file's coloring and behaviors are contingent upon its syntax. If you need to change the syntax mode, just choose Text → Syntax Mode from the file menu.

Changing Line Endings

Got one of those files where the line endings are all out of whack and you need to fix them? You can stop sharing files with Windows users, or instead, you can go to Text → Line Endings and choose from the options available to fix the file.

Changing File Encodings

Likewise, if you have a file that needs its text encoding altered, go to Text → Encoding and pick from the available options.

Reference Books

Coda has built-in references books for HTML, CSS, Python, Ruby on Rails, PHP, and more. Need to look something up? Just click on the big plus sign in the upper righthand of Coda, select Book, and pick a reference. Also, double-clicking a keyword while holding down the Command key will open up a search for that keyword in the reference book, provided the keyword you're looking up corresponds to the syntax mode you're in. For example, you won't be able to use the easy lookup for PHP while in an HTML document. Rather, you'll need to open the PHP book and search inside it.

Text Processing Commands

There are several commands available from the Text → Processing menu to do some heavy lifting for you within a file. Some favorites include Convert Spaces to Tabs (and vice-versa) and convert URL to link (a personal favorite). Review the commands available so you're aware of what might be handy in a pinch.

Preview a Document

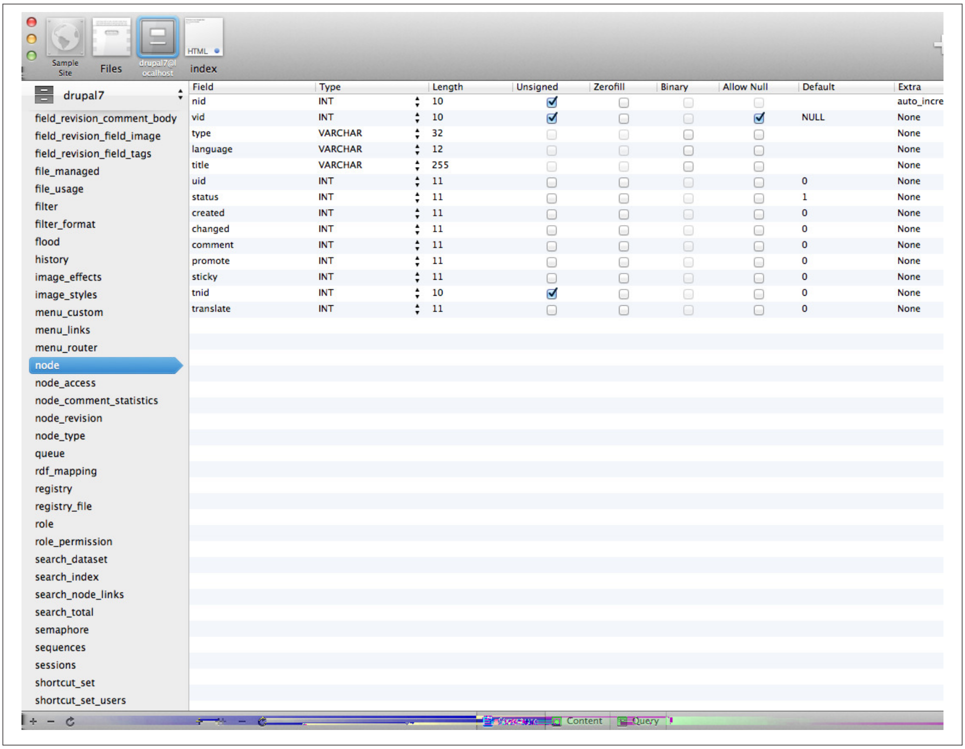
Especially handy for designers, you can view what your document looks like by choosing View → Show Preview. Coda uses Apple's Safari WebKit engine to preview pages, so what you see is the visual equivalent of looking at your page in Safari or Google Chrome. Using PHP or another server-side language? You'll need to make sure your Site is configured with a remote root URL for that. Additionally, a Preview button is available in the path bar; click it for the same result.

> As of 10.7 (Lion), the WebKit used by third-party applications is not identical to Safari's WebKit. Chrome also uses a customized WebKit which differs from Safari. You may see subtle deviations between Coda's built-in Preview feature and the aforementioned browsers.

A great way to take advantage of Preview *and* get some work done is to split a file using the Split pane button in the right corner above your editing area. With the text view in one pane and the preview in the other, you can make changes, see them rendered instantly, and get instant gratification. Just don't forget to put all those changes under version control. We'll discuss another feature for previewing—AirPreview, and how it works with Coda's sibling, Diet Coda—a little later.

Using the MySQL Editor

Coda ships with a powerful MySQL editor, allowing you to connect to a local or remote database server for viewing and editing a MySQL database (Figure 5-4).



Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Default	Extra
nid	INT	10	☑	□	□	□		auto_incre
vid	INT	10	☑	□	□	☑	NULL	None
type	VARCHAR	32	□	□	□	□		None
language	VARCHAR	12	□	□	□	□		None
title	VARCHAR	255	□	□	□	□		None
uid	INT	11	□	□	□	□	0	None
status	INT	11	□	□	□	□	1	None
created	INT	11	□	□	□	□	0	None
changed	INT	11	□	□	□	□	0	None
comment	INT	11	□	□	□	□	0	None
promote	INT	11	□	□	□	□	0	None
sticky	INT	11	□	□	□	□	0	None
tnid	INT	10	☑	□	□	□	0	None
translate	INT	11	□	□	□	□	0	None

Figure 5-4. Viewing the structure of the node table of the Drupal 7 content management system database

Of course, you can configure a Site with a database, but you also can skip that and connect as you go by entering in the required information (hostname, credentials, etc.).

To fire up the MySQL editor, click the plus sign in the upper righthand side of Coda's tab bar and select MySQL. Fill out the necessary details, and once connected, select a database (or create a new one) and you'll be able to see the structure and content of the database.

With a database selected, all the tables of the database are listed on the left side of the editing window in a sidebar. With a table selected, at the bottom of the editing window are buttons for Structure, Content, and Query.

Structure Mode
: Here you're able to view all the rows in a table, and double-clicking a cell will allow you to edit it. You can add or delete tables using the plus or minus symbols at the bottom of the window or reload the database.

Content Mode
: View and edit the data in the table, using the same options available in structure mode.

Query Mode
: Learning MySQL and teaching SQL queries is outside the scope of this book. For those who are versed in structured query language, you can perform a valid query in this mode on your database/table.

Sharing Documents

As briefly mentioned earlier, Coda lets multiple people simultaneously edit the same file in real time.

How does it do this? Coda takes advantage of the engine of SubEthaEdit (a simultaneous text editor from the Germany-based Mac programmers, The Coding Monkeys) which powers the company's own SubEthaEdit Text Editor.

Announcing a Document

To make the document editable by outsiders, we'll need to announce it first. To begin, we'll take an open file and click on the Share button at the bottom of the editing window. This will give us two options: Bonjour and Invite.

Clicking the Bonjour button uses Apple's Bonjour networking technology to share your document. Bonjour will search for others on the network who are using Coda (or, more specifically, the SubEthaEdit Engine) to share a document, then list all those available. For Coda users, it will be in their Shared sidebar, for SubEthaEdit users, in their Bonjour window.

The sharing also allows Coda and SubEthaEdit users outside your local network to edit your document, provided they know your IP address.

Users are identified by the name and Mac OS X user icon for each user. Your icon will be the same as the one you use for your Mac OS X user account. As mentioned in Chapter 4, you can set another image if you like.

Choose whom you want to share the document with, and they will be prompted in Coda to accept the invitation. The invitee can accept or reject your invitation. If they accept, then it's all gravy. Let the editing begin.

Kicking Someone Off Your Documents

You've gone the extra mile to share your documents and have someone else slice away at your code, but now it's time for him to leave. You could cut off his Internet service or get the power company to kill the juice, but Coda has a way that is much more civilized.

Click on the Share button that is located at the bottom of the editing window. Those connected will be listed to the right in the pop-up window. Hovering over a person's name will bring up an "x." Click on it, and that person will be instantly cut off. And, you don't even have to send an email: the disconnected user will receive a notification on her machine about the disconnection.

When this happens, you make multiple copies of the document you were working on, with one on your machine and one on each user that was connected to it. But users will only have the changes made before the disconnection if they save the files to their machine.

Joining a Shared Document

Sharing a document is only one side of the coin. What if it's YOU who are on the receiving end? Or, what if you want to join a document that's shared, but you didn't get your invitation?

Aside from sharing, you can also look for documents that are being shared by others on your network. In the Sidebar is a sharing icon similar to the one from our editing window. Click on it, and the Shared Document area will appear. Then you'll see a list of users on your network who are sharing, and the documents they've made available. Double-click on the document and it will open it in Coda for you to edit.

As mentioned before, you also can connect to shared documents using the Sharing sidebar. If you or the remote user doesn't know the IP address, visit *http://whatismyip.com* to get the information. Panic's documentation also includes this useful nugget for those using both Coda and SubEthEdit at the same time on their computer:

> If the person you are connecting to has both Coda and SubEthaEdit open on the same computer, the person joining may need to increment the default port number (for example, from 6942 to 6943). To specify a port, add a colon at the end of the address, followed by the port number, such as 10.0.1.1:6943.

To make connection easier, the person hosting the document may choose File → Copy Sharing URL, which will place a URL for the shared document on the clipboard. This URL can then be sent to the person attempting to join.

Monitor Changes to Shared Documents

When several people are editing the same document, it can be hard to track who is changing what. Coda makes this process easier to manage by assigning each user with a random color that highlights his changes and the position of his cursor in Coda's preferences.

You can open the Share banner to see the assigned colors. By default, the editing marks are visible, but you can hide them by clicking on View → Hide Change Marks.

Troubleshooting Connection Issues

If you're having problems finding or connecting to shared documents on your network, there are a few things you can try. First, make sure the other user has announced the document or ask the user to invite you to connect.

If you are behind a firewall, Coda must be given permission to use ports ranging from 6942 to 6951.

