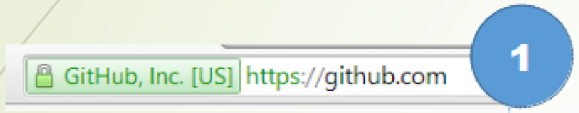
ECSE 321 Tutorial 2

Git and Github

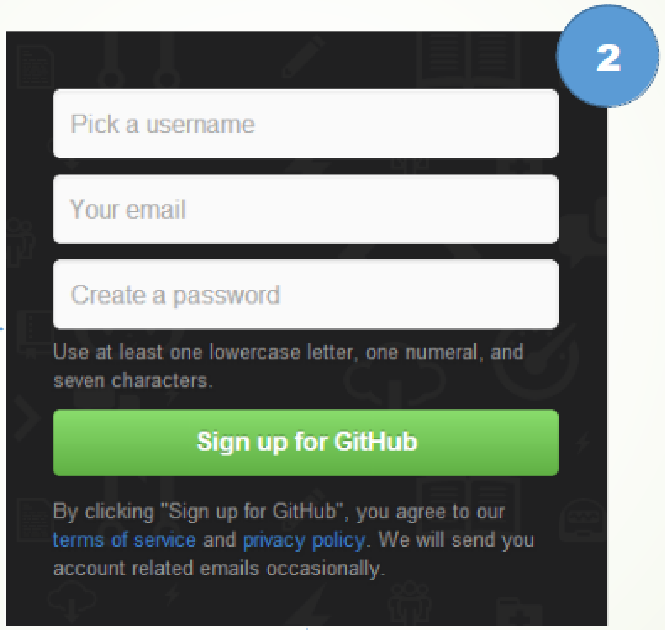
Today's plan

- Creating a Github profile
- Configuring Git
- Basics of Git
- Putting our code in the cloud

Register for Github



GitHub, Inc. [US] https://github.com



Pick a username

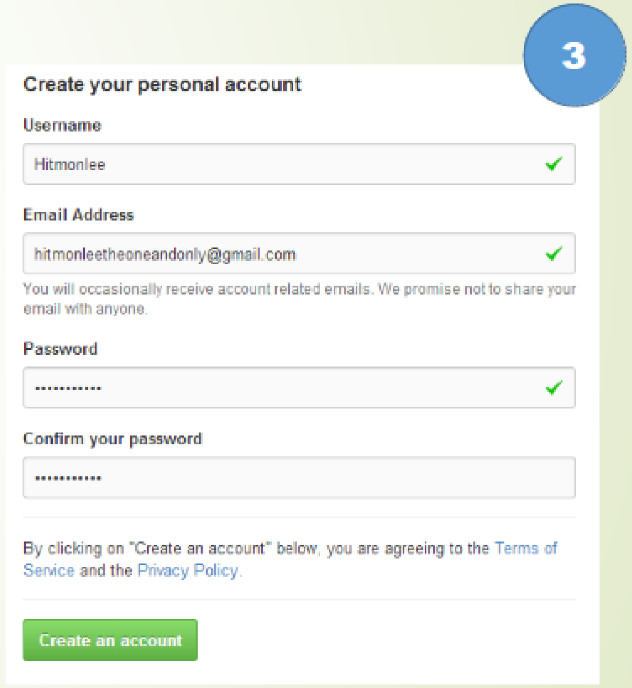
Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.



Create your personal account

Username

Hitmonlee ✓

Email Address

hitmonleetheoneandonly@gmail.com ✓

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

••••••••••• ✓

Confirm your password

•••••••••••

By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

Create an account

Your TA this Semester

Shabbir Hussain

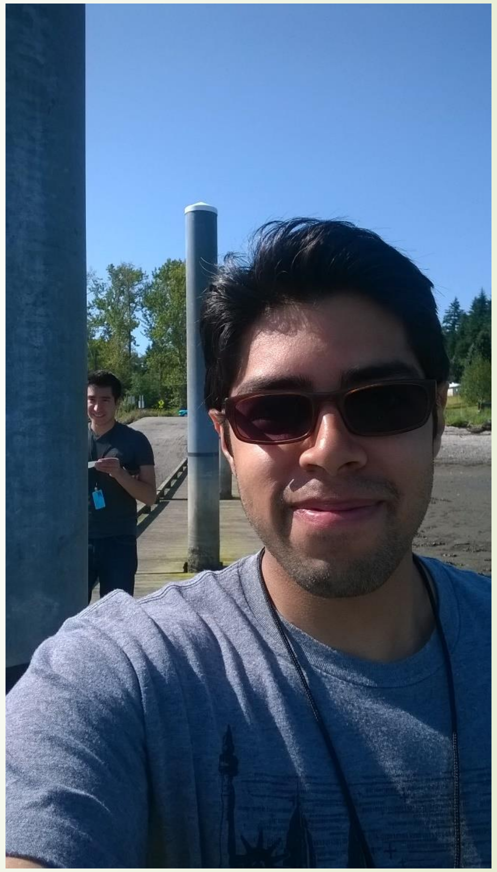
Graduated EE @ McGill Winter 2015

1st yr Masters in Comp Eng

TAed this course three times before

Contact:

- Shabbir.hussain@mail.mcgill.ca
- https://github.com/shabbir-hussain/
- MyCourses



Your TA this Semester

Ossama Ahmed

Third Year Software Eng Undergraduate

Contact:

- Ossama.Ahmed@mail.mcgill.ca
- MyCourses

What is Version Control

- Version Control software is a tool used to **keep track** of different **versions** of files
- **Revert** to an old **version**
- **Branch** to create multiple versions
- **Merge** two different versions together
- **Synchronize** files on different **machines**

Pop Quiz

- Would it be a good idea to use **version control** on your **photo album**?
- When would you want to **revert** to a previous version of a file?
- Why would you want to **branch** your files?

Why are you going to use git

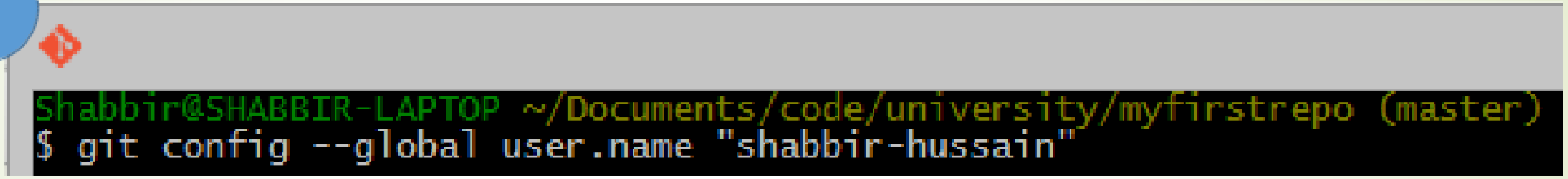
- You **<u>need</u>** it for this class
- It's a great way to **sync** your code with your team
- I will post code **examples** on my Github
- Because using Dropbox to sync your code is <u>so last semester</u>

Did I install it correctly?

