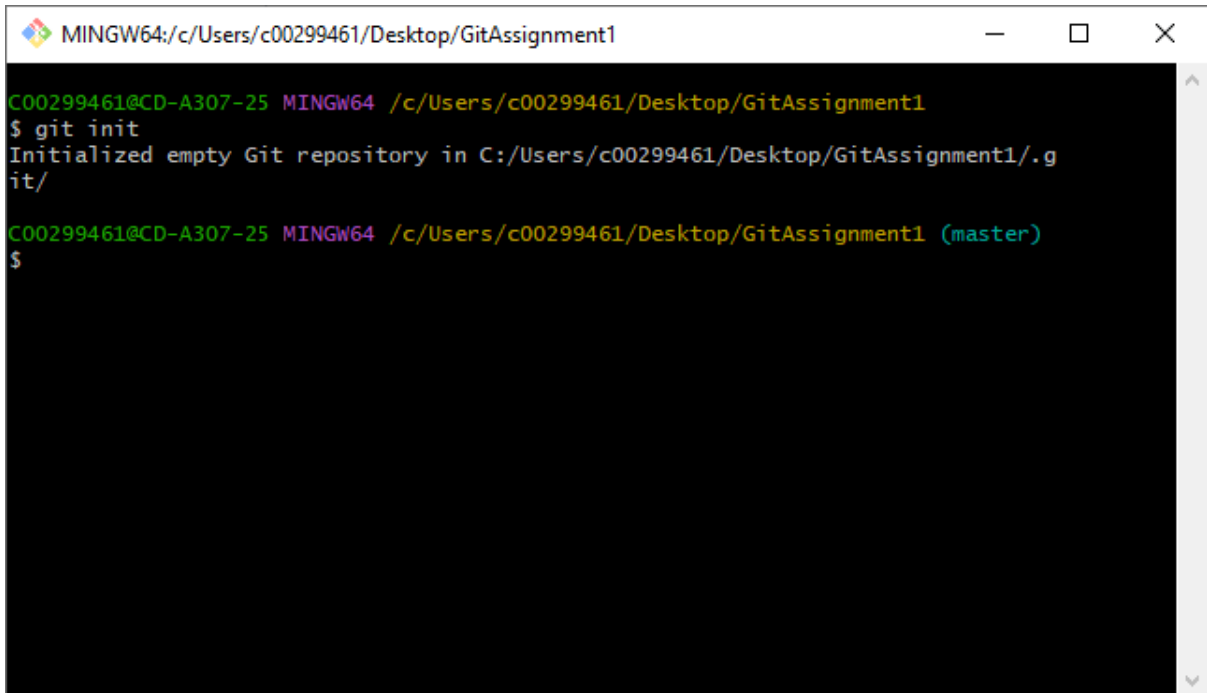


Init

git init is used to create a new repository in the place where you started your Git Bash from. It creates a hidden .git folder which is the directory for this project. It will also create an empty branch for you to work in. It will mainly be used at the beginning of a project, but you can use it in an existing repository without issue.

Syntax: git init

A screenshot of a Git Bash terminal window. The title bar shows the path 'MINGW64:/c:/Users/c00299461/Desktop/GitAssignment1'. The terminal output shows the command '\$ git init' being entered, followed by the message 'Initialized empty Git repository in C:/Users/c00299461/Desktop/GitAssignment1/.git/'. The prompt then changes to '\$' and the branch '(master)' is indicated in parentheses next to the path.

```
MINGW64:/c:/Users/c00299461/Desktop/GitAssignment1
C00299461@CD-A307-25 MINGW64 /c:/Users/c00299461/Desktop/GitAssignment1
$ git init
Initialized empty Git repository in C:/Users/c00299461/Desktop/GitAssignment1/.git/
C00299461@CD-A307-25 MINGW64 /c:/Users/c00299461/Desktop/GitAssignment1 (master)
$
```

Add

git add is used to bring a file found in your bash folder to the 'stage,' making it ready for the next commit. This acts as a 'screenshot' of the file, so if you update it before committing you will need to use this command to stage it again. You can check if it worked with the git status command, after which the file will be highlighted in green if it's staged.