If users outside of your local network can't access shared documents, the problem could be because of your router. If the person trying to connect outside of your network gets stopped by the router, you need find a way to pass that person through your router to your machine. Port forwarding could be a possible fix.

Port forwarding is where you open up ports on your router to allow incoming and outgoing traffic to your computer. To do this, you'll need to be familiar with the administration panel for your router. The address for administering your router can vary by manufacturer, so contact your user manual for further information.

In Coda's case, we must enable one of the user's routers so that ports in the 6942 to 6951 range are forwarded to her computer. Then, the other user should be able to connect and edit the shared documents. If you or the remote user doesn't know the IP address, visit *http://whatismyip.com* to get the information.

If, after all these attempts, you still can't connect to shared documents, contact the Panic support staff.

CHAPTER 6

Working in Coda

Abraham Lincoln once said, “Give me six hours to chop down a tree and I will spend the first four sharpening the axe.”

Up to this point, we’ve spent most of our time configuring Coda to our specifications, preparing it to work efficiently with our own workflow. Now it’s time to move into how we’re actually going to work in Coda.

We’ll start with the Sample Site we created. Open it up and we’ll start cutting down some proverbial trees.

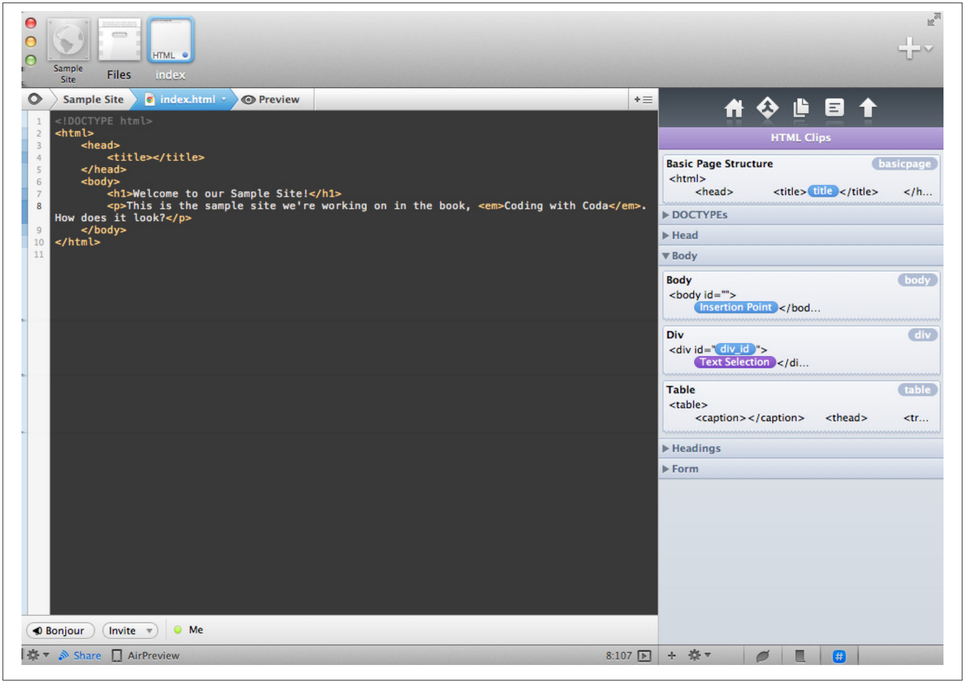
The sole file we have in Sample Site, *index.html*, is empty. We’ll fix that now, adding the following code to create a barebones, HTML5 page.

Constructing a Page

With *index.html* open, we’ll use the Clips sidebar to get our page ready for markup. Since all the cool kids are doing HTML5 these days, we grab the HTML5 DOCTYPE clip and drop it in the page.

We also need a basic structure, but we see that there is an HTML clip with a tab trigger called basicpage. Typing “basicpage” after the DOCTYPE markup and pressing Tab gives us the same effect as dragging the clip to the page.

We’ll need an h1 heading, so let’s tab trigger h1, putting “Welcome to our Sample Site!” at the insertion point. And this page needs an intro paragraph, so directly below it, we’ll code a paragraph tag and some content. Your end result should look like Figure 6-1.



```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're working on in the book, <em>Coding with Coda</em>.
How does it look?</p>
    </body>
</html>
```

Basic Page Structure basicpage
<html>
<head> <title> title </title> </h...

▶ DOCTYPEs

▶ Head

▼ Body

Body body
<body id="">
Insertion Point </bod...

Div div
<div id=" div_id ">
Text Selection </di...