Terminal
Git Bash

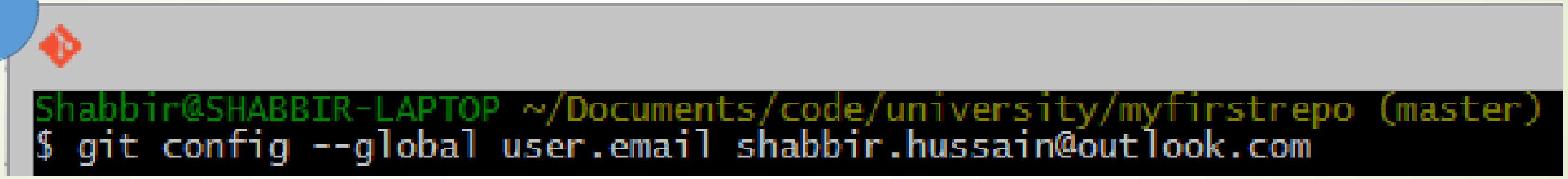
Making git work with GitHub

1



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git config --global user.name "shabbir-hussain"
```

2

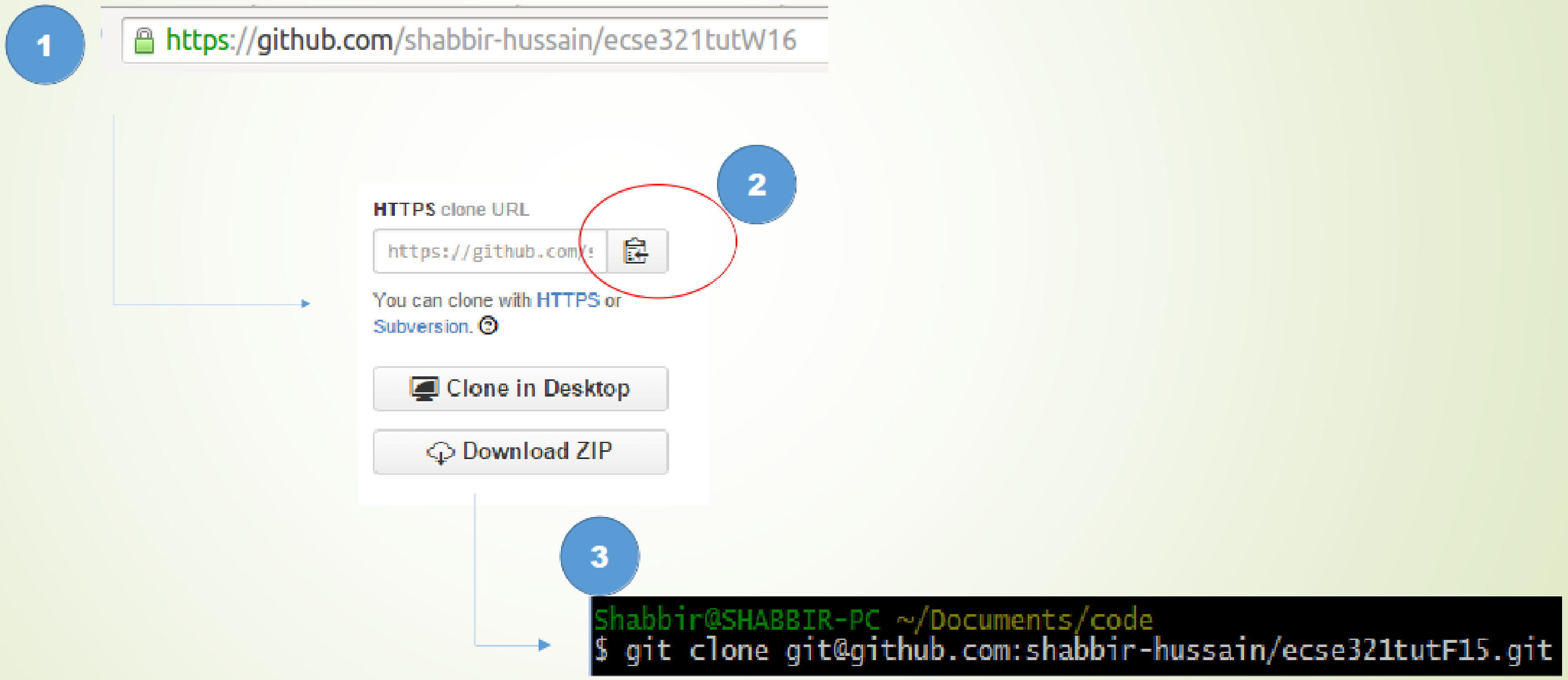


```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git config --global user.email shabbir.hussain@outlook.com
```

3

https://help.github.com/articles/generating-ssh-keys

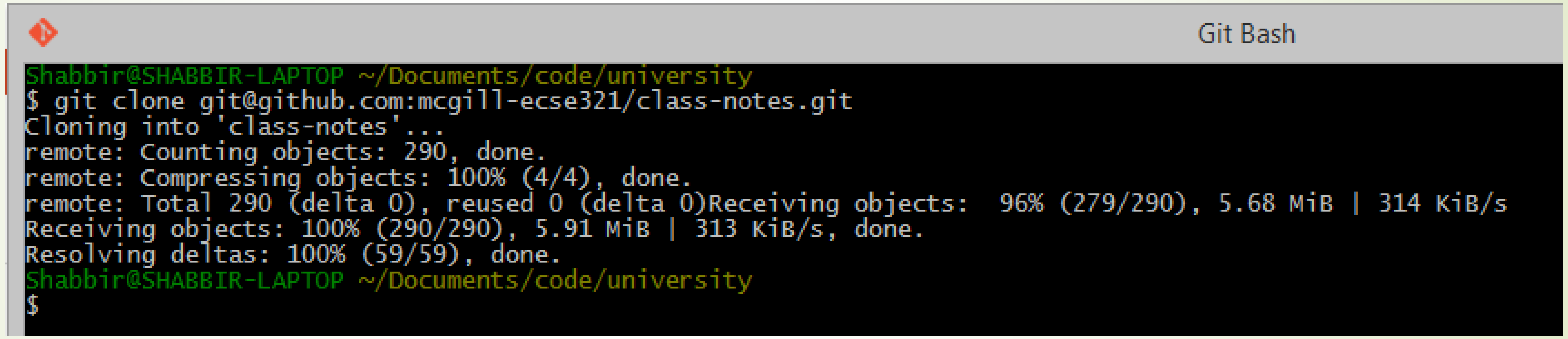
Getting the tutorial notes



https://github.com/shabbir-hussain/ecse321tutW16

