

How to be a git

Dominic Mitchell

Git!

- It's new!
- Everybody's talking about it!
- What is it?
 - Distributed Version Control

The Git logo is displayed in a white rectangular box. It features three red dashes above three green plus signs, followed by the word "git" in a green, lowercase, monospace font.

Why Git?

- Fast
- Local
- Toolkit
- Widely used
- Github

Toolkit lets other people build tools for you easily, even if you don't personally care about it.

Why version control?

- Let me rephrase that...

Why not version control?

- Because I like to lose files
- Because I like hundreds of project.old directories
- Because I don't care how my project has changed

Basically, if you're coding, there's no excuse for not using some form of version control. Git is a good choice.

History

- Invented by Linus Torvalds in 2005
 - Y'know, the Linux guy
- When BitKeeper took their toys away
- Basically a weekend hack that grew



Prerequisites

- Comfortable using the command line

There are some GUI tools (demo later), but there are still some things that require a level of command line familiarity.

You may have heard it's complicated. It's not (well, no more than any other VCS).

Installing Git

- <http://code.google.com/p/msysgit/> (Windows)
- <http://code.google.com/p/git-osx-installer/>
- `apt-get install git-core` (Debian)
- <http://www.kernel.org/pub/software/scm/git/>
(DIY)

Also in MacPorts, Fink, FreeBSD ports, etc. It's unlikely you'll have difficulty installing.

Setup

```
git config --global user.email 'me@example.com'  
git config --global user.name 'My Name'
```

Starting a project

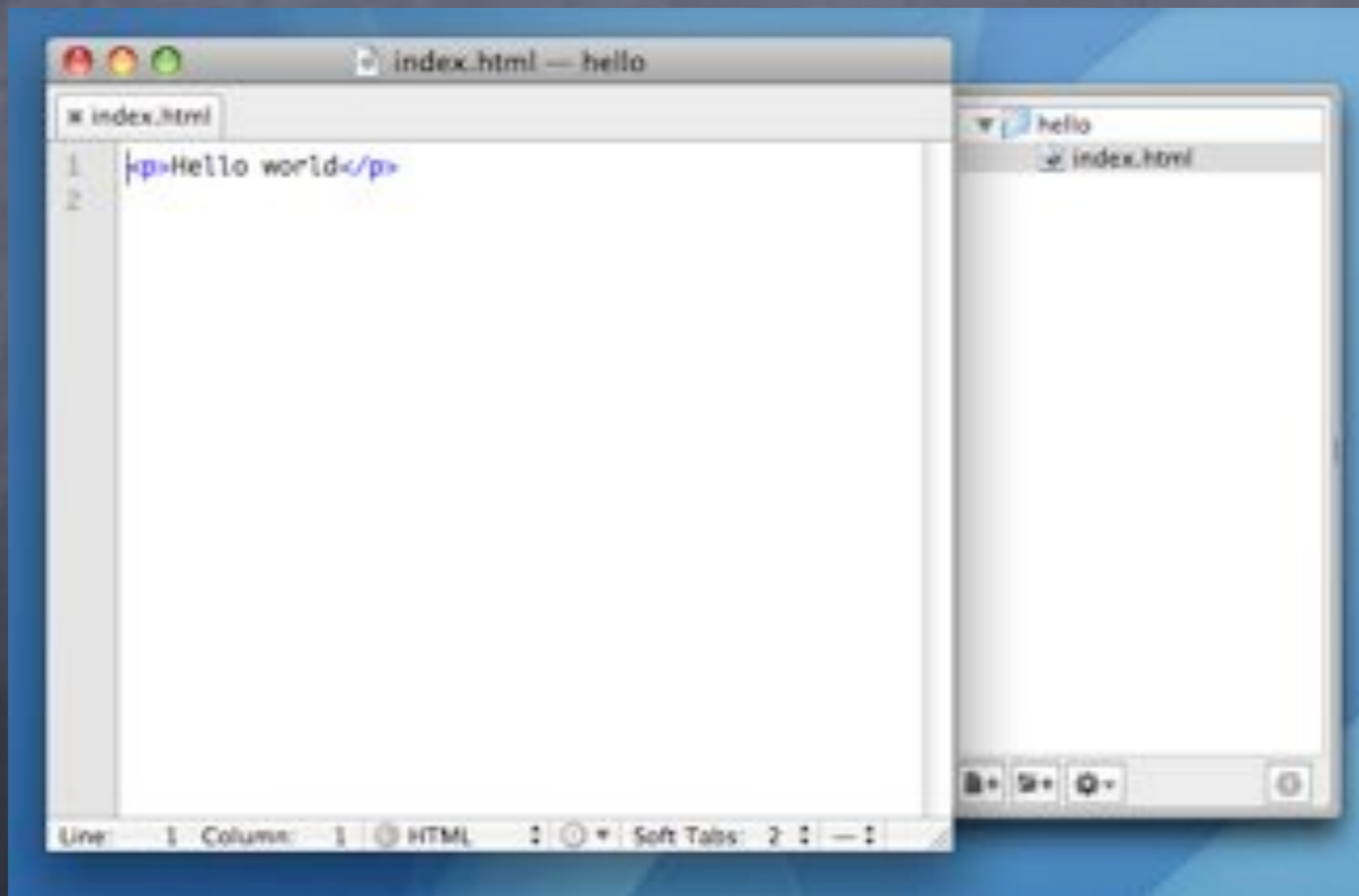
1. Make a new repository

```
$ mkdir hello  
$ cd hello  
$ git init  
Initialized empty Git repository in /Users/dom/work/hello/.git/
```

