

Let's  
Learn  
Git.  
No  
Excuses!



dwayne@pantheon.io

@mcdwayne on Twitter

Slides at mcdwayne.com



Developer Advocate At Pantheon  
Since 2014

[dwayne@pantheon.io](mailto:dwayne@pantheon.io)

@mcdwayne on Twitter

Some things I enjoy:

- Webcomics and Comic Books
- Crochet and learning knitting
- Karaoke!!!

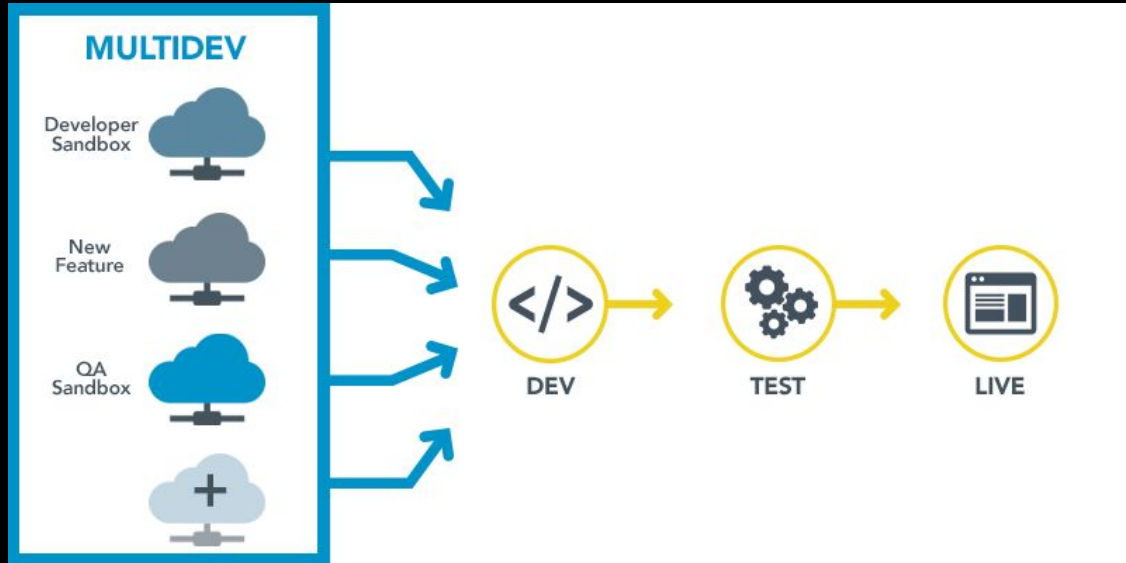


Slides at: [mcdwayne.com](http://mcdwayne.com)



# PANTHEON®

*Marketers and  
developers build, host,  
and manage their most  
important websites on  
Pantheon*



# Poll time





# "FINAL".doc



FINAL.doc!



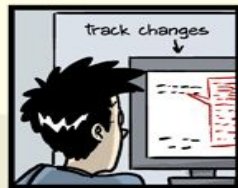
FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10. #@\$%WHYDID  
ICOMETOGRADSCHOOL?????.doc