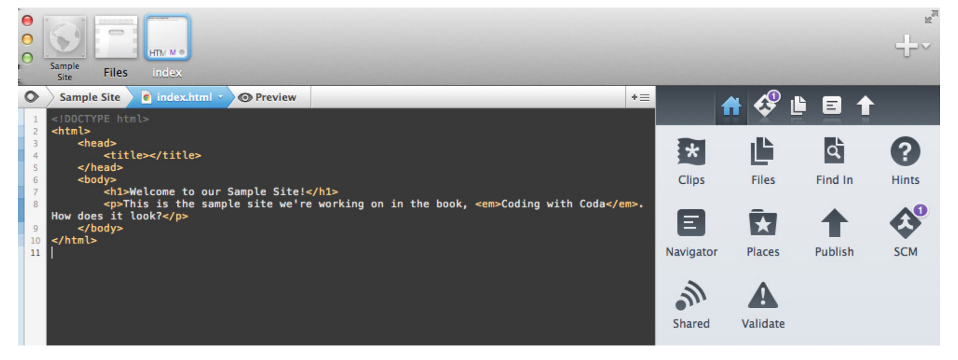
Table table
<table>
<caption></caption> <thead> <tr...

▶ Headings

▶ Form

Figure 6-1. Our index page with some basic structure

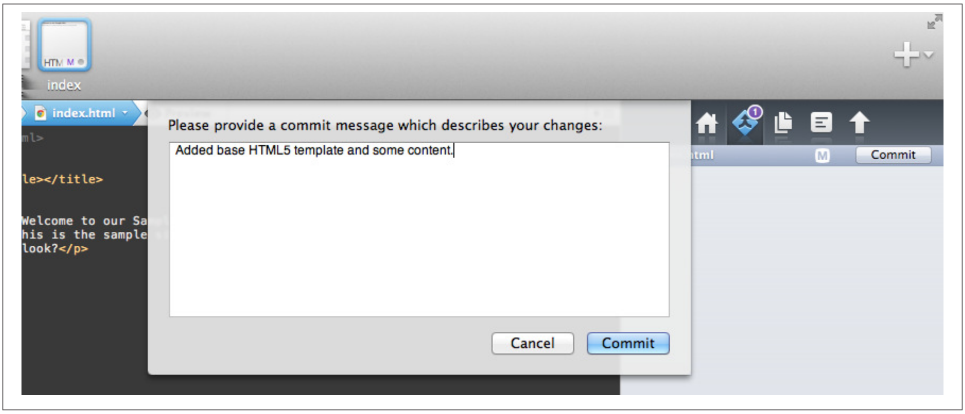
We save the file, and shortly after saving, notice the indicator on the SCM button in the sidebar (Figure 6-2).



```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're working on in the book, <em>Coding with Coda</em>.
How does it look?</p>
    </body>
</html>
```

Figure 6-2. SCM sidebar gives a notification after saving

Clicking on the button brings up the SCM information we discussed earlier. Click the Add button, then the Commit button. Make the message "Added base HTML5 template and some content." and then commit that message. It's easy to see how the workflow can be established from here on out: edit, save, add, commit—repeat as necessary (Figure 6-3).



Please provide a commit message which describes your changes:

Added base HTML5 template and some content.

```
ml>
le></title>
Welcome to our Sa
his is the sample
look?</p>
```

Figure 6-3. The commit message dialog box

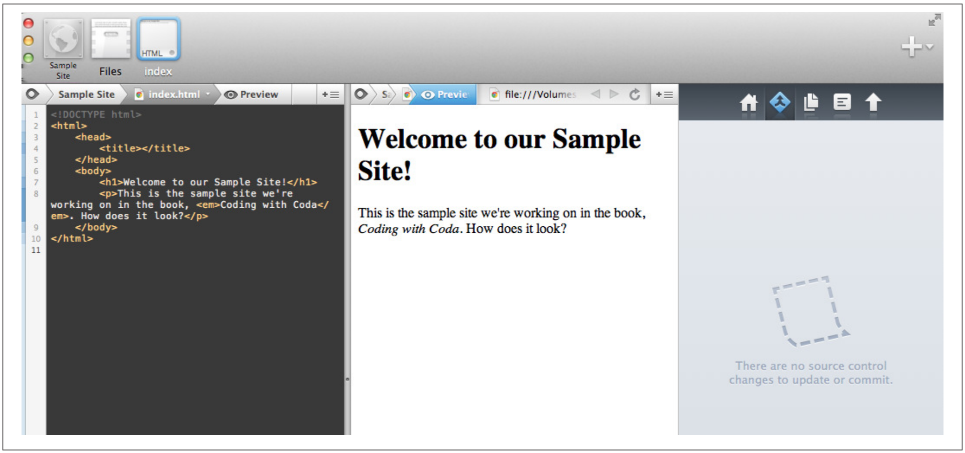
Repetition is great when working with source control, but can be a pain when you're using the same bits of code over and over. Let's see how Coda uses a fun feature to make repetitious code easy to use.

Let's go ahead and preview the page by splitting the viewing window (Figure 6-4). Holding down the Option key while clicking the Split button puts our preview next to our code. The result is…not pretty. We need some CSS.

> You can open multiple files in a single tab in Coda by right-clicking a file in the File Browser, then selecting Open in Split. Also, you can drag a file to an existing tab or path bar, which will replace the focused document in that tab. This is useful when editing HTML and CSS while previewing the HTML file.

Go back to the HTML Clips one more time, and we'll take the Stylesheet (external) clip and insert it before the closing `</head>` tag on our page. We'll come back to it in a minute and link it to a .css file we're about to create.