“This directory is managed by git”

Starting a project

2. Insert content



Any content will do, but let's keep it simple for now.

Starting a project

3. Commit!

```
$ git add .  
$ git commit -m 'Here beginneth the lesson.'  
[master (root-commit) add872b] Here beginneth the lesson.  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 index.html
```

Woohoo! You are now a git master!

“git add .” adds all files in the directory (+ subdirs)
Normally, just say “git commit” and be prompted for a commit message.
NB: SHA1 commit id

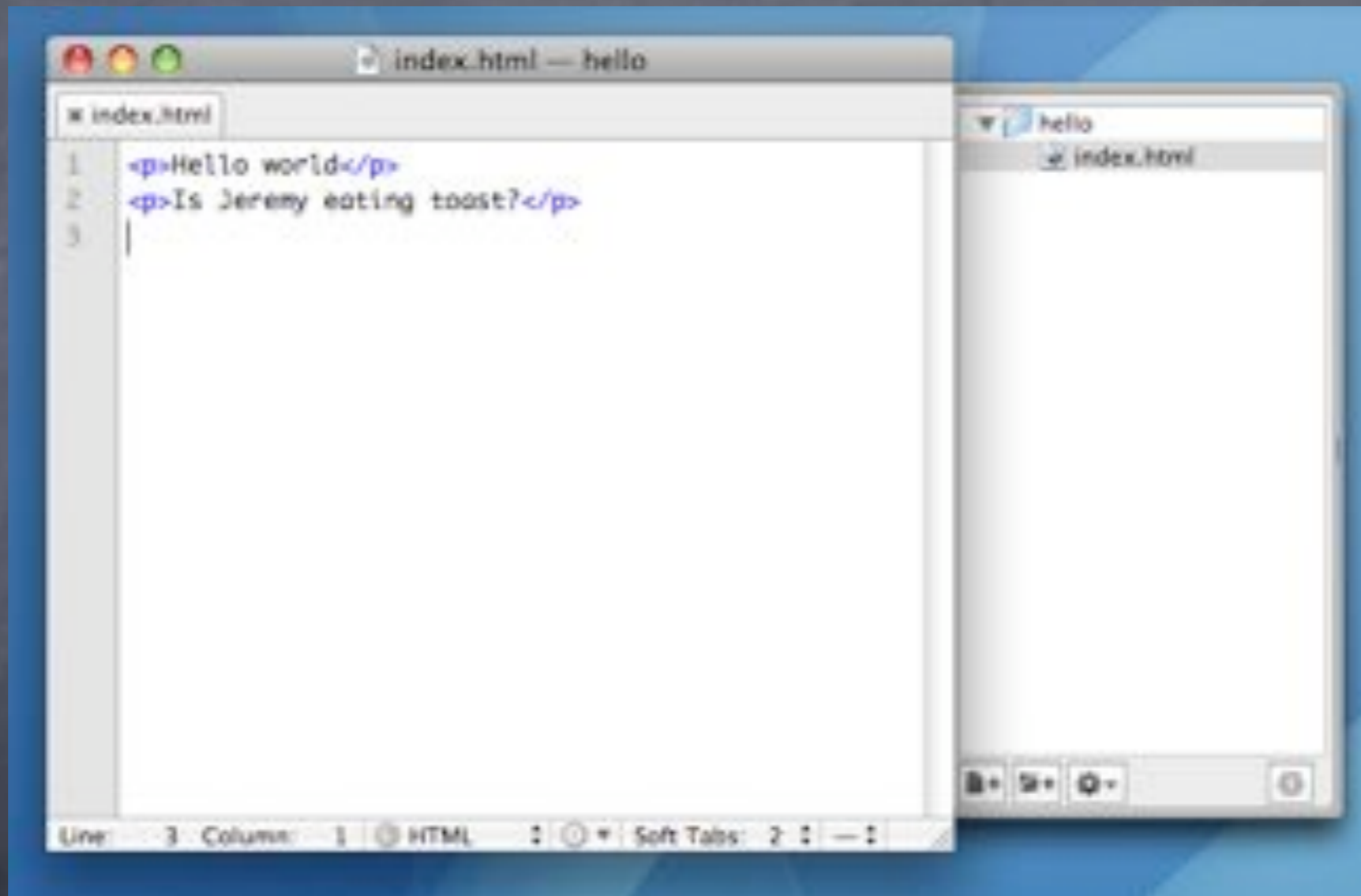
Look what I made!

```
$ git log
```

```
commit add872bbbc04502ced6c6c12d7b0f0cd179a0335  
Author: Dominic Mitchell <dom@happygiraffe.net>  
Date: Thu Sep 3 14:38:19 2009 +0100
```