```
Shabbir@SHABBIR-PC ~/Documents/code
$ git clone git@github.com:shabbir-hussain/ecse321tutF15.git
```

Done correctly



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$ git clone git@github.com:mcgill-ecse321/class-notes.git
Cloning into 'class-notes'...
remote: Counting objects: 290, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 290 (delta 0), reused 0 (delta 0)Receiving objects:  96% (279/290), 5.68 MiB | 314 KiB/s
Receiving objects: 100% (290/290), 5.91 MiB | 313 KiB/s, done.
Resolving deltas: 100% (59/59), done.
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$
```

Understanding the basics

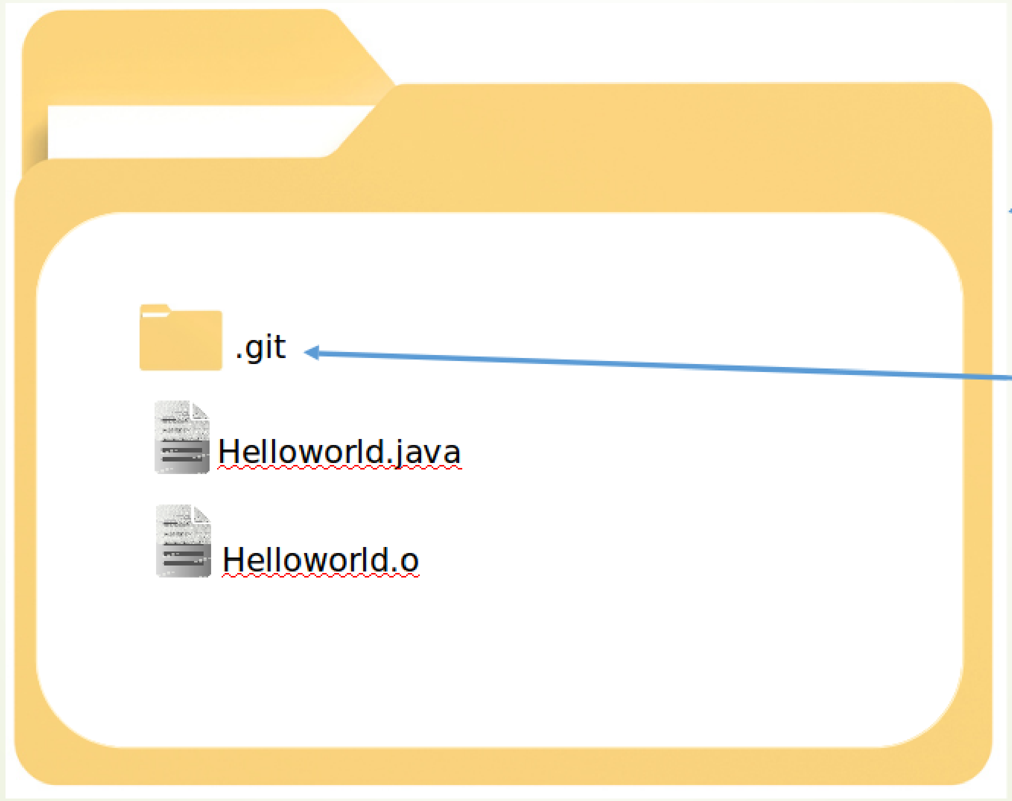
Going from this

To this

Glossary

- **Git** is your **version control software**
- **Git Hub** is a **website** that hosts your repositories
- **Repositories** are a collection of **files** and file **histories**
- **Commits** are like **save states**

What is a repository

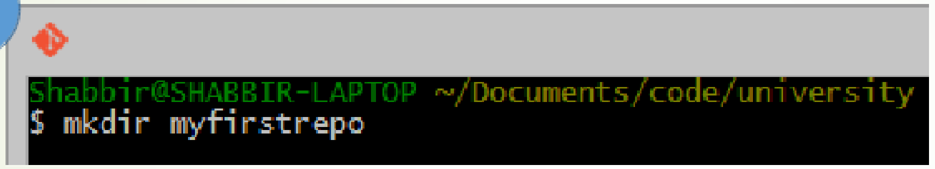


This is what a Repository Looks like

All the data about a repository lives in this folder

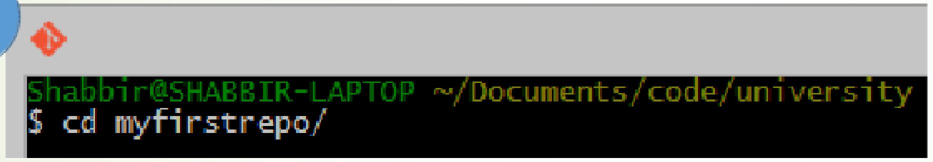
Creating our first Repo

1



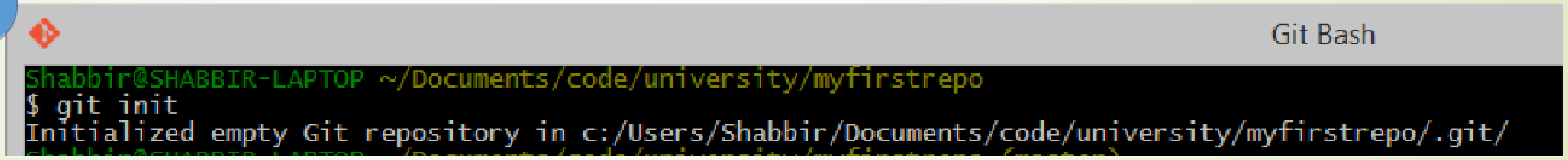
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$ mkdir myfirstrepo
```

2



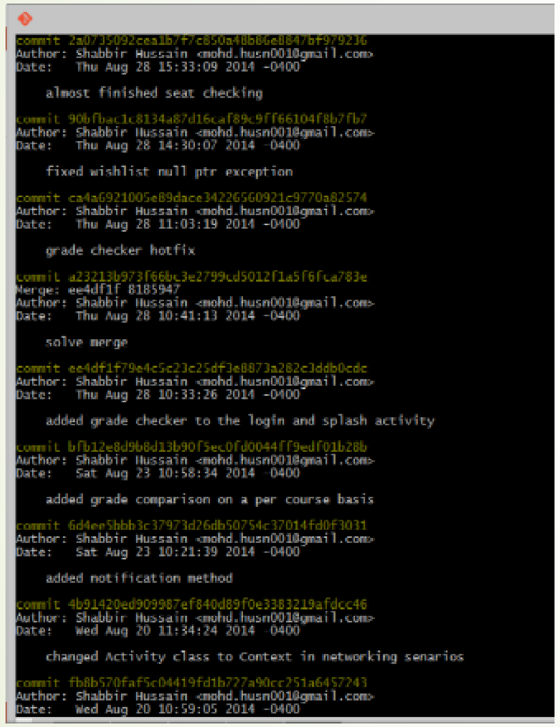
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$ cd myfirstrepo/
```

