

Git Introduction I - Local Repositories

Luis Ibanez original modified by Moorthy

Git

Introduction

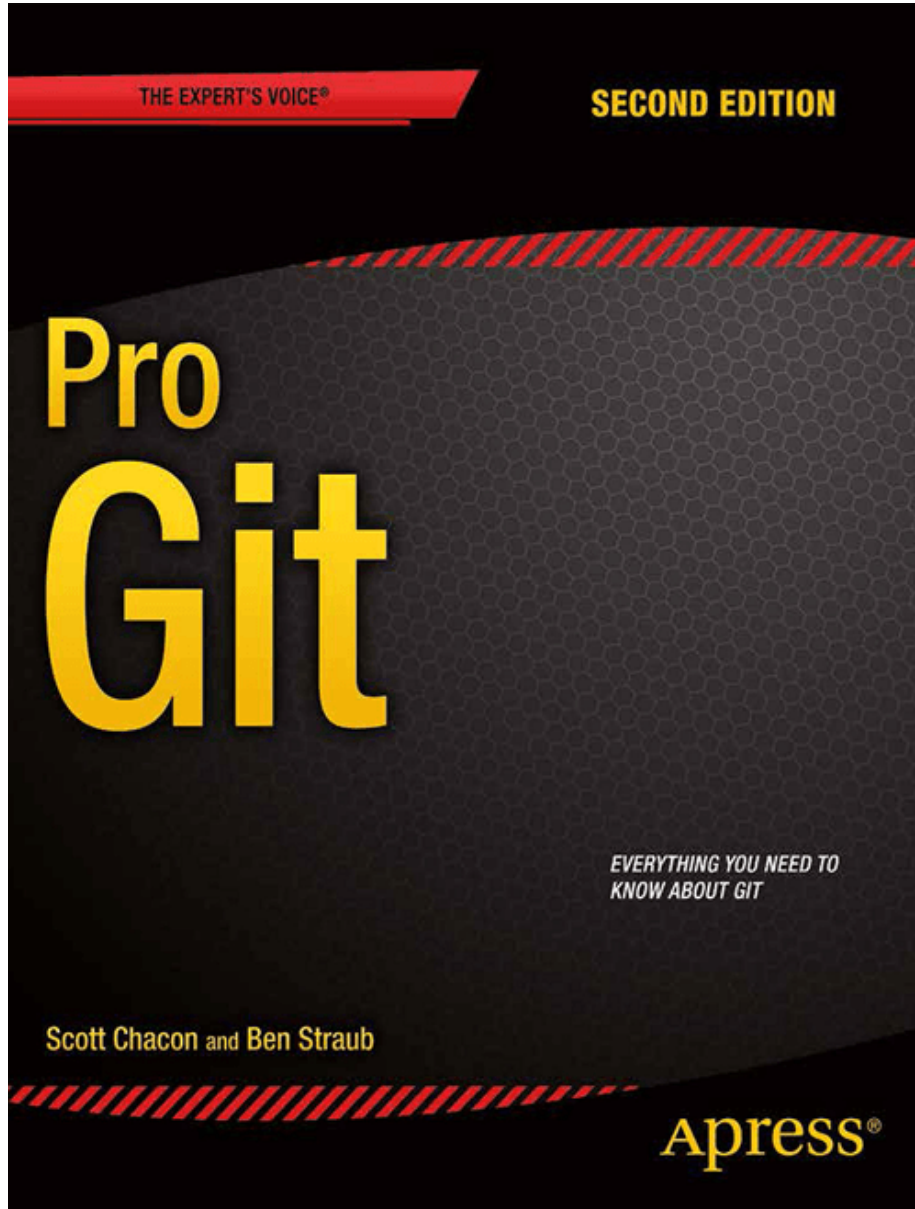
Part I

Created by Luis Ibanez / @luisibanez, modified by Moorthy /@mskmoorthy and Wesley Turner / @wdturner



Git Introduction by Luis Ibanez (original author), modified by Moorthy and Wesley Turner is licensed under a Creative Commons Attribution 3.0 Unported License.

