Introduction to Git

Download GitHub Desktop & Create Account

Google “download github desktop”

https://desktop.github.com/

Click the download link for you operating system

Google “GitHub”

https://github.com/

Create account

You could use Bitbucket and Source Tree instead, but I will be using GitHub and GitHub Desktop

Why are we learning the GUI?

GUI = Graphical User Interface = GitHub Desktop, Source Tree

There are a lot of people in industry that use GUIs and a lot that use the terminal

Some feel strongly about using the terminal

Learning the terminal is too much to cover in an thirty minutes

So we’re using the GUI

Create a Repository

Create a **remote repository** in GitHub on the web

GitHub.com is where everyone's collective changes are at (remote repository)

On your personal machine is where *your* changes are at (**local repository**)

You and your group members all have your own local repository

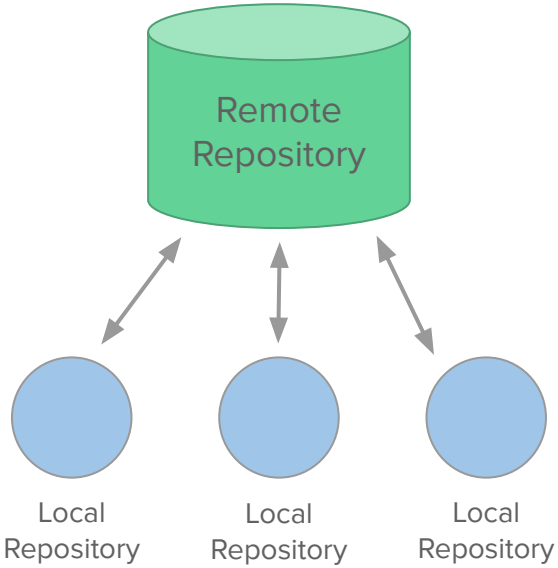
You will put your changes on the remote repository by **merging**

Clone the Repository

Add your group members to the repository

Clone the repository you (or your group member) just created

This is your **local repository**



Branches

Branch **master** is where the current project is at

Everyone in your group should create their own branch
to work on

You and your group members should only work on your
individual branches

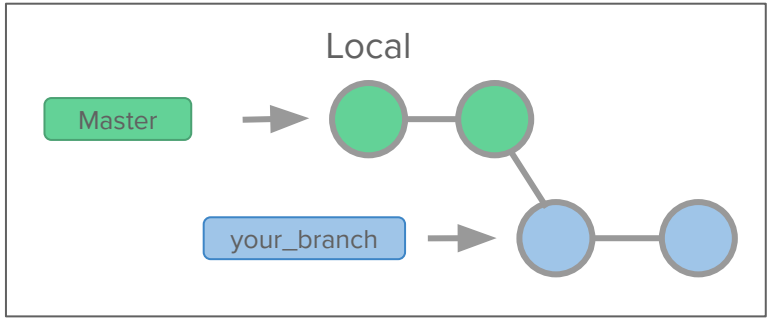
When your changes are ready to **merge** with the current project you merge your branch and the master branch

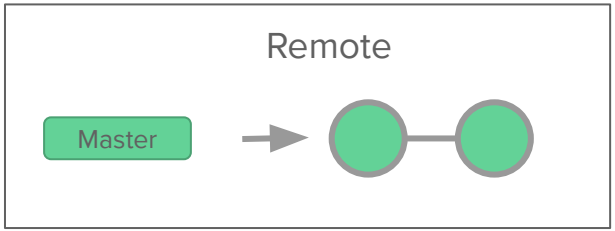
Creating a branch

How to create a branch

Where is your branch at right now?

Let’s add it to the remote repo





Make changes on your branch

Let’s add a file

Create new text file, put some stuff in that text file, save the file

Go back to the GUI

Notice that you have a change now

Commit the changes

Commits include a message that should be descriptive of what you did:
“Fixed bug <describe bug> by updating <describe the changes you made>”

