

# SPEAKER\_NOTES.md - git

Speaker Notes for the 2016-01-11 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` )

**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?
  - How do you get round it?
  - MS Office: track changes
  - Office 365: simultaneous edits?
  - DropBox: now locks files in use
  - Google Docs: simultaneous editing
- **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

## SLIDE (Multiple editors - branching)

- Talk around slide

## SLIDE (Combining changes - merging)

- Talk around slide

## SLIDE (What version control systems do)

- Talk around slide

# 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`
  - 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!**

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

- **Check settings at any time**
  - You can reset these settings at any point

- check with `git config --list`

```
1 lpritch@Totoro:~$ git config --list
2 user.name=Leighton Pritchard
3 user.email=leighton.pritchard@hutton.ac.uk
4 push.default=simple
5 color.ui=auto
6 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

```
1 lpritch@Totoro:~$ cd
2 lpritch@Totoro:~$ mkdir planets
3 lpritch@Totoro:~$ cd planets
4 lpritch@Totoro:planets$
```

- **Make this directory a repository**

```
1 lpritch@Totoro:planets$ git init
2 Initialized empty Git repository in /Users/lpritch/planets/.git/
3 lpritch@Totoro:planets$ ls
4 lpritch@Totoro:planets$ ls -a
5 ./    ../    .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

```
1 | lpritc@Totoro:planets$ git status
2 | # On branch master
3 | #
4 | # Initial commit
5 | #
6 | 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
  - We now have a single file in this directory

```
1 | lpritc@Totoro:planets$ nano mars.txt
2 | Cold and dry. Everything a nice colour. Matt Damon.
3 | lpritc@Totoro:planets$ ls
4 | mars.txt
5 | lpritc@Totoro:planets$ cat mars.txt
6 | Cold and dry. Everything a nice colour. Matt Damon.
```

- **Check repository status**
  - Use `git status` to see what `git` thinks.
  - It has noticed that the file exists
  - Git says it is not 'tracking' the file - i.e. not recording changes

```

1 lprtc@Totoro:planets$ git status
2 # On branch master
3 #
4 # Initial commit
5 #
6 # Untracked files:
7 #   (use "git add <file>..." to include in what will be committed)
8 #
9 #   mars.txt
10 nothing added to commit but untracked files present (use "git add" to track)

```

**SLIDE** (My first `git` commit)

- Talk around slide
- Tell `git` to track the file
  - Git sees the new file as a 'change' that is not yet committed
  - The changes are **STAGED**

```

1 lprtc@Totoro:planets$ git add mars.txt
2 lprtc@Totoro:planets$ git status
3 # On branch master
4 #
5 # Initial commit
6 #
7 # Changes to be committed:
8 #   (use "git rm --cached <file>..." to unstage)
9 #
10 #   new file:   mars.txt
11 #

```

- **Commit the file to the repository**
  - Committing the change to the repository stores it
  - We add a message to keep notes on changes
  - We commit *all* current changes simultaneously
  - Note the short identifier for this commit

```

1 lprtc@Totoro:planets$ git commit -m "Start notes on Mars as a base"
2 [master (root-commit) d22195b] Start notes on Mars as a base
3   1 file changed, 2 insertions(+)
4   create mode 100644 mars.txt
5 lprtc@Totoro:planets$ git status
6 # On branch master
7 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`

```

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

```

## modify-add-commit

### SLIDE (modify-add-commit)

- Talk around slide
- **Make some changes to the file**
  - `git` tracks `mars.txt` so sees the changes are made
  - Changed file is "not staged for commit"

```

1 lpritch@Totoro:planets$ nano mars.txt
2 lpritch@Totoro:planets$ cat mars.txt
3 Cold and dry. Everything a nice colour. Matt Damon.
4 Two moons. This may be an issue for werewolves!
5 lpritch@Totoro:planets$ git status
6 # On branch master
7 # Changes not staged for commit:
8 #   (use "git add <file>..." to update what will be committed)
9 #   (use "git checkout -- <file>..." to discard changes in working directory)
10 #
11 #    modified:   mars.txt
12 #
13 no changes added to commit (use "git add" and/or "git commit -a")

```

- **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
  - 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)

```

1 lpritch@Totoro:planets$ git diff
2 diff --git a/mars.txt b/mars.txt
3 index 87b4a99..96b2def 100644
4 --- a/mars.txt
5 +++ b/mars.txt
6 @@ -1,2 +1,2 @@
7   Cold and dry. Everything a nice colour. Matt Damon.
8 -
9 +Two moons. This may be an issue for werewolves!