3



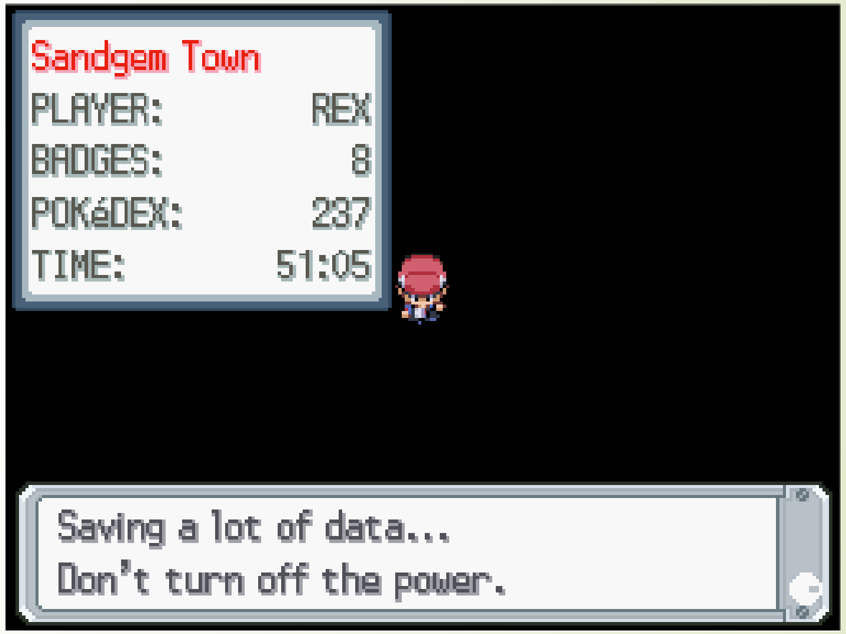
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo
$ git init
Initialized empty Git repository in c:/Users/Shabbir/Documents/code/university/myfirstrepo/.git/
```

What is a commit?



```
commit 2a0735092cea1b7f7c850a48b06e8847bf979236
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Thu Aug 28 15:33:09 2014 -0400

    almost finished seat checking

commit 90bfbac1c8134a87d16caf89c9ff66104f8b7fb7
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Thu Aug 28 14:30:07 2014 -0400

    fixed wishlist null ptr exception

commit ca4a6921005e89dace34226560921c9770a82574
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Thu Aug 28 11:03:19 2014 -0400

    grade checker hotfix

commit a23213b973f66bc3e2799cd5012f1a5f6fca783e
Merge: ee4df1f 8185947
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Thu Aug 28 10:41:13 2014 -0400

    solve merge

commit ee4df1f79e4c5c23c25df3e8873a282c3ddb0cdc
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Thu Aug 28 10:33:26 2014 -0400

    added grade checker to the login and splash activity

commit bfb12e8d9b8d13b90f5ec0fd0044ff9edf01b28b
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Sat Aug 23 10:58:34 2014 -0400

    added grade comparison on a per course basis

commit 6d4ee5bbb3c37973d26db50754c37014fd0f3031
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Sat Aug 23 10:21:39 2014 -0400

    added notification method

commit 4b91420ed909987ef840d89f0e3383219afdcc46
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Wed Aug 20 11:34:24 2014 -0400

    changed Activity class to Context in networking senarios

commit fb8b570faf5c04419fd1b727a90cc251a6457243
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Wed Aug 20 10:59:05 2014 -0400
```

Commits



Sandgem Town
PLAYER: REX
BADGES: 8
POKéDEX: 237
TIME: 51:05

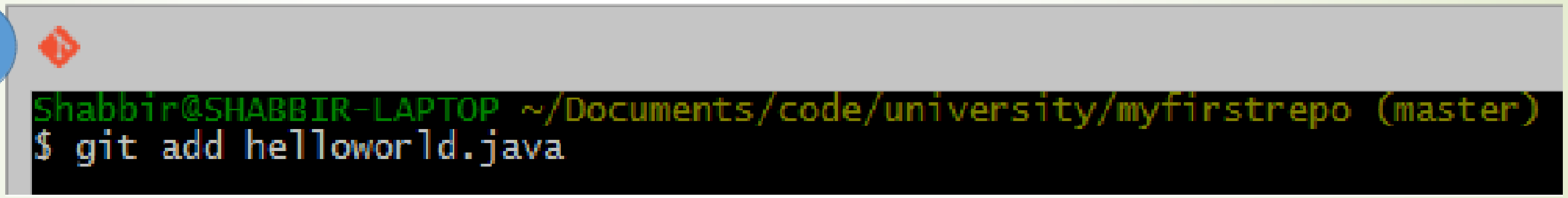
Saving a lot of data...
Don't turn off the power.

Save states

Creating and committing a file

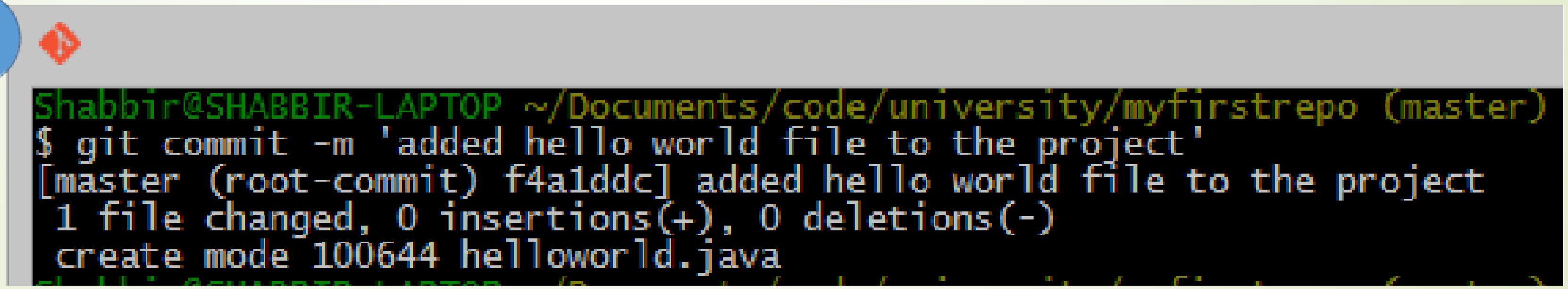
1 Create a file

2



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git add helloworld.java
```