Reading Material



Pro Git

by Scott Chacon and Ben Straub

Chapters 1 - 3.3 and 4.3

Install Git

Instructions Here

Global Configuration

Author Information

```
git config --global user.name "mskmoorthy"  
git config --global user.email mskmoorthy@gmail.com
```

Global Configuration

Color Preferences

```
git config --global color.ui "auto"
```

Check Configuration

```
git config --global --list
```

Generating ssh-key

- Retains the global configuration
- No need to type your password each time
- Go to top right corner of your github page - select settings
- Select ssh keys from your left column menu

Generating a new SSH key

ssh-key

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"  
# Creates a new ssh key, using the provided email as a label  
Generating public/private rsa key pair.
```

Generating a new SSH key (contd)

ssh-key

```
Enter a file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]  
Enter passphrase (empty for no passphrase): [Type a passphrase]  
Enter same passphrase again: [Type passphrase again]
```

Generating a new SSH key (contd)

ssh-key addition to your github account

```
cd .ssh
```

- Copy the contents of id_rsa.pub to your clipboard
- In your github page, click new ssh key
- In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "ubuntulinuxopen-source".
- Paste your key into the "Key" field.

Check whether your ssh keys have been entered properly

```
ssh -T git@github.com
```

Create Directory

- Make new directory "exercise01"
- Change into new directory "exercise01"

Create Directory

```
mkdir exercise01
```

```
cd exercise01
```

Create Local

Git Repository

- Initialize local Repository

Create Local

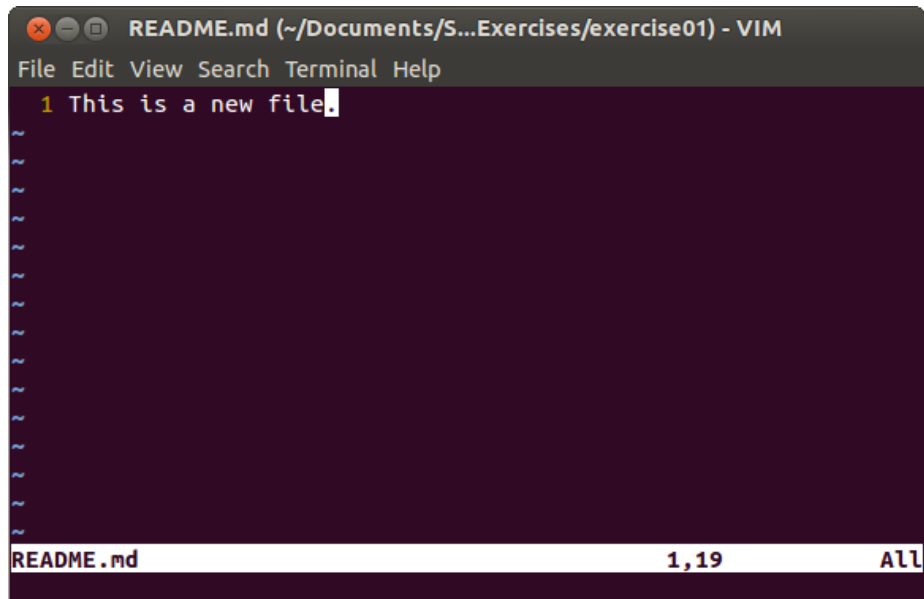
Git Repository

