http://tinyurl.com/MDW-Lesson1

Intro to GitHub

An open community of software developers

Adapted from the full tutorial at:
http://www.tutorialspoint.com/git/git_tutorial.pdf

GitHub: https://github.com/

Distributed Version Control System (DVCS)

A copy of your software exists in two (or more) places - on your own computer, and on the GitHub server.

A copy also exists on any computer who has “cloned” your software - maybe another computer owned by you, maybe another person’s computer

Why Git?

Multiple people can work on the same code at the same time

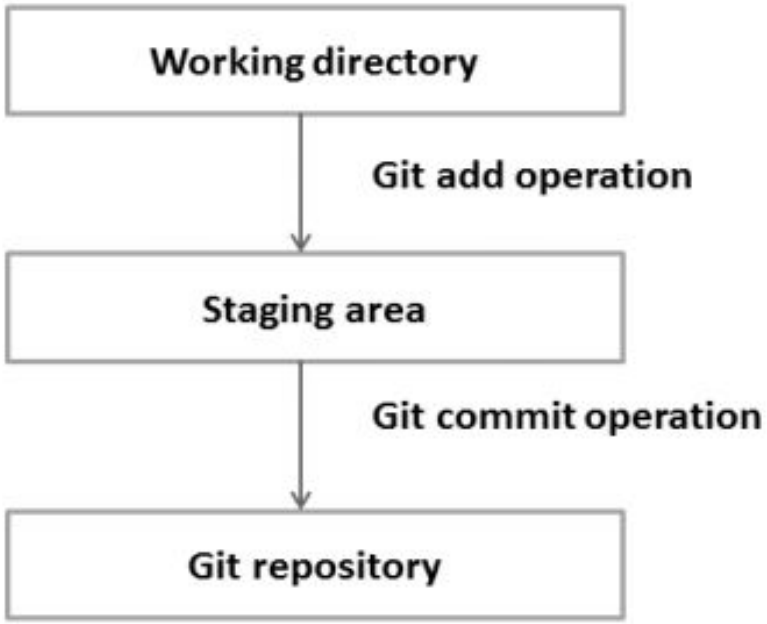
Conflicts are highlighted, and the owner of the code is made aware of them

All changes are recorded, and you can go back to old versions

You can work on “new features” of a project, AND work on the main project, without conflicts (this is called “branching”)

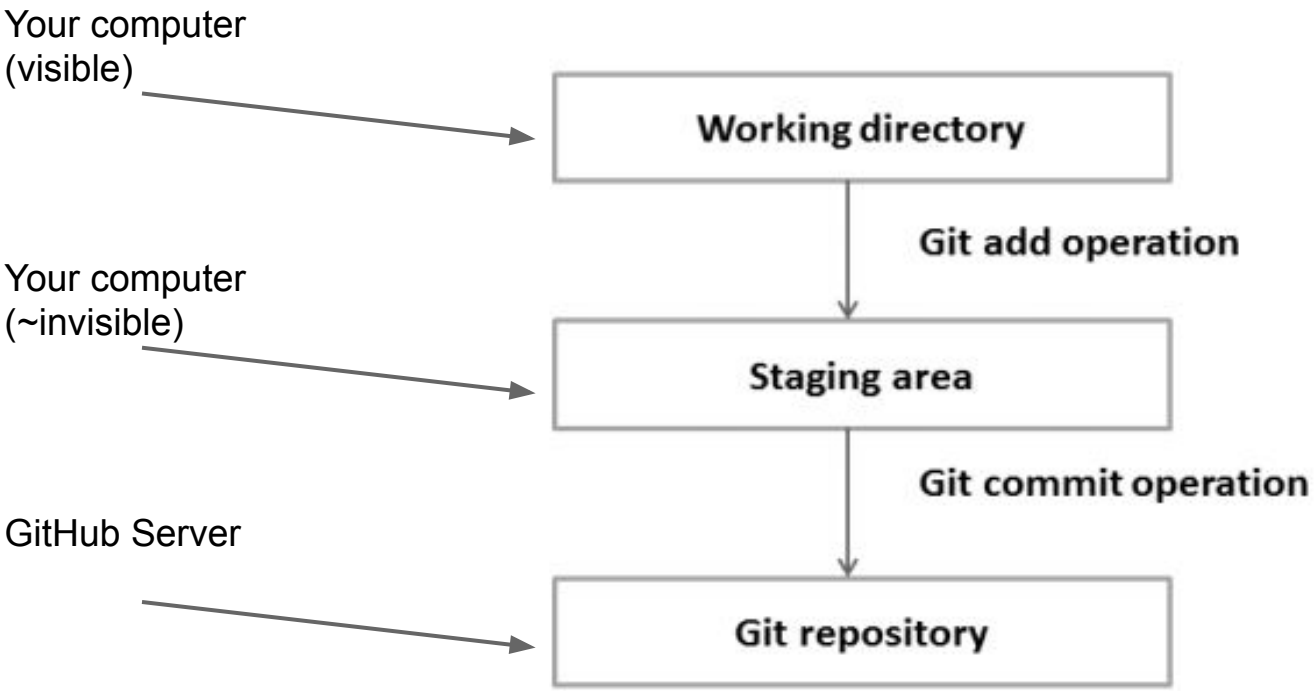
When the new features are finished, you merge them into the main project (“merging”)