Uploads DocV2.txt at 11:58



Uploads DocV2.txt at 11:59



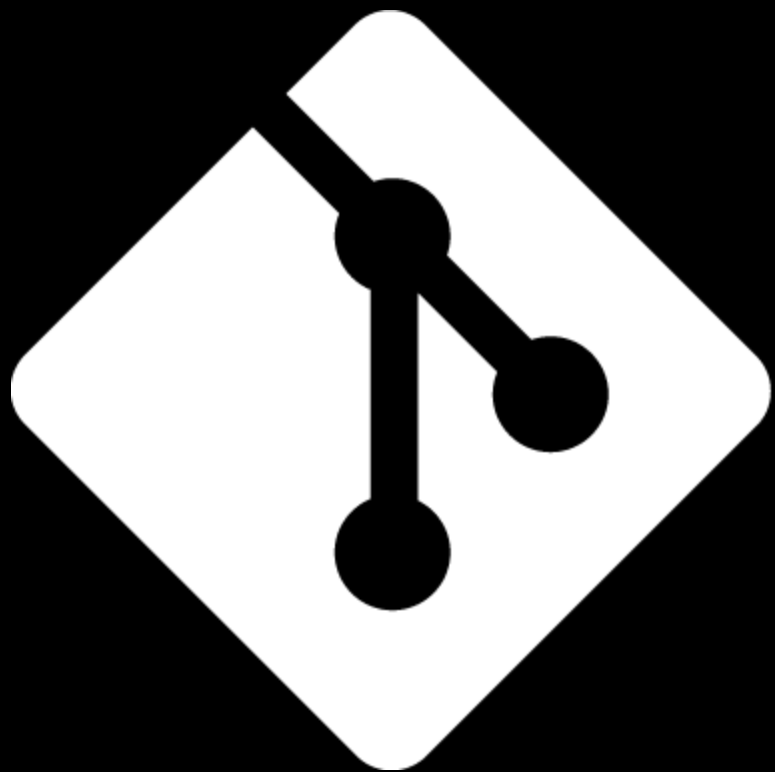






Command+Z is not a strategy

**WHAT?**



**git**

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

**NAME**

git - the stupid content tracker

**SYNOPSIS**

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>]
      <command> [<args>]
```

**DESCRIPTION**

Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

See **gittutorial(7)** to get started, then see **giteveryday(7)** for a useful minimum set of commands. The **Git User's Manual[1]** has a more in-depth introduction.

After you mastered the basic concepts, you can come back to this page



Created by Linus Torvalds in 2005 for development of the Linux kernel



Maintained since 2005 by Junio Hamano

# Knowledge Worker



Anyone working with items  
that will be saved, then  
modified and saved again

**NOT JUST FOR DEVS**





Basic idea:

The document is created

The document is saved

The document is edited

The changes are saved

Keep track of every time modifications are saved.

Include a history of the state of the document before and after the modification.

Provide you, the author or editor, with the opportunity to add a note explaining the nature of each change.

<https://premium.wpmudev.org/blog/git-for-wordpress-development/>

Really though:

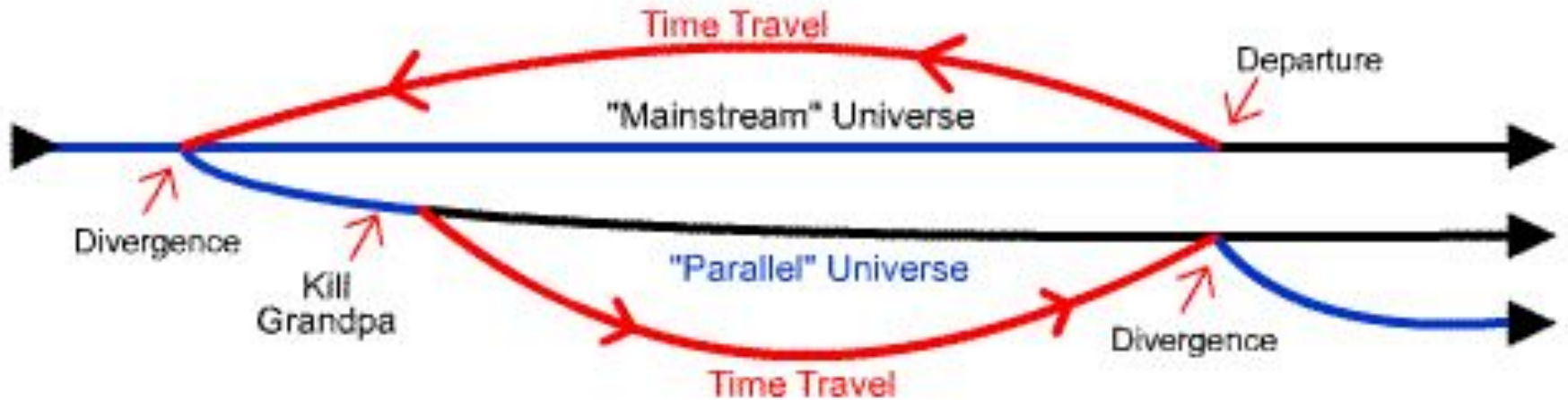
The project is created

The project is saved

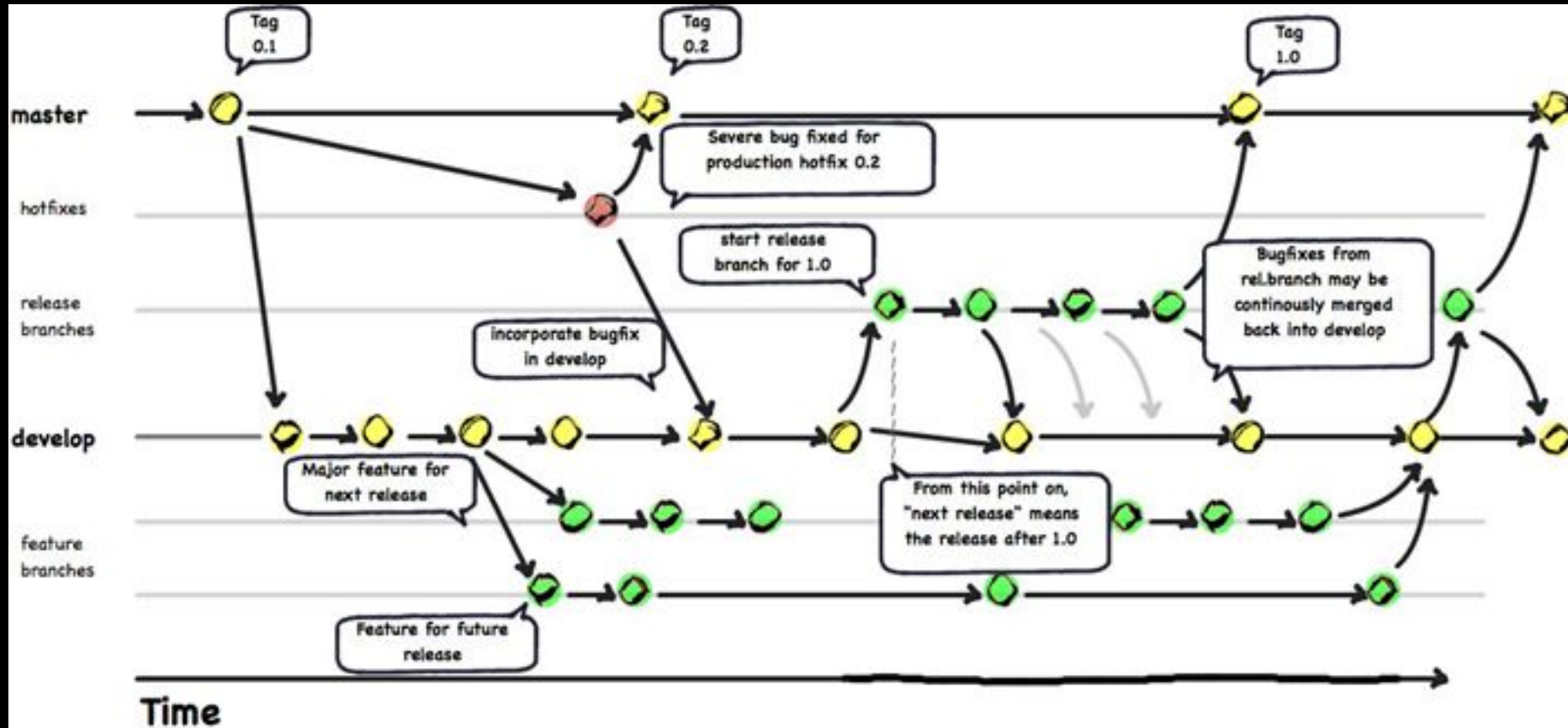
The project is edited

The project changes are saved

# The Basics Of ~~Time Travel~~ Git



# Graphical Based



Everything is local...

*Everything*

Git is lightweight

Git is NOT a backup tool



There are a LOT of community resources  
And very well documented



# Install Git

<https://git-scm.com>

[Download git for OSX](#)

[Download git for Windows](#)

[Download git for Linux](#)

# Git CLI

```
-bash
David@Crunchifier ~/scratch/test [master : +0 ~1 -0 | +1 ~1 -1]
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commit each, respectively.
(use "git pull" to merge the remote branch into yours)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   bar.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    deleted.txt
        modified:   modified.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        added.txt

David@Crunchifier ~/scratch/test [master : +0 ~1 -0 | +1 ~1 -1]
$
```

# Git GUI

**GitHub** Desktop



Atlassian  
**SourceTree**



# Create a Repo

Create a new repo locally

```
git init
```

```
~/Documents/demogitfolder $git init  
Initialized empty Git repository in /Users/dwaynemcdaniel/  
Documents/demogitfolder/.git/  
~/Documents/demogitfolder $
```

# Create a Repo

## Create a new repo locally

`git init`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ pwd  
/Users/dwaynemcdaniel/Documents/gitdemo  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ ls  
index.html      style.css      text1.txt      text2.txt      text3.txt  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ ls -a  
.               index.html    text1.txt      text3.txt  
..             style.css     text2.txt  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```















Q Search



10



0

Name	^	Date Modified	Size	Kind
▶  branches		Jul 6, 2017, 12:35 PM	--	Folder
 COMMIT_EDITMSG		Jul 18, 2017, 4:57 PM	16 bytes	TextEd...ument
 config		Jul 18, 2017, 4:54 PM	308 bytes	TextEd...ument
 description		Jul 6, 2017, 12:35 PM	73 bytes	TextEd...ument
 HEAD		Jul 6, 2017, 12:35 PM	23 bytes	TextEd...ument
▶  hooks		Jul 6, 2017, 12:35 PM	--	Folder
 index		Jul 18, 2017, 4:57 PM	10 KB	TextEd...ument
▶  info		Jul 6, 2017, 12:35 PM	--	Folder
▶  logs		Jul 6, 2017, 12:36 PM	--	Folder
▶  objects		Jul 18, 2017, 4:57 PM	--	Folder
 ORIG_HEAD		Jul 18, 2017, 4:57 PM	41 bytes	TextEd...ument
▶  refs		Jul 6, 2017, 12:39 PM	--	Folder



# See what is going on (you will use this often)

List which files are staged,  
unstaged, and untracked

`git status`