```
git init
```

Edit New File

```
vim README.md
```

Edit New File



Edit New File

Vim Commands

```
i
This is a new file
ESC
:wq
```

Edit New File

Vim Commands

- i = insert mode
- ...write something...
- ESC = command mode
- :wq = write and quit

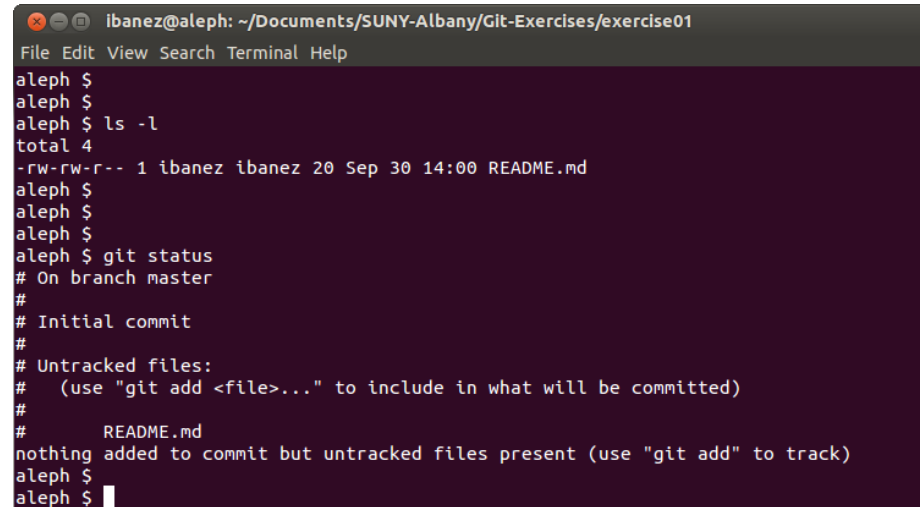
What is going on?

```
ls -l
```

What is going on?

```
git status
```

What is going on?

A terminal window with a dark purple background and light green text. The title bar shows the user 'ibanez@aleph' and the path '~/Documents/SUNY-Albany/Git-Exercises/exercise01'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a series of commands and their outputs: 'aleph \$' followed by another 'aleph \$', then 'aleph \$ ls -l' which outputs 'total 4' and a file entry for 'README.md'. This is followed by another 'aleph \$' and then 'aleph \$ git status'. The output of 'git status' indicates the user is on the 'master' branch, it's an initial commit, and there is one untracked file, 'README.md'. It suggests using 'git add' to track the file. The prompt 'aleph \$' is shown at the bottom with a cursor.

```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $
aleph $
aleph $ ls -l
total 4
-rw-rw-r-- 1 ibanez ibanez 20 Sep 30 14:00 README.md
aleph $
aleph $
aleph $ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       README.md
nothing added to commit but untracked files present (use "git add" to track)
aleph $
aleph $
```

Add File to
Repository

```
git add README.md
```

What is going on?

```
git status
```

What is going on?

```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $ git add README.md
aleph $
aleph $
aleph $ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   README.md
#
aleph $
```

Commit File into Repository