```

## • Commit the change

- `git` won't let us commit without staging
- We don't always want to commit all changes to all files at the same time

```

1 lpritch@Totoro:planets$ git commit -m "Lycanthropy concerns"
2 # On branch master
3 # Changes not staged for commit:
4 #   (use "git add <file>..." to update what will be committed)
5 #   (use "git checkout -- <file>..." to discard changes in working directory)
6 #
7 #   modified:   mars.txt
8 #
9 no changes added to commit (use "git add" and/or "git commit -a")

```

## • Stage the changes, then commit

```

1 lpritch@Totoro:planets$ git add mars.txt
2 lpritch@Totoro:planets$ git commit -m "Lycanthropy concerns"
3 [master ea59e91] Lycanthropy concerns
4 1 file changed, 1 insertion(+), 1 deletion(-)

```

## • Check the log

```

1 lpritch@Totoro:planets$ git log
2 commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
3 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
4 Date:   Fri Jan 8 16:18:55 2016 +0000
5     Lycanthropy concerns
6 commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
7 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
8 Date:   Fri Jan 8 16:05:57 2016 +0000
9     Start notes on Mars as a base

```

## • Stage another change

- This time, don't do `git diff` until after staging
- The file has changed, but `git diff` doesn't show this

```

1 lprtc@Totoro:planets$ nano mars.txt
2 lprtc@Totoro:planets$ cat mars.txt
3 Cold and dry. Everything a nice colour. Matt Damon.
4 Two moons. This may be an issue for werewolves!
5 Not much atmosphere, mind.
6 lprtc@Totoro:planets$ git add mars.txt
7 lprtc@Totoro:planets$ git diff

```

- Use the `--staged` flag

- We need `git diff --staged` to see staged file differences
- Commit the change

```

1 lprtc@Totoro:planets$ git diff --staged
2 diff --git a/mars.txt b/mars.txt
3 index 96b2def..919cb89 100644
4 --- a/mars.txt
5 +++ b/mars.txt
6 @@ -1,2 +1,3 @@
7   Cold and dry. Everything a nice colour. Matt Damon.
8   Two moons. This may be an issue for werewolves!
9 +Not much atmosphere, mind.
10 lprtc@Totoro:planets$ git commit -m "Climate issues"
11 [master 43faba5] Climate issues
12   1 file changed, 1 insertion(+)
13 lprtc@Totoro:planets$ git log
14 commit 43faba5250cf67208f16e94f143a270f584a2e1b
15 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
16 Date:   Fri Jan 8 16:27:29 2016 +0000
17     Climate issues
18 commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
19 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
20 Date:   Fri Jan 8 16:18:55 2016 +0000
21     Lycanthropy concerns
22 commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
23 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
24 Date:   Fri Jan 8 16:05:57 2016 +0000
25     Start notes on Mars as a base
26 lprtc@Totoro:planets$ git status
27 # On branch master
28 nothing to commit, working directory clean

```

## SLIDE (Challenge 1)

Solution:

```

1 lprtc@Totoro:planets$ nano mars.txt
2 lprtc@Totoro:planets$ cat mars.txt
3 Cold and dry. Everything a nice colour. Matt Damon.

```



```
4 Two moons. This may be an issue for werewolves!
5 Not much atmosphere, mind.
6 A bit dusty. Bring a Hoover.
7 lpritch@Totoro:planets$ nano earth.txt
8 lpritch@Totoro:planets$ cat earth.txt
9 Mostly harmless
10 lpritch@Totoro:planets$ git status
11 # On branch master
12 # Changes not staged for commit:
13 #   (use "git add <file>..." to update what will be committed)
14 #   (use "git checkout -- <file>..." to discard changes in working directory)
15 #
16 #    modified:   mars.txt
17 #
18 # Untracked files:
19 #   (use "git add <file>..." to include in what will be committed)
20 #
21 #   earth.txt
22 no changes added to commit (use "git add" and/or "git commit -a")
23 lpritch@Totoro:planets$ git add mars.txt
24 lpritch@Totoro:planets$ git add earth.txt
25 lpritch@Totoro:planets$ git status
26 # On branch master
27 # Changes to be committed:
28 #   (use "git reset HEAD <file>..." to unstage)
29 #
30 #   new file:   earth.txt
31 #   modified:   mars.txt
32 #
33 lpritch@Totoro:planets$ git commit -m "Hoover needed. Ignore Earth."
34 [master 1f4de17] Hoover needed. Ignore Earth.
35 2 files changed, 2 insertions(+)
36 create mode 100644 earth.txt
37 lpritch@Totoro:planets$ git status
38 # On branch master
39 nothing to commit, working directory clean
40 lpritch@Totoro:planets$ git log
41 commit 1f4de176dc72c3ba0746e9e6a33b39f8f7e9fd75
42 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
43 Date:   Fri Jan 8 16:34:09 2016 +0000
44     Hoover needed. Ignore Earth.
45 commit 43faba5250cf67208f16e94f143a270f584a2e1b
46 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
47 Date:   Fri Jan 8 16:27:29 2016 +0000
48     Climate issues
49 commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
50 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
51 Date:   Fri Jan 8 16:18:55 2016 +0000
52     Lycanthropy concerns
```