Clicking on the Files tab, click the plus button at the bottom of the Files sidebar where the index.html file is showing. When the blank file appears, name it *styles.css*. Let's open that file and make this page more attractive.



```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're
working on in the book, <em>Coding with Coda</
em>. How does it look?</p>
    </body>
</html>
```

Welcome to our Sample Site!

This is the sample site we're working on in the book, *Coding with Coda*. How does it look?

There are no source control changes to update or commit.

Figure 6-4. The index in split screen, but it needs some style

Add this code to the stylesheet and save the file:

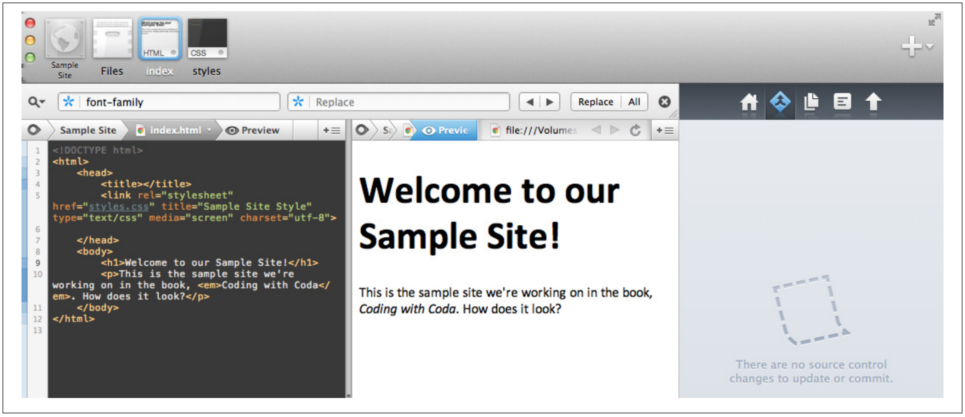
```
+body {
    font-family: "Calibri", Arial, Helvetica, Verdana, sans-serif;
        line-height: 1.2em;
}

h1 {
        font-size: 2.8em;
        line-height: 1.2em;
}+
```

Then visit the SCM sidebar and commit your changes. A simple message of “Added style sheet and body style” will be fine.

Go back to the index file, clicking on its icon in the tab bar. Change the href (and title) code so that it pulls in the new style sheet we just created, like this:

```
<link rel="stylesheet" href="styles.css" title="Sample Site Style"
type="text/css" media="screen" charset="utf-8">
```

And of course, make your commits to version control in the SCM sidebar. Things are starting to look better (Figure 6-5). It’s just a start, but it’s a quick overview of how to work with what we’ve learned in this book.



```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <link rel="stylesheet"
href="styles.css" title="Sample Site Style"
type="text/css" media="screen" charset="utf-8">

    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're
working on in the book, <em>Coding with Coda</
em>. How does it look?</p>
    </body>
</html>
```

Welcome to our Sample Site!

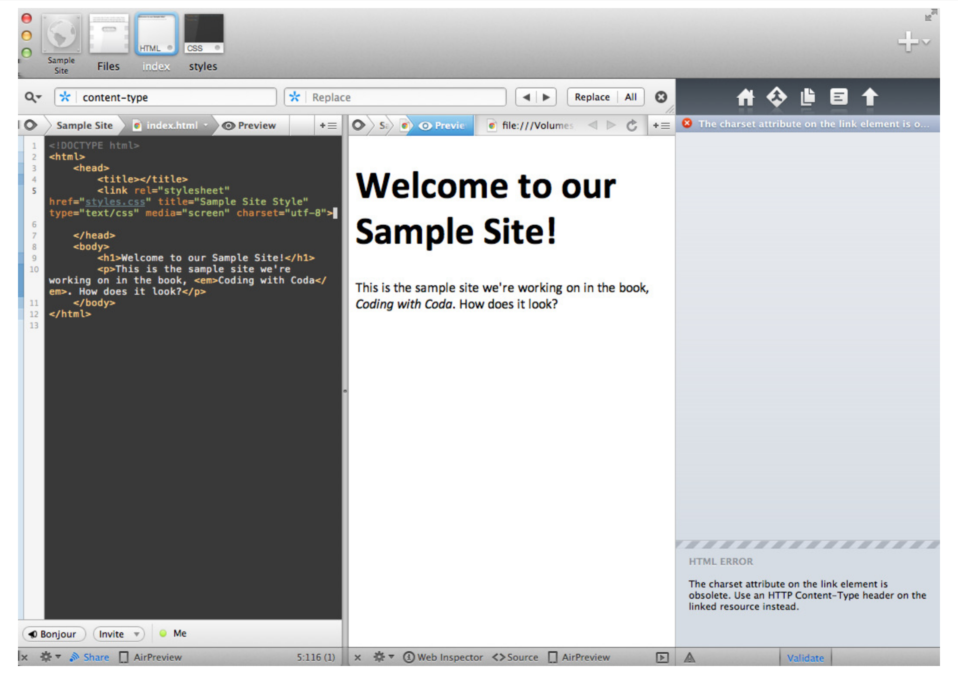
This is the sample site we're working on in the book, *Coding with Coda*. How does it look?