```
~/Documents/demogitfolder $git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file1.txt
        file2.txt
        file3.txt
        index.html
        pic1.png
        pic2.png
        pic3.png

nothing added to commit but untracked files present (use "git add" to track)
```

See what is going on (you will use this often)

List which files are staged,  
unstaged, and untracked

`git status`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ ls  
index.html      style.css      text1.txt      text2.txt      text3.txt  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

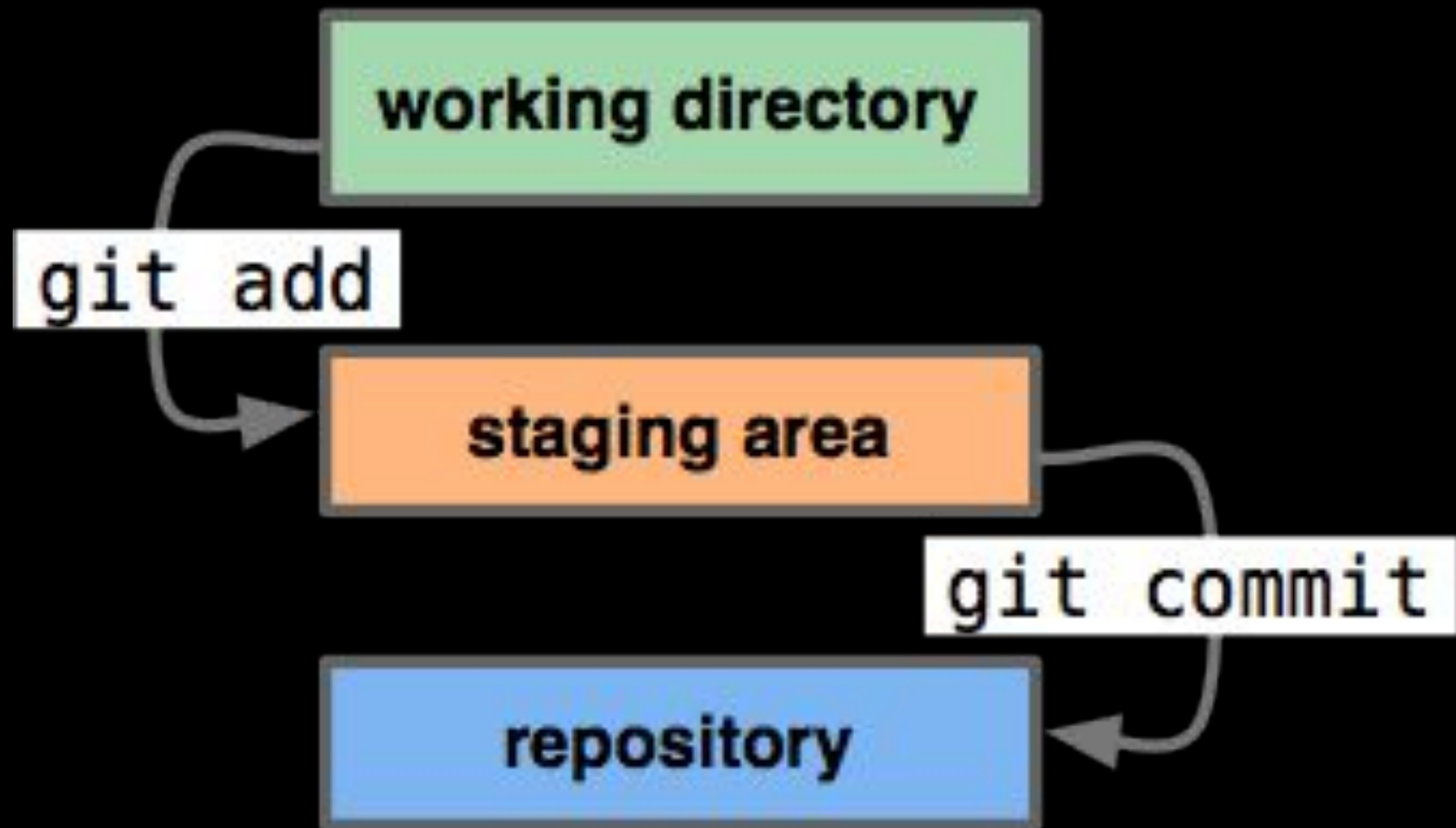
**working directory**

`git add`

**staging area**

`git commit`

**repository**



# Add files to the staging area

## Add individual files

`git add filename`

```
~/Documents/demogitfolder $git add file1.txt
```

```
~/Documents/demogitfolder $gs
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
    new file:   file1.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
file2.txt
```

```
file3.txt
```

```
index.html
```

```
pic1.png
```

```
pic2.png
```

```
pic3.png
```

# Add files to the staging area

## Add individual files

`git add filename`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ ls
index.html      style.css      text1.txt      text2.txt      text3.txt
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status
On branch master
```

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

```
index.html
style.css
text1.txt
text2.txt
text3.txt
```

nothing added to commit but untracked files present (use "git add" to track)

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# Add files to the staging area

## Add all the files

```
git add .
```

```
~/Documents/demogitfolder $git add .  
~/Documents/demogitfolder $git status  
On branch master
```

```
No commits yet
```

```
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)
```

```
new file:   file1.txt  
new file:   file2.txt  
new file:   file3.txt  
new file:   index.html  
new file:   pic1.png  
new file:   pic2.png  
new file:   pic3.png
```

# Add files to the staging area

## Add all the files

```
git add .
```

```
text1.txt  
text2.txt  
text3.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git add index.html  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status  
On branch master
```

Initial commit

Changes to be committed:  
(use "git rm --cached <file>..." to unstage)

```
new file:   index.html
```

Untracked files:  
(use "git add <file>..." to include in what will be committed)

```
style.css  
text1.txt  
text2.txt  
text3.txt
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# Commit things to the repository

Make an official record of the  
change in your history

```
git commit -m "message"
```

```
~/Documents/demogitfolder $git commit -m "intial commit fo  
r all files"  
[master (root-commit) 0d16e24] intial commit for all files  
7 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 file1.txt  
create mode 100644 file2.txt  
create mode 100644 file3.txt  
create mode 100644 index.html  
create mode 100644 pic1.png  
create mode 100644 pic2.png  
create mode 100644 pic3.png  
~/Documents/demogitfolder $
```



# Commit things to the repository

Make an official record of the  
change in your history

```
git commit -m "message"
```

```
Untracked files:
(use "git add <file>..." to include in what will be committed)
```

```
style.css
text1.txt
text2.txt
text3.txt
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git add .
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status
On branch master
```

Initial commit

```
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
```

```
new file:   index.html
new file:   style.css
new file:   text1.txt
new file:   text2.txt
new file:   text3.txt
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

## The vi editor - you will eventually run into it

If you don't use -m "message" you  
get dropped into vi editor

'i' to insert then esc  
' :wq' to write & quit

```
Added a new file4.txt
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   new file:   file4.txt
#
```

# The vi editor - you will eventually run into it

If you don't use -m "message" you  
get dropped into vi editor

'i' to insert then esc  
' :wq' to write & quit

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ su
```

# When you have modified a file, repeat add and commit

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status
On branch master
nothing to commit, working tree clean
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ s
```