3 “layers” of Git



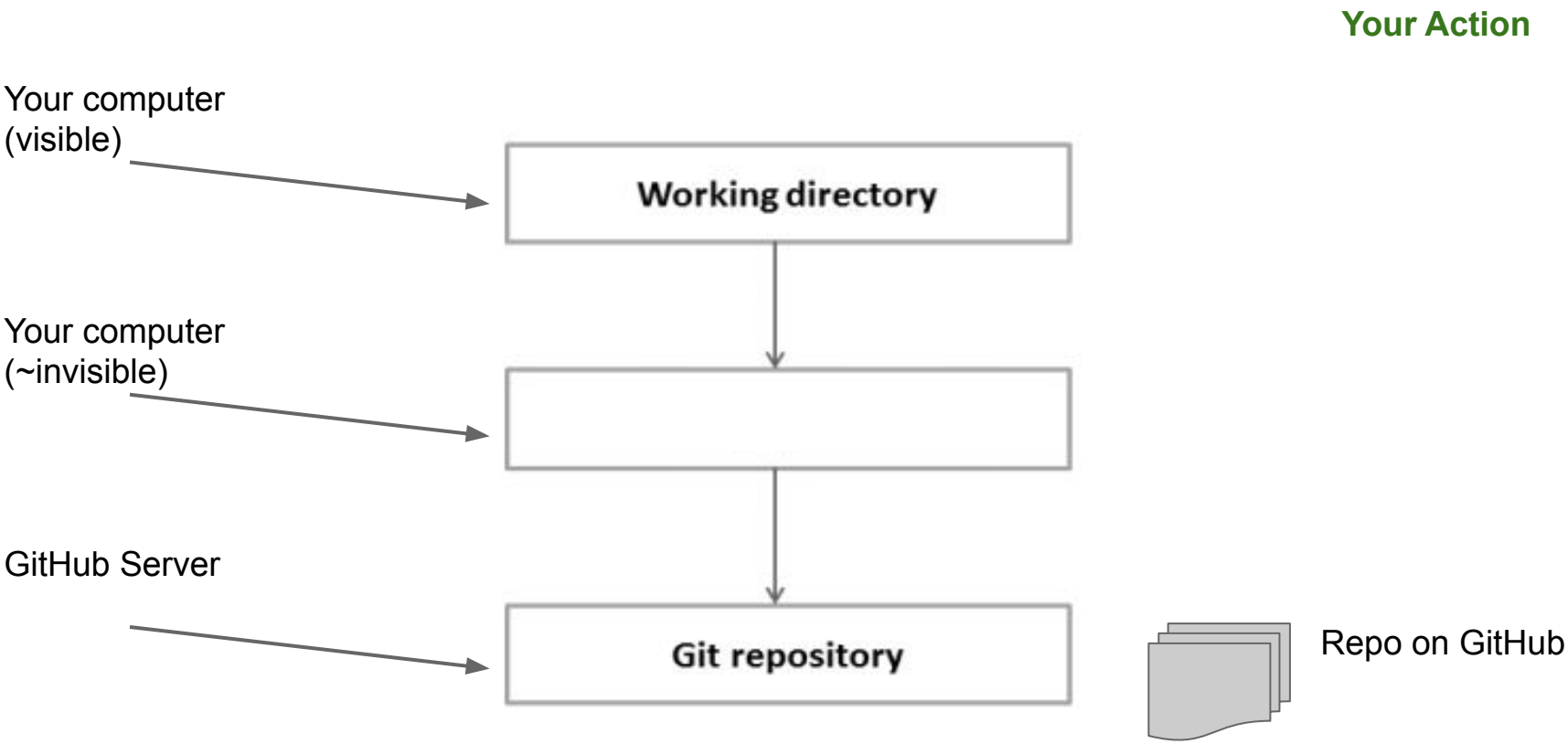
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