```
git commit
```

Editor should open

```

COMMIT_EDITMSG (~/.Documents/SUNY...it-Exercises/exercise01/.git) - VIM
File Edit View Search Terminal Help
1
2 # Please enter the commit message for your changes. Lines starting
3 # with '#' will be ignored, and an empty message aborts the commit.
4 # On branch master
5 #
6 # Initial commit
7 #
8 # Changes to be committed:
9 #   (use "git rm --cached <file>..." to unstage)
10 #
11 # new file:   README.md
12 #

```

.git/COMMIT_EDITMSG 1,0-1 All

Write Commit Message

It should look like an email

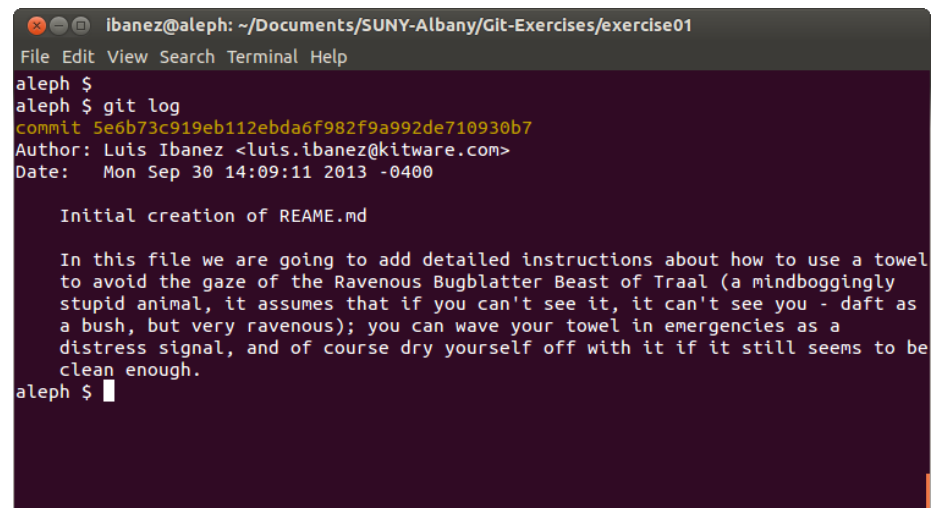
- First line is the SUBJECT
- Second line is empty
- The rest, are paragraphs with the BODY of the email.

Write Commit Message

Git can already answer

- Who
- When
- What

`git log`

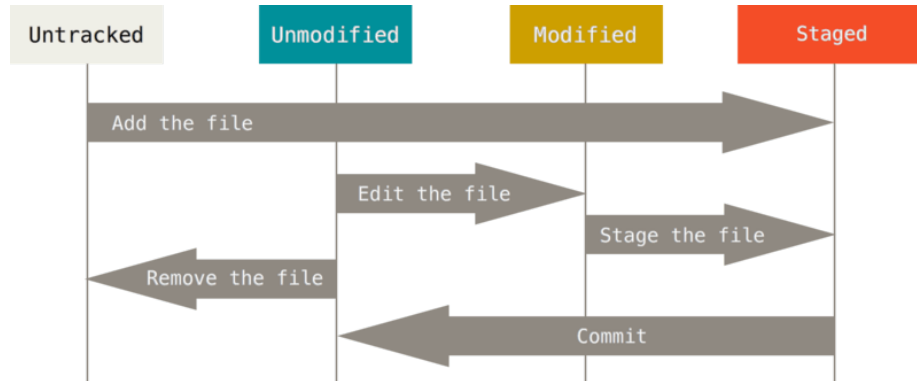
A terminal window with a dark background and light text. The title bar shows 'Ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01'. The terminal content shows the command 'git log' being executed, resulting in a commit hash, author information, date, and a commit message about a README file.

```
Ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $
aleph $ git log
commit 5e6b73c919eb112ebda6f982f9a992de710930b7
Author: Luis Ibanez <luis.ibanez@kitware.com>
Date: Mon Sep 30 14:09:11 2013 -0400

