Quiz

1. Getting setup with Git

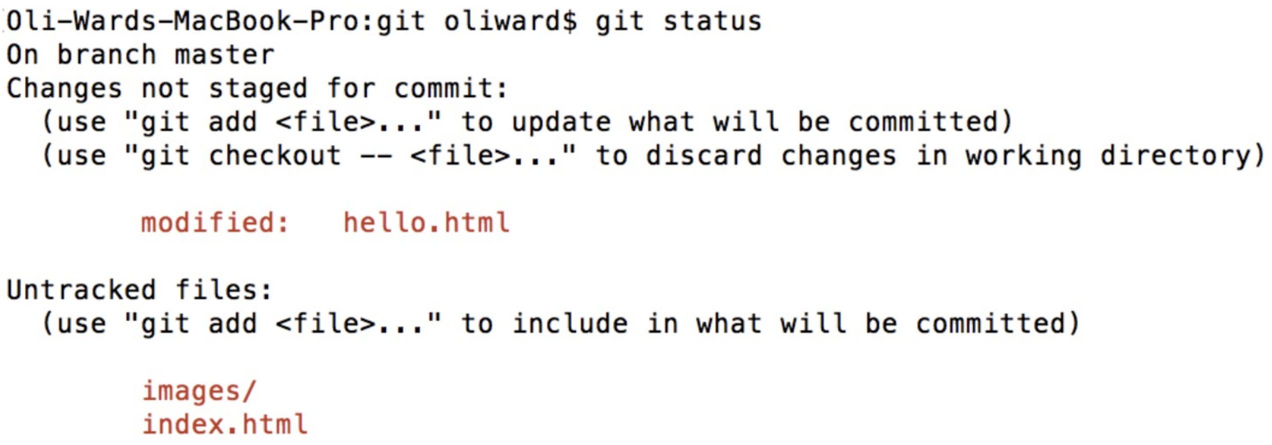
What does this command do?

```
$ git remote add origin git@github.com:develop-me/drupal1.git
```

This sets up a relationship between an (existing) local repository and the remote repository at the given origin URL.

2. Committing files

How would I stage the images folder and index.html for a commit with message "Apple" and then commit hello.html with message "Banana"?

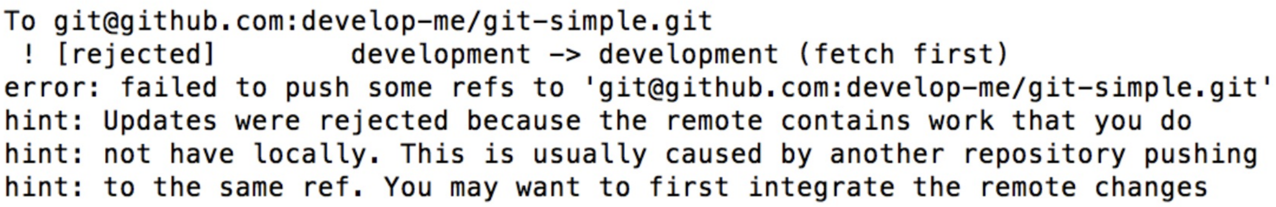


```
Oli-Wards-MacBook-Pro:git oliward$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   hello.html

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

        images/
        index.html
```

```
# add untracked file and folder
$ git add images/ index.html
# commit the currently staged items
$ git commit -m "Apple"

# commit the latest version hello.html
$ git commit -am "Banana"
```

3. Oh noes no.1

What command have I just run and what has gone wrong? What should I do next?