# See all changes over time

Display entire commit history

`git log`

```
[~/Documents/demogitfolder $git log
commit 46bfc8ecaff57ab95a450baf48b3fe26441db87f (HEAD -> m
aster)
Author: mcdwayne <1dwayne.mcdaniel@gmail.com>
Date:    Sat Apr 6 11:24:23 2019 -0400

    Added file4.txt

commit 0d16e249fc8f676d2b13ff979b106f084d220ffb
Author: mcdwayne <1dwayne.mcdaniel@gmail.com>
Date:    Sat Apr 6 11:17:43 2019 -0400

    intial commit for all files
```

## See all changes over time

Display entire commit history

```
git log (--oneline)
```

```
~/Documents/demogitfolder $git log --oneline
3422148 (HEAD -> master) modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $
```

# See all changes over time

## Display entire commit history

`git log (--oneline)`

```
[~/Documents/kevinthulldotcom $gl  
5d17200 (HEAD -> master, origin/master) moved the donate button above the fold and added a suggestio  
n for people to make their own PRs if they are not Kevin Thull  
0eabc63 using local pic of kevin  
0bde664 Added a an optimized pic of kevin for performance  
4dd3d67 Added <html lang="en"> for aXe standards  
7c7c2f1 Delete CNAME  
6218df6 Create CNAME  
35f932e Merge pull request #2 from notkevinthull/patch-1  
e3bdd75 Add donate link  
3d559a3 Merge pull request #1 from mherchel/master  
5ba8607 Adding source documents.  
40fbdd4 gave mike credit for photo  
746dcc2 changed width  
378b8a6 added head and body  
0f788d2 initial commit of index of pic  
[~/Documents/kevinthulldotcom $git oull  
git: 'oull' is not a git command. See 'git --help'.  
  
The most similar command is  
    pull  
[~/Documents/kevinthulldotcom $git pull  
Already up to date.  
~/Documents/kevinthulldotcom $
```

# See all what changed in your working directory

## Show what changed where

`git diff`

```
~/Documents/demogitfolder $git diff
diff --git a/file1.txt b/file1.txt
index f30fab4..829305f 100644
--- a/file1.txt
+++ b/file1.txt
@@ -1,5 +1,5 @@
  This is the first line

-This is a second line
-
  And this is a 3rd line
+
+This is a second line
~/Documents/demogitfolder $
```



# See all what changed in your working directory

## Show what changed where

```
git diff
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ pwd  
/Users/dwaynemcdaniel/Documents/gitdemo  
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ █
```

**BACK**  
**TO THE FUTURE**

# Undoing things (safe)

Let's undo a specific commit

```
git revert <commit ID>
```

```
~/Documents/demogitfolder $git log --oneline
edb8895 (HEAD -> master) changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $git revert edb8895
[master 0dd125e] Revert "changed order in file1.txt"
 1 file changed, 2 insertions(+), 2 deletions(-)
~/Documents/demogitfolder $git log --oneline
0dd125e (HEAD -> master) Revert "changed order in file1.txt"
edb8895 changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $
```

# Undoing things (safe)

## Let's undo a specific commit

```
git revert <commit ID>
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status
On branch master
nothing to commit, working tree clean
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git log --oneline
f4c096a made a 4th change
9a7fe92 made a 3rd change
82389b4 made a send line to index.html
e8b1513 Hello World added to index.html
e788d07 initial commit, added 5 files
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# Undoing things (safe)

Let's undo a specific commit even  
reverts!

```
git revert <commit ID>
```

```
~/Documents/demogitfolder $git log --oneline
0dd125e (HEAD -> master) Revert "changed order in file1.txt"
edb8895 changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $git revert 0dd125e
[master 2eef0ef] Revert "Revert "changed order in file1.txt"
"
1 file changed, 2 insertions(+), 2 deletions(-)
~/Documents/demogitfolder $git log --oneline
2eef0ef (HEAD -> master) Revert "Revert "changed order in fi
le1.txt""
0dd125e Revert "changed order in file1.txt"
edb8895 changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $
```

# Undoing things (safe)

Let's undo a specific commit even  
reverts!

```
git revert <commit ID>
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git log --oneline
3f5b7a5 Revert "made a 4th change"
f4c096a made a 4th change
9a7fe92 made a 3rd change
82389b4 made a send line to index.html
e8b1513 Hello World added to index.html
e788d07 initial commit, added 5 files
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

I



## Undoing things (dangerous)

Let's go backwards in time and forget everything after that point

```
git reset --hard <commit ID>
```

```
~/Documents/demogitfolder $git log --oneline
2eef0ef (HEAD -> master) Revert "Revert "changed order in fi
le1.txt""
0dd125e Revert "changed order in file1.txt"
edb8895 changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $git reset --hard edb8895
HEAD is now at edb8895 changed order in file1.txt
~/Documents/demogitfolder $git log --oneline
fatal: unrecognized argument: --online
~/Documents/demogitfolder $git log --oneline
edb8895 (HEAD -> master) changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
```

# Undoing things (dangerous)

Let's go backwards in time and forget everything after that point

```
git reset --hard <commit ID>
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git log --oneline
```

```
3f5b7a5 Revert "made a 4th change"
```

```
f4c096a made a 4th change
```

```
9a7fe92 made a 3rd change
```

```
82389b4 made a send line to index.html
```

```
e8b1513 Hello World added to index.html
```

```
e788d07 initial commit, added 5 files
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git revert 3f5b7a5
```

```
[master 7c6126a] Revert "Revert "made a 4th change""
```

```
1 file changed, 3 insertions(+), 1 deletion(-)
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git log --oneline
```

```
7c6126a Revert "Revert "made a 4th change""
```

```
3f5b7a5 Revert "made a 4th change"
```

```
f4c096a made a 4th change
```

```
9a7fe92 made a 3rd change
```

```
82389b4 made a send line to index.html
```

```
e8b1513 Hello World added to index.html
```

```
e788d07 initial commit, added 5 files
```