    Initial creation of REAME.md

    In this file we are going to add detailed instructions about how to use a towel
    to avoid the gaze of the Ravenous Bugblatter Beast of Traal (a mindboggingly
    stupid animal, it assumes that if you can't see it, it can't see you - daft as
    a bush, but very ravenous); you can wave your towel in emergencies as a
    distress signal, and of course dry yourself off with it if it still seems to be
    clean enough.
aleph $
```

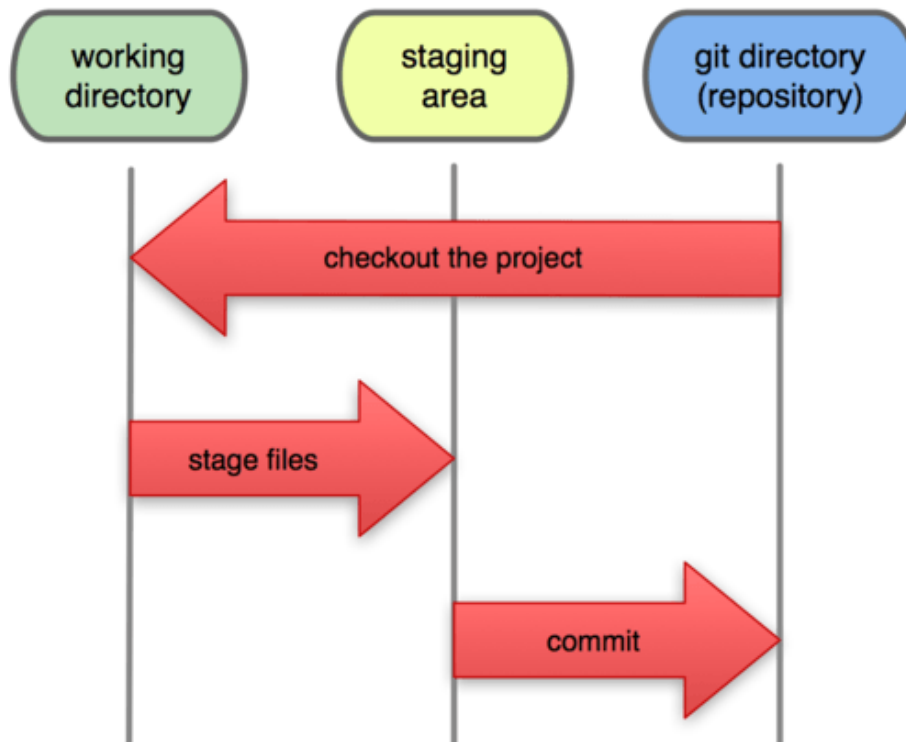

File Status Lifecycle



Courtesy of git-scm Book.

The Three States

Local Operations



Courtesy of git-scm Book.

Go to your Github Page

Create a Repository "Exercise01" under your github account (same name)

```
git remote add origin http://github.com/mskmoorthy/Exercise01.git
git push -u origin
```

Your Turn !

Changes bring

Happiness!

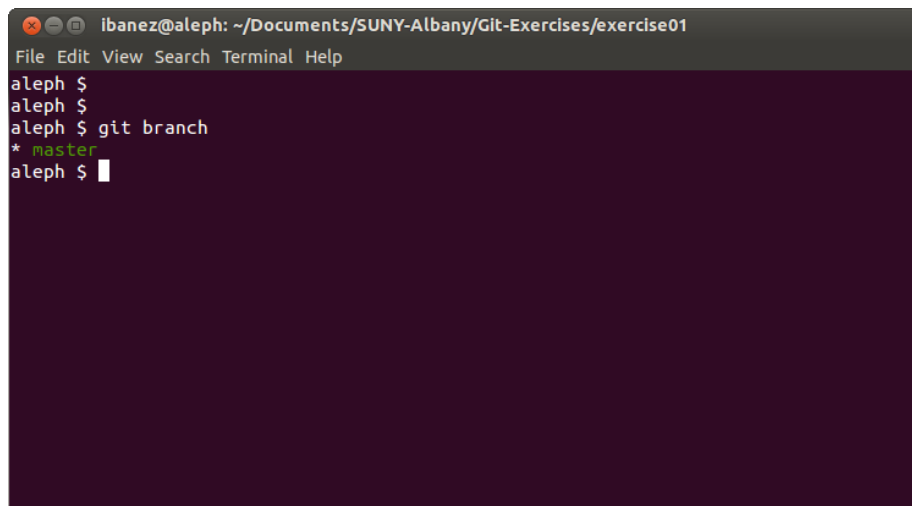
Modify and Commit the file

- Edit the file
- Git add the changes
- Git commit the changes
- Git check the log

Branching

Inspect Branches

