

Exercise 4: Adding a File to the Repository

For this exercise, you will be creating a new text file, adding it into Git, committing it, and then uploading it to GitHub.

1. First open your text editor (for example, Notepad++ for Windows, Atom for Mac, etc.). Create a new file and then write two paragraphs of documentation for a product that you make up. Separate the paragraphs with a blank line.

```
This documentation describes how to use the Pavel robotic  
toothbrush.
```

```
Note that this toothbrush is not safe to use for children,  
adults, or pets.
```

2. Save it in your `exercise` folder that you cloned the repository into. Give it the name `docs.md`.
3. Open your terminal app and `cd` to the `exercise` folder in your home directory. Do an `ls` (list) command and you should see `docs.md` listed. `README.md` is also listed because that was created when you cloned the repository.

```
cd ~/exercise
```

```
ls
```

4. Do a `git status` and it should list `docs.md` as “untracked”. This means it’s unstaged.

```
git status
```

5. Now stage all files in the current directory with the `git add` command. Only `docs.md` will be staged, because you haven’t made any changes to `README.md`.

```
git add .
```

6. Do a `git status` to verify that `docs.md` is staged.
7. If this is the first time you have used Git, before you do a commit, you need to give git some information about yourself so that when it does a commit, it can include that information. Type these two lines, replacing “Jane Doe” and `jane@example.com` with your own name and email. (If you have already been using Git, you will not need to do that.)

```
git config --global user.name "Jane Doe"
git config --global user.email jane@example.com
```

Note: You may get a message suggesting that you type `git config --edit`. I do not recommend doing that, since it tends to bring up an ancient, very confusing editor called Vim. If you do end up doing that, type `:x` to exit, and do the `git config` lines above instead.

8. Now commit all of the staged files, which is just `docs.md`. Use the `-m` option to add a message. The example below shows one possible message, but you can put whatever you like between the quotation marks.

```
git commit -m "Added docs.md"
```

9. Again, do a `git status` to verify that everything is committed.

Note: Notice that you haven’t had to sign into your GitHub account in any way. The reason for this is that you created a public repository, so anyone can read it. However, it’s not true that anyone can make changes to it. You have to have access in order to do that. When you push your changes to GitHub, it will require that you sign in if you haven’t already.

10. Type `git push origin` to upload your changes. Note that the first time you use Git, you will be asked for your username and password. This happens differently on different operating systems. On Windows, a dialog will pop up. On Mac OSX, you will be asked in the command line.

```
git push origin
```

11. Open up the browser with your GitHub account. Go to your exercise repository by clicking on the **exercise** link. (Or if you are already there, refresh the page.) You should see the file has been added, now you have two files: **docs.md** and **README.md**.