3



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git commit -m 'added hello world file to the project'
[master (root-commit) f4a1ddc] added hello world file to the project
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 helloworld.java
```

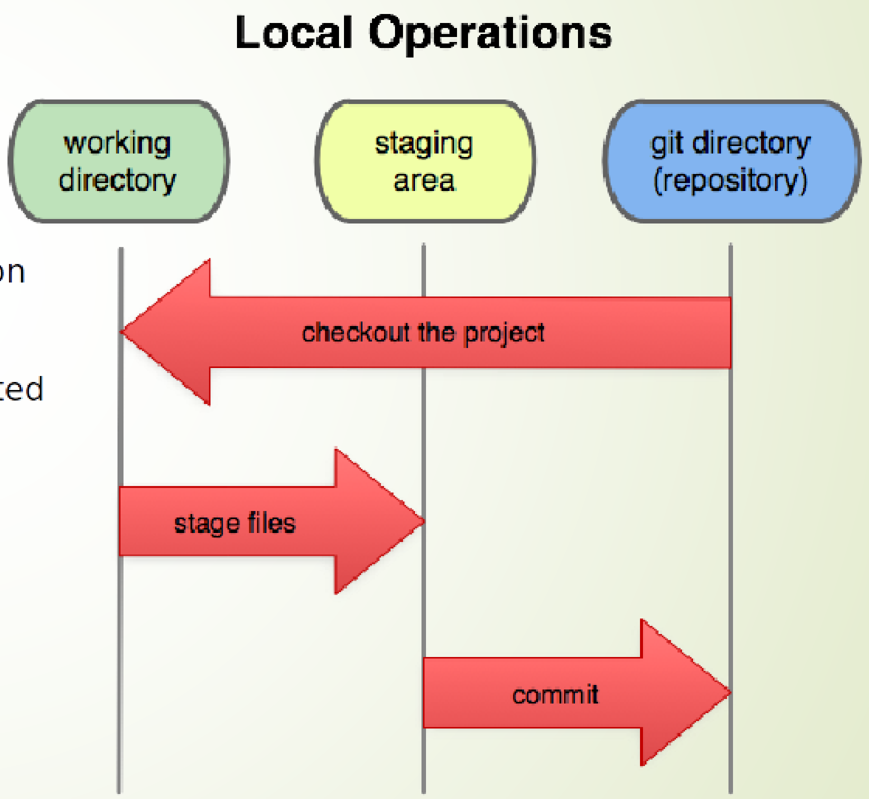
Inside a Repository

•Working Directory: Files being worked on right now

•Staging area: Files ready to be committed

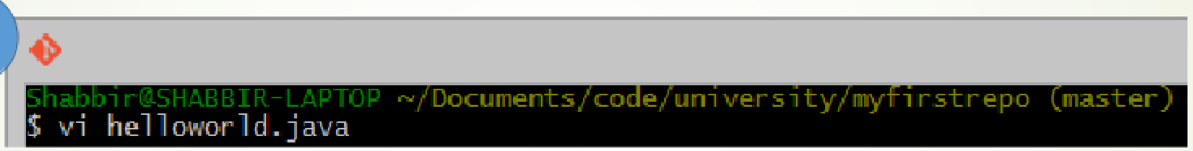
•Repository: A collection of commits

Why should we stage files instead of directly committing them?



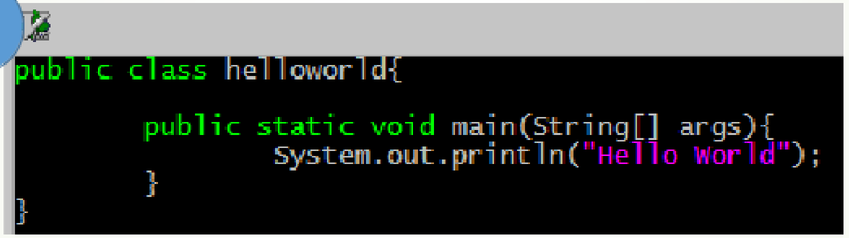
File status update

1



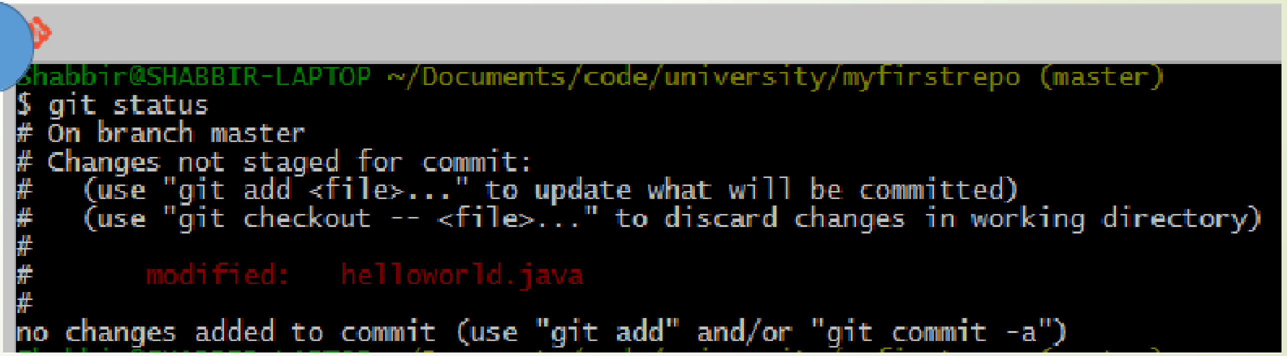
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ vi helloworld.java
```