Syntax: git add

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git add Git1.txt

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ |
```

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git add Git1.txt

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Git1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Git1.docx

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ |
```

```
MINGW64:/c/Users/c00299461/Desktop/GitAssignment1
hint: "git config advice.addEmptyPaths false"

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git add Git1.txt

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git add Git1.txt

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git remote add origin https://github.com/PeterLowe/git-one-SophiaStanley.git

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git1.docx
        Git1.txt
        ~$Git1.docx

nothing added to commit but untracked files present (use "git add" to track)
```

Status

git status is used to show the details of your repository. This includes what branch you're currently on and details on your files, for example if they've been modified but not staged. If you have a remote repository, it will also tell you how much this version of the repository differs between the remote one.

Syntax: git status

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Git1.docx

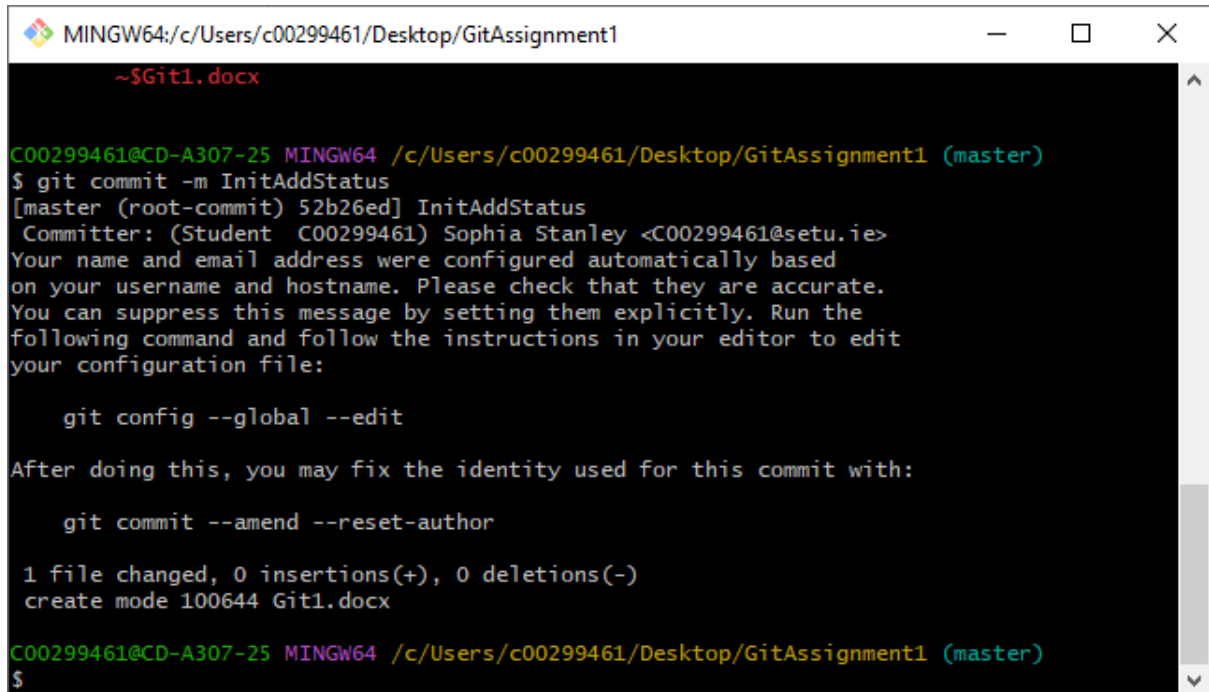
no changes added to commit (use "git add" and/or "git commit -a")

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$
```

Commit

git commit is used to create a save point for your repository. If adding is a screenshot of your files, then committing could be seen as a screenshot of your whole repository. You cannot commit until you've made some changes since the previous time, such as by staging another file.

Syntax: git commit



```
MINGW64:/c/Users/c00299461/Desktop/GitAssignment1
~$Git1.docx

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git commit -m InitAddStatus
[master (root-commit) 52b26ed] InitAddStatus
Committer: (Student C00299461) Sophia Stanley <C00299461@setu.ie>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