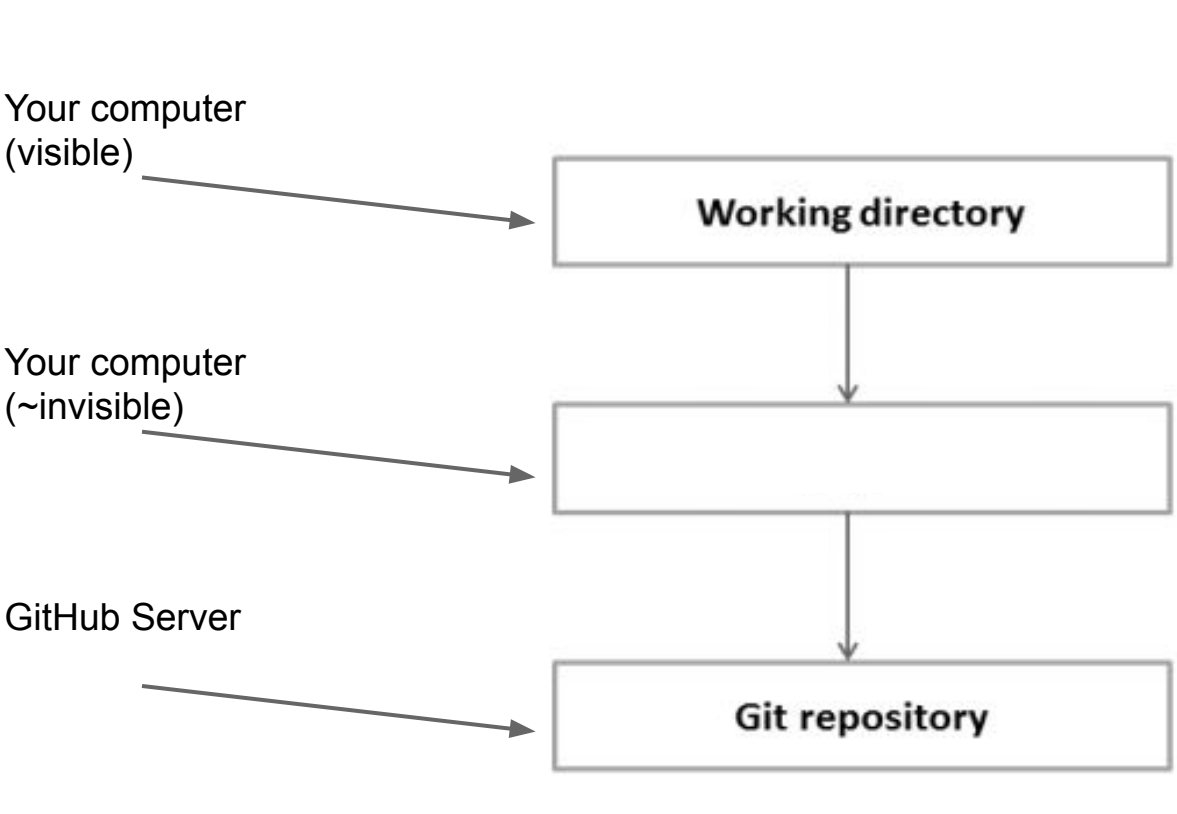
http://www.tutorialspoint.com/git/git_tutorial.pdf