2



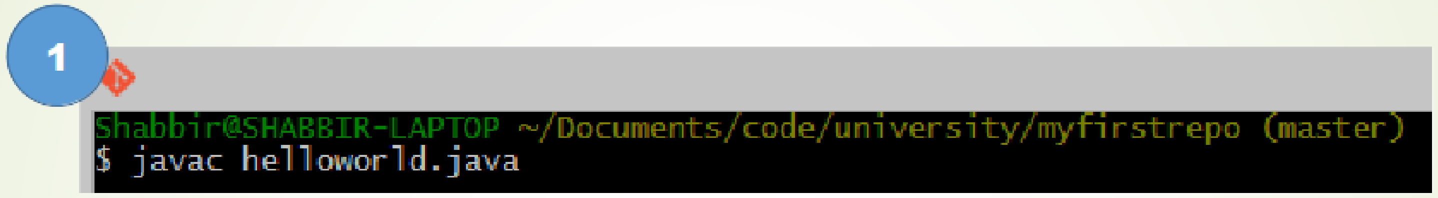
```
public class helloworld{

        public static void main(String[] args){
                System.out.println("Hello World");
        }
}
```

3

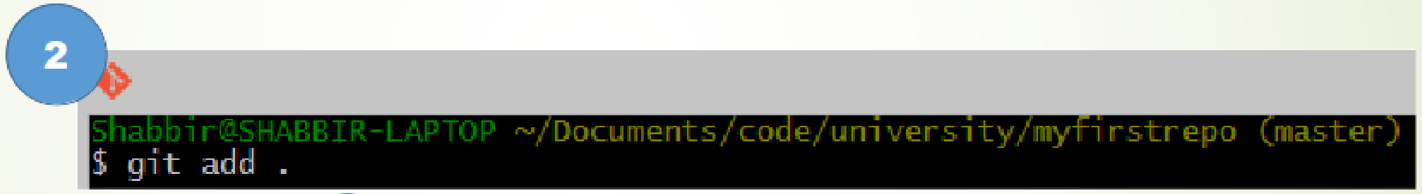


```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   helloworld.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

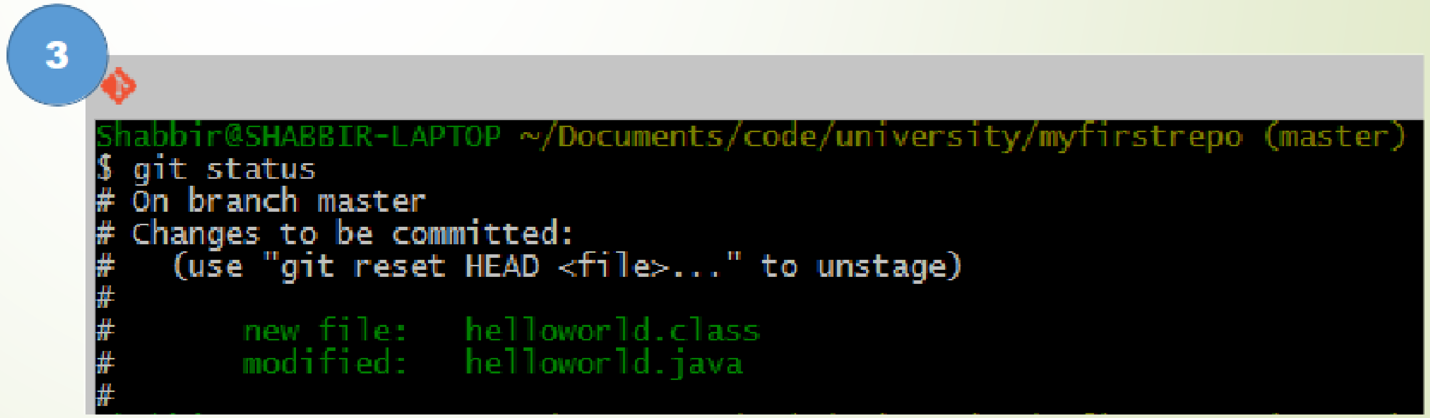
Staging and Unstaging files



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ javac helloworld.java
```

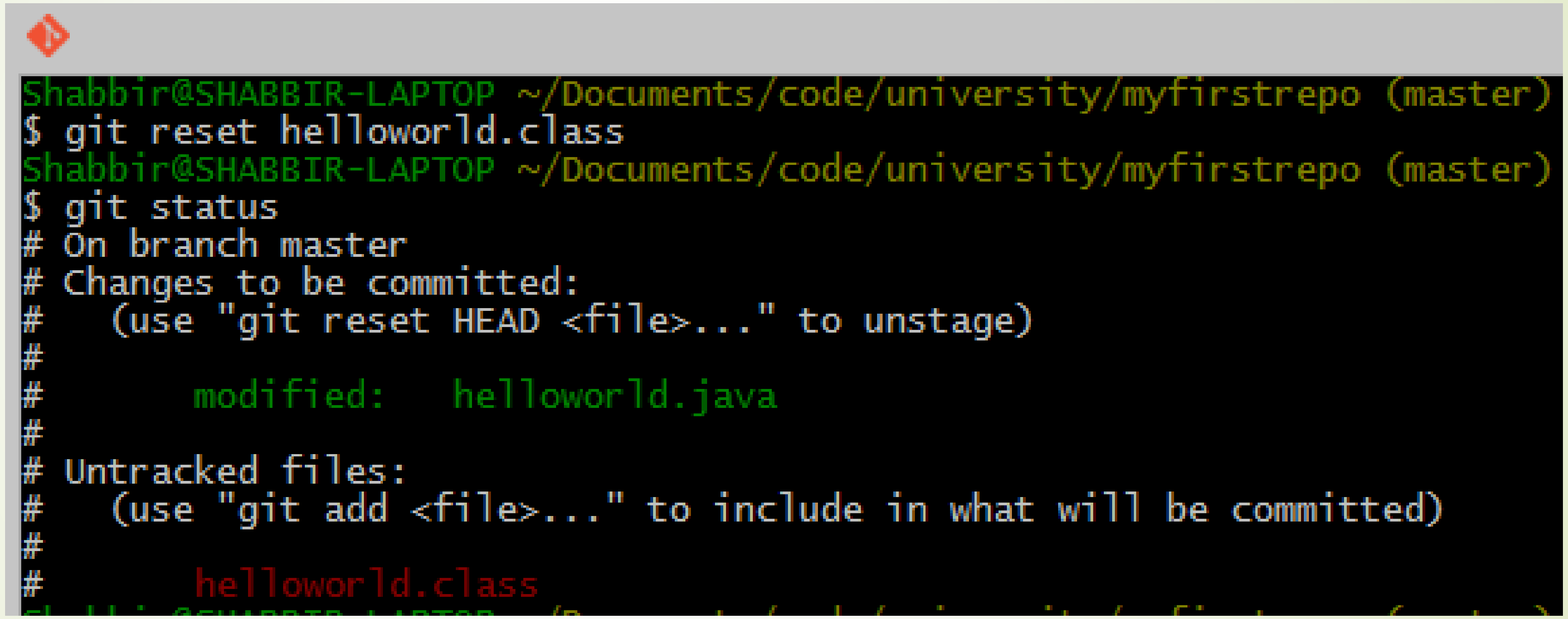


```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git add .
```