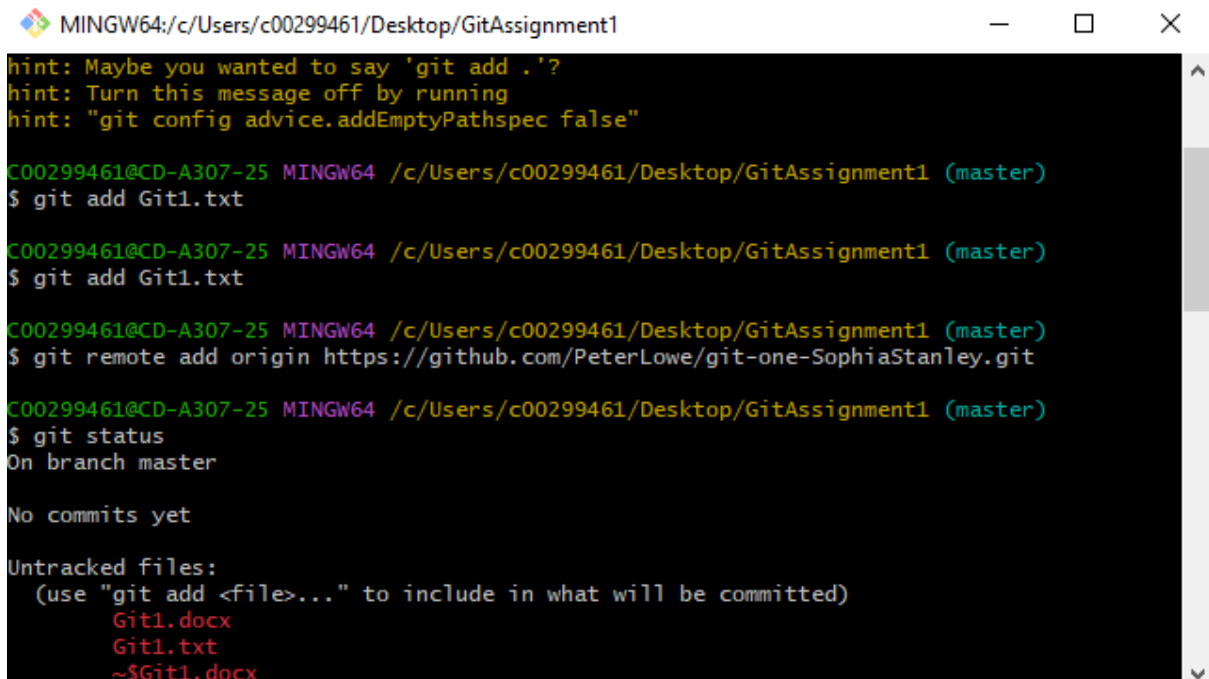
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Git1.docx

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$
```



```
MINGW64:/c/Users/c00299461/Desktop/GitAssignment1

hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathsSpec false"

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git add Git1.txt

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git add Git1.txt

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git remote add origin https://github.com/PeterLowe/git-one-SophiaStanley.git

C00299461@CD-A307-25 MINGW64 /c/Users/c00299461/Desktop/GitAssignment1 (master)
$ git status
On branch master

