

GitHub : For Pizza

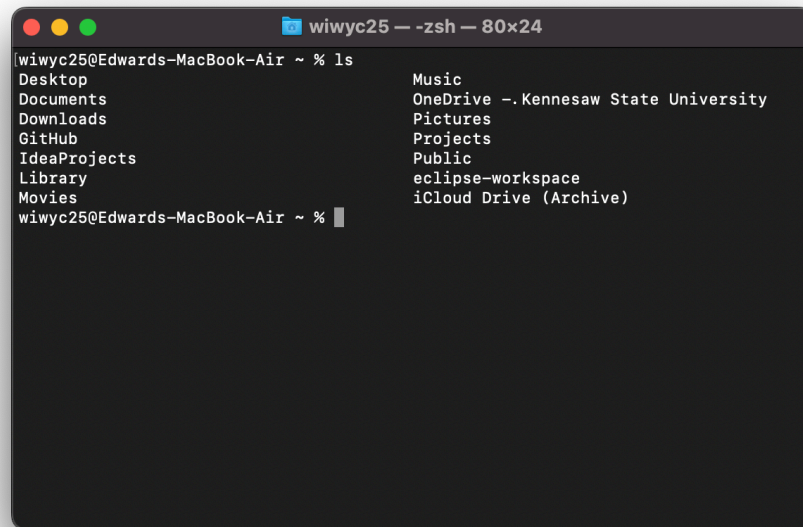
Basic Git Commands:

Command	Action
ls	Lists all documents in current terminal location.
cd nameOfFileHere	Changes directory to specified file, or return to top level of directory if no file is specified.
git clone repositoryAddress	Copies all the files of a repository to the destination of your terminal location.
git status	Displays status of current brand and version of the project.
git checkout -b 0000E-NameOfBranch	Create a new branch, generally from the main branch, of a project.
git checkout 0000E-NameOfBranch	Checkout a version of the project that is not available through the main branch.
git add NameOfFile	Add a specific file to the staging area, in preparation to be saved.
git commit -m "Message about changes made"	"Commits" are synonymous with "Saving". This command saves the changes a developer makes in their branch.
git push origin 0000E-NameOfBranch	Sends the saved version of your branch back to the main (origin) version of your project. Main == Origin (generally speaking)
git pull	Will pull down all updated changes made to the main branch of a project.

How to Clone a Project:

First, think about where you want this project to be located on your computer.

I'm going to first list my directory so I understand where I am and what's available.

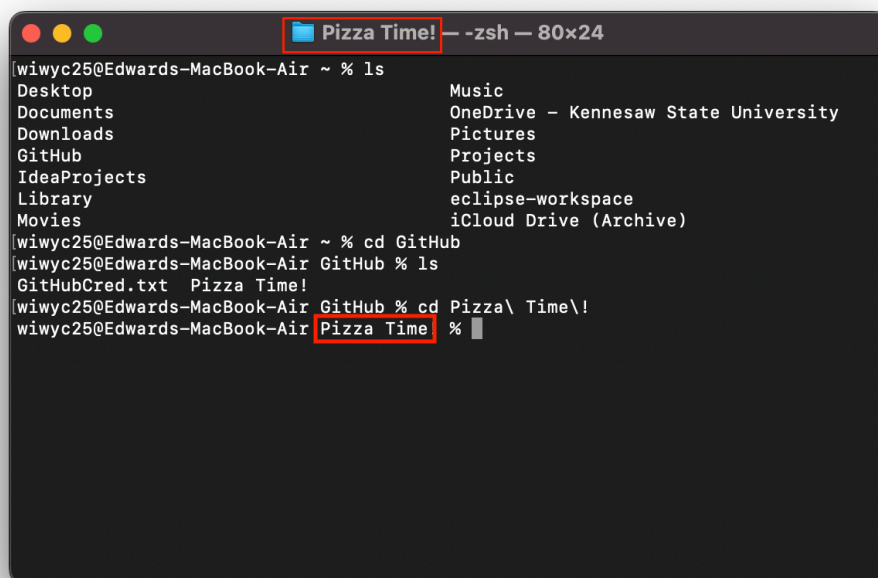
A terminal window titled "wiwyc25 - zsh - 80x24" showing the output of the 'ls' command. The output lists various directories and files in two columns. The first column includes Desktop, Documents, Downloads, GitHub, IdeaProjects, Library, Movies, and wiwyc25@Edwards-MacBook-Air ~ %. The second column includes Music, OneDrive -- Kennesaw State University, Pictures, Projects, Public, eclipse-workspace, and iCloud Drive (Archive).

```
wiwyc25@Edwards-MacBook-Air ~ % ls
Desktop                               Music
Documents                            OneDrive -- Kennesaw State University
Downloads                             Pictures
GitHub                                Projects
IdeaProjects                          Public
Library                               eclipse-workspace
Movies                                iCloud Drive (Archive)
wiwyc25@Edwards-MacBook-Air ~ %
```

I have already created a folder titled "GitHub" with files inside. I want to save the clone of the project in a specific location. To do that I'm going to navigate my way there.