There are no source control changes to update or commit.

Figure 6-5. The index page with better style

And lastly, because we care about such things, let's hop into the Validate sidebar to make sure this code checks out as valid.

Bummer. We pressed the Validate button hoping our code was perfect. Instead, that `charset="utf-8"` isn't valid thanks to the changes in HTML5 (Figure 6-6).



```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <link rel="stylesheet"
href="styles.css" title="Sample Site Style"
type="text/css" media="screen" charset="utf-8">

    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're
working on in the book, <em>Coding with Coda</
em>. How does it look?</p>
    </body>
</html>
```

Welcome to our Sample Site!

This is the sample site we're working on in the book, *Coding with Coda*. How does it look?

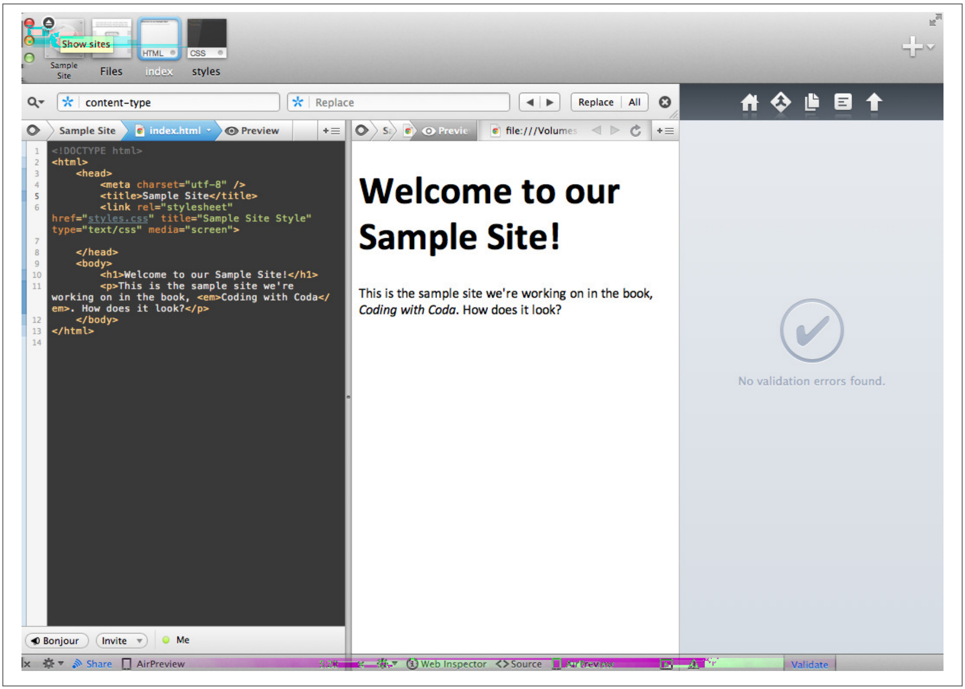
The charset attribute on the link element is o...

HTML ERROR

The charset attribute on the link element is obsolete. Use an HTTP Content-Type header on the linked resource instead.

Figure 6-6. Coda's built-in validation tool gives us an error

So, we remove that code from the stylesheet link. Instead, we put it inside a meta tag above our `<title></title>` (which needs an *actual* title; add something in there now, too). After we revalidate the page, what do we get? (See Figure 6-7.)



```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>Sample Site</title>
        <link rel="stylesheet"
href="styles.css" title="Sample Site Style"
type="text/css" media="screen">

    </head>
    <body>
        <h1>Welcome to our Sample Site!</h1>
        <p>This is the sample site we're
working on in the book, <em>Coding with Coda</
em>. How does it look?</p>
    </body>
</html>
```

Welcome to our Sample Site!

This is the sample site we're working on in the book, *Coding with Coda*. How does it look?

No validation errors found.

Figure 6-7. Our page passes validation with flying colors

Excellent. We have valid HTML5 code that looks decent; at least, decent enough to start building further.

Wrapping Up

Since this isn't a book about web development, we won't go much deeper. But you can see where we're headed.

With your Site configured, you can build this out in the server-side language of your choice, test it on your local machine, hook it up to a MySQL, and edit the database. Then you can commit changes to version control and push everything to a remote server using an FTP or SSH—all within the one window of Coda.

Coda is very flexible, so you'll have to figure out what works best for you and then make it your own. For example, I know some who prefer to commit changes in Git only through a Terminal window. There's no reason why you couldn't do the same with a new Terminal tab opened if you wanted.

It all comes down to how you like to work. But there might be a few more things to check out before you get to coding, as we'll see in the next chapter.

CHAPTER 7

Extras

The more you work with Coda, the more you might find you have questions about existing features, compatibility issues with your system, or just general questions about how Coda works.

Fortunately, there are options for taking your education further than this book.

