SPEAKER_NOTES.md - git

Speaker Notes for the 2016-10-17 Software Carpentry git lesson

TYPE ALL EXAMPLES AS YOU GO. THIS KEEPS THE SPEED SANE, AND ALLOWS YOU TO EXPLAIN EVERY STEP.

START SLIDES WITH `reveal-md slides.md --theme=white`

Version control with `git`

SLIDE (Version control with `git`)

- Build good software engineering **habits** (needs repetition and practice)
- **Transferable skill**: what's good for software also good for other documents

SLIDE (Etherpad)

- Share code snippets
- Ask/answer questions
- Communal note-taking

SLIDE (Let me tell you a story)

SLIDE (Once upon a time)

- Talk around slide

SLIDE (Advantage of version control)

- Talk around slide
- credit and blame?

SLIDE (Version control with `git`)

SLIDE (What lies ahead)

- Talk around slide
- Who has used `git`
- Who has heard about `git` but not used it?
- Who's heard horror stories about `git`?
- `git` actually elegant and powerful - some truth in jokes, though
- SWC focus more on *version control*, just that `git` is the tool we're using

SLIDE (Learning objectives)

- Talk around slide

SLIDE (Do you recognise this?)

- Talk around slide

- **Things get much more complicated when more people are involved**
- **Ask the audience**
- Who has been in that situation?
- Fundamental problems:
  - synchronising changes
  - tracking change times
- How do you get round it?
- MS Office: track changes
- Office 365: simultaneous edits?
- DropBox: now locks files in use
- Google Docs: simultaneous editing - but for code?
- **Version control systems are a solution to this**
- Excellent for code
- Useful for most activities
- **How version control works**
- Keeps only one document
- Saves records of all changes made to the document
- Who made the changes, and when
- Can reconstruct the entire editing history

SLIDE (How version control works)

- Talk around slide
- **for mathematicians!**
- can treat original document as a value (d)
- changes are then functions, acting on that value, e.g. f(d)
- functions can be compounded: g(f(d))
- **draw on whiteboard?**

SLIDE (Multiple editors - branching)

- Talk around slide
- **for mathematicians**
- f(d) != g(d)

SLIDE (Combining changes - merging)

- Talk around slide
- **for mathematicians**

- h(f(d), g(d))

SLIDE (What version control systems do)

- Talk around slide
- *commits* are not exactly *changes* - more like checkpoints

Setting up `git`

SLIDE (Setting up `git`)

SLIDE (Learning objectives)

- Talk around slide

SLIDE (Setting global options)

- Talk around slide
- **`git` command structure**
- `git <command> <options>`
- i.e. `git` **verb** *options*
- The name and email address will be associated with all our actions in `git`
- It's helpful to have some colour to interpret `git` output
- The `--global` flag means that every project on the computer will see these settings
- **Use your own name and email address!**

```
lpritc@Totoro:~$ git config --global user.name "Leighton Pritchard"
lpritc@Totoro:~$ git config --global user.email "leighton.pritchard@hutton.ac.uk"
lpritc@Totoro:~$ git config --global color.ui "auto"
lpritc@Totoro:~$ git config --global core.editor "nano -w"
(git config --global core.editor "'C:\Program Files (x86)\Notepad++\notepad++.exe' \
                                  -multiInst -notabbar -nosession -noPlugin")
```

- **Check settings at any time**
- You can reset these settings at any point
- check with `git config --list`

```
lpritc@Totoro:~$ git config --list
user.name=Leighton Pritchard
user.email=leighton.pritchard@hutton.ac.uk
push.default=simple
color.ui=auto
core.editor=emacs
```

Creating a repository

SLIDE (Creating a repository)

SLIDE (Learning objectives)

- Talk around slide

SLIDE (Creating a `git` repository)

- Talk around slide
- **Create a new directory to hold the project**
- Create it somewhere convenient for you

```
lpritc@Totoro:~$ cd
lpritc@Totoro:~$ mkdir planets
lpritc@Totoro:~$ cd planets
lpritc@Totoro:planets$
```

- point out that we are in `planets` - maybe do an `ls`
- **Make this directory a repository**

```
lpritc@Totoro:planets$ git init
Initialized empty Git repository in /Users/lpritc/planets/.git/
lpritc@Totoro:planets$ ls
lpritc@Totoro:planets$ ls -a
./    ../   .git/
```

