

[← Pre-Course Work](#)

14 / 28

Committing File Changes to Git

We've made some changes to our project, but we haven't committed them to Git. Remember, `git status` will compare what we've done to what Git has saved from the last commit:

```
Terminal — -zsh — 68x12
[example1 master % 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:   contact.html
        modified:   home.html

no changes added to commit (use "git add" and/or "git commit -a")
example1 master %
```

Both `home.html` and `contact.html` have been modified.

In order to commit our changes to Git, we'll need to:

1. Stage the files using `git add`. (If you remember, we did this when we added our files to our repo. We have to do it before each commit.)
2. Commit our changes using `git commit`.

So, let's add `home.html` to staging:

```
git add home.html
git status
```

You should see the following:



```
git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   home.html

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:   contact.html

example1 master %
```

If we were to commit our changes right now, only `home.html` would be saved.

Let's test this out:

```
git commit -m"Added text to home page"
git status
```

After `git status`, you'll see something like this:

```
Terminal — -zsh — 68x14
example1 master % git commit -m"Added text to home page"
[git status
[master 177d3cf] Added text to home page
 1 file changed, 1 insertion(+)
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:   contact.html

no changes added to commit (use "git add" and/or "git commit -a")
example1 master %
```

Git is telling us that the changes to `home.html` were committed, but `contact.html` is still considered modified.

To commit our changes to `contact.html`, type:



```
git status
```

```
git status
```

 will show you:

```
On branch master  
nothing to commit, working tree clean
```

This means that all of your changes are in sync with Git!

