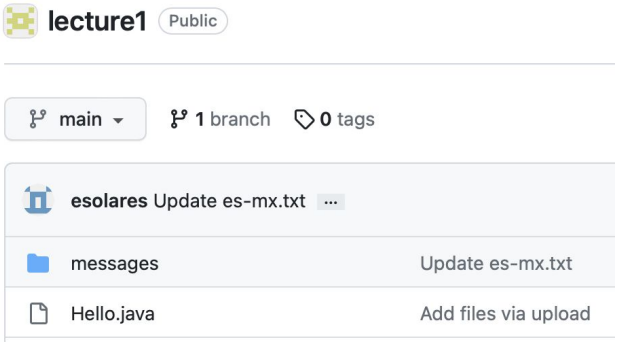
Name: Kevin Tran	**PID:** A16809193

Downloading Code from Github at the command line

`git clone <repository-url>`

Downloads ("clones") the code from the given repository-url. The repository with the example from this lecture is at
https://github.com/ucsd-cse15l-f23/lecture1



Example file structure:

```
/home/
|- lecture1/
    |- Hello.java
    |- README
    |- messages/
       |- en-us.txt (English, USA)
       |- es-mx.txt (Espanol, Mexico)
       |- zh-cn.txt (Chinese, PRC)
```

Absolute path: Where a single file or folder is within a filesystem Example: `/home/lecture1/README`
Root folder/directory: The folder that isn't contained in any other folder. Often written as `/` or `C:\\`
Relative path: A part of a path that doesn't start with the root. Example: `lecture1/README`
Working directory: An absolute path to a directory that a program or terminal uses to resolve relative paths.
Parent directory: The directory "above" or "outside" the current directory. In paths, written `..`
No directory/no path: Single dots `.` in absolute paths are ignored (e.g. `/home/./lecture1` means `/home/lecture1`)

Some Key Commands

"display"
`cat <path1> <path2> ...` - "Concatenate" Used to print the contents of one or more files given by the paths

`ls <path>` - "List" Used to list the files and folders the given path

`pwd` - "Print working directory" Used to display the current working directory

`cd <path>` - "Change Directory" Used to switch the current working directory to the given path

Consider the following working directories, relative paths, and corresponding absolute paths. Fill in the blanks.

To resolve a relative path, join it with the current working directory, then remove single dots and collapse directory names that are followed by `..`.

Working Directory (pwd)	**Relative Path**	**Absolute Path**
`/home`	`./lecture1`	`/home/lecture1`
`/home/lecture1/messages`	`../Hello.java`	`/home/lecture1/Hello.java`
`/home/lecture1`	/messages/en-us.txt	`/home/lecture1/messages/en-us.txt`
`/`	home	/home
`/home/lecture1`	`..`	/home
`/home/lecture1`	`Goodbye.java`	/home/lecture1
`/home/`	`messages/`	/home/lecture1/messages

Starting from the working directory `/home/lecture1/messages/`, what absolute path or file do each of these refer to?

`/home/lecture1/Hello.java`

`../Hello.java`

`../lecture1/Hello.java`

`/home/lecture1/messages/../Hello.java`

`./Hello.java`

`home/lecture1/Hello.java`

`./../Hello.java`

`~/lecture1/messages/Hello.java`

`~/lecture1/Hello.java`

`~/../joe/lecture1/Hello.java`

Starting from the web page `https://website.org/lab-reports/index.html`, what full URL do each of these link paths refer to?

`/images/picture.png`

`../images/picture.png`

`./picture.png`

`images/picture.png`

`https://website.org/lab-reports/picture.png`

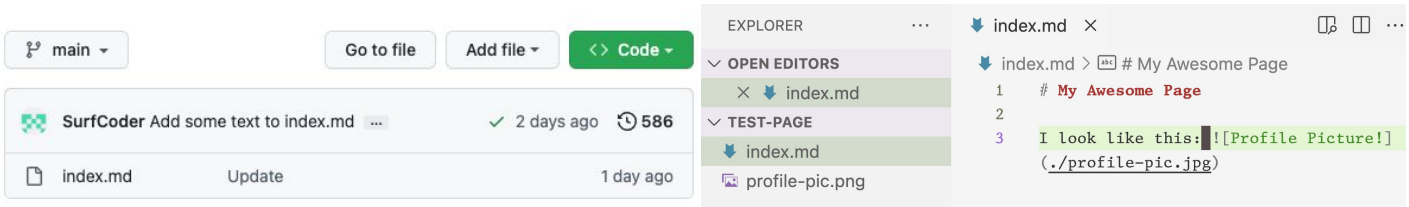
`https://website.org/images/picture.png`

`./images/picture.png`

`/`

`./` `../`

A developer is working on a Github Pages site. They are adding an image to their index.md file. The image looks great in their Visual Studio Code preview, but on their Github Pages site it appears as a broken image link. Here's their VScode and their Github repository:



What's going on? What mistake or mistakes might they have made?