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

## Exploring history

**SLIDE** (Exploring history)

**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**

```
1 lprtc@Totoro:planets$ git diff HEAD~1 mars.txt
2 diff --git a/mars.txt b/mars.txt
3 index 919cb89..dfd5875 100644
4 --- a/mars.txt
5 +++ b/mars.txt
6 @@ -1,3 +1,4 @@
7   Cold and dry. Everything a nice colour. Matt Damon.
8   Two moons. This may be an issue for werewolves!
9   Not much atmosphere, mind.
10 +A bit dusty. Bring a Hoover.
11 lprtc@Totoro:planets$ git diff HEAD~2 mars.txt
12 diff --git a/mars.txt b/mars.txt
13 index 96b2def..dfd5875 100644
14 --- a/mars.txt
15 +++ b/mars.txt
16 @@ -1,2 +1,4 @@
17   Cold and dry. Everything a nice colour. Matt Damon.
18   Two moons. This may be an issue for werewolves!
19 +Not much atmosphere, mind.
20 +A bit dusty. Bring a Hoover.
```

**SLIDE** (History with commit IDs)

- Talk around slide

- Get IDs with `git log`

- 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)

```
1 lpritch@Totoro:planets$ git log
2 commit 1f4de176dc72c3ba0746e9e6a33b39f8f7e9fd75
3 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
4 Date:   Fri Jan 8 16:34:09 2016 +0000
5     Hoover needed. Ignore Earth.
6 commit 43faba5250cf67208f16e94f143a270f584a2e1b
7 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
8 Date:   Fri Jan 8 16:27:29 2016 +0000
9     Climate issues
10 commit ea59e9169633eb7f9083ee79c2cc7ebf20efaf98
11 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
12 Date:   Fri Jan 8 16:18:55 2016 +0000
13     Lycanthropy concerns
14 commit d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3
15 Author: Leighton Pritchard <leighton.pritchard@hutton.ac.uk>
16 Date:   Fri Jan 8 16:05:57 2016 +0000
17     Start notes on Mars as a base
18 lpritch@Totoro:planets$ git diff d22195b9ec3c8fb4c2ce0f52f344b95ce5d0d0e3 mars.txt
19 diff --git a/mars.txt b/mars.txt
20 index 87b4a99..dfd5875 100644
21 --- a/mars.txt
22 +++ b/mars.txt
23 @@ -1,2 +1,4 @@
24     Cold and dry. Everything a nice colour. Matt Damon.
25     -
26     +Two moons. This may be an issue for werewolves!
27     +Not much atmosphere, mind.
28     +A bit dusty. Bring a Hoover.
29 lpritch@Totoro:planets$ git diff d221 mars.txt
30 diff --git a/mars.txt b/mars.txt
31 index 87b4a99..dfd5875 100644
32 --- a/mars.txt
33 +++ b/mars.txt
34 @@ -1,2 +1,4 @@
35     Cold and dry. Everything a nice colour. Matt Damon.
36     -
37     +Two moons. This may be an issue for werewolves!
38     +Not much atmosphere, mind.
39     +A bit dusty. Bring a Hoover.
40 lpritch@Totoro:planets$ git diff 43faba mars.txt
41 diff --git a/mars.txt b/mars.txt
42 index 919cb89..dfd5875 100644
43 --- a/mars.txt
```

```
43 --- a/mars.txt
44 +++ b/mars.txt
45 @@ -1,3 +1,4 @@
46 Cold and dry. Everything a nice colour. Matt Damon.
47 Two moons. This may be an issue for werewolves!
48 Not much atmosphere, mind.
49 +A bit dusty. Bring a Hoover.
```