No commits yet

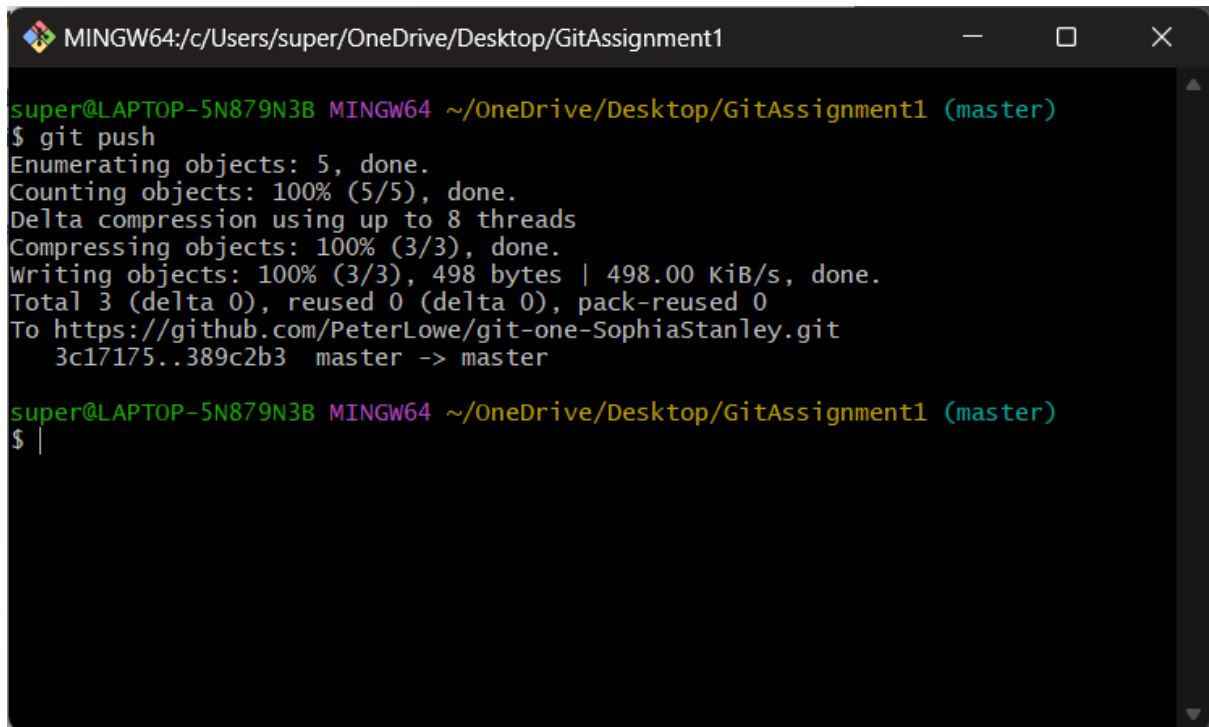
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git1.docx
    Git1.txt
    ~$Git1.docx
```

Push

The git push command updates any remote repositories with the most recent commit to ensure that they're both up to date, by sending all of the new content within the local copy to the remote one.

You should specify which repository you want to update, as if you have more than one remote it will either be sent to the one configured as the 'default' or the one named origin.

Syntax: `git push [repository]`

A terminal window titled 'MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1' with standard window controls. The prompt is 'super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)'. The command '\$ git push' is entered, followed by output: 'Enumerating objects: 5, done.', 'Counting objects: 100% (5/5), done.', 'Delta compression using up to 8 threads', 'Compressing objects: 100% (3/3), done.', 'Writing objects: 100% (3/3), 498 bytes | 498.00 KiB/s, done.', 'Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', 'To https://github.com/PeterLowe/git-one-SophiaStanley.git', '3c17175..389c2b3 master -> master'. The prompt returns to '\$ |'.

Remote

You can use `git remote` to create additional versions of your repository that exist outside of your local drive. For creating a new one, you will need to include a name and the URL of where the new copy will be stored. Simply typing '`git remote`' brings up a list of your remote copies, and adding `-v` gives a more detailed list.

Syntax: `git remote add [name] [URL]`

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git remote add origin https://github.com/PeterLowe/git-one-SophiaStanley

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git remote -v
origin https://github.com/PeterLowe/git-one-SophiaStanley (fetch)
origin https://github.com/PeterLowe/git-one-SophiaStanley (push)

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ |
```

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git remote
origin

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$
```

Log

git log displays a list of every commit you have made in this repository, starting from the earliest. Depending on your screen size, this can fill up the whole screen after about 5 commits, after which you can press the space bar to scroll down to the end, and q to quit out of the list.

Syntax: git log

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git log
commit 7e5e96e4dc89e31936739ade6b12f8e88a67fb75 (HEAD -> master)
Author: Soph Stanley <c00299461@setu.ie>
Date: Mon Oct 23 02:48:45 2023 +0100

    Push

commit 389c2b3fe13663331baacf062687b36582d95527 (origin/master)
Author: Soph Stanley <c00299461@setu.ie>
Date: Mon Oct 23 02:44:38 2023 +0100

    InitAddStatusAgain

commit 3c171758cfcbdcce691d9442bfc03afc0b8b7db
Author: (Student C00299461) Sophia Stanley <C00299461@setu.ie>
Date: Fri Oct 20 10:17:46 2023 +0100

    PushLog

commit 52b26edd2e706170c95202d74f256c671937cb26
Author: (Student C00299461) Sophia Stanley <C00299461@setu.ie>
Date: Fri Oct 20 09:36:58 2023 +0100
```

Stash

git stash is used for when you want to create a record of your current directory, without needing to commit. You can use git stash pop to retrieve the contents from a stash to put in the repository.