- Nothing appears to have changed, at first
- Using `ls -a` shows the hidden `.git` subdirectory
- All information about the repository is stored in this subdirectory
- **Check all is well**
- Get a report on repository status
- We have not added or committed any files, so we get this statement
- Describe `master` branch - default

```
lpritc@Totoro:planets$ git status
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
```

Tracking changes

SLIDE (Tracking changes)

SLIDE (Learning objectives)

- Talk around slide

My first `git` commit

SLIDE (My first untracked file)

- Talk around slide
- **Create a file**
- We'll be discussing Mars' suitability as a base
- Add some suitable text and save the file

```
lpritc@Totoro:planets$ nano mars.txt
Cold and dry. Everything a nice colour. Matt Damon.
lpritc@Totoro:planets$ ls
mars.txt
lpritc@Totoro:planets$ cat mars.txt
Cold and dry. Everything a nice colour. Matt Damon.
```

- We now have a single file in this directory
- **Check repository status**
- Use `git status` to see what `git` thinks.

```
lpritc@Totoro:planets$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   mars.txt
nothing added to commit but untracked files present (use "git add" to track)
```

- It has noticed that the file exists
- Git says it is not 'tracking' the file - i.e. not recording changes

SLIDE (My first `git` commit)

- Talk around slide
- **Tell `git` to track the file**

```
lpritc@Totoro:planets$ git add mars.txt
lpritc@Totoro:planets$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#   new file:   mars.txt
#
```

- Git sees the new file as a 'change' that is not yet committed
- The changes are **STAGED**
- **Commit the file to the repository**
- Commiting the change to the repository stores it
- We add a message to keep notes on changes
  - short (<50 char) imperative one-liner
  - details in a second text block
- We commit *all* current changes simultaneously
- Note the short identifier for this commit

```
lpritc@Totoro:planets$ git commit -m "Start notes on Mars as a base"
[master (root-commit) d22195b] Start notes on Mars as a base
 1 file changed, 2 insertions(+)
 create mode 100644 mars.txt
lpritc@Totoro:planets$ git status
# On branch master
nothing to commit, working directory clean
```

- **Use `git log` to see the history**
- `git log` shows commits in reverse chronological order
- Note the full identifier for this commit
- The log message is also included
- Only the file exists in the current working directory - all the log information etc. is in `.git`

```
lpritc@Totoro:planets$ git log
commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:05:57 2016 +0000
    Start notes on Mars as a base
lpritc@Totoro:planets$ ls
mars.txt
```

SLIDE (The staging area)

- Talk around slide

modify-add-commit

SLIDE (modify-add-commit)

- Talk around slide
- **Make some changes to the file**

```
lpritc@Totoro:planets$ nano mars.txt
lpritc@Totoro:planets$ cat mars.txt
Cold and dry. Everything a nice colour. Matt Damon.
Two moons. This may be an issue for werewolves!
lpritc@Totoro:planets$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#   modified:   mars.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- `git` tracks `mars.txt` so sees the changes are made
- Changed file is "not staged for commit"
- **Inspect changes since last commit**
- `git diff` notes differences between the current version of the file in the working directory, and the latest in the repository

```
lpritc@Totoro:planets$ git diff
diff --git a/mars.txt b/mars.txt
index 87b4a99..96b2def 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,2 +1,2 @@
 Cold and dry. Everything a nice colour. Matt Damon.
-
+Two moons. This may be an issue for werewolves!
```

- Line 1: output is similar to `diff` command
- Line 2: unique identifiers for files being compared
- Lines 3,4: which versions are compared (a: repo; b: directory)

- The rest of the lines show what the differences are between the files (-: lines removed; +: lines added)
- **Commit the change**
- `git` won't let us commit without staging (with `git add`)
- We don't always want to commit all changes to all files at the same time

```
lpritc@Totoro:planets$ git commit -m "Lycanthropy concerns"
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#   modified:   mars.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- **Stage the changes, then commit**

```
lpritc@Totoro:planets$ git add mars.txt
lpritc@Totoro:planets$ git commit -m "add lycanthrope concerns"
[master ea59e91] add lycanthrope concerns
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- **Check the log**

```
lpritc@Totoro:planets$ git log
commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:18:55 2016 +0000
    add lycanthrope concerns
commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:05:57 2016 +0000
    start notes on Mars as a base
```

- **Stage another change**
- This time, we won't do `git diff` until after staging