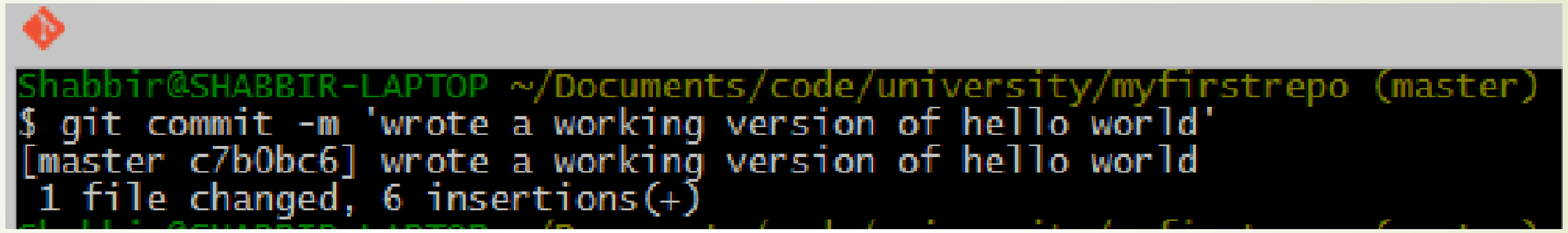
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   helloworld.class
#       modified:   helloworld.java
#
```

Staging and unstaging files



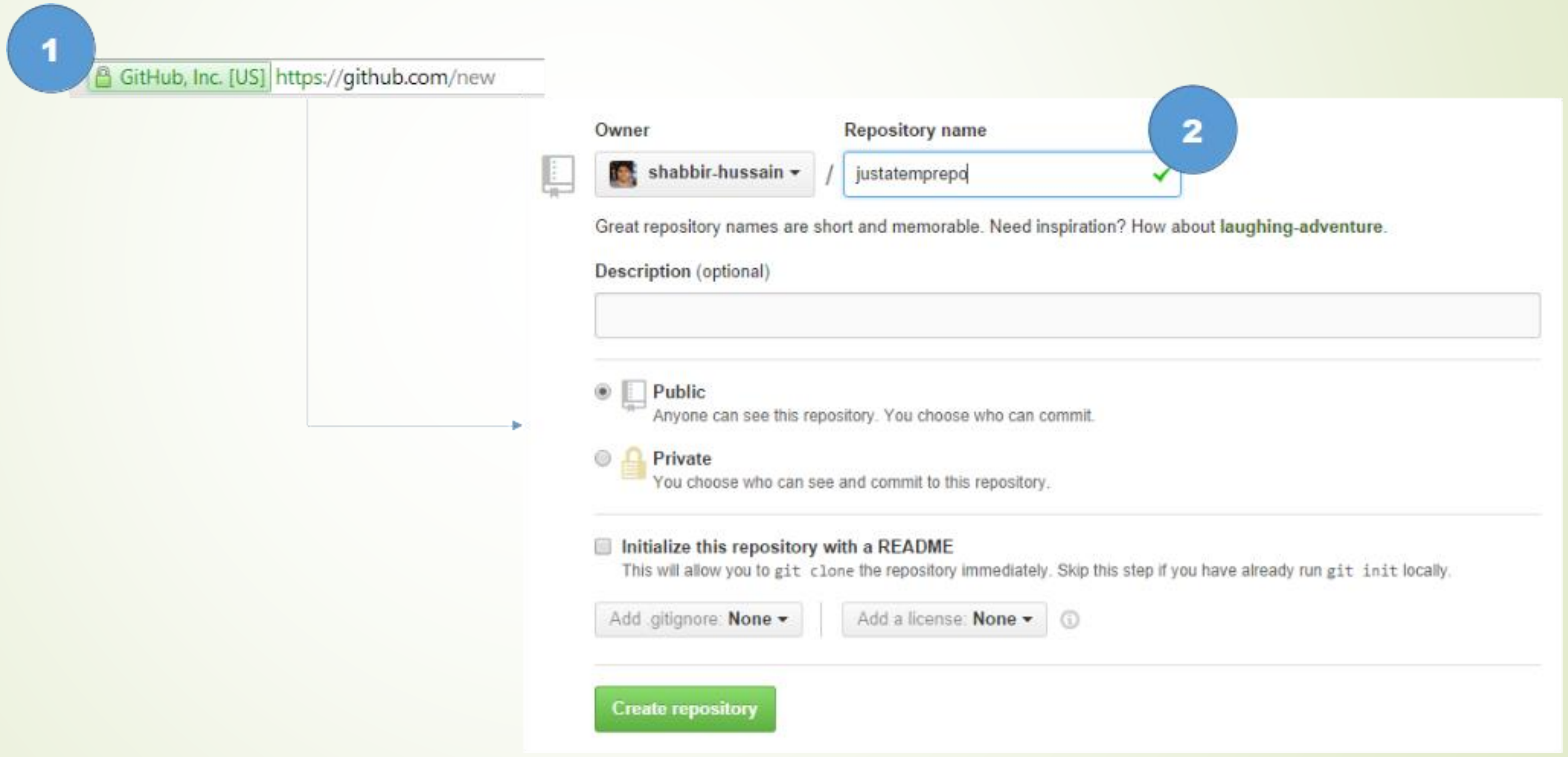
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git reset helloworld.class
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   helloworld.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       helloworld.class
```

Committing (cont.)


```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git commit -m 'wrote a working version of hello world'
[master c7b0bc6] wrote a working version of hello world
 1 file changed, 6 insertions(+)
```

Only Staged files are committed.

Putting our code in the cloud



GitHub, Inc. [US] https://github.com/new

Owner: shabbir-hussain / Repository name: justatemprepo

Great repository names are short and memorable. Need inspiration? How about laughing-adventure.

Description (optional)

[✓ Public] Anyone can see this repository. You choose who can commit.