Merging changes

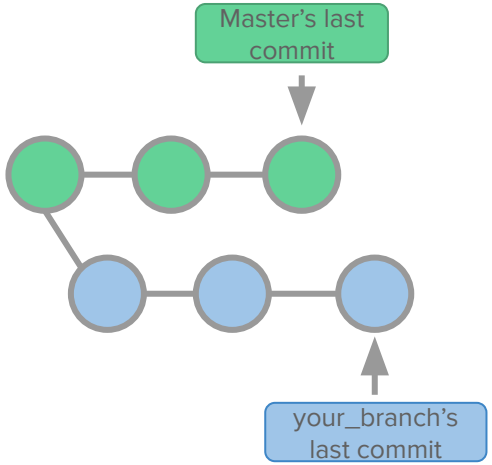
Update the master branch to reflect the changes made in your branch

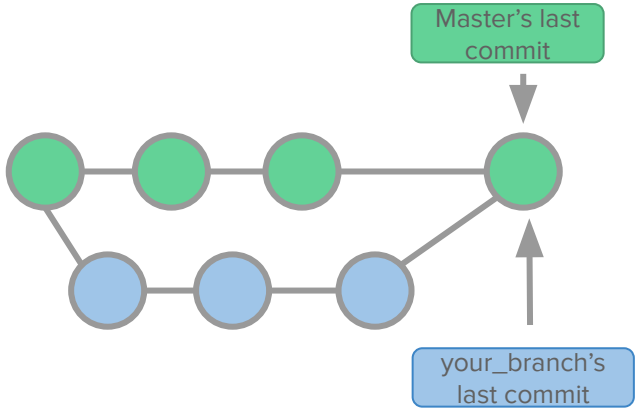
First compare the branches to see if you will have any conflicts

Click “History” and note that your branch is *ahead* by one commit

There are no conflicts, so now we will switch to master branch and merge

Now your local version of master has your changes

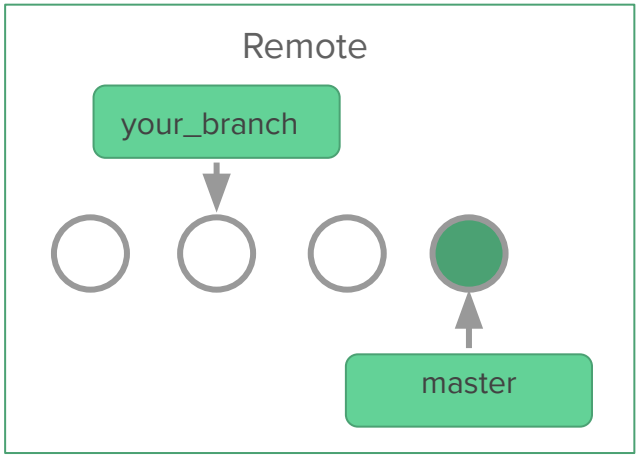


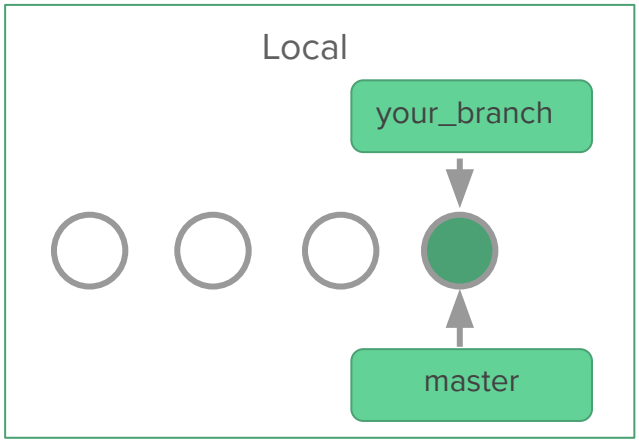


Pushing

We **push** things to the remote repository so others can see and use our changes

Click “Push origin” on master branch and your own branch





Master has been changed

You should not make changes directly on master

BUT...to simulate a file change we're going to do it just this once.

Update the text file, commit, and push master

Change back to your branch and change the same text file

Commit your changes to your branch and push them

Merge conflicts

Compare your branch with master

Notice the warning that tells you there will be a **merge conflict**

Deciphering a conflict

```
<<<<<<<<< HEAD
Your branch's content
==================
Other branch's content
>>>>>>>>>> other_branch
```

Note: this will appear in the *actual file*

1. Remove <<<<<Head, =====, and >>>>master lines
2. Merge the text manually
3. Save
4. Commit
5. Finish merging (appears to be automatic on GUI)

Note: this should NOT be done on master

Switch to master and merge

Questions?