There are a couple of indicators (highlighted in red below) that discern what location I'm in. Once the terminal is in the location you want to save your project, it's time to clone the repository.

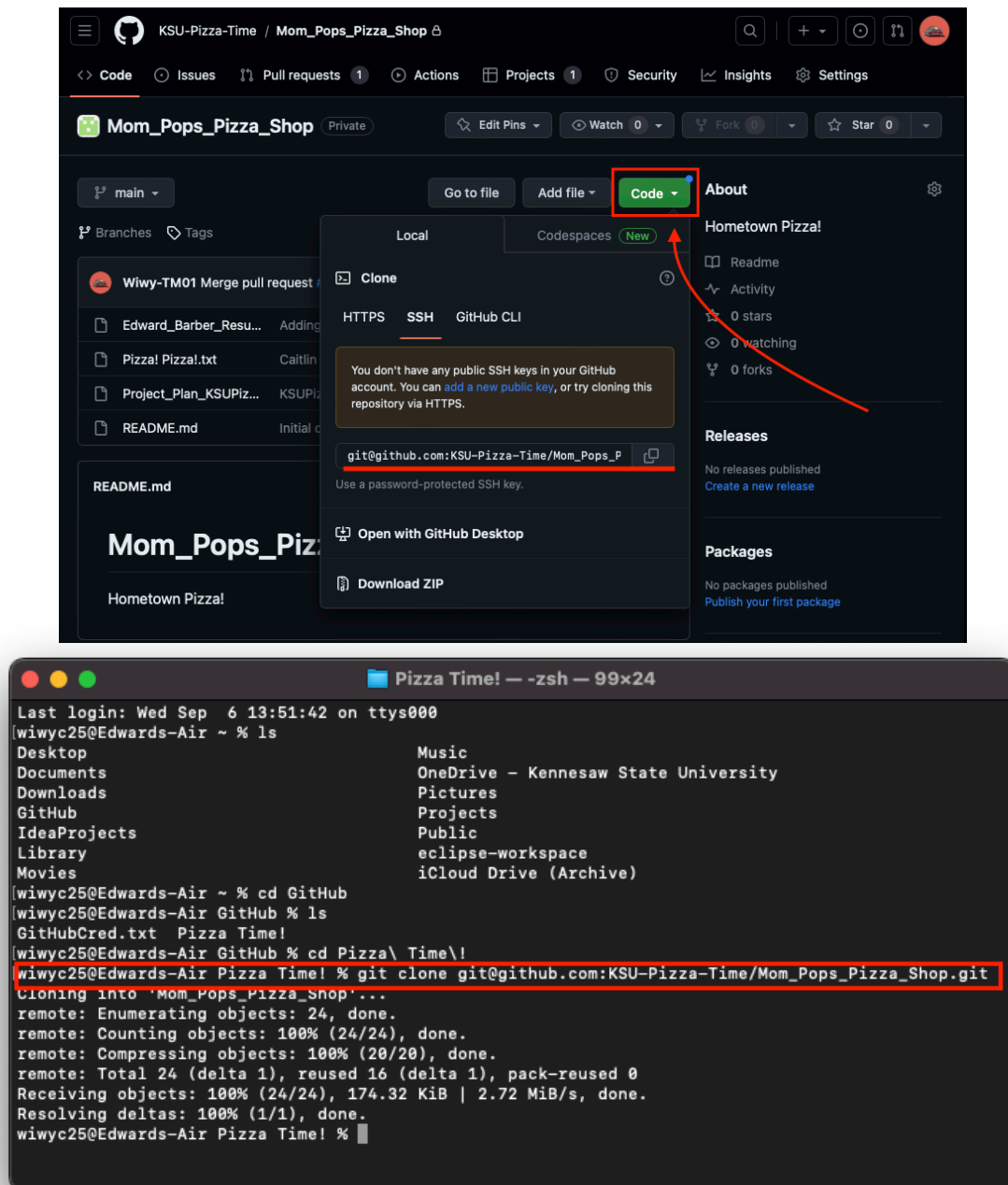
TIP: Once a file has been partially spelled out in the terminal, tapping TAB will auto-complete the file that best matches the input.