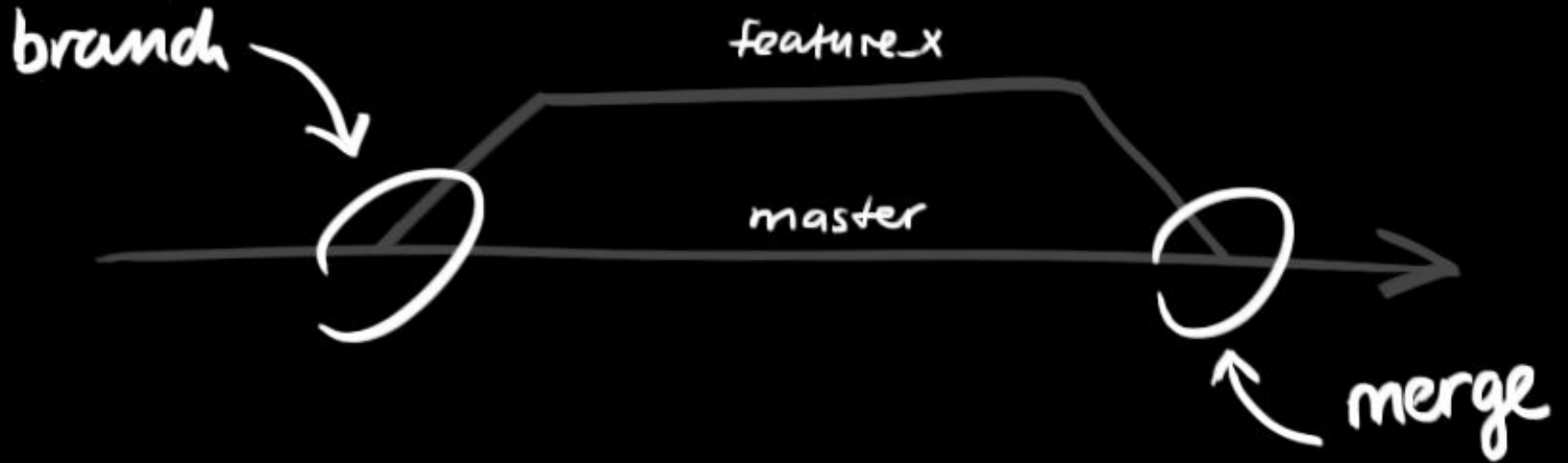
```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```



**Do YOU believe in  
Parallel Universes?**



# Working with branches



# Working with branches

See what exists and  
create a new one

```
git branch (newBranchName)
```

```
~/Documents/demogitfolder $git branch
* master
~/Documents/demogitfolder $git branch newbranch
~/Documents/demogitfolder $git branch
* master
  newbranch
~/Documents/demogitfolder $
```

# Working with branches

See what exists and  
create a new one

```
git branch (newBranchName)
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ █
```

# Working with branches

## Switch between branches

`git checkout branchname`

```
~/Documents/demogitfolder $git branch
* master
  newbranch
~/Documents/demogitfolder $git checkout newbranch
Switched to branch 'newbranch'
~/Documents/demogitfolder $git branch
  master
* newbranch
~/Documents/demogitfolder $
```

# Working with branches

## Switch between branches

`git checkout branchname`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git branch
* master
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git branch newbranch
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git branch
* master
  newbranch
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# Working with branches

Your changes in a branch are not reflected in 'master'

git branch

```
~/Documents/demogitfolder $git checkout newbranch
Switched to branch 'newbranch'
~/Documents/demogitfolder $git log --oneline
165cc84 (HEAD -> newbranch) mod to file1.txt and new file5.txt
eddb8895 (master) changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
~/Documents/demogitfolder $git checkout master
Switched to branch 'master'
~/Documents/demogitfolder $git log --oneline
eddb8895 (HEAD -> master) changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
```

# Working with branches

Your changes in a branch are not reflected in 'master'

`git branch`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ gl
9a7fe92 made a 3rd change
82389b4 made a send line to index.html
e8b1513 Hello World added to index.html
e788d07 initial commit, added 5 files
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git branch
  master
* newbranch
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ subl
```



# Working with merging

Merging branches, make sure you  
are on master first

```
git merge branchToMergeIn
```

```
~/Documents/demogitfolder $git branch
* master
  newbranch
~/Documents/demogitfolder $git merge newbranch
Updating edb8895..165cc84
Fast-forward
 file1.txt | 3 ---
 file5.txt | 0
2 files changed, 3 deletions(-)
create mode 100644 file5.txt
~/Documents/demogitfolder $git log --oneline
165cc84 (HEAD -> master, newbranch) mod to file1.txt and new
file5.txt
edb8895 changed order in file1.txt
3422148 modified file1.txt
46bfc8e Added file4.txt
0d16e24 intial commit for all files
```

# Working with merging

Merging branches, make sure you  
are on master first

```
git merge branchToMergeIn
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ █
```

# Merge Conflicts

This will happen sometimes  
Changes conflict, looks like this

```
git merge branchToMergeIn
```

```
~/Documents/demogitfolder $git branch
* master
  newbranch
~/Documents/demogitfolder $git merge newbranch
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
~/Documents/demogitfolder $cat file1.txt
<<<<<< HEAD
This is a new first line

and this is done in master
=====
This is a new 1st line

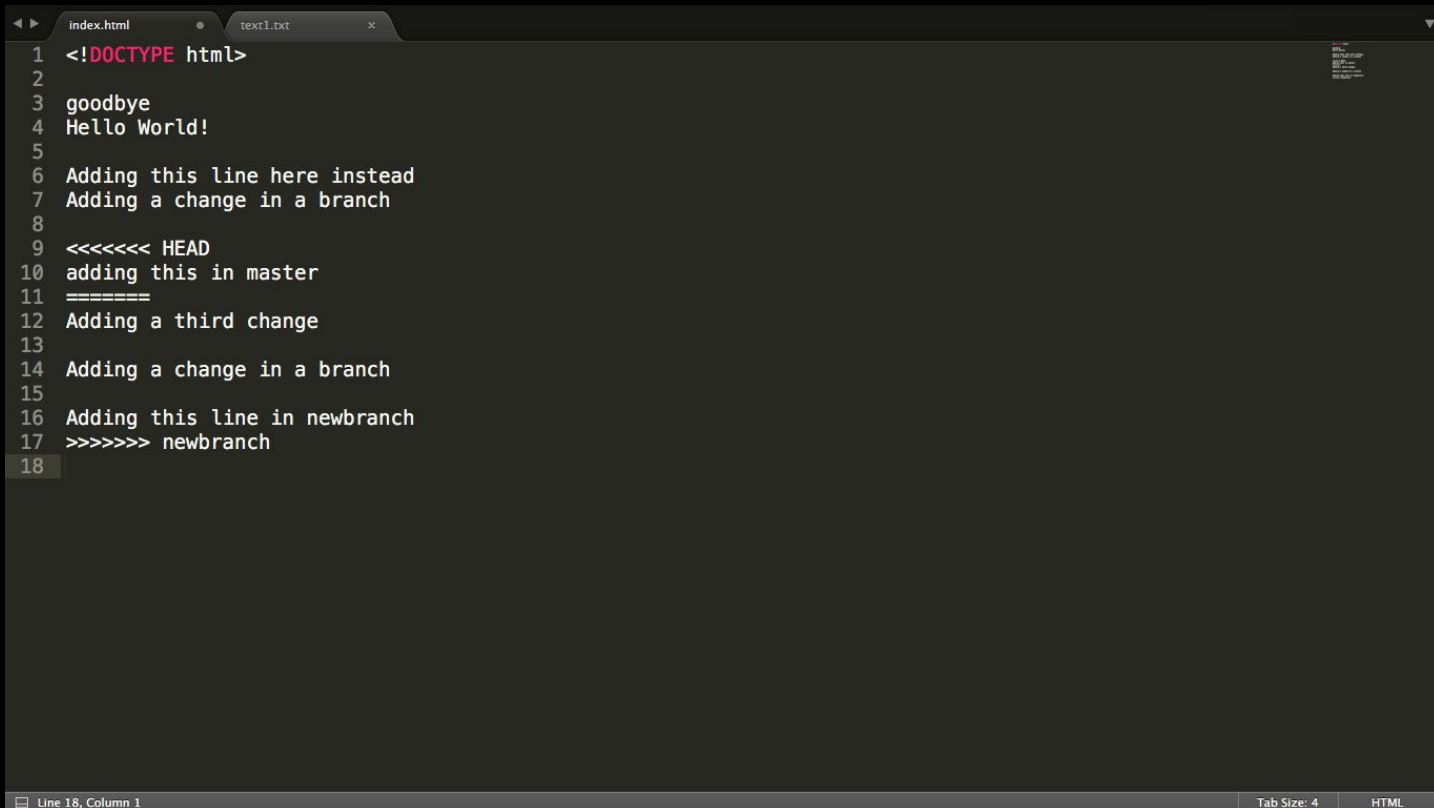
This was done in the newbranch
>>>>>> newbranch

And this is a 3rd line
```