```
lpritc@Totoro:planets$ nano mars.txt
lpritc@Totoro:planets$ cat mars.txt
Cold and dry. Everything a nice colour. Matt Damon.
Two moons. This may be an issue for werewolves!
Not much atmosphere, mind.
lpritc@Totoro:planets$ git add mars.txt
lpritc@Totoro:planets$ git diff
```

- The file has changed, but `git diff` doesn't show this
- **Use the `--staged` flag**

- We need `git diff --staged` to see staged file differences

```
lpritc@Totoro:planets$ git diff --staged
diff --git a/mars.txt b/mars.txt
index 96b2def..919cb89 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,2 +1,3 @@
 Cold and dry. Everything a nice colour. Matt Damon.
 Two moons. This may be an issue for werewolves!
+Not much atmosphere, mind.
```

- Commit the change

```
lpritc@Totoro:planets$ git commit -m "Climate issues"
[master 43faba5] Climate issues
 1 file changed, 1 insertion(+)
lpritc@Totoro:planets$ git log
commit 43faba5250cf67208f16e94f143a270f584a2e1b
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:27:29 2016 +0000
    Climate issues
commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:18:55 2016 +0000
    Lycanthropy concerns
commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:05:57 2016 +0000
    Start notes on Mars as a base
```

- Check repository status

```
lpritc@Totoro:planets$ git status
# On branch master
nothing to commit, working directory clean
```

SLIDE (Question)

1. would only commit if files were staged
2. would try to create a new repo
3. solution
4. would try to commit a file `"my recent changes"` with the message `myfile.txt`

SLIDE (Challenge 1)

Solution:

```
lpritc@Totoro:planets$ nano mars.txt
lpritc@Totoro:planets$ cat mars.txt
```

```
Cold and dry. Everything a nice colour. Matt Damon.
Two moons. This may be an issue for werewolves!
Not much atmosphere, mind.
A bit dusty. Bring a Hoover.
lpritc@Totoro:planets$ nano earth.txt
lpritc@Totoro:planets$ cat earth.txt
Mostly harmless
lpritc@Totoro:planets$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#   modified:   mars.txt
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   earth.txt
no changes added to commit (use "git add" and/or "git commit -a")
lpritc@Totoro:planets$ git add mars.txt
lpritc@Totoro:planets$ git add earth.txt
lpritc@Totoro:planets$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   earth.txt
#   modified:   mars.txt
#
lpritc@Totoro:planets$ git commit -m "Hoover needed. Ignore Earth."
[master 1f4de17] Hoover needed. Ignore Earth.
 2 files changed, 2 insertions(+)
 create mode 100644 earth.txt
lpritc@Totoro:planets$ git status
# On branch master
nothing to commit, working directory clean
lpritc@Totoro:planets$ git log
commit 1f4de176dc72c3ba0746e9e6a33b39f8f7e9fd75
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:34:09 2016 +0000
    Hoover needed. Ignore Earth.
commit 43faba5250cf67208f16e94f143a270f584a2e1b
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:27:29 2016 +0000
    Climate issues
```

```
commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:18:55 2016 +0000
    Lycanthropy concerns
commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:05:57 2016 +0000
    Start notes on Mars as a base
```

- To show files in a commit: `git show --pretty="" --name-only`

SLIDE (The Modify-Add-Commit lifecycle)

- Talk around slide

SLIDE (In which I predict the future)

- Talk around slide
- Can be difficult to think of commit messages
- Good practice: short messages < 50 chars, imperative
- Good practice: detail in longer paragraph(s) as secondary message

Exploring history

SLIDE (Exploring history)

SLIDE (Is history bunk?)

- Talk around slide

SLIDE (Learning objectives)

- Talk around slide

SLIDE (Commit history)

- Talk around slide

SLIDE (History with `git diff`)

- Talk around slide
- **See differences between named commits**

```
lpritc@Totoro:planets$ git diff HEAD~1 mars.txt
diff --git a/mars.txt b/mars.txt
index 919cb89..dfd5875 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,3 +1,4 @@
 Cold and dry. Everything a nice colour. Matt Damon.
 Two moons. This may be an issue for werewolves!
```