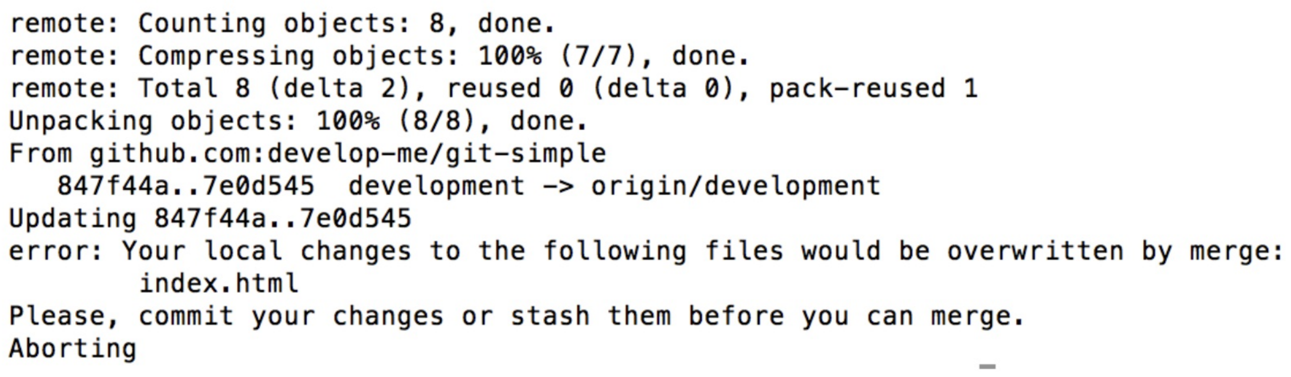


```
To git@github.com:develop-me/git-simple.git
 ! [rejected]        development -> development (fetch first)
error: failed to push some refs to 'git@github.com:develop-me/git-simple.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
```

You've just done a git push from a local branch, but the remote branch has work that your local branch is missing. You need to run a `git pull`. Then run the `git push` again.

4. Oh noes no.2

What command have I just run and what has gone wrong? What should I do next?

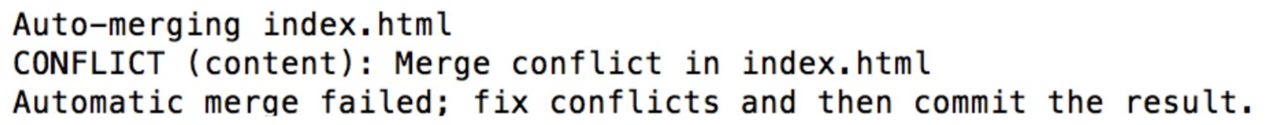


```
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 2), reused 0 (delta 0), pack-reused 1
Unpacking objects: 100% (8/8), done.
From github.com:develop-me/git-simple
   847f44a..7e0d545  development -> origin/development
Updating 847f44a..7e0d545
error: Your local changes to the following files would be overwritten by merge:
        index.html
Please, commit your changes or stash them before you can merge.
Aborting
```

You've tried to run a `git pull` without committing recent changes made on your local repo. Commit your changes, then run `git pull` again.

5. Oh noes no.3

What command have I just run and what has gone wrong? What should I do next?



```
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

You've just run `git pull` but your changes conflict with the remote changes for the same lines of code, and Git doesn't know which changes to keep or discard. You need to resolve the conflict manually in your text editor. Then commit the resulting clean files.

6. Burn it all!

How do you 'throw away' all work since last commit, and revert to how the files were at that moment?