Books

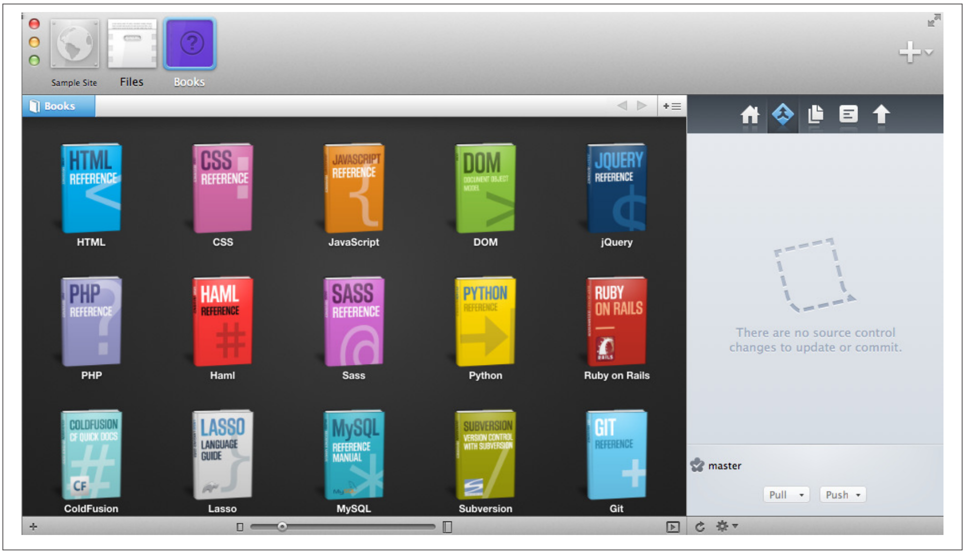
As discussed in Chapter 6, Coda has built-in reference books for HTML, CSS, Python, Ruby on Rails, PHP, and more.

You can add your own custom book very easily. Open up the Book tab by clicking on the plus sign in the upper righthand corner of the window and then selecting Book from the options (Figure 7-1).

From the window that opens, you'll notice a plus sign in the lower lefthand corner of the window. Click it, and a dialog box will give you fields for the book title, URL, a cover image, a choice to search the book directly for a given syntax mode, and a search URL, if available.

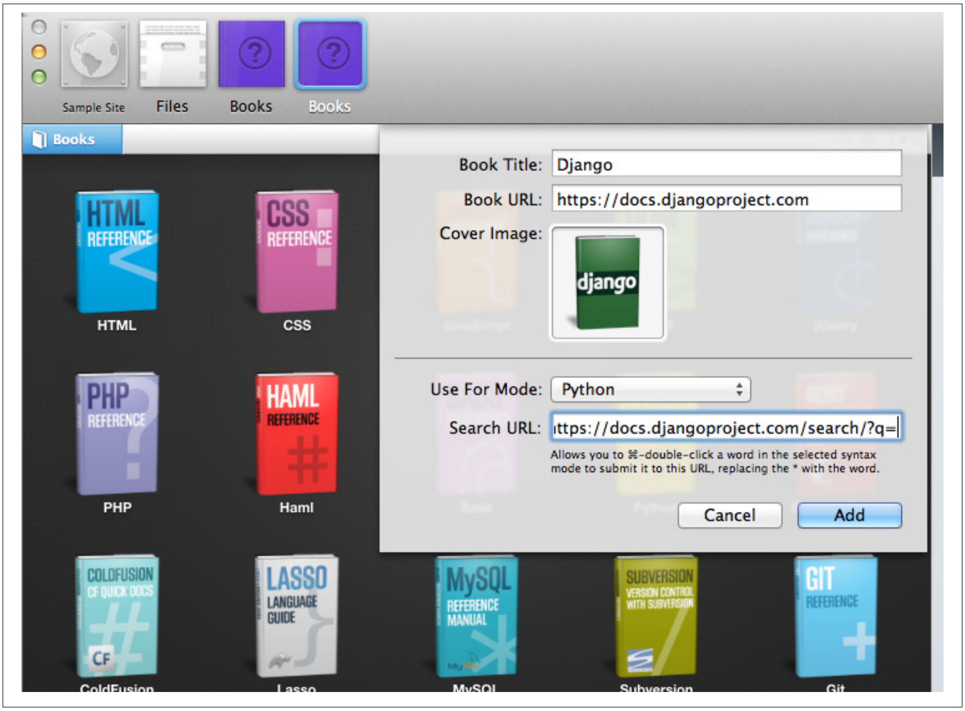
To show how this is done, we'll add a reference book for the popular web development framework, Django. The dialog box will read as follows (see Figure 7-2):

> Book Title: Django
>
> Book URL: *https://docs.djangoproject.com*
>
> Cover Image: (I made one myself)
>
> Use for Mode: Python
>
> Search URL: *https://docs.djangoproject.com/search/?q=**



There are no source control changes to update or commit.

Figure 7-1. The Books in Coda have loads of references at your fingertips.