```
git branch
```

A terminal window with a dark background and light text. The title bar at the top reads "ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01". Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows the user "aleph" at the prompt "\$" typing "git branch". The output is "* master", where "master" is highlighted in green. The prompt returns to "aleph \$".

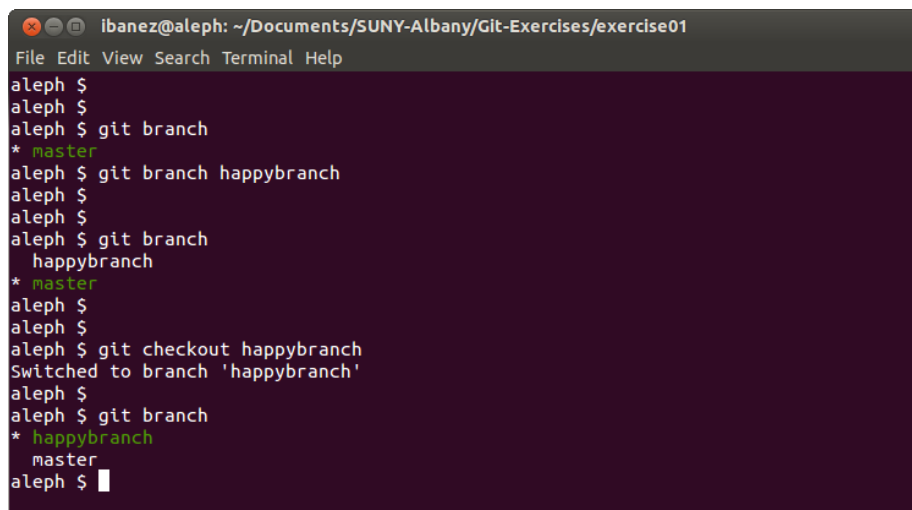
```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $
aleph $
aleph $ git branch
* master
aleph $
```

Create New Branch

```
git branch happybranch
git branch
git checkout happybranch
git branch
```

Move to New Branch

```
git checkout happybranch
```

A terminal window with a dark purple background and a light gray title bar. The title bar contains the text 'ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01'. The terminal shows a series of commands and their outputs. The commands are: 'aleph \$', 'aleph \$', 'aleph \$ git branch', 'aleph \$ git branch happybranch', 'aleph \$', 'aleph \$ git branch', 'aleph \$ git checkout happybranch', 'aleph \$ git branch', and 'aleph \$'. The outputs are: 'aleph \$', 'aleph \$', '* master', 'aleph \$', 'happybranch', '* master', 'aleph \$', 'aleph \$', 'Switched to branch \'happybranch\'', 'aleph \$', '* happybranch', 'master', and 'aleph \$'. The cursor is at the end of the last line.

```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $
aleph $
aleph $ git branch
* master
aleph $ git branch happybranch
aleph $
aleph $ git branch
happybranch
* master
aleph $
aleph $
aleph $ git checkout happybranch
Switched to branch 'happybranch'
aleph $
aleph $ git branch
* happybranch
master
aleph $
```

Changes bring

Happiness!

Modify and Commit a file

- Edit the file
- Git add the changes
- Git commit the changes
- Git check the log

File Lifecycle

```
vim README.md
git add
git commit
git branch
```

```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $ vim README.md
aleph $ git add README.md
aleph $ git commit
[happybranch 7d90101] This file is getting older.
1 file changed, 3 insertions(+)
aleph $ git branch
* happybranch
  master
aleph $
```

Merge Branch

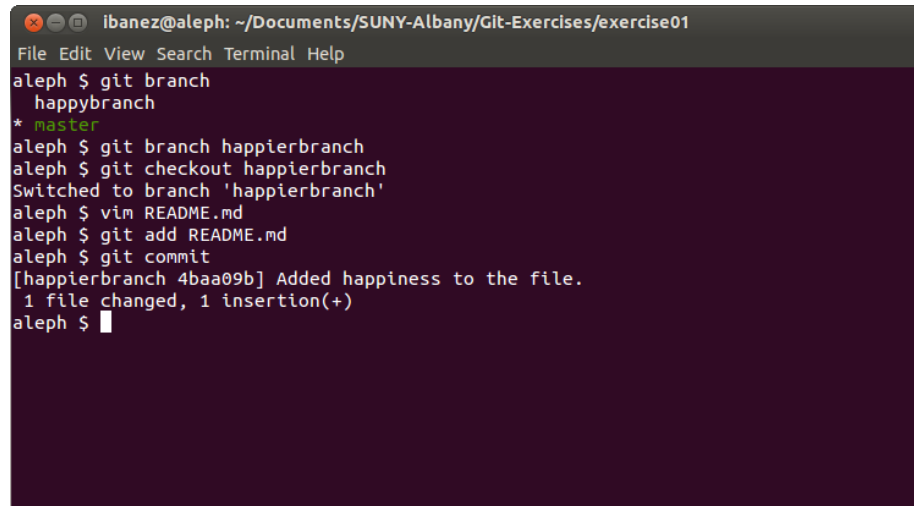
```
git branch
git checkout master
git branch
git merge happybranch
```