# Merge Conflicts

Git tries to fix things and shows you  
where it went bad

`git merge`



```
1 <!DOCTYPE html>
2
3 goodbye
4 Hello World!
5
6 Adding this line here instead
7 Adding a change in a branch
8
9 <<<<<<<< HEAD
10 adding this in master
11 =====
12 Adding a third change
13
14 Adding a change in a branch
15
16 Adding this line in newbranch
17 >>>>>> newbranch
18
```

Line 18, Column 1      Tab Size: 4      HTML

# Merge Conflicts

Git tries to fix things and shows you  
where it went bad

`git merge`

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git branch
master
* newbranch
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ sublime index.html
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git add .
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git commit -m "added new line in newbranch"
[newbranch 9d520a0] added new line in newbranch
1 file changed, 3 insertions(+), 1 deletion(-)
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ sublime index.html
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git add .
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git commit -m "Added a line in master"
[master 939435a] Added a line in master
1 file changed, 2 insertions(+)
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git merge newbranch
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ :(
```

# Merge Conflicts

There are built tools you can have  
git run to help fix conflicts

`git mergetool`

```
[~/Documents/demogitfolder $git mergetool
```

```
This message is displayed because 'merge.tool' is not config  
ured.
```

```
See 'git mergetool --tool-help' or 'git help config' for mor  
e details.
```

```
'git mergetool' will now attempt to use one of the following  
tools:
```

```
opendiff tortoisemerge emerge vimdiff
```

```
Merging:
```

```
file1.txt
```

```
Normal merge conflict for 'file1.txt':
```

```
  {local}: modified file
```

```
  {remote}: modified file
```

```
Hit return to start merge resolution tool (opendiff): █
```

## Remote repositories



GitLab



# Remote repositories

GitHub, Inc. [US] <https://github.com/mcdwayne/gitdemorepo>



This repository

Search

Pull requests

Issues

Marketplace

Gist



mcdwayne / gitdemorepo

Unwatch

1

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

No description, website, or topics provided.

Edit

Add topics

4 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



mcdwayne made a 3rd change

Latest commit 9a7fe92 2 hours ago

index.html

made a 3rd change

2 hours ago

style.css

initial commit, added 5 files

2 hours ago

text1.txt

initial commit, added 5 files

2 hours ago

text2.txt

initial commit, added 5 files

2 hours ago

text3.txt

initial commit, added 5 files

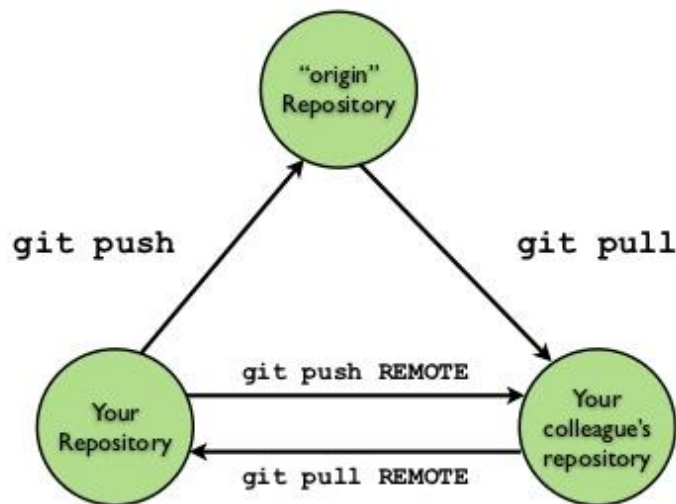
2 hours ago

Help people interested in this repository understand your project by adding a README.

Add a README



# The Git model

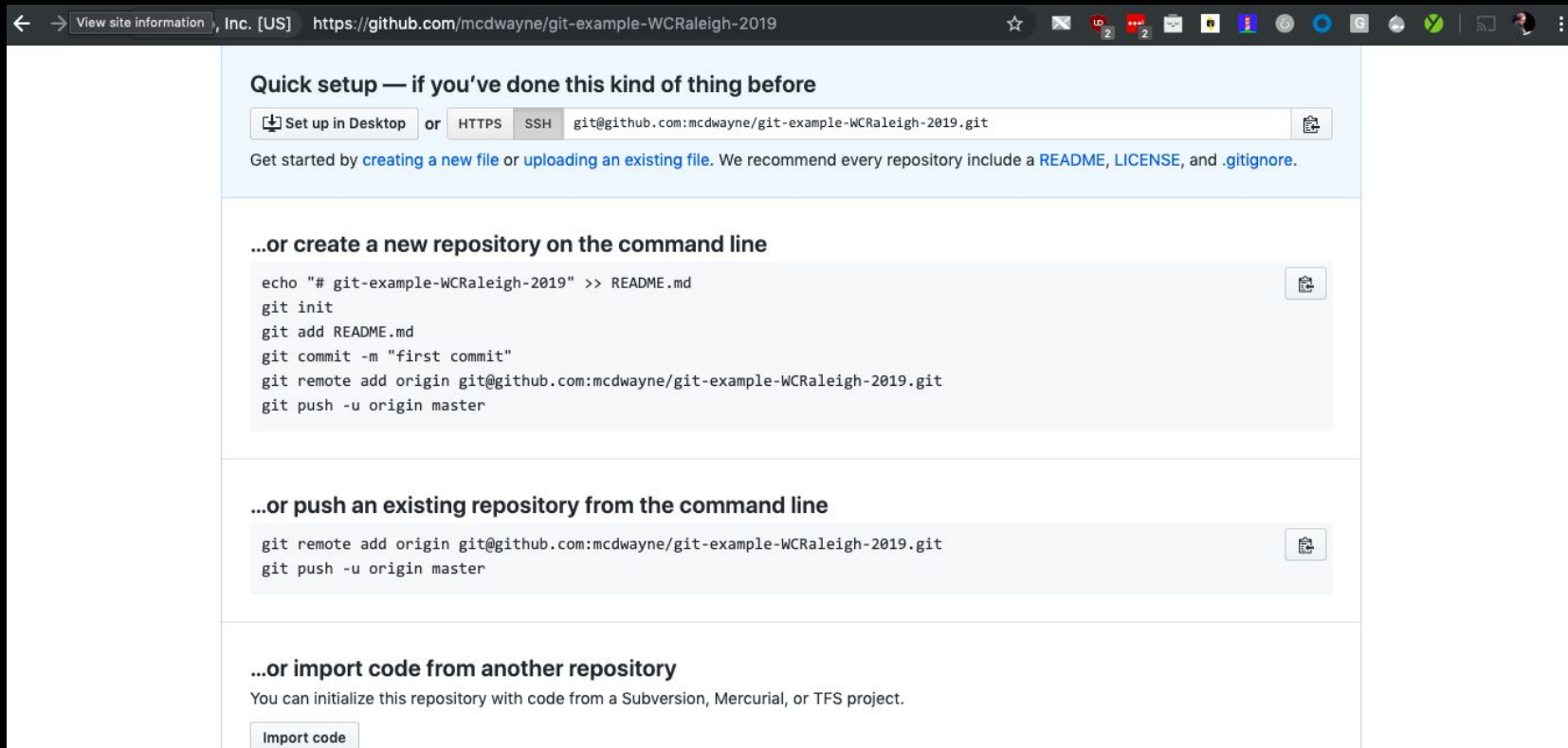


You can push/pull commits to any remote repository, there is no difference between server and client.

# Let's add a remote repository

Name the remote (origin) and  
address of the online repo