A terminal window titled "Pizza Time! - zsh - 80x24" showing the output of several commands. The first command is 'ls', which lists the same directories and files as the previous terminal window. The second command is 'cd GitHub', which changes the current directory to GitHub. The third command is 'ls', which lists the contents of the GitHub directory: GitHubCred.txt and Pizza Time!. The fourth command is 'cd Pizza\ Time!', which changes the current directory to Pizza Time!. The fifth command is 'ls', which lists the contents of the Pizza Time! directory. The output of the fifth command is partially visible, showing 'Pizza Time' followed by a space and a cursor.

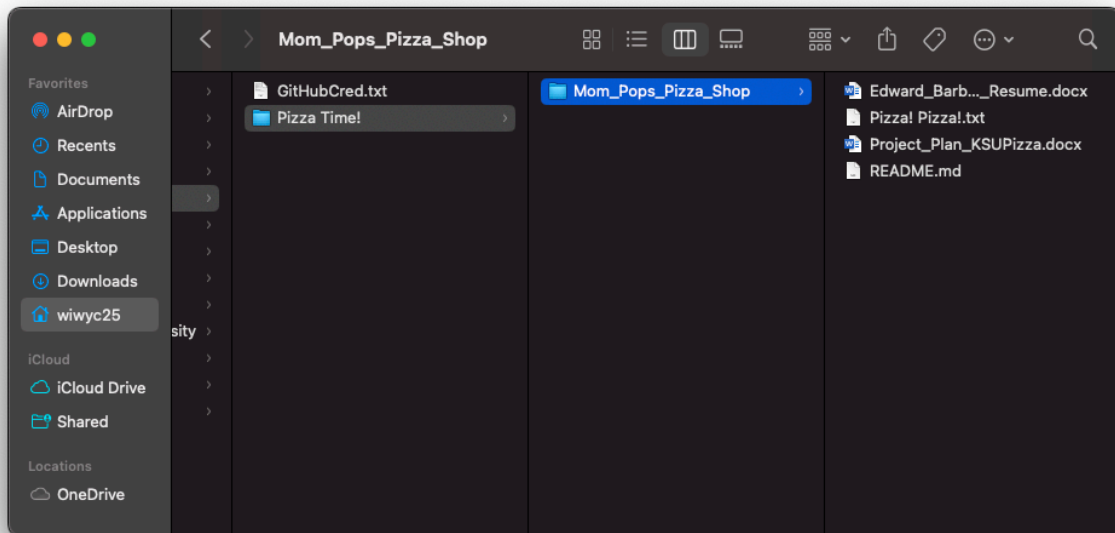
```
wiwyc25@Edwards-MacBook-Air ~ % ls
Desktop                               Music
Documents                            OneDrive -- Kennesaw State University
Downloads                             Pictures
GitHub                                Projects
IdeaProjects                          Public
Library                               eclipse-workspace
Movies                                iCloud Drive (Archive)
wiwyc25@Edwards-MacBook-Air ~ % cd GitHub
wiwyc25@Edwards-MacBook-Air GitHub % ls
GitHubCred.txt  Pizza Time!
wiwyc25@Edwards-MacBook-Air GitHub % cd Pizza\ Time!
wiwyc25@Edwards-MacBook-Air Pizza Time %
```

There are a few ways to download the files attached to a repo, all of which GitHub provides under the code tab. For git, we want the autogenerated SSH command, which we can copy.

Once we have the command address, we make the request through the terminal. If we have permission, it will be followed with the download packaging information.



These changes will also reflect in your computer's file system. The repository that was just cloned will be the most up to date version of the project available on the main branch.



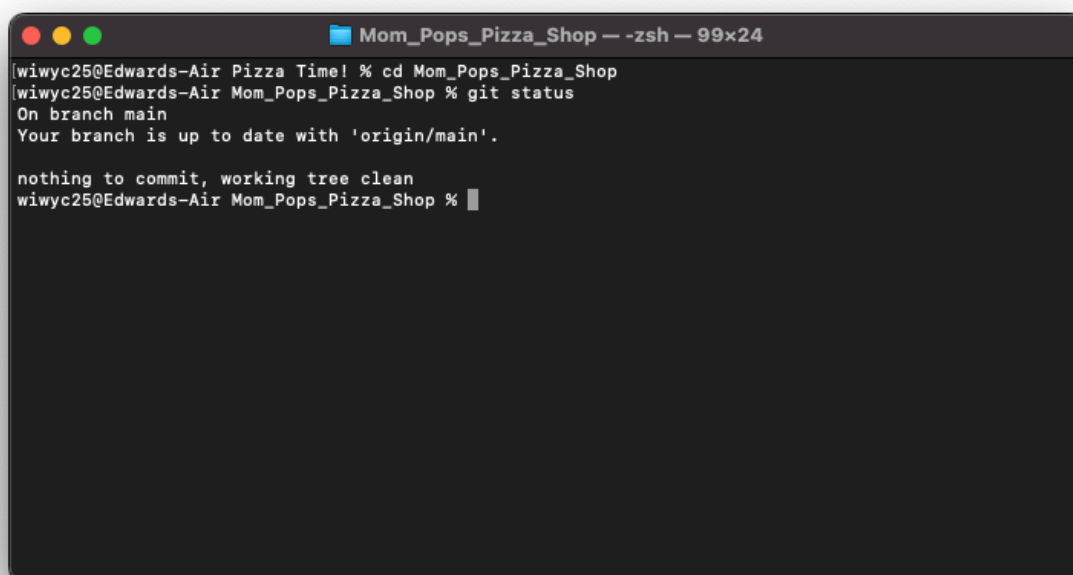
How to Create a New Branch to Create Changes:

Now that there is a copy of the project available, a developer can create and push changes.

The first step is to checkout a new branch, so any changes made will be neatly organized until a developer is ready to merge those changes into the main branch.

A helpful command for understanding the current status of your branch (and whether or not changes have been made to the project) is the “git status” command.

TIP: Once the project has been cloned, make sure to navigate to the newly cloned project by changing the directory (cd). This doesn't automatically happen once the project is finished being cloned.

A terminal window titled "Mom_Pops_Pizza_Shop - zsh - 99x24" is shown. The terminal output is as follows:

```
wiwyc25@Edwards-Air Pizza Time! % cd Mom_Pops_Pizza_Shop
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch main
Your branch is up to date with 'origin/main'.

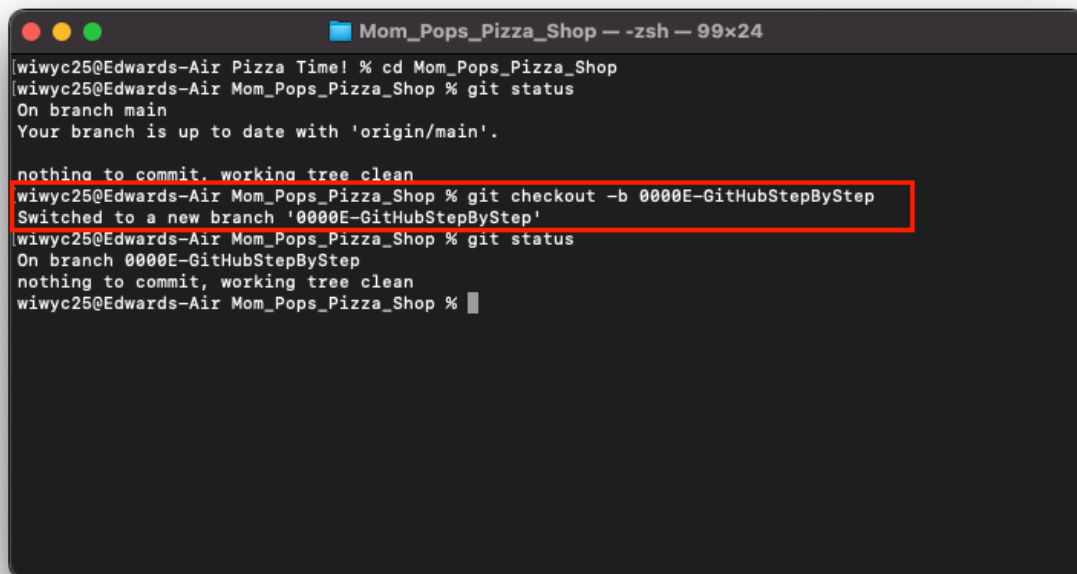
nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %
```

Above the terminal informs us that we are on the main branch, the main branch is up to date with 'origin/main' (essentially another way to say main branch), and our working tree is clean. These are the desirable conditions to have before creating a new branch.

NOTE: Depending on the size and scale of the project, constant updating will be needed to keep the local version (computer clone) of the project in-line with the remote version. SEE GIT PULL SECTION.

To create a new branch, we use the checkout command.

The “-b” part of this command specifies that the branch that is being checked out is new. I named the new branch 0000E-GitHubStepByStep.

A terminal window titled "Mom_Pops_Pizza_Shop -- zsh -- 99x24" showing a series of git commands and their outputs. The commands are: "cd Mom_Pops_Pizza_Shop", "git status", "git checkout -b 0000E-GitHubStepByStep", and "git status". The output of the first "git status" shows the user is on the 'main' branch. The output of the "git checkout" command shows a switch to a new branch. The second "git status" shows the user is on the '0000E-GitHubStepByStep' branch. A red rectangle highlights the "git checkout -b 0000E-GitHubStepByStep" command and its output.

```
wiwyc25@Edwards-Air Pizza Time! % cd Mom_Pops_Pizza_Shop
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git checkout -b 0000E-GitHubStepByStep
Switched to a new branch '0000E-GitHubStepByStep'
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %
```