Here beginneth the lesson.

Let's do some more



What's changed?

```
$ git status
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
# modified:   index.html
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Note helpful commands embedded in output

What's really changed

```
$ git diff
diff --git a/index.html b/index.html
index a04fa3c..c94ed68 100644
--- a/index.html
+++ b/index.html
@@ -1,2 @@
   <p>Hello world</p>
+<p>Is Jeremy eating toast?</p>
```

Commit

```
$ git commit -a  
[master 5817b70] Say what Jeremy's up to.  
1 files changed, 1 insertions(+), 0  
deletions(-)
```

-a == all files

What's in a commit?

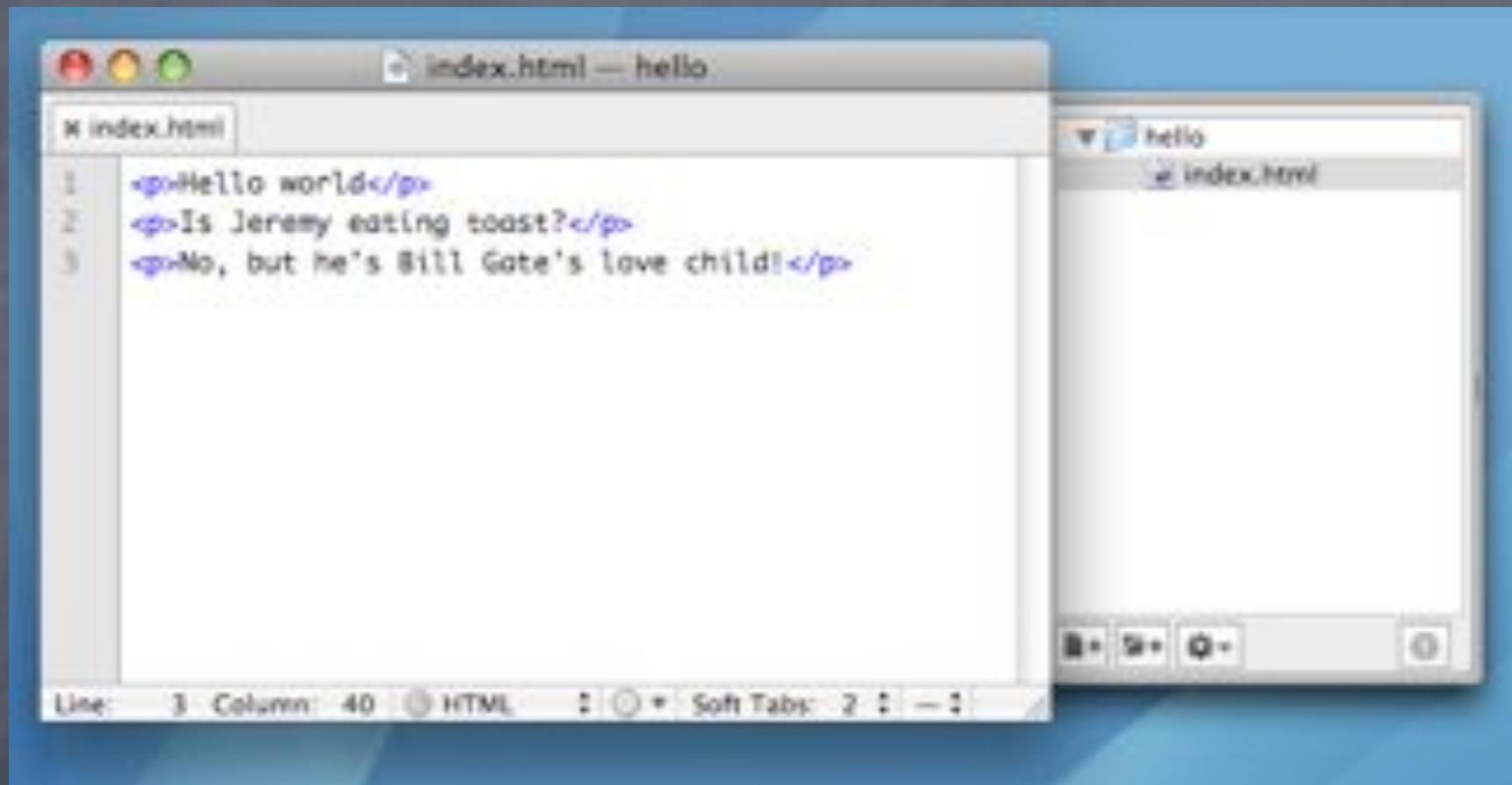
```
$ git show 5817b70
```

```
commit 5817b70c6b011ccfbdeae72a2ad1a05119decacc  
Author: Dominic Mitchell <dom@happygiraffe.net>  
Date: Sat Sep 5 11:51:33 2009 +0100
```

Say what Jeremy's up to.

```
diff --git a/index.html b/index.html  
index a04fa3c..c94ed68 100644  
--- a/index.html  
+++ b/index.html  
@@ -1,2 @@  
 <p>Hello world</p>  
+<p>Is Jeremy eating toast?</p>
```

More change



Revert

```
$ git diff
diff --git a/index.html b/index.html
index c94ed68..5f9a18f 100644
--- a/index.html
+++ b/index.html
@@ -1,2 +1,3 @@
  <p>Hello world</p>