```
ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01
File Edit View Search Terminal Help
aleph $ git branch
* happybranch
  master
aleph $ git checkout master
Switched to branch 'master'
aleph $ git branch
  happybranch
* master
aleph $ git merge happybranch
Updating 5e6b73c..7d90101
Fast-forward
 README.md | 3 +++
1 file changed, 3 insertions(+)
aleph $
```

Let's do it Again!

```
git branch happierbranch
git checkout happierbranch
```

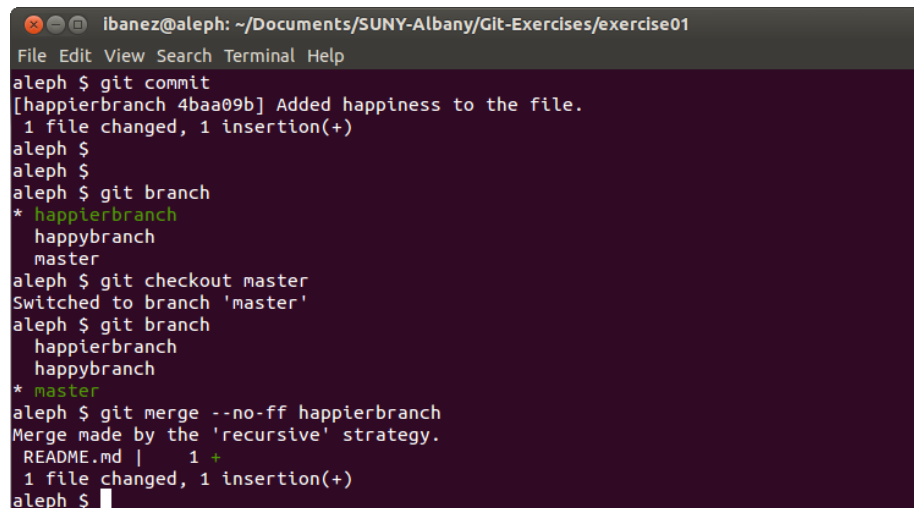
```
vim README.md
git add README.md
git commit
```

A terminal window titled 'ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
aleph $ git branch
  happybranch
* master
aleph $ git branch happierbranch
aleph $ git checkout happierbranch
Switched to branch 'happierbranch'
aleph $ vim README.md
aleph $ git add README.md
aleph $ git commit
[happierbranch 4baa09b] Added happiness to the file.
1 file changed, 1 insertion(+)
aleph $
```