Book Title: Django

Book URL: https://docs.djangoproject.com

Cover Image:

Use For Mode: Python

Search URL: ıttps://docs.djangoproject.com/search/?q=

Allows you to ⌘-double-click a word in the selected syntax mode to submit it to this URL, replacing the * with the word.

Figure 7-2. Adding the Django "book" cover

By clicking the Add button, there is now a Django reference available within the collection of other books. Poking around on the Django website, I copied the search query from the Django documentation and added an asterisk at the end (*https://docs.djangoproject.com/search/?q=**). Coupled with my assignment of this Book to the Python syntax mode, now I can press the Command key and double click on a word in any Python file and Coda will search this documentation for a reference.

Support

If you find yourself with a technical question you can't figure out on your own, go to Panic's support page. Alternatively, you can shoot them an email at *coda@panic.com*, or find them on Twitter.

Learning

The *Panic Blog* is a fantastic way to keep up to date about the latest developments from Panic, and occasionally find some great tidbits as well. One post you'll want to read is *Top 20 Secrets of Coda 2*.

The mailing list available at Google Groups, *Coda-Users*, is a fantastic resource for getting assistance from other Coda users. If you have a problem that others might have run into before, search the archives, or throw out your own question and wait for a response. You won't be waiting long; the group is active and filled with top-notch information.

Stack Overflow is another great resource for finding answers not only to your Coda questions, but general design and programming questions spanning a multitude of disciplines. Search for Coda in the Stack Overflow search box, or visit this link. And yet another fine resource is *Stack Exchange* with its Apple-centric answers for Coda.

Extending Coda

You can make Coda even more powerful by reviewing the growing list of third-party **plug-ins** from the Panic website. There you will find plug-ins such as Hipster Ipsum, which inserts a more modern version of Lorem Ipsum placeholder text; Markdown, a converter of Markdown-formatted text to HTML; and PHP & Web Toolkit, a PHP, CSS, HMTL, and JavaScript validation tool; and many others.

Additional syntax modes, syntax highlighting themes, and a command-line tool for opening Coda from the Terminal are also available.

If you're interesting in making your own plug-in for Coda, visit the Panic plug-in development page. There you'll be provided with samples and source code, some basic instructions, and details on using the Cocoa plug-in API. You can also download the Coda 2 Plug-in Creator app, free of charge.

AirPreview and Diet Coda

When Panic launched Coda 2, the company also launched Diet Coda, the iOS version of its desktop web editors for iPad.

Diet Coda is available for $19.99 in the App Store. The app offers a lot of great features that are available in Coda: syntax highlighting, Clips, Sites, find and replace, an SSH Terminal, and more.

Perhaps one of the biggest drawbacks to Diet Coda is that there is no local copy, so editing with it means editing live on a web server. This isn't a good practice; it takes your editing outside of source control management, but there are times when, in a pinch, work like that must be done.

On the flip side, one incredible feature is *AirPreview*.

With AirPreview, you can see the Site you're developing on your Mac mirrored on your iPad (note that your Mac and iPad will need to be on the same wireless network). Diet Coda is Retina-ready, so you can see how your Site looks on a tablet—and on a Retina display if you have a Retina iPad—simply by activating AirPreview from the bottom of the Coda editor tab with a Site open (Figure 7-3).

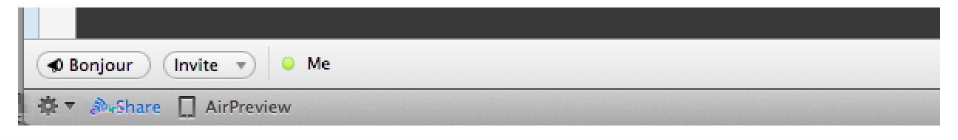


Figure 7-3. The AirPreview button

You'll be prompted to open Diet Coda on your iPad (or purchase it if you haven't already), and then follow the instructions to preview your Site.

You can make changes in Coda, and when you save the changes, they are pushed instantly for viewing in AirPreview.

File Transfers

Although Coda isn't billed as an FTP client, it has very powerful file transfer features built in. As an added bonus, files and folders can be uploaded to a server by dragging them directly to or from a Finder window.

As such, Coda ends up working just like a typical FTP application, although the dedicated FTP client, *Transmit*, is highly recommended and has a lot of powerful features.

Wrapping Up

Working in Coda can be a really fun experience, and a few preference tweaks to your liking can make it a highly efficient tool for you to have in your development toolbox. It's certainly been one of my favorites for a long time, and I look forward to future changes and additions from Panic in the future.

Thank you for reading *Coding with Coda*. It has been as much a pleasure to write as Coda is to use. Feel free to write me at *eric@rumblestrut.com*; I would love to hear your feedback.

About the Author

Eric J. Gruber works full-time as a web developer for the city of Olathe, Kansas.

He lives in Lawrence, Kansas with his wife and two daughters and is co-organizer of a local web development Meetup group, Lawrence Coders.

In his spare time, he plays guitar and rides motorcycles.

Colophon

The animal on the cover of *Coding with Coda* is the guinea pig (*Cavia porcellus*).

The cover image is from Wood's *Animate Creation*. The cover font is Adobe ITC Garamond. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.