“Cloning” a repository



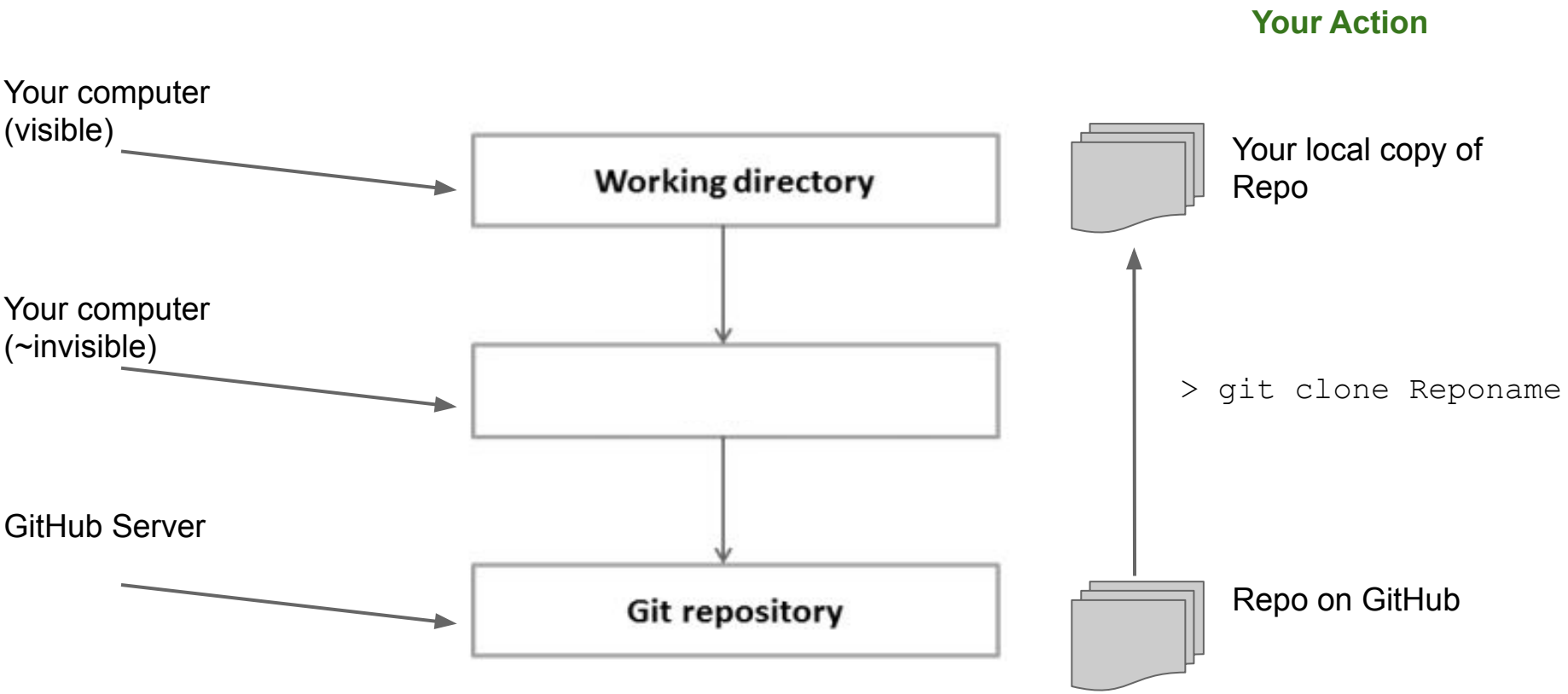
http://www.tutorialspoint.com/git/git_tutorial.pdf

“Cloning” a repository



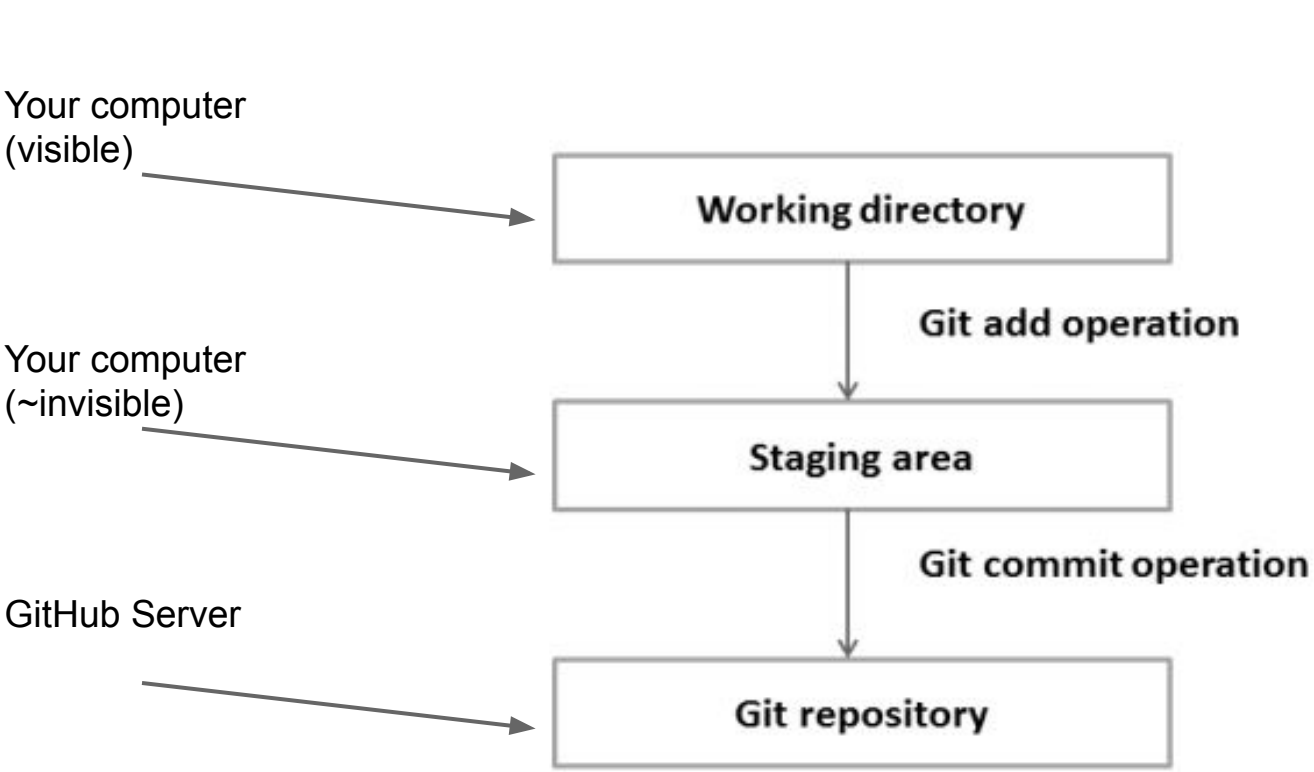
http://www.tutorialspoint.com/git/git_tutorial.pdf

