

Understanding `git status`

Consider the following Terminal output run from a `git` repository:

```
git status
```

```
1 On branch master
2 Your branch is up to date with 'origin/master'.
3
4 Changes not staged for commit:
5   (use "git add/rm <file>..." to update what will be committed)
6   (use "git restore <file>..." to discard changes in working directory)
7       modified:   _quarto.yml
8       modified:   lectures/_metadata.yml
9       deleted:    pdf/_git_exam_questions.md
10
11 Untracked files:
12   (use "git add <file>..." to include in what will be committed)
13       _learning.qmd
14       gfx/capstone.png
15       gfx/github_setup_branches.png
16       pdf/git_practice_status.qmd
17
18 no changes added to commit (use "git add" and/or "git commit -a")
```

1. What is the name of the current branch?
2. Explain what the information on line 2 means.
3. Why is the file on line 7 described as `modified`?
4. Does the file on line 9 listed as `deleted` exist in the current working directory? Does it exist in the repository on GitHub?
5. Are we ready to commit? Why or why not.
6. If we run `git add -u`, which files will be staged for commit?

7. What command would we type if we want to bring back the file listed as **deleted** on line 9?
8. What does it mean for files to be listed as **untracked**? Where do those file exist?
9. How could we stage the two PNG files listed as **untracked** to the commit in one command *without* also staging the **.qmd** files for commit?

Making a commit

Suppose that you run the following commands:

```
git restore pdf/_git_exam_questions.md
git add -u
git add pdf/git_practice_status.qmd
git status
```

```
1 On branch master
2 Your branch is up to date with 'origin/master'.
3
4 Changes to be committed:
5   (use "git restore --staged <file>..." to unstage)
6       modified:   _quarto.yml
7       modified:   lectures/_metadata.yml
8       new file:   pdf/git_practice_status.qmd
9
10 Untracked files:
11   (use "git add <file>..." to include in what will be committed)
12       _learning.qmd
13       gfx/capstone.png
14       gfx/github_setup_branches.png
```

1. Describe what each of the previous four commands did.
2. Write a command to commit these changes.
3. After we commit, does GitHub have a record of the changes we made? Can our teammates see them?