Syntax: git stash

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git stash
Saved working directory and index state WIP on master: 23214c6 Log

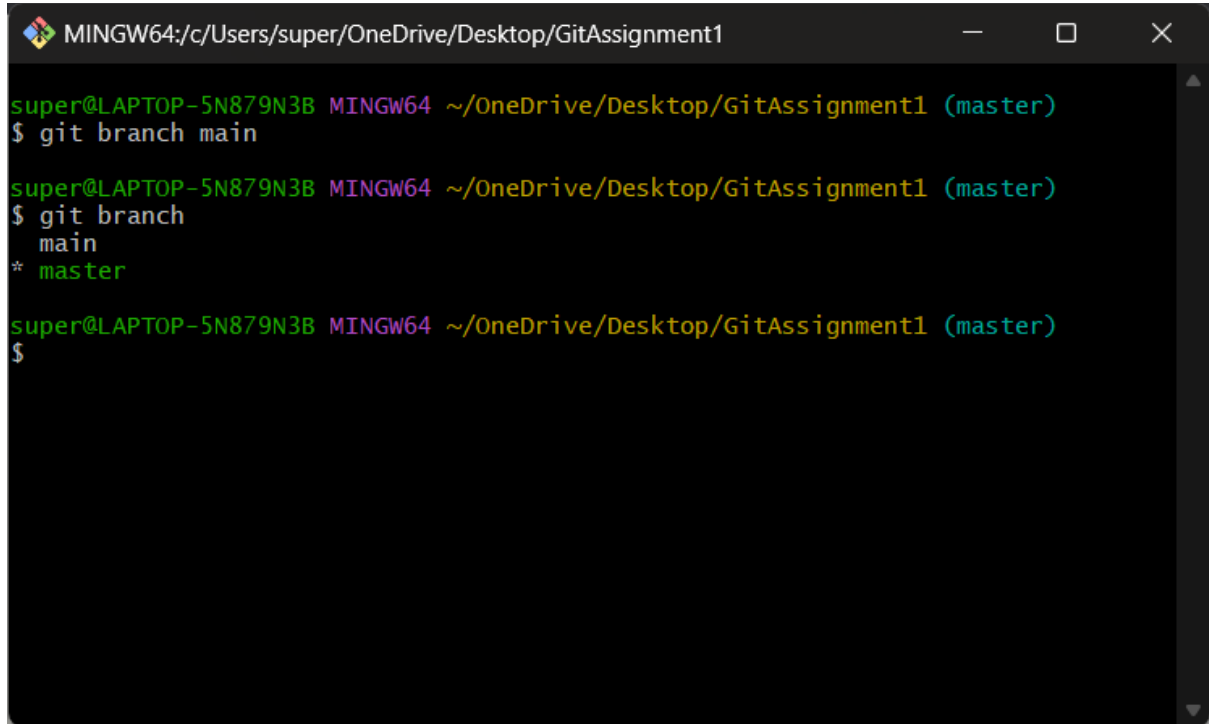
super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git stash list
stash@{0}: WIP on master: 23214c6 Log
stash@{1}: WIP on master: 23214c6 Log
stash@{2}: WIP on master: 23214c6 Log
stash@{3}: WIP on master: 3c17175 PushLog
stash@{4}: WIP on master: 3c17175 PushLog
stash@{5}: WIP on master: 52b26ed InitAddStatus

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$
```

Branch

git branch can be used to create another branch in a repository. This is a great way of sectioning off certain parts to do different things rather than having everything combined into one branch.

Syntax: git branch [name]

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1'. The terminal shows three lines of command execution. The first line shows the user 'super' at 'LAPTOP-5N879N3B' running 'git branch main' in the 'master' branch. The second line shows the user running 'git branch' again, which lists 'main' and marks 'master' with an asterisk to indicate it is the current branch. The third line shows the user pressing the enter key at the prompt, resulting in another prompt line.

```
super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git branch main

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git branch
  main
* master

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$
```

Checkout

If you're working with multiple branches, you can use the git checkout command to specify which one you want to be on currently. Make sure your current branch has been fully committed or stashed, otherwise it will not allow you to run the command.

Syntax: git checkout [branch]


```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git checkout main
Switched to branch 'main'

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (main)
$ git status
On branch main
nothing to commit, working tree clean

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (main)
$
```

```
MINGW64:/c/Users/super/OneDrive/Desktop/GitAssignment1

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$ git checkout main
error: Your local changes to the following files would be overwritten by checkout:
    Git1.docx
Please commit your changes or stash them before you switch branches.
error: The following untracked working tree files would be overwritten by checkout:
    ~$Git1.docx
Please move or remove them before you switch branches.
Aborting

super@LAPTOP-5N879N3B MINGW64 ~/OneDrive/Desktop/GitAssignment1 (master)
$
```