```
git remote add
```



The screenshot shows the GitHub 'Add new repository' page. At the top, there's a navigation bar with the GitHub logo and the text 'Add new repository'. Below this, there's a section titled 'Quick setup — if you've done this kind of thing before'. This section contains a form with a dropdown menu set to 'HTTPS', a text input field containing 'git@github.com:mcdwayne/git-example-WCRaleigh-2019.git', and a 'Copy' button. Below the form, there's a link to 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' Below this, there's a section titled '...or create a new repository on the command line' which contains a code block with the following commands: 

```
echo "# git-example-WCRaleigh-2019" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:mcdwayne/git-example-WCRaleigh-2019.git
git push -u origin master
```

 Below this, there's a section titled '...or push an existing repository from the command line' which contains a code block with the following commands: 

```
git remote add origin git@github.com:mcdwayne/git-example-WCRaleigh-2019.git
git push -u origin master
```

 At the bottom, there's a section titled '...or import code from another repository' with the text 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and an 'Import code' button.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:mcdwayne/git-example-WCRaleigh-2019.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# git-example-WCRaleigh-2019" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:mcdwayne/git-example-WCRaleigh-2019.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:mcdwayne/git-example-WCRaleigh-2019.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

## Let's add a remote repository

Name the remote (origin) and  
address of the online repo

```
git remote add
```

```
~/Documents/demogitfolder $git remote add origin git@github.  
com:mcdwayne/git-example-WCRaleigh-2019.git  
~/Documents/demogitfolder $git remote  
origin  
~/Documents/demogitfolder $
```

## Let's add a remote repository

Name the remote (origin) and  
address of the online repo

```
git remote add
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

## Let's push to a remote repository

Name the remote (origin) and name  
of branch (master)

```
git push
```

```
git push origin master
```

```
git push origin branch
```

```
git push other-remote branch
```

## Let's push to a remote repository

Name the remote (origin) and name  
of branch (master)

`git push`

```
~/Documents/demogitfolder $git push -u origin master
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (17/17), 1.51 KiB | 775.00 KiB/s, done
.
Total 17 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), done.
To github.com:mcdwayne/git-example-WCRaleigh-2019.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from
'origin'.
```

# Let's push to a remote repository

Name the remote (origin) and name of branch (master)

git push

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git remote add origin git@github.com:mcdwayne/gitdemorepo.git
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git remote -v
origin  git@github.com:mcdwayne/gitdemorepo.git (fetch)
origin  git@github.com:mcdwayne/gitdemorepo.git (push)
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# Let's pull changes into our repository

## Using Github's editor and pulling changes locally

Editing on Github

The screenshot shows the GitHub web editor interface. At the top, the browser address bar displays the URL: `https://github.com/mcdwayne/git-example-WCRaleigh-2019/edit/master/file2.txt`. Below the browser bar is the GitHub navigation bar with the search bar and links for Pull requests, Issues, Marketplace, and Explore. The main content area shows the repository `mcdwayne / git-example-WCRaleigh-2019` with options to Unwatch, Star (0), and Fork (0). A secondary navigation bar includes links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The file editor section shows the file `git-example-WCRaleigh-2019 / file2.txt` with a copy icon and the text "or cancel". The editor has two tabs: "Edit file" (active) and "Preview changes". On the right side of the editor, there are settings for "Spaces" (set to 2), "Soft wrap" (checked), and a line number "2". The code area contains two lines of text: "1 Added a first line on Github" and "2".



# Let's pull changes into our repository

Using Github's editor and pulling changes locally

`git pull`

```
~/Documents/demogitfolder $git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused
Unpacking objects: 100% (3/3), done.
From github.com:mcdwayne/git-example-WCRaleigh-2019
   7db279f..aadafa8  master    -> origin/master
Merge made by the 'recursive' strategy.
 file2.txt | 1 +
 1 file changed, 1 insertion(+)
~/Documents/demogitfolder $
```

# Let's pull changes into our repository

Using Github's editor and pulling  
changes locally

```
git pull
```

The screenshot shows the GitHub interface for a repository named 'mcdwayne / gitdemorepo'. At the top, there's a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. Below this, the repository name is displayed along with statistics: 1 star, 0 forks, and 0 watchers. The 'Code' tab is selected, showing a list of files: 'index.html', 'style.css', 'text1.txt', 'text2.txt', and 'text3.txt'. Each file has a commit history entry: 'mcdwayne made a 3rd change' for 'index.html' (2 hours ago) and 'Initial commit, added 5 files' for the others (3 hours ago). A 'Clone or download' button is visible. At the bottom, there's a footer with copyright information and links for 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

This repository

Search

Pull requests Issues Marketplace Gist

mcdwayne / gitdemorepo

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Edit

Add topics

4 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

mcdwayne made a 3rd change Latest commit 9a7fe92 2 hours ago

index.html	made a 3rd change	2 hours ago
style.css	Initial commit, added 5 files	3 hours ago
text1.txt	Initial commit, added 5 files	3 hours ago
text2.txt	Initial commit, added 5 files	3 hours ago
text3.txt	Initial commit, added 5 files	3 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

# Let's push local back up again

Name the remote (origin) and name of branch (master)

git push

```
~/Documents/demogitfolder $vi file2.txt
~/Documents/demogitfolder $git add .
~/Documents/demogitfolder $git commit -m "mod file2.txt locally"
[master 09cf8a5] mod file2.txt locally
 1 file changed, 2 insertions(+)
~/Documents/demogitfolder $git push origin master
Enumerating objects: 20, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 1.27 KiB | 1.27 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:mcdwayne/git-example-WCRaleigh-2019.git
   aadafa8..09cf8a5  master -> master
```

# Let's push local back up again

Name the remote (origin) and name of branch (master)

git push

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:mcdwayne/gitdemorepo
 * branch                master      -> FETCH_HEAD
    506e56a..ed4751a      master      -> origin/master
Updating 506e56a..ed4751a
Fast-forward
 text1.txt | 1 +
 1 file changed, 1 insertion(+)
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$
```

# We can clone a copy locally from any remote repository

We just need to specify where from  
and it creates the folder

`git clone`