  <p>Is Jeremy eating toast?</p>
+<p>No, but he's Bill Gate's love child!</p>
$ git checkout index.html
$ git diff
```

Stuff happens

The screenshot shows a Git GUI window titled 'hello'. At the top, there are controls for 'View', 'Branch' (set to 'All branches'), 'Create Branch', and a search bar containing 'Subject'. Below this is a commit history table with columns for Subject, Author, and Date. The most recent commit is highlighted in green.

Subject	Author	Date
Tableicious	Dominic Mitchell	2009-09-05 12:06:23
Make heading.	Dominic Mitchell	2009-09-05 12:04:58
Make proper HTML.	Dominic Mitchell	2009-09-05 12:04:42
Say what Jeremy's up to.	Dominic Mitchell	2009-09-05 11:51:33
Here beginneth the lesson.	Dominic Mitchell	2009-09-03 14:38:19

Below the table, the details for the selected commit are shown:

SHA: 208f79a8b643709ac446ffdc3f48b37fc70047ae
Author: Dominic Mitchell <dom@happygiraffe.net>
Date: Sat Sep 05 2009 12:06:23 GMT+0100 (BST)
Subject: **Tableicious**
Refs: **master**
Parent: [50a72bc071ff315fef477dc7a10b343c1b858fd9](#)

The commit message is 'Tableicious', and the file 'index.html' is listed as changed. A diff view for 'index.html' is partially visible below.

5 commits loaded

Throw away commits