“Cloning” a repository



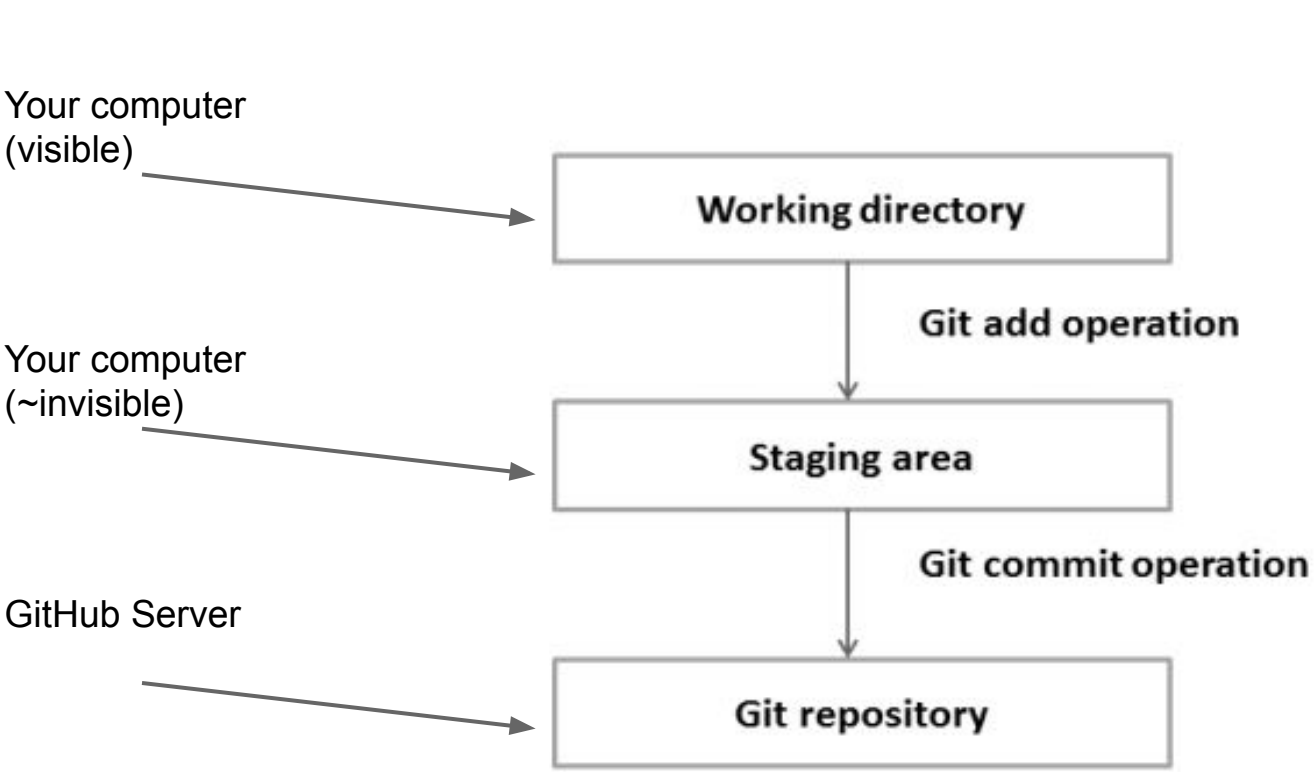
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git