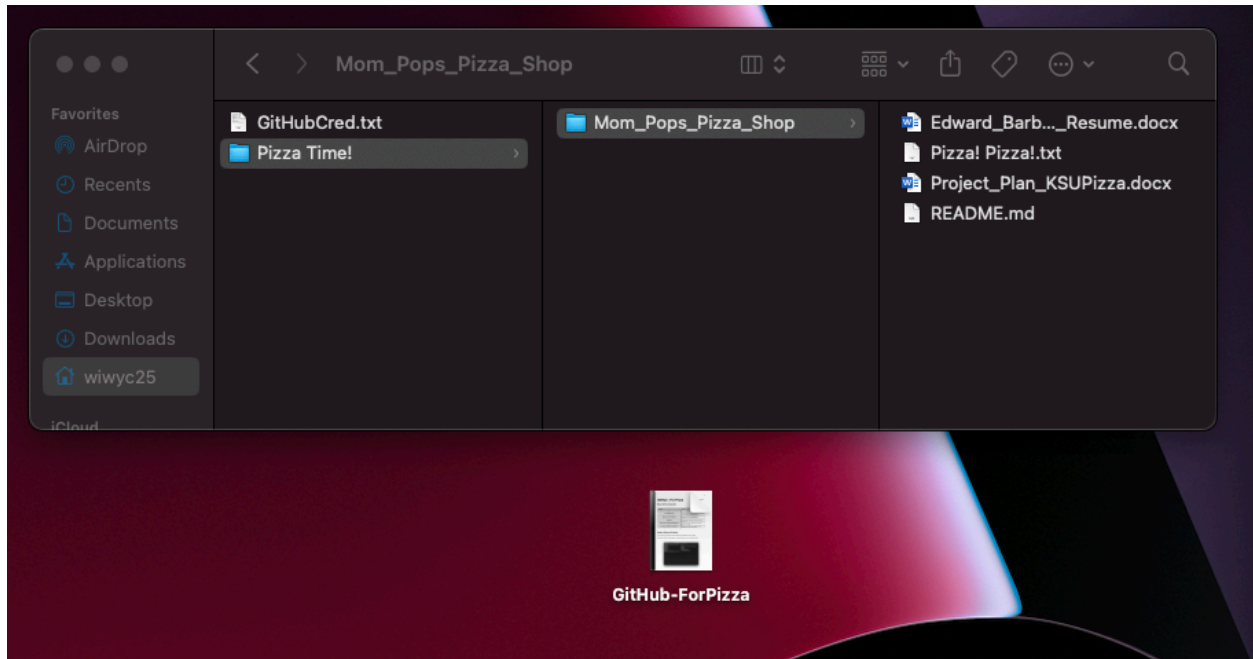
Developers can checkout branches that have already been created by omitting the -b, and just reference the branch name. This will be important later when we peer review our work on the project.

For Example: git checkout 0000E-GitHubStepByStep, would allow a developer to pull a branch that has changes not available on the main project. This includes branches made by other developers on the project.

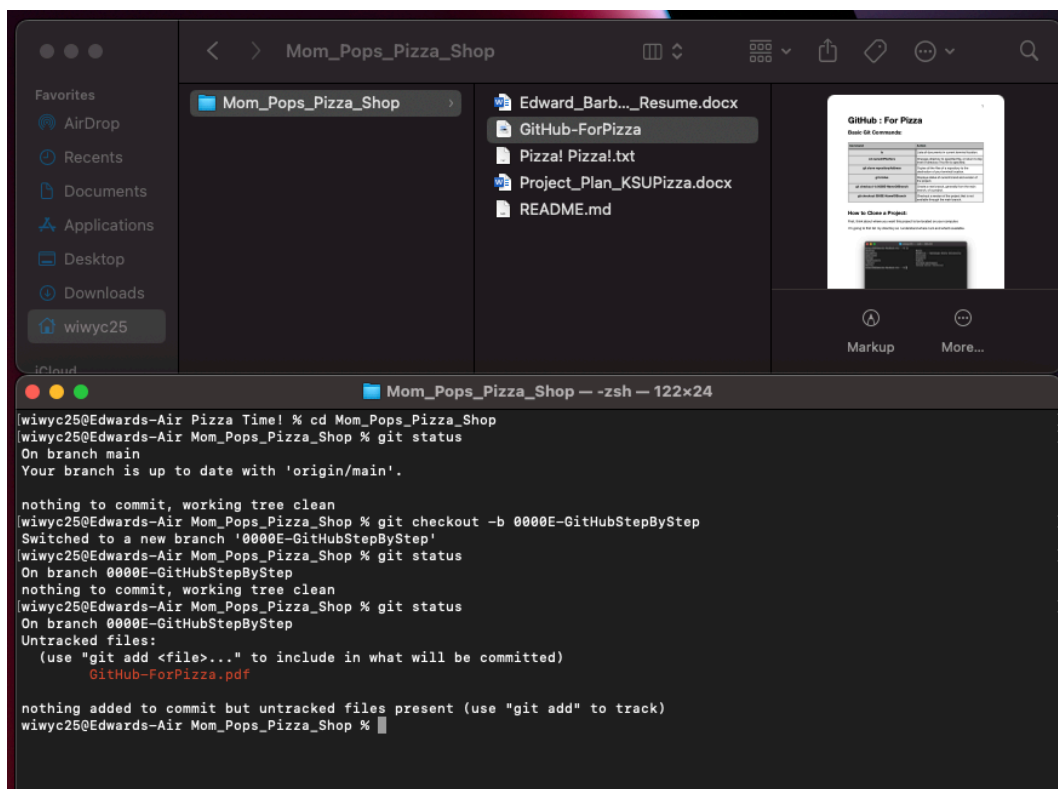
Creating, Saving, Pushing Changes:

Making changes to a project is simple, as adding a single word is enough to be considered a different version of the main project. In this case I will show how to add a new document to the project.

In the image below, I have my project location pulled up on my file finder. Underneath that is the document that I want to add to my project.



By dragging the PDF document into my Mom_Pops_Pizza_Shop folder, my git status will change to show these amendments.

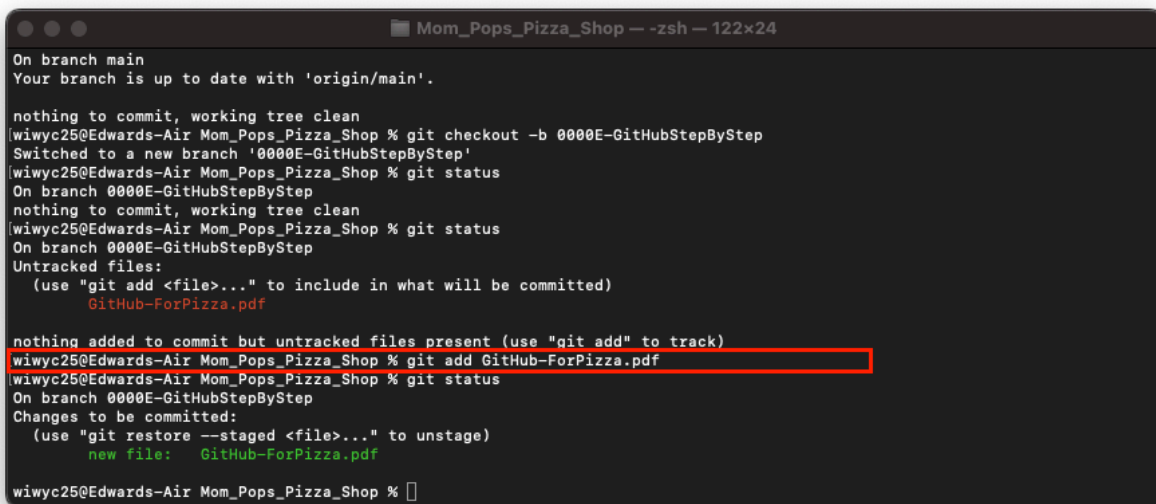


My status now claims I have an untracked file, which makes sense because I added that file and it has changed the project.

In order for these changes to merge into the main branch there are 3 steps.

1. Stage: Add all files that are desired to be included in these changes
2. Save: those changes to the local branch
3. Push: those changes to the main / remote branch

STAGE : all files that have successfully been staged will show up in green with a git status command.



```

Mom_Pops_Pizza_Shop -- zsh -- 122x24

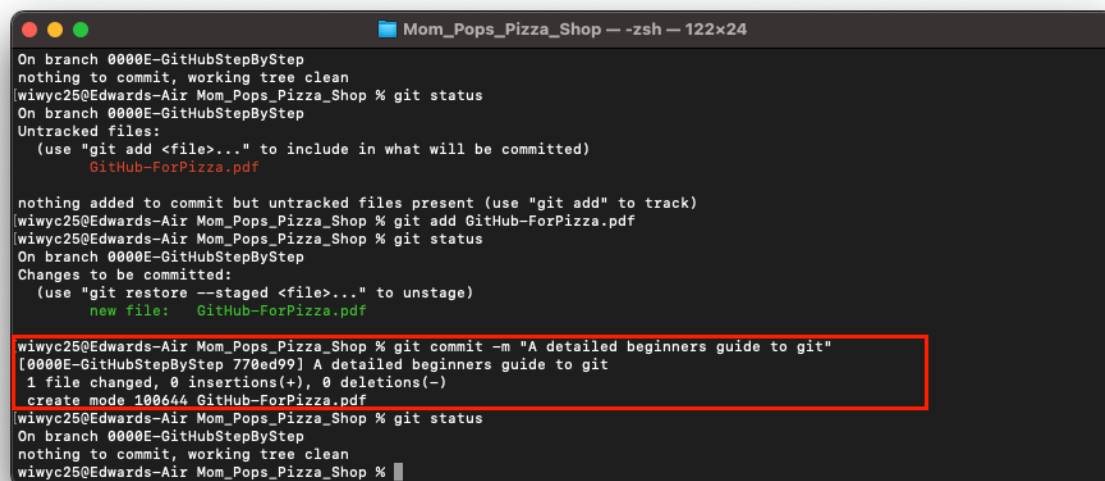
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git checkout -b 0000E-GitHubStepByStep
Switched to a new branch '0000E-GitHubStepByStep'
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GitHub-ForPizza.pdf

nothing added to commit but untracked files present (use "git add" to track)
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git add GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %

```

SAVE : Now that we have determined what files we want to add/change in our project, we need to save those changes. Once those changes are saved, our branch will reflect that it is up to date and no new changes have since been made.



```

Mom_Pops_Pizza_Shop -- zsh -- 122x24

On branch 0000E-GitHubStepByStep
nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        GitHub-ForPizza.pdf

nothing added to commit but untracked files present (use "git add" to track)
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git add GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git commit -m "A detailed beginners guide to git"
[0000E-GitHubStepByStep 770ed99] A detailed beginners guide to git
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %

```


PUSH: Now we share our changes with everyone on the project. To do this we want to send it back to the main branch as a Pull Request (PR).

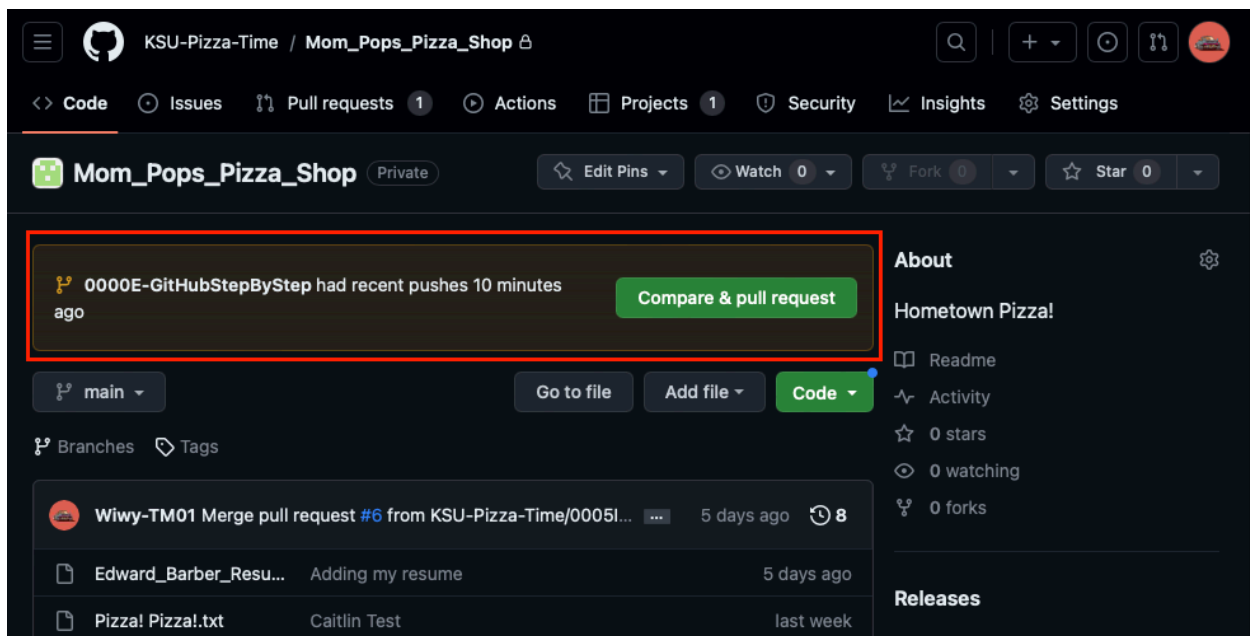
```

new file:   GitHub-ForPizza.pdf

[wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git commit -m "A detailed beginners guide to git"
[0000E-GitHubStepByStep 770ed99] A detailed beginners guide to git
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 GitHub-ForPizza.pdf
[wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch 0000E-GitHubStepByStep
nothing to commit, working tree clean
[wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git push origin 0000E-GitHubStepByStep
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 851.26 KiB | 20.76 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for '0000E-GitHubStepByStep' on GitHub by visiting:
remote:   https://github.com/KSU-Pizza-Time/Mom_Pops_Pizza_Shop/pull/new/0000E-GitHubStepByStep
remote:
To github.com:KSU-Pizza-Time/Mom_Pops_Pizza_Shop.git
 * [new branch]      0000E-GitHubStepByStep -> 0000E-GitHubStepByStep
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %

```

There is some nuance to the “git push origin” command, but for our purposes this will work fine. You can see the changes made in the terminal are now available for review on gitHub via a PR.