```
$ git reset --hard HEAD^  
HEAD is now at 50a72bc Make heading.
```

Gone!

The screenshot shows a Git GUI window titled "hello". At the top, there are controls for "View", "Branch" (set to "All branches"), "Create Branch", and a search bar labeled "Subject". Below these is a table of commit history:

Subject	Author	Date
<input checked="" type="radio"/> Make heading.	Dominic Mitchell	2009-09-05 12:04:58
<input type="radio"/> Make proper HTML.	Dominic Mitchell	2009-09-05 12:04:42
<input type="radio"/> Say what Jeremy's up to.	Dominic Mitchell	2009-09-05 11:51:33
<input type="radio"/> Here beginneth the lesson.	Dominic Mitchell	2009-09-03 14:38:19

Below the table, the details for the selected commit are shown:

SHA: 50a72bc071ff315fef477dc7a10b343c1b858fd9 Ctrl B
Author: Dominic Mitchell <domi@happygiraffe.net>
Date: Sat Sep 05 2009 12:04:58 GMT+0100 (BST)
Subject: **Make heading.**
Refs: **master**
Parent: [56cf4e2f77de50ec5097c4128e64c18806b566a6](#)

The file `index.html` is shown as changed. A diff view for `index.html` is partially visible below, showing a change from `-3,7 +3,7`.

At the bottom, it says "4 commits loaded" and has navigation icons.

Branching

- What's a branch?
- "Parallel line of development"
- Alternative line of work

Git makes branches really cheap and easy to use. They're most important for things you're not sure about. It's trivial to experiment and throw stuff away.

Let's branch!

```
$ git checkout -b eyesore
Switched to a new branch 'eyesore'
$ git commit -a
[eyesore 344a957] Oooh! How 1996!
 1 files changed, 1 insertions(+), 1
deletions(-)
```

New branch

The screenshot shows a Git GUI window titled 'hello'. At the top, there is a 'View' button, a 'Branch' dropdown menu set to 'master', and a 'Create Branch' button. A search bar on the right contains the text 'Subject'. Below this is a table of commit history:

Subject	Author	Date
eyebrow Oooh! How 1996!	Dominic Mitchell	2009-09-05 12:13:19
headline Make heading.	Dominic Mitchell	2009-09-05 12:04:58
Make proper HTML.	Dominic Mitchell	2009-09-05 12:04:42
Say what Jeremy's up to.	Dominic Mitchell	2009-09-05 11:51:33
Here beginneth the lesson.	Dominic Mitchell	2009-09-03 14:38:19

Below the table, the details for the selected commit are shown:

SHA: 344a957c7b61097735ad131bfb95bdf9c95289f7 Copy
Author: Dominic Mitchell <dom@happygiraffe.net>
Date: Sat Sep 05 2009 12:13:19 GMT+0100 (BST)
Subject: **Oooh! How 1996!**
Refs: **eyebrow**
Parent: [50a72bc071ff315fef477dc7a10b343c1b858fd9](#)

The commit message 'Oooh! How 1996!' is displayed, followed by a list of changed files: **changed** index.html. Below this, the content of index.html is shown with line numbers 1-5 and diff markers.

5 commits loaded

Parallel work