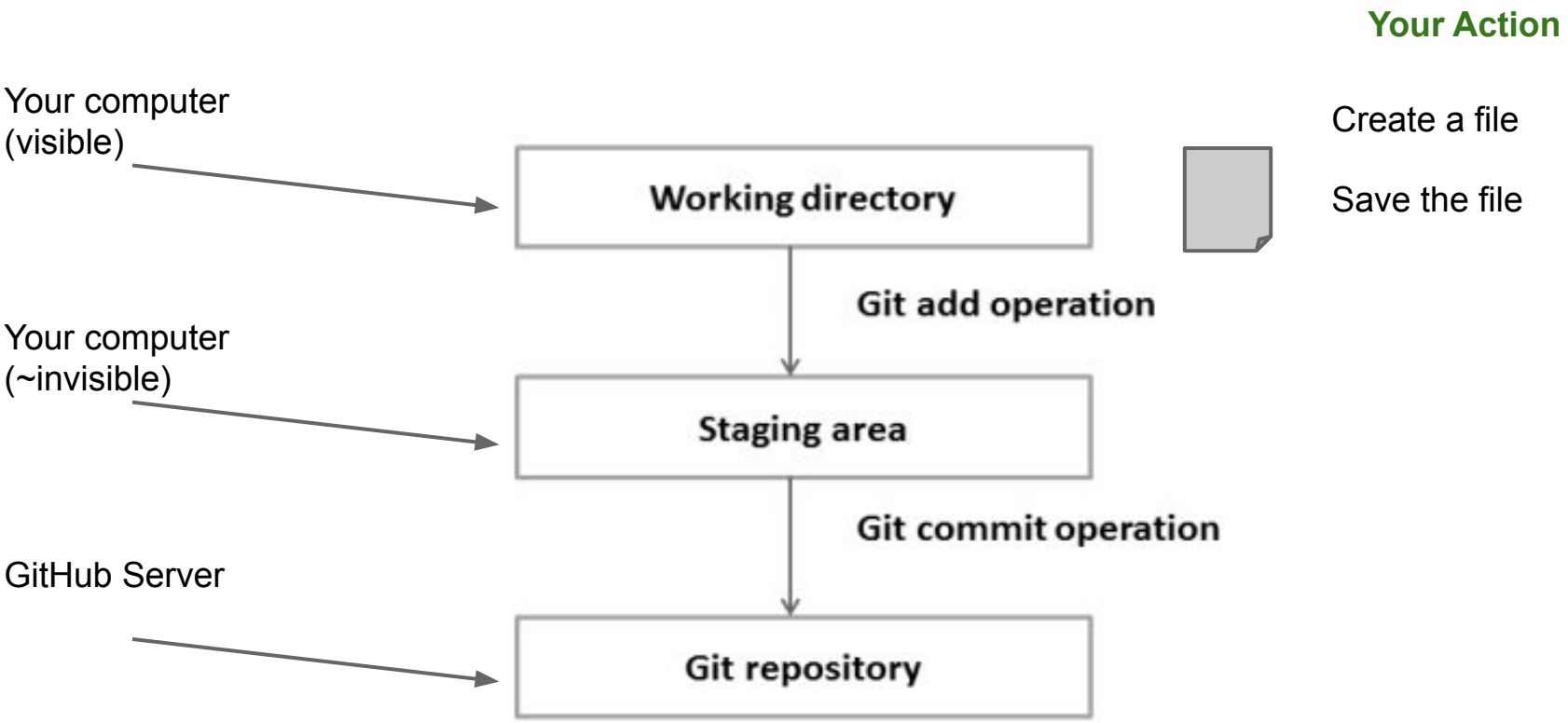


Your Action

Create a file

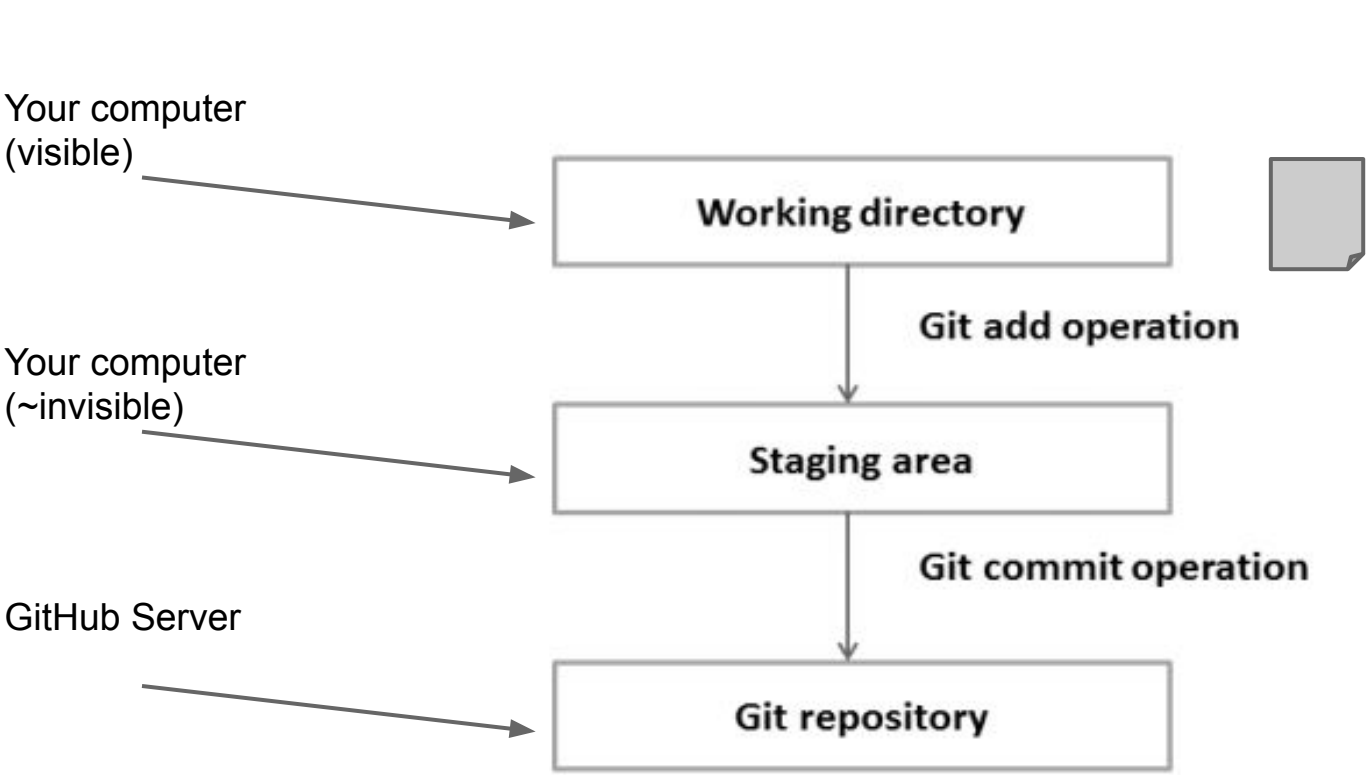