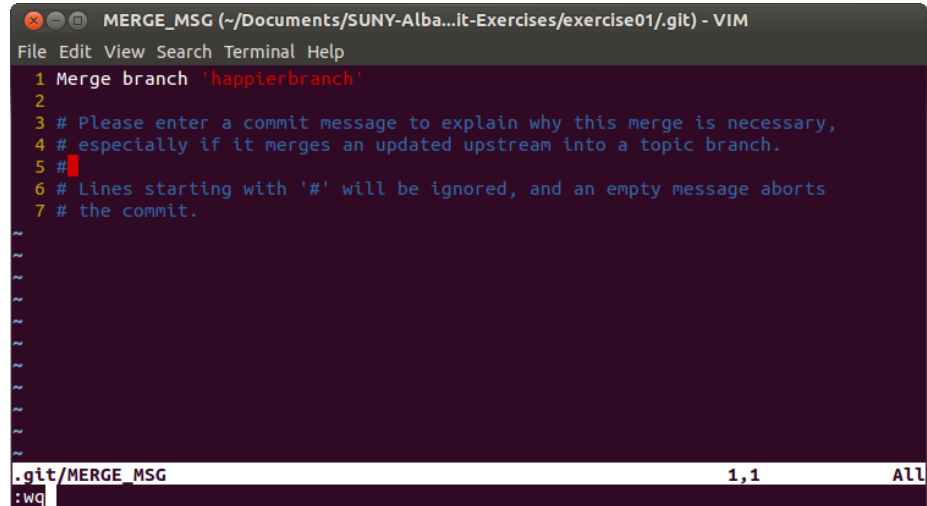
Merge Differently

```
git checkout master
git merge --no-ff happierbranch
```

A terminal window titled 'ibanez@aleph: ~/Documents/SUNY-Albany/Git-Exercises/exercise01' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
aleph $ git commit
[happierbranch 4baa09b] Added happiness to the file.
1 file changed, 1 insertion(+)
aleph $
aleph $
aleph $ git branch
* happierbranch
  happybranch
  master
aleph $ git checkout master
Switched to branch 'master'
aleph $ git branch
  happierbranch
  happybranch
* master
aleph $ git merge --no-ff happierbranch
Merge made by the 'recursive' strategy.
 README.md | 1 +
1 file changed, 1 insertion(+)
aleph $
```

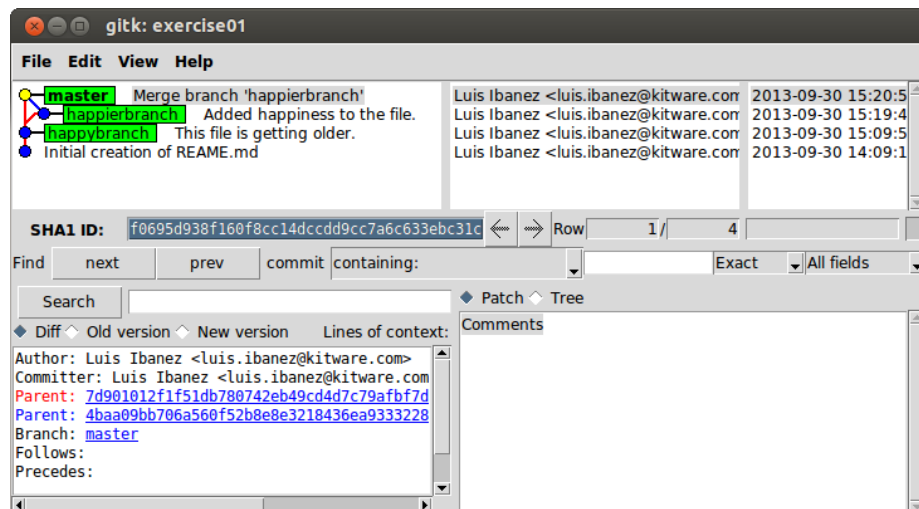
Explain the Merge



```
1 Merge branch 'happierbranch'
2
3 # Please enter a commit message to explain why this merge is necessary,
4 # especially if it merges an updated upstream into a topic branch.
5 #
6 # Lines starting with '#' will be ignored, and an empty message aborts
7 # the commit.
```

Inspect Branches

gitk



Your Turn !

Branches bring

Happiness!

Branch and Merge

- Create new branch
- Move into new branch
- Edit and a commit file
- Move to master branch
- Merge new branch into master

Complete Three Merges!

THE END

BY Luis Ibanez some modifications by Moorthy and Wes Turner