```
$ git checkout master
```

```
Switched to branch 'master'
```

```
$ git commit -a
```

```
[master fbbc32e] Protect intellectual property
```

```
1 files changed, 4 insertions(+), 1 deletions(-)
```

Visualisation

The screenshot shows a Git GUI window for a repository named 'hello'. The top bar includes a 'View' button, a 'Branch' dropdown set to 'master', a 'Create Branch' button, and a search field containing 'Subject'. Below this is a table of commit history:

Subject	Author	Date
master Protect intellectual property	Dominic Mitchell	2009-09-05 12:18:21
feature Oooh! How 1996!	Dominic Mitchell	2009-09-05 12:13:19
Make heading.	Dominic Mitchell	2009-09-05 12:04:58
Make proper HTML.	Dominic Mitchell	2009-09-05 12:04:42
Say what Jeremy's up to.	Dominic Mitchell	2009-09-05 11:51:33
Here beginneth the lesson.	Dominic Mitchell	2009-09-03 14:38:19

Below the table, the details for the selected commit are shown:

SHA: [fbbc32e23362b03f9190a7628f399793ff58ca28](#) Copy
Author: Dominic Mitchell <dom@happygiraffe.net>
Date: Sat Sep 05 2009 12:18:21 GMT+0100 (BST)
Subject: **Protect intellectual property**
Refs: **master**
Parent: [50a72bc071ff315fef477dc7a10b343c1b858fd9](#)

The commit message is "Protect intellectual property". A file named "index.html" is listed as changed. The bottom status bar indicates "6 commits loaded".

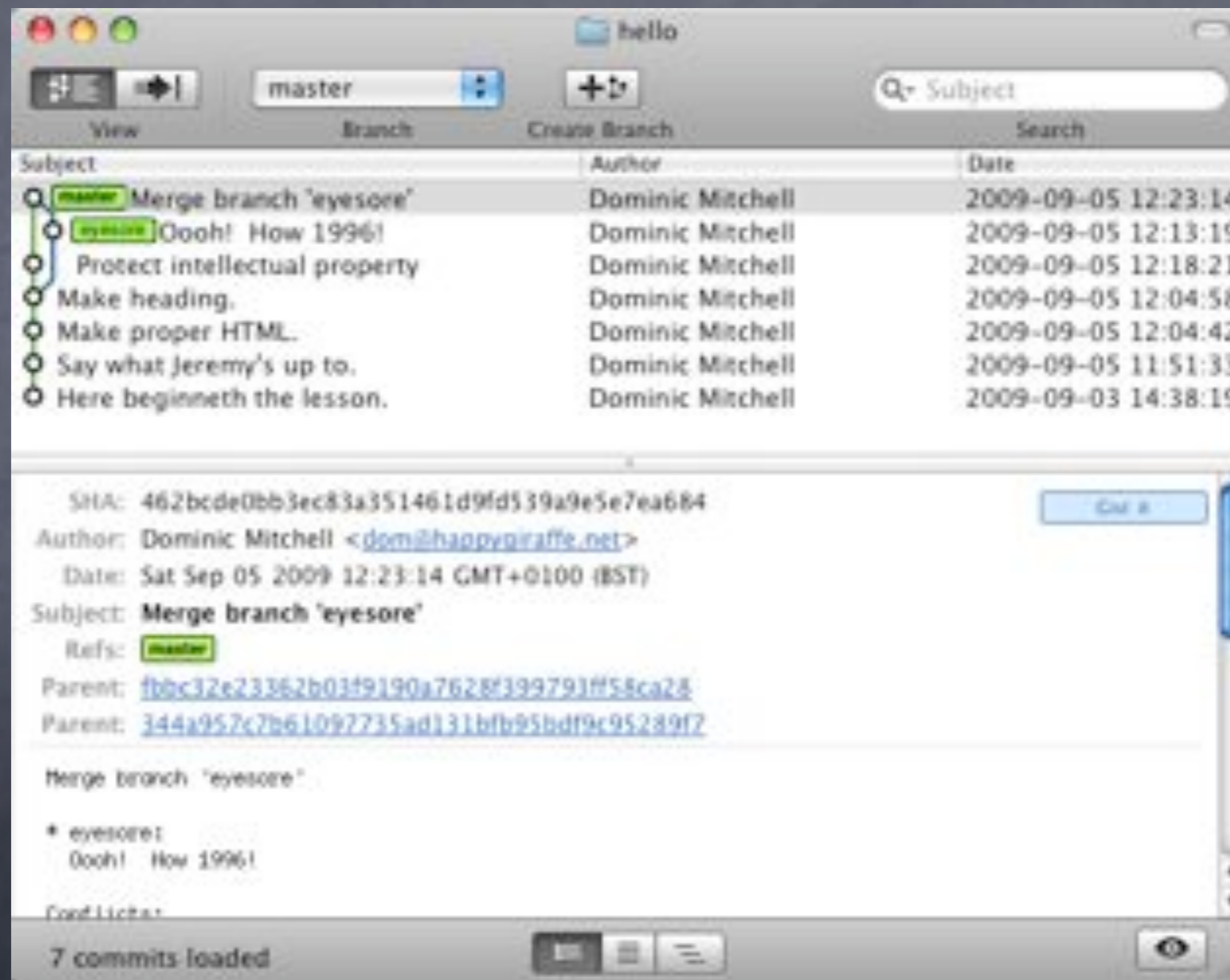
You can clearly see the parallel lines of development going on.

Merging