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



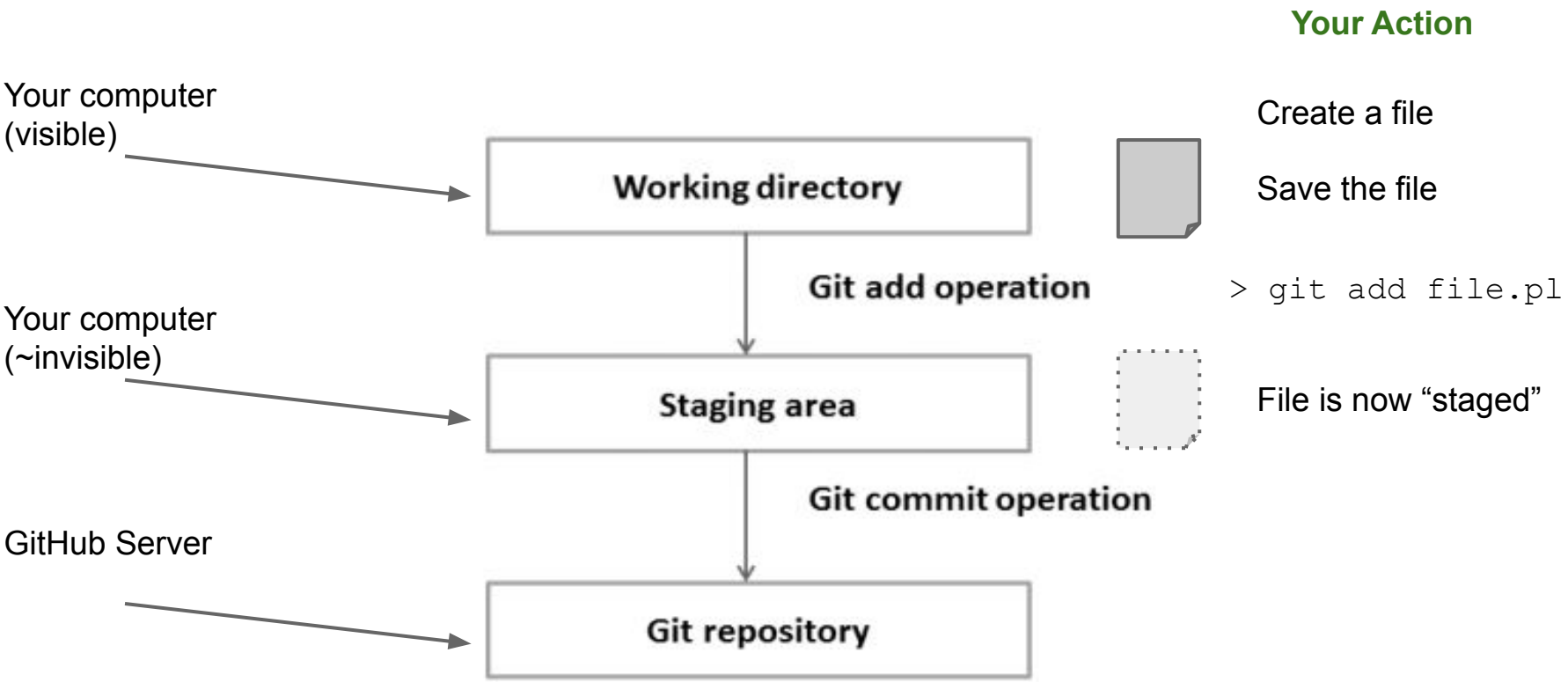
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