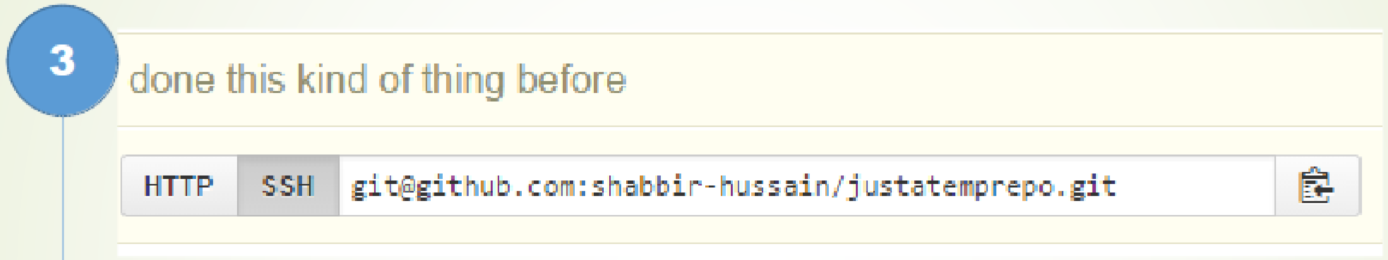
Private — You choose who can see and commit to this repository.

Initialize this repository with a README — This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: None | Add a license: None

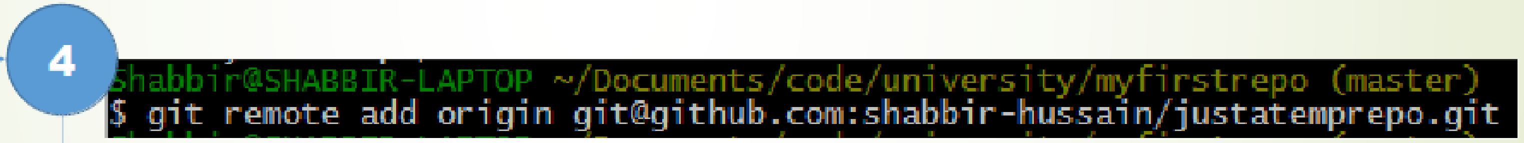
Create repository

Putting our code in the cloud (cont.)

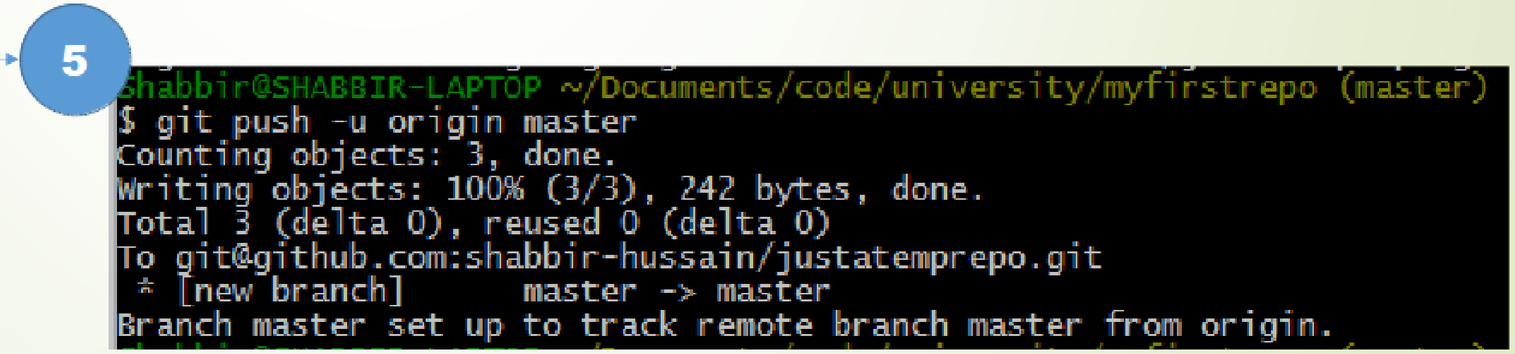


done this kind of thing before

HTTP SSH `git@github.com:shabbir-hussain/justatemprepo.git`

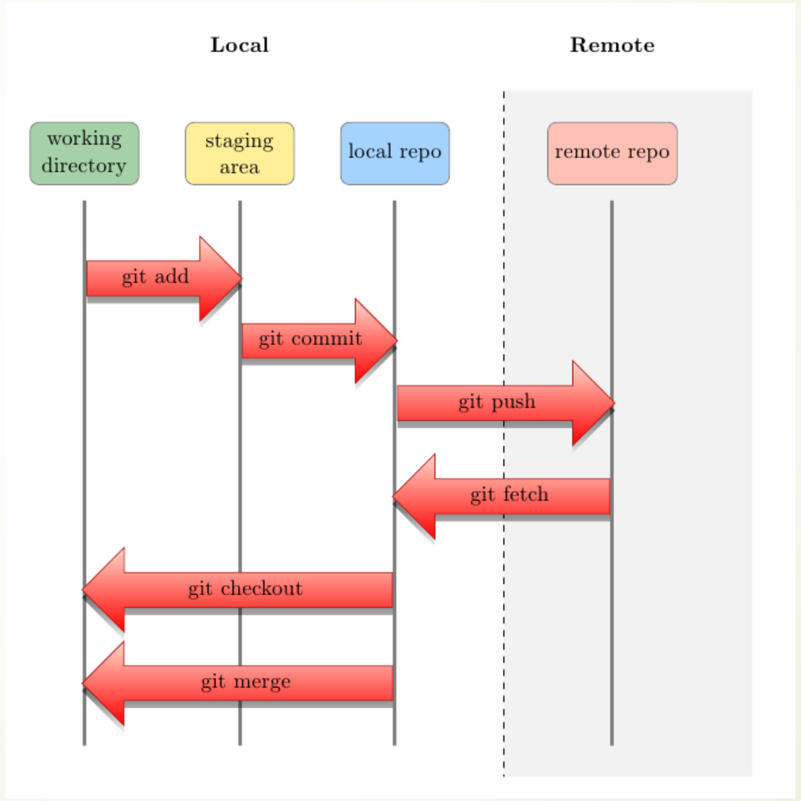


```
shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git remote add origin git@github.com:shabbir-hussain/justatemprepo.git
```



```
shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 242 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:shabbir-hussain/justatemprepo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Revisiting the model



Extra Resources

- Try github Interactive Tutorial: https://try.github.io/levels/1/challenges/1
- Visual Explanation: http://www.wei-wang.com/ExplainGitWithD3/

References

- Vogella Reference: http://www.vogella.com/tutorials/Git/article.html
- SSH keys tutorial: https://help.github.com/articles/generating-ssh-keys
- Git Cheetsheet: https://raw.githubusercontent.com/nerdgirl/git-cheatsheet-visual/master/gitcheatsheet.png
- Dominic's Tutorial slides (Previous TA): http://slides.com/dominiccharleyroy/tutorial-1-git#/

See you next week!