```
$ git merge eyesore
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
$ vim index.html
$ git commit -a
[master 462bcde] Merge branch 'eyesore'
```

Damn! The client wants the “eyesore” work...

Merge commit



What about the branch? Delete it or leave it, doesn't matter.

Rebase

- Merge alternative
- Re-apply each change at new point in time

Distributed

- clone somebody else's repository
- ... hack, hack, hack ...
- push back or mail patches

Clone

```
$ git clone git@github.com:happygiraffe/irccat.git
Initialized empty Git repository in /private/tmp/irccat/.git/
remote: Counting objects: 111, done.
remote: Compressing objects: 100% (81/81), done.
remote: Total 111 (delta 20), reused 0 (delta 0)
Receiving objects: 100% (111/111), 2.28 MiB | 100 KiB/s, done.
Resolving deltas: 100% (20/20), done.
```

This is a complete copy — you need never go back to the original again.

Complete Copy

irccat

View Branch Create Branch Search Subject

Subject	Author	Date
Merge bran... origin/HEAD origin/master	Dominic Mitchell	2009-01-15 22:22:09
change .isEmpty to restore java 1.5 ...	rj	2009-01-15 14:46:32
Build a tar.gz file.	Dominic Mitchell	2009-01-15 22:13:44
Add a shell script to run irccat.	Dominic Mitchell	2009-01-15 22:09:26
Enable line numbers in stack traces.	Dominic Mitchell	2009-01-15 21:08:55
Clean up unused imports.	Dominic Mitchell	2009-01-15 21:08:03
Remove unnecessary @SuppressWarnings...	Dominic Mitchell	2009-01-15 21:06:41
Add missing @Override annotations.	Dominic Mitchell	2009-01-15 21:02:30
Ignore created directories.	Dominic Mitchell	2009-01-15 20:56:02
add link to trac project	rj	2009-01-08 12:09:42

SHA: 61ce0f1c6fe999e054081ccda7469306a5570955 [Copy](#)

Author: Dominic Mitchell <dom@semantico.com>

Date: Thu Jan 15 2009 22:22:09 GMT+0000 (GMT)

Subject: Merge branch 'master' of <git://github.com/RJ/irccat>

Refs: [master](#) [origin/HEAD](#) [origin/master](#)

Parent: [bf48d46526a8c92787f77b359c0447361ff4c7c2](#)

Parent: [6df909f341ba736451e6a9e0b93fa21c875c1c6](#)

Merge branch "master" of <git://github.com/RJ/irccat>

* "master" of <git://github.com/RJ/irccat>:
change .isEmpty to restore java 1.5 connectivity

12 commits loaded

Push me pull you

- `git push` — send your changes to origin repo
- `git pull` — update your copy from origin
 - Really, `git fetch` + `git merge`

Patches

- Sometimes you can't push
- Patch & email instead

Making Patches

```
$ git commit -m 'Format XML.' build.xml
[master 69f615e] Format XML.
 1 files changed, 53 insertions(+), 64 deletions(-)
rewrite build.xml (74%)
$ git format-patch origin/master
0001-Format-XML.patch
```

Publishing

- <http://github.com> — social coding
- gitjour — share via bonjour / rendezvous

Cool Git Features

- `git rebase -i` — rewrite history
- `git add -p` — commit parts of a file
- `git svn` — use git with subversion

GUIs

- Gitk (Tk — cross platform)
- gitx (OSX only)
- TortoiseGit (Windows)

Links

- <http://git-scm.com/>
- <http://whygitisbetterthanx.com/>