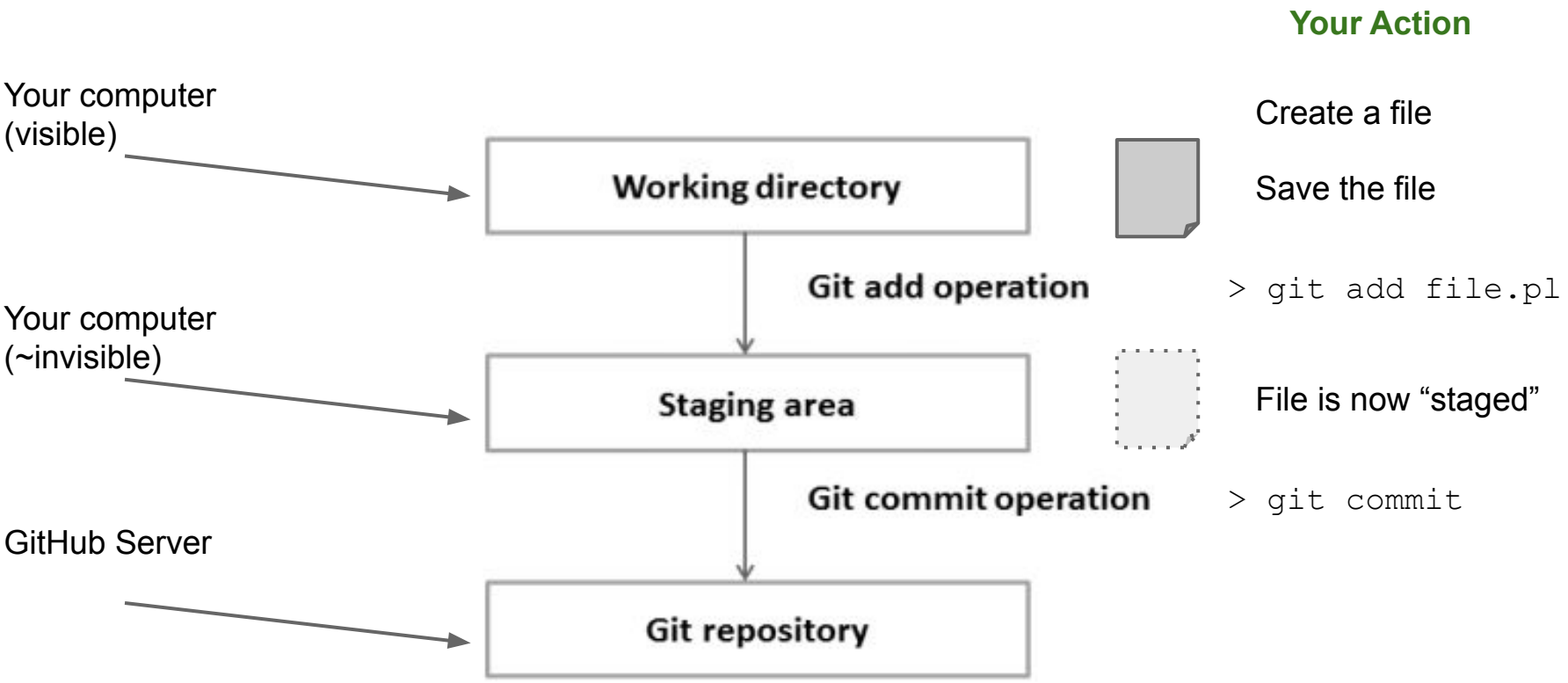
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