```
 Not much atmosphere, mind.
+A bit dusty. Bring a Hoover.
lpritc@Totoro:planets$ git diff HEAD~2 mars.txt
diff --git a/mars.txt b/mars.txt
index 96b2def..dfd5875 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,2 +1,4 @@
 Cold and dry. Everything a nice colour. Matt Damon.
 Two moons. This may be an issue for werewolves!
+Not much atmosphere, mind.
+A bit dusty. Bring a Hoover.
```

SLIDE (History with commit IDs)

- Talk around slide
- **Get IDs with `git log`**
- `git log --pretty=oneline`
- Use a long ID in same way as above
- We don't want to have to type out 40char IDs all the time
- Use two short IDs (different lengths)

```
lpritc@Totoro:planets$ git log
commit 1f4de176dc72c3ba0746e9e6a33b39f8f7e9fd75
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:34:09 2016 +0000
    Hoover needed. Ignore Earth.
commit 43faba5250cf67208f16e94f143a270f584a2e1b
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:27:29 2016 +0000
    Climate issues
commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:18:55 2016 +0000
    Lycanthropy concerns
commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Fri Jan 8 16:05:57 2016 +0000
    Start notes on Mars as a base
lpritc@Totoro:planets$ git diff d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3 mars.txt
diff --git a/mars.txt b/mars.txt
index 87b4a99..dfd5875 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,2 +1,4 @@
```

```
 Cold and dry. Everything a nice colour. Matt Damon.
-
+Two moons. This may be an issue for werewolves!
+Not much atmosphere, mind.
+A bit dusty. Bring a Hoover.
lpritc@Totoro:planets$ git diff d221 mars.txt
diff --git a/mars.txt b/mars.txt
index 87b4a99..dfd5875 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,2 +1,4 @@
 Cold and dry. Everything a nice colour. Matt Damon.
-
+Two moons. This may be an issue for werewolves!
+Not much atmosphere, mind.
+A bit dusty. Bring a Hoover.
lpritc@Totoro:planets$ git diff 43faba mars.txt
diff --git a/mars.txt b/mars.txt
index 919cb89..dfd5875 100644
--- a/mars.txt
+++ b/mars.txt
@@ -1,3 +1,4 @@
 Cold and dry. Everything a nice colour. Matt Damon.
 Two moons. This may be an issue for werewolves!
 Not much atmosphere, mind.
+A bit dusty. Bring a Hoover.
```

Restoring older versions

SLIDE (Restoring older versions)

- Talk around slide
- **Accidentally overwrite a file**

```
lpritc@Totoro:planets$ nano mars.txt
lpritc@Totoro:planets$ cat mars.txt
POTATOES!
```

- Once overwritten, `git status` says the changes were made, but not staged.

```
lpritc@Totoro:planets$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
```

```
#   modified:   mars.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- **Restore the last commit**
- Use `git checkout HEAD`
- No file named: `git` warns you
- Name the file, `git` thinks you mean it!

```
lpritc@Totoro:planets$ git checkout HEAD mars.txt
lpritc@Totoro:planets$ cat mars.txt
Cold and dry. Everything a nice colour. Matt Damon.
Two moons. This may be an issue for werewolves!
Not much atmosphere, mind.
A bit dusty. Bring a Hoover.
```

SLIDE (`git checkout`)

- Talk around slide

SLIDE (Question)

Solution: 2 & 4

1. will not overwrite (will warn)
2. solution
3. will overwrite `data-cruncher.py` with the change before last
4. solution

Ignoring things

SLIDE (Ignoring things)

SLIDE (Learning objectives)

- Talk around slide
- **Ask the audience** - which files would you ignore?

SLIDE (Not all files are useful)

- Talk around slide
- **Create dummy files**
- Create dummy files and subdirectory

```
lpritc@Totoro:planets$ mkdir results
lpritc@Totoro:planets$ touch a.dat b.dat c.dat results/a.out results/b.out
lpritc@Totoro:planets$ ls
a.dat      b.dat      c.dat      earth.txt  mars.txt   results/
lpritc@Totoro:planets$ ls results/
a.out  b.out
```

- These files are no use to us
- `git` says they're not being tracked
- `git` doesn't tell us about contents of untracked directories

```
lpritc@Totoro:planets$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   a.dat
#   b.dat
#   c.dat
#   results/
nothing added to commit but untracked files present (use "git add" to track)
```