## Restoring older versions

### SLIDE (Restoring older versions)

- Talk around slide
- **Accidentally overwrite a file**
  - Once overwritten, `git status` says the changes were made, but not staged.

```
1 lprtc@Totoro:planets$ nano mars.txt
2 lprtc@Totoro:planets$ cat mars.txt
3 POTATOES!
4 lprtc@Totoro:planets$ git status
5 # On branch master
6 # Changes not staged for commit:
7 #   (use "git add <file>..." to update what will be committed)
8 #   (use "git checkout -- <file>..." to discard changes in working directory)
9 #
10 #    modified:   mars.txt
11 #
12 no changes added to commit (use "git add" and/or "git commit -a")
```

- **Restore the last commit**
  - Use `git checkout HEAD`

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

### SLIDE ( `git checkout` )

- Talk around slide

### SLIDE (Question)

## Ignoring things

---

**SLIDE** (Ignoring things)

**SLIDE** (Learning objectives)

- Talk around slide

**SLIDE** (Not all files are useful)

- Talk around slide
- **Create dummy files**
  - Create dummy files and subdirectory
  - These files are no use to us
  - `git` says they're not being tracked

```
1  lprtc@Totoro:planets$ mkdir results
2  lprtc@Totoro:planets$ touch a.dat b.dat c.dat results/a.out results/b.out
3  lprtc@Totoro:planets$ ls
4  a.dat      b.dat      c.dat      earth.txt  mars.txt   results/
5  lprtc@Totoro:planets$ ls results/
6  a.out  b.out
7  lprtc@Totoro:planets$ git status
8  # On branch master
9  # Untracked files:
10 #   (use "git add <file>..." to include in what will be committed)
11 #
12 #   a.dat
13 #   b.dat
14 #   c.dat
15 #   results/
16 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)
  - `git` now ignores the new files, but notices `.gitignore`

```

1 lprtc@Totoro:planets$ nano .gitignore
2 lprtc@Totoro:planets$ cat .gitignore
3 # Exclude all files ending in .dat
4 *.dat
5 # Exclude all files below results/
6 results/
7 lprtc@Totoro:planets$ git status
8 # On branch master
9 # Untracked files:
10 #   (use "git add <file>..." to include in what will be committed)
11 #
12 #   .gitignore
13 nothing added to commit but untracked files present (use "git add" to track)

```

### • Commit `.gitignore`

```

1 lprtc@Totoro:planets$ git add .gitignore
2 lprtc@Totoro:planets$ git commit -m "Added .gitignore file"
3 [master f0232b1] Added .gitignore file
4   1 file changed, 5 insertions(+)
5   create mode 100644 .gitignore
6 lprtc@Totoro:planets$ git status
7 # On branch master
8 nothing to commit, working directory clean

```

### • List ignored files

```

1 lprtc@Totoro:planets$ git status --ignored
2 # On branch master
3 # Ignored files:
4 #   (use "git add -f <file>..." to include in what will be committed)
5 #
6 #   a.dat
7 #   b.dat
8 #   c.dat
9 #   results/
10 nothing to commit, working directory clean

```

### • Add an ignored file

- `git` will refuse
- Can override with `-f`, `--force`

```

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

```

# Remotes in GitHub

---

**SLIDE** (Remotes in GitHub)

**SLIDE** (Learning objectives)

- Talk around slide

## Remote repositories

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

```
1 lprtc@Totoro:planets$ git remote add origin https://github.com/widdowquinn/planets.g
2 lprtc@Totoro:planets$ git remote -v
3 origin https://github.com/widdowquinn/planets.git (fetch)
4 origin https://github.com/widdowquinn/planets.git (push)
```

- **Push to remote repo**

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

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

**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

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

## GitHub collaboration

---

**SLIDE** (GitHub collaboration)

**SLIDE** (Learning objectives)

- Talk around slide

**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`

```
1 lprtc@Totoro:planets$ cd /tmp/
2 lprtc@Totoro:tmp$ git clone https://github.com/widdowquinn/planets.git
3 Cloning into 'planets'...
4 remote: Counting objects: 16, done.
5 remote: Compressing objects: 100% (10/10), done.
6 remote: Total 16 (delta 2), reused 16 (delta 2), pack-reused 0
7 Unpacking objects: 100% (16/16), done.
8 Checking connectivity... done
9 lprtc@Totoro:tmp$ ls planets/
10 earth.txt mars.txt
11 lprtc@Totoro:planets$ git remote -v
12 origin https://github.com/widdowquinn/planets.git (fetch)
13 origin https://github.com/widdowquinn/planets.git (push)
```

## SLIDE (Make a collaborative change)

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

```
1 lprtc@Totoro:tmp$ cd planets
2 lprtc@Totoro:planets$ nano pluto.txt
3 lprtc@Totoro:planets$ cat pluto.txt
4 Never a planet. Lovely, but not a planet.
```

- **Commit the file**

```
1 lprtc@Totoro:planets$ git add pluto.txt
2 lprtc@Totoro:planets$ git commit -m "Notes on Pluto"
3 [master 4907645] Notes on Pluto
4 1 file changed, 1 insertion(+)
5 create mode 100644 pluto.txt
```

- **Push the change to GitHub**

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

- **View the change on your own GitHub repo as \*host\***
  - Talk through changes on GitHub

**SLIDE** (Pull a collaborator's changes)

- Talk around slide
- **Pull the new change to your repository as host**
  - Change directory to your host repository
  - Check it's the right one with `git remote -v`
  - `git status` does not show remote changes
  - Sync with `git pull`

```
1 lprtc@Totoro:planets$ cd ~/planets/
2 lprtc@Totoro:planets$ git remote -v
3 origin https://github.com/widdowquinn/planets.git (fetch)
4 origin https://github.com/widdowquinn/planets.git (push)
5 lprtc@Totoro:planets$ git status
6 # On branch master
7 nothing to commit, working directory clean
8 lprtc@Totoro:planets$ git pull origin master
9 From https://github.com/widdowquinn/planets
10  * branch                master      -> FETCH_HEAD
11 Updating f0232b1..4907645
12 Fast-forward
13  pluto.txt | 1 +
14  1 file changed, 1 insertion(+)
15  create mode 100644 pluto.txt
```

## Resolving `git` conflicts

**SLIDE** (Resolving `git` conflicts)

**SLIDE** (Learning objectives)

- Talk around slide

**SLIDE** (Why conflicts occur)

- Talk around slide
- `git` encourages ways of working
  - modular structure - small files (reusability)
  - small, incremental changes (reproducibility)

- good planning
- interaction/communication

## Creating a conflict

**SLIDE** (Let's make a conflict)

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

```
1 lprtc@Totoro:planets$ git remote -v
2 origin https://github.com/widdowquinn/planets.git (fetch)
3 origin https://github.com/widdowquinn/planets.git (push)
4 lprtc@Totoro:planets$ nano mars.txt
5 lprtc@Totoro:planets$ git add mars.txt
6 lprtc@Totoro:planets$ git commit -m "Slandering Pluto"
7 [master d83cb5a] Slandering Pluto
8 1 file changed, 1 insertion(+)
9 lprtc@Totoro:planets$ git push origin master
10 Username for 'https://github.com': widdowquinn
11 Password for 'https://widdowquinn@github.com':
12 Counting objects: 5, done.
13 Delta compression using up to 4 threads.
14 Compressing objects: 100% (3/3), done.
15 Writing objects: 100% (3/3), 341 bytes | 0 bytes/s, done.
16 Total 3 (delta 2), reused 0 (delta 0)
17 To https://github.com/widdowquinn/planets.git
18 4907645..d83cb5a master -> master
```

- **Create the collaborator conflict**
  - Change to collaborator repo
  - Check with `git remote -v`
  - Add a line to `mars.txt`
  - Commit the change

```
1 lprtc@Totoro:planets$ git remote -v
2 origin https://github.com/widdowquinn/planets.git (fetch)
3 origin https://github.com/widdowquinn/planets.git (push)
4 lprtc@Totoro:planets$ cd /tmp/planets/
5 lprtc@Totoro:planets$ nano mars.txt
6 lprtc@Totoro:planets$ git add mars.txt
7 lprtc@Totoro:planets$ git commit -m "Praising Pluto"
8 [master 60f892f] Praising Pluto
9 1 file changed, 1 insertion(+)
```

- **Push the change**

- This should raise a conflict!

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

**SLIDE** (The conflict message)

- Talk around the slide

**SLIDE** (The conflicting changes)

- Talk around the slide

## 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

```

1 lprtc@Totoro:planets$ git pull origin master
2 remote: Counting objects: 3, done.
3 remote: Compressing objects: 100% (1/1), done.
4 remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
5 Unpacking objects: 100% (3/3), done.
6 From https://github.com/widdowquinn/planets
7   * branch                master       -> FETCH_HEAD
8     4907645..d83cb5a      master       -> origin/master
9 Auto-merging mars.txt
10 CONFLICT (content): Merge conflict in mars.txt
11 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)

```

1 lprtc@Totoro:planets$ cat mars.txt
2 Cold and dry. Everything a nice colour. Matt Damon.
3 Two moons. This may be an issue for werewolves!
4 Not much atmosphere, mind.
5 A bit dusty. Bring a Hoover.
6 <<<<<<< HEAD
7 I wish Pluto was a planet, like this one.
8 =====
9 But at least it's a planet, unlike Pluto.
10 >>>>>>> d83cb5a5ef115c4718a685121182c27a874143a8

```

- **Edit the file to resolve the change**

- Until you add/commit, `git status` will warn about "unmerged paths"

```

1 lprtc@Totoro:planets$ nano mars.txt
2 lprtc@Totoro:planets$ cat mars.txt
3 Cold and dry. Everything a nice colour. Matt Damon.
4 Two moons. This may be an issue for werewolves!
5 Not much atmosphere, mind.
6 A bit dusty. Bring a Hoover.
7 But at least it's a planet, unlike Pluto. Which is definitely interesting.
8 lprtc@Totoro:planets$ git status
9 # On branch master
10 # Your branch and 'origin/master' have diverged,
11 # and have 1 and 1 different commit each, respectively.
12 # (use "git pull" to merge the remote branch into yours)
13 #
14 # You have unmerged paths.
15 # (fix conflicts and run "git commit")
16 #
17 # Unmerged paths:
18 # (use "git add <file>..." to mark resolution)
19 #
20 # both modified: mars.txt
21 #
22 no changes added to commit (use "git add" and/or "git commit -a")

```

- **Add and commit, then push**

```

1 lprtc@Totoro:planets$ git add mars.txt
2 lprtc@Totoro:planets$ git commit -m "merged changes from GitHub"
3 [master 665ed3f] merged changes from GitHub
4 lprtc@Totoro:planets$ git status
5 # On branch master
6 # Your branch is ahead of 'origin/master' by 2 commits.
7 # (use "git push" to publish your local commits)
8 #
9 nothing to commit, working directory clean
10 lprtc@Totoro:planets$ git push origin master
11 Username for 'https://github.com': widdowquinn
12 Password for 'https://widdowquinn@github.com':
13 Counting objects: 10, done.
14 Delta compression using up to 4 threads.
15 Compressing objects: 100% (6/6), done.
16 Writing objects: 100% (6/6), 671 bytes | 0 bytes/s, done.
17 Total 6 (delta 4), reused 0 (delta 0)
18 To https://github.com/widdowquinn/planets.git
19 d83cb5a..665ed3f master -> master

```

- **Change back to your host repo, and pull**

- Check with `git remote -v`

- The conflict is resolved

```
1 lprtc@Totoro:planets$ cd ~/planets/
2 lprtc@Totoro:planets$ git remote -v
3 origin https://github.com/widdowquinn/planets.git (fetch)
4 origin https://github.com/widdowquinn/planets.git (push)
5 lprtc@Totoro:planets$ git pull origin master
6 remote: Counting objects: 6, done.
7 remote: Compressing objects: 100% (2/2), done.
8 remote: Total 6 (delta 4), reused 6 (delta 4), pack-reused 0
9 Unpacking objects: 100% (6/6), done.
10 From https://github.com/widdowquinn/planets
11  * branch          master      -> FETCH_HEAD
12   d83cb5a..665ed3f  master      -> origin/master
13 Updating d83cb5a..665ed3f
14 Fast-forward
15  mars.txt | 2 +-
16  1 file changed, 1 insertion(+), 1 deletion(-)
17 lprtc@Totoro:planets$ cat mars.txt
18 Cold and dry. Everything a nice colour. Matt Damon.
19 Two moons. This may be an issue for werewolves!
20 Not much atmosphere, mind.
21 A bit dusty. Bring a Hoover.
22 But at least it's a planet, unlike Pluto. Which is definitely interesting.
```

## Wrapping up

---

- Talk around slide