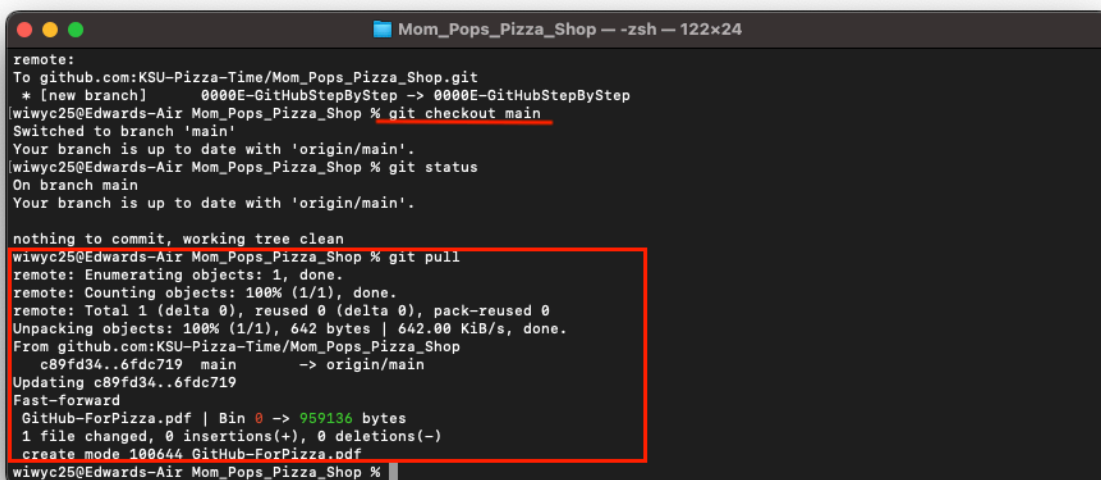
Be sure to write descriptively about the changes that are encompassed with your branch on your pull request. Then press “Create pull request” once you’re finished. Once everything is approved, go ahead and merge your changes into the project.

The screenshot shows the GitHub 'Open a pull request' page for the repository **KSU-Pizza-Time / Mom_Pops_Pizza_Shop**. The page is in dark mode. At the top, there's a navigation bar with links to Code, Issues, Pull requests (1), Actions, Projects (1), Security, Insights, and Settings. Below this, the main heading is 'Open a pull request' with a subtext: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below the heading, there's a comparison bar showing 'base: main' and 'compare: 0000E-GitHubStepByStep' with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The main content area is titled '0000E-GitHubStepByStep' and has two tabs: 'Write' (active) and 'Preview'. The 'Write' tab shows a rich text editor with the text 'A new document to better understand the process and implementation of gitHub for this project.' Below the text area, there's a placeholder for attaching files: 'Attach files by dragging & dropping, selecting or pasting them.' At the bottom right of the main content area is a green button labeled 'Create pull request'. On the right side of the page, there's a sidebar with sections: 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), 'Development' (Use [Closing keywords](#) in the description to automatically close issues), and 'Helpful resources' (a link to [GitHub Community Guidelines](#)). At the bottom left of the main content area, there's a small information icon and a note: 'Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).'

How to use the Git Pull Command:

Now that we have made our changes, it's important that developers update their local files. That way when work for new changes comes down the line, developers are ready with the latest version of the project.

The first step to do this is to checkout the main branch. If we do a git status it will not always give us a heads up that new changes have since been made. As long as we're in the main branch, we are safe to pull down the changes.



```
Mom_Pops_Pizza_Shop -- zsh -- 122x24
remote:
To github.com:KSU-Pizza-Time/Mom_Pops_Pizza_Shop.git
* [new branch]      0000E-GitHubStepByStep -> 0000E-GitHubStepByStep
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop % git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 642 bytes | 642.00 KiB/s, done.
From github.com:KSU-Pizza-Time/Mom_Pops_Pizza_Shop
   c89fd34..6fdc719  main       -> origin/main
Updating c89fd34..6fdc719
Fast-forward
 GitHub-ForPizza.pdf | Bin 0 -> 959136 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 GitHub-ForPizza.pdf
wiwyc25@Edwards-Air Mom_Pops_Pizza_Shop %
```