SLIDE (`.gitignore`)

- Talk around slide
- **Create `.gitignore` file**
- Two patterns: `*.dat`, `results/` (all files below this point)

```
lpritc@Totoro:planets$ nano .gitignore
lpritc@Totoro:planets$ cat .gitignore
# Exclude all files ending in .dat
*.dat
# Exclude all files below results/
results/
lpritc@Totoro:planets$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   .gitignore
nothing added to commit but untracked files present (use "git add" to track)
```

- `git` now ignores the new files, but notices `.gitignore`
- **Commit `.gitignore`**

```
lpritc@Totoro:planets$ git add .gitignore
lpritc@Totoro:planets$ git commit -m "add .gitignore file"
[master f0232b1] Added .gitignore file
 1 file changed, 5 insertions(+)
 create mode 100644 .gitignore
lpritc@Totoro:planets$ git status
# On branch master
nothing to commit, working directory clean
```

- **List ignored files**

```
lpritc@Totoro:planets$ git status --ignored
# On branch master
# Ignored files:
#   (use "git add -f <file>..." to include in what will be committed)
#
#   a.dat
#   b.dat
#   c.dat
#   results/
nothing to commit, working directory clean
```

- **Add an ignored file**
- `git` will refuse
- Can override with `-f`, `--force`

```
lpritc@Totoro:planets$ git add b.dat
The following paths are ignored by one of your .gitignore files:
b.dat
Use -f if you really want to add them.
fatal: no files added
```

Remotes in GitHub

SLIDE (Remotes in GitHub)

SLIDE (Learning objectives)

- Talk around slide

SLIDE (Remote repositories)

- Talk around slide

SLIDE (Log in to GitHub)

- Talk around slide

SLIDE (Create a remote repository)

- **Create new repository**
- `+(create new) -> New Repository`
- Repository name: `planets`
- `Create repository`
- We then see a page of quick setup hints

SLIDE (A freshly-made GitHub repository)

- Talk around slide

SLIDE (Connecting local and remote repositories)

- Talk around slide

- **Get remote repository URL**
- Copy the `https://` link
- Add the remote repository (`git remote add origin`)

```
lpritc@Totoro:planets$ git remote add origin https://github.com/widdowquinn/planets.git
```

- Check it worked (`git remote -v`)

```
lpritc@Totoro:planets$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
```

- **Push to remote repo**
- `master` refers to the *branch* we're on

```
lpritc@Totoro:planets$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
Counting objects: 16, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 1.48 KiB | 0 bytes/s, done.
Total 16 (delta 2), reused 0 (delta 0)
To https://github.com/widdowquinn/planets.git
 * [new branch]      master -> master
```

- Show changes on GitHub site

SLIDE (Remote GitHub repo after first push)

- Talk around slide

SLIDE (My first remote *pull*)

- Talk around slide
- **Pull from remote repo**
- `git` confirms we're up to date

```
lpritc@Totoro:planets$ git pull origin master
From https://github.com/widdowquinn/planets
 * branch            master     -> FETCH_HEAD
Already up-to-date.
```

GitHub collaboration

SLIDE (GitHub collaboration)

SLIDE (Learning objectives)

- Talk around slide
- It's fine to use GitHub/other remote hosts as a personal store (even a private store)
- At some point you may want/need to collaborate - and this is requires a few more skills

SLIDE (Starting a collaboration)

- **Pair off learners**
- **Demonstrate giving a colleague access**
- `Settings -> Collaborators` - add your partner's username/ID
- `cd` to a new directory (e.g. `tmp/`)
- clone *your colleague's* repo
- Check the remote repo location with `git remote -v`

```
bash-3.2$ cd ../..
bash-3.2$ mkdir lesson_collaborator
bash-3.2$ cd lesson_collaborator
bash-3.2$ git clone https://github.com/widdowquinn/planets.git
Cloning into 'planets'...
remote: Counting objects: 16, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 16 (delta 3), reused 16 (delta 3), pack-reused 0
Unpacking objects: 100% (16/16), done.
Checking connectivity... done
bash-3.2$ cd planets
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
```

SLIDE (Make a collaborative change)

- Talk around slide
- **Make a new file**

```
bash-3.2$ cd planets
bash-3.2$ nano pluto.txt
bash-3.2$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   pluto.txt
nothing added to commit but untracked files present (use "git add" to track)
bash-3.2$ git diff
bash-3.2$ cat pluto.txt
```

```
Not really a planet. Lovely, and all, but not a planet.
```

- **Commit the file**

```
bash-3.2$ git add pluto.txt
bash-3.2$ git commit -m "add notes on Pluto"
[master a416b49] add notes on Pluto
 1 file changed, 1 insertion(+)
 create mode 100644 pluto.txt
bash-3.2$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 331 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To https://github.com/widdowquinn/planets.git
   b9cf6a3..a416b49  master -> master
bash-3.2$ git status
# On branch master
nothing to commit, working directory clean
```

- **Push the change to GitHub**

```
lpritc@Totoro:planets$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/widdowquinn/planets.git
   f0232b1..4907645  master -> master
```

- Check the change on your collaborator's repo.

SLIDE (Pull a collaborator's changes)

- Talk around slide
- **View the change on your own GitHub repo as *owner*.**
- Talk through changes on GitHub
- **Pull the new change to your repository as host**
- Change directory to your *own* repository
- Check it's the right one with `git remote -v`

```
bash-3.2$ cd ../../lesson_owner/planets/
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
bash-3.2$ git status
# On branch master
nothing to commit, working directory cleanclean
```

- `git status` does not show remote changes
- Sync with `git pull`

```
bash-3.2$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/widdowquinn/planets
 * branch            master     -> FETCH_HEAD
   b9cf6a3..a416b49  master     -> origin/master
Updating b9cf6a3..a416b49
Fast-forward
 pluto.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 pluto.txt
```

Resolving `git` conflicts

SLIDE (Resolving `git` conflicts)

SLIDE (Learning objectives)

- What do I do when my changes conflict with someone else's?
- Talk around slide

SLIDE (Why conflicts occur)

- Talk around slide
- **`git` encourages particular ways of working**
- modular structure - small files (reusability)
- small, incremental changes (reproducibility)
- commits should pass all tests!
- good planning
- interaction/communication

SLIDE (Seriously, `git push` when done)

SLIDE (Let's make a conflict)

- Talk around the slide
- **Create the owner conflict**
- Check you're in the right repo
- Add a line to `mars.txt`
- Commit and push

```
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
bash-3.2$ git status
# On branch master
nothing to commit, working directory clean
bash-3.2$ pwd
/Users/lpritc/lesson_owner/planets
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
bash-3.2$ nano mars.txt
bash-3.2$ cat mars.txt
Cold and dry. Everything a nice colour. Evidence of Matt Damon.
Two moons! This might be a problem for werewolves.
Not much atmosphere, mind. Windy.
Dusty. Bring a Hoover. Or a Dyson.
Definitely a planet. Not like that space rock, Pluto!
bash-3.2$ git add mars.txt
bash-3.2$ git commit -m "add slander against Pluto"
[master 9d2082c] add slander against Pluto
 1 file changed, 1 insertion(+)
bash-3.2$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 358 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/widdowquinn/planets.git
   a416b49..9d2082c  master -> master
```

- **Create the collaborator conflict**
- Change to collaborator repo

- Check with `git remote -v`
- Add a line to `mars.txt`
- Commit the change

```
bash-3.2$ cd ../../lesson_collaborator/planets/
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
bash-3.2$ nano mars.txt
bash-3.2$ cat mars.txt
Cold and dry. Everything a nice colour. Evidence of Matt Damon.
Two moons! This might be a problem for werewolves.
Not much atmosphere, mind. Windy.
Dusty. Bring a Hoover. Or a Dyson.
This is nearly as nice a planet as Pluto. Which is DEFINITELY A PLANET!!!
bash-3.2$ git add mars.txt
bash-3.2$ git commit -m "adds praise for Pluto"
[master 955bfca] adds praise for Pluto
 1 file changed, 1 insertion(+)
```

- **Push the change**
- As the collaborator, this should raise a conflict!

```
bash-3.2$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
To https://github.com/widdowquinn/planets.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/widdowquinn/planets.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

SLIDE (The conflict message)

- Talk around the slide
- local conflicts with remote, so you cannot push!
- advice from `git`: integrate remote with `git pull` before `push`ing again.

SLIDE (The conflicting changes)

- Talk around the slide
- **COMMUNICATION BETWEEN DEVELOPERS**

Resolving a conflict

SLIDE (Resolving a conflict)

- Talk around slide
- **Pull the remote changes**
- `git` tries to merge automatically, and will if it can
- If not, it marks the conflict in the affected file
- `git` tells us about the problem

```
bash-3.2$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/widdowquinn/planets
 * branch            master     -> FETCH_HEAD
   a416b49..9d2082c  master     -> origin/master
Auto-merging mars.txt
CONFLICT (content): Merge conflict in mars.txt
Automatic merge failed; fix conflicts and then commit the result.
```

- **Explain conflict syntax**
- The local change in `HEAD` is preceded by `<<<<<<<`
- Then there's a separator `=========`
- Then the remote change, followed by `>>>>>>>`
- We have to decide which change to keep (if either)

```
bash-3.2$ cat mars.txt
Cold and dry. Everything a nice colour. Evidence of Matt Damon.
Two moons! This might be a problem for werewolves.
Not much atmosphere, mind. Windy.
Dusty. Bring a Hoover. Or a Dyson.
<<<<<<< HEAD
This is nearly as nice a planet as Pluto. Which is DEFINITELY A PLANET!!!
=======
Definitely a planet. Not like that space rock, Pluto!
>>>>>>> 9d2082c3d0209af6b8e51c01993350f6014610d6
```

- **Edit the file to resolve the change**
- Until you add/commit, `git status` will warn about "unmerged paths"

```
bash-3.2$ nano mars.txt
bash-3.2$ cat mars.txt
Cold and dry. Everything a nice colour. Evidence of Matt Damon.
Two moons! This might be a problem for werewolves.
```

```
Not much atmosphere, mind. Windy.
Dusty. Bring a Hoover. Or a Dyson.
Definitely a planet. Not like that space rock, Pluto!
bash-3.2$ 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)
#
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#   both modified:      mars.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- **Add and commit, then push**

```
bash-3.2$ git add mars.txt
bash-3.2$ git commit -m "merged changes from GitHub"
[master 0581d44] merged changes from GitHub
bash-3.2$ git push origin master
Username for 'https://github.com': widdowquinn
Password for 'https://widdowquinn@github.com':
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 571 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/widdowquinn/planets.git
   9d2082c..0581d44  master -> master
```

- **Change back to your host repo, and pull**
- Check with `git remote -v`
- The conflict is resolved

```
bash-3.2$ cd ../../lesson_owner/planets
bash-3.2$ git remote -v
origin  https://github.com/widdowquinn/planets.git (fetch)
origin  https://github.com/widdowquinn/planets.git (push)
bash-3.2$ git pull origin master
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 2), reused 4 (delta 2), pack-reused 0
```

```
Unpacking objects: 100% (4/4), done.
From https://github.com/widdowquinn/planets
 * branch            master     -> FETCH_HEAD
   9d2082c..0581d44  master     -> origin/master
Updating 9d2082c..0581d44
Fast-forward
bash-3.2$ git log
commit 0581d443b61c2c433b9ab398188eac93b003c57a
Merge: 955bfca 9d2082c
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 15:50:54 2016 +0100

    merged changes from GitHub

commit 955bfca6492367f72f82f44816f979b88141dba4
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 15:44:26 2016 +0100

    adds praise for Pluto

commit 9d2082c3d0209af6b8e51c01993350f6014610d6
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 15:42:48 2016 +0100

    add slander against Pluto

commit a416b49acd10498530ed6efd74bd1261af88e306
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 15:35:55 2016 +0100

    add notes on Pluto

commit b9cf6a312eb223b09bd4dbacf7d8513c8e394fe1
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 15:12:54 2016 +0100

    add .gitignore file

commit 68d0ec8ed8cdff90cd468e26c2a3615638b1a0b3
Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
Date:   Sat Oct 15 14:54:11 2016 +0100

    add notes on Earth, and Mars cleaning

commit 8e848b150c647b576bb9333121264f940416bf01
bash-3.2$ git status
```

```
# On branch master
nothing to commit, working directory clean
bash-3.2$ cat mars.txt
Cold and dry. Everything a nice colour. Evidence of Matt Damon.
Two moons! This might be a problem for werewolves.
Not much atmosphere, mind. Windy.
Dusty. Bring a Hoover. Or a Dyson.
Definitely a planet. Not like that space rock, Pluto!
```

Wrapping up

- Talk around slide