```
~/Documents/demogitfolder $git clone git@github.com:mcdwayne
/mcdwayne-wp-cli-demos.git wpclidemo
Cloning into 'wpclidemo'...
remote: Enumerating objects: 187, done.
remote: Total 187 (delta 0), reused 0 (delta 0), pack-reused
187
Receiving objects: 100% (187/187), 42.50 KiB | 478.00 KiB/s,
done.
Resolving deltas: 100% (105/105), done.
~/Documents/demogitfolder $ls
file1.txt          file5.txt          pic3.png
file2.txt          index.html         wpclidemo
file3.txt          pic1.png
file4.txt          pic2.png
~/Documents/demogitfolder $
```

We can clone a copy locally from any remote repository

We just need to specify where from  
and it creates the folder

```
git clone
```

```
Dwaynes-MacBook-Air:gitdemo dwaynemcdaniel$ █
```

```
I
```

# Enter the Pull Request!

## This is is branching and collaboration in one concept



### Commit changes

Update index.html

Add an optional extended description...

- ☐ Commit directly to the `master` branch.
- ☒ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

# Enter the Pull Request!



## Branch

Develop features on a branch and create a pull request to get changes reviewed.



## Discuss

Discuss and approve code changes related to the pull request.

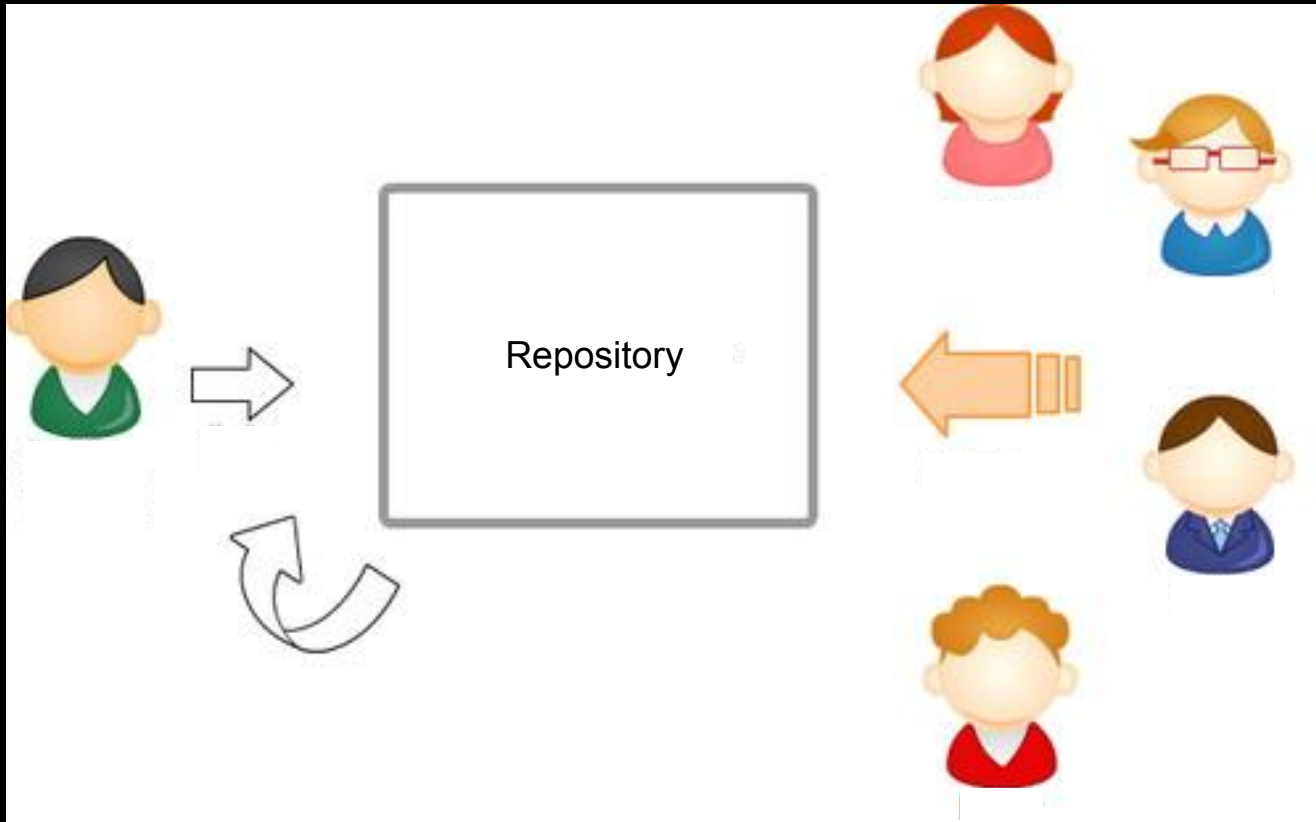


## Merge

Merge the branch with the click of a button.



**Anyone with push access to the repository can complete the merge.  
Normally this is assigned to the repo owner.**



# What I memorized

git init

git status

git add

git commit

git log

git diff

git revert

git reset

git branch

git checkout

git merge

git remote

git push

git pull

git clone

# For everything else!

Privacy, simplified. ▾



DuckDuckGo

The search engine that doesn't track you. [Learn More.](#)



# 100s of 'how to' guides and docs

Official Documents:

<https://git-scm.com/doc>

<https://guides.github.com/introduction/flow/>

Articles I Like:

<https://calderaforms.com/2015/09/who-should-be-using-version-control-for-wordpress-development/>

<https://premium.wpmudev.org/blog/git-for-wordpress-development/>

<https://www.sitepoint.com/wordpress-version-control-git/>

<http://www.makeuseof.com/tag/git-version-control-youre-developer/>

<https://blog.wppusher.com/demystifying-git-for-wordpress-developers/>

<https://wptavern.com/git-and-wordpress-3-tips-to-do-it-better>

Cheat Sheets!

<https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

<https://www.git-tower.com/blog/git-cheat-sheet/>

Fixing things:

<https://sethrobertson.github.io/GitFixUm/fixup.html>

<http://ohshitgit.com/>



THANK YOU!!!! I am Dwayne

[dwayne@pantheon.io](mailto:dwayne@pantheon.io)

@mcdwayne on Twitter

Some things I enjoy:

- Webcomics and Comic Books
- Crochet and learning knitting
- Karaoke!!!



Slides at: [mcdwayne.com](http://mcdwayne.com)

## Forward in time

```
git init  
git status  
git add  
git commit  
git log  
git diff
```

## Back in time

```
git revert  
git reset
```

## Parallel universes

```
git branch  
git checkout  
git merge
```

## Collaborating

```
git remote  
git push  
git pull  
git clone
```



[dwayne@pantheon.io](mailto:dwayne@pantheon.io)

@mcdwayne on [Twitter](#)

Slides at: [mcdwayne.com](http://mcdwayne.com)

# Thanks





## Forward in time

```
git init  
git status  
git add  
git commit  
git log  
git diff
```

## Back in time

```
git revert  
git reset
```

## Parallel universes

```
git branch  
git checkout  
git merge
```

## Collaborating

```
git remote  
git push  
git pull  
git clone
```



[dwayne@pantheon.io](mailto:dwayne@pantheon.io)

@mcdwayne on Twitter

Slides at: [mcdwayne.com](http://mcdwayne.com)