```
$ git reset --hard HEAD
```

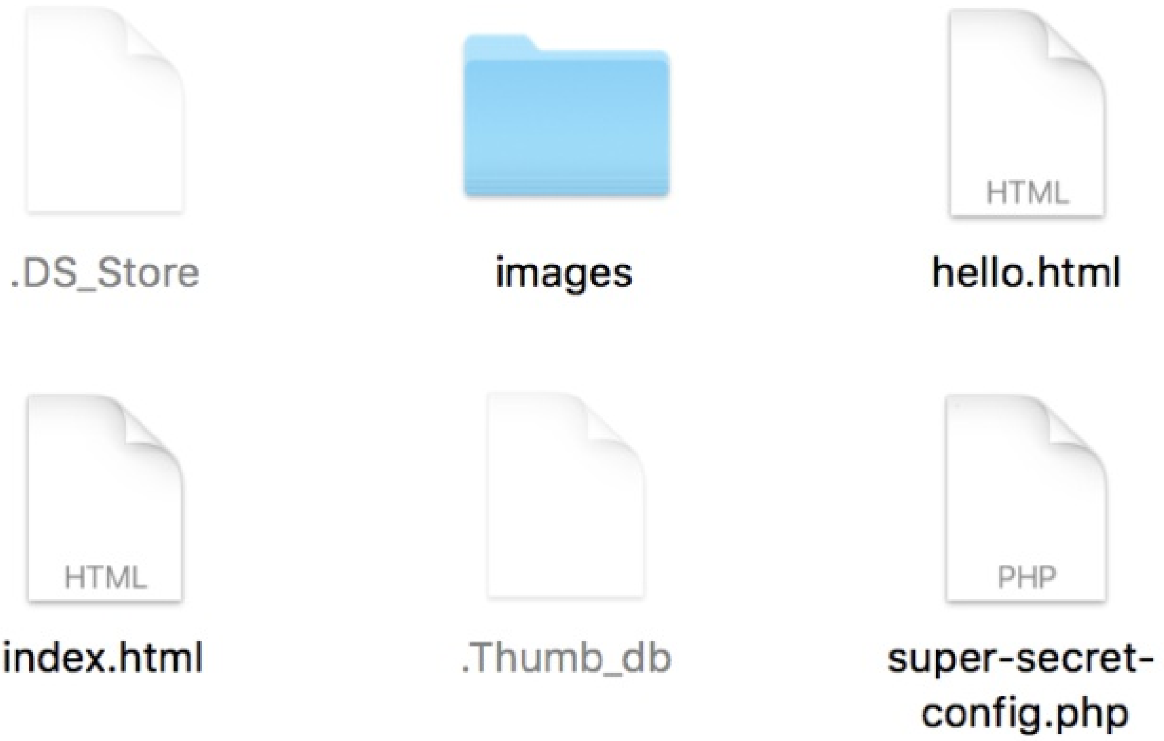
7. (Sort of) Burn it all!

How do you 'throw away' the last commit (bbbbbb), going back to the commit before last (aaaaaa), but keep the files as they are (in commit bbbbbb), to not lose that work.

```
$ git reset HEAD~
```

8. Ignore it all!

Create a .gitignore file that will cause Git to only track hello.html in my folder:



```
# remove currently tracked files from tracking
$ git rm --cached images/ .DS_Store index.html .Thumb_db super-secret-config.php
$ touch .gitignore
$ nano .gitignore
# Add above files to .gitignore in nano editor
```

9. Describe the process

You're working in a team that uses Gitflow. Document the steps, and git commands, from starting a new piece of work, to submitting it as a pull request.

```
# Clone remote repo into local folder
$ git clone {repository URL} {folder to create}
# Checkout the development branch
$ git checkout development
# Pull the development branch to your local repo
$ git pull
# Create a feature branch to work on
$ git branch my-feature-branch
# Checkout new feature branch
$ git checkout my-feature-branch
...
```

Do work / make changes

```
# Add any new files
$ git add {new file name}
# Commit your changes
$ git commit -am "commit message"
# Push your local branch to remote branch
$ git push origin my-feature-branch
```

```
...
```

Next, go to GitHub.

Find your branch in Code => Branches

Create pull request for your branch (make sure it's going into the development branch)

10. npm & gulp

We're starting working on a new project and want to bring in some of our favourite workflow from a previous project.

What steps should we take and what commands will we need to run?

1. Initalise npm and install npm packages for gulp

```
$ npm init
$ npm install {module name}
...
```

2. Create gulpfile.js with gulp tasks

3. Run tasks

Alternatively, you could bring `package.json` and `gulpfile.js` over from the previous project, then use `npm install`.

11. Git, npm & gulp

Our fresh project needs a .gitignore file, what should its contents be?

The `node_modules` folder and `super-secret-config.php` file.

12. Gulp and git

Our project uses gulp tasks to compile our source sass files into a styles.min.css file. In merging another developer's work we end up with this, what should we do with each?

- `/scss/_headings.scss`

  Add `_headings.scss` to scss import file, then run gulp 'styles' tasks to merge and minify into `styles.min.css`

- `/css/styles.min.css`

  Do nothing

- `about.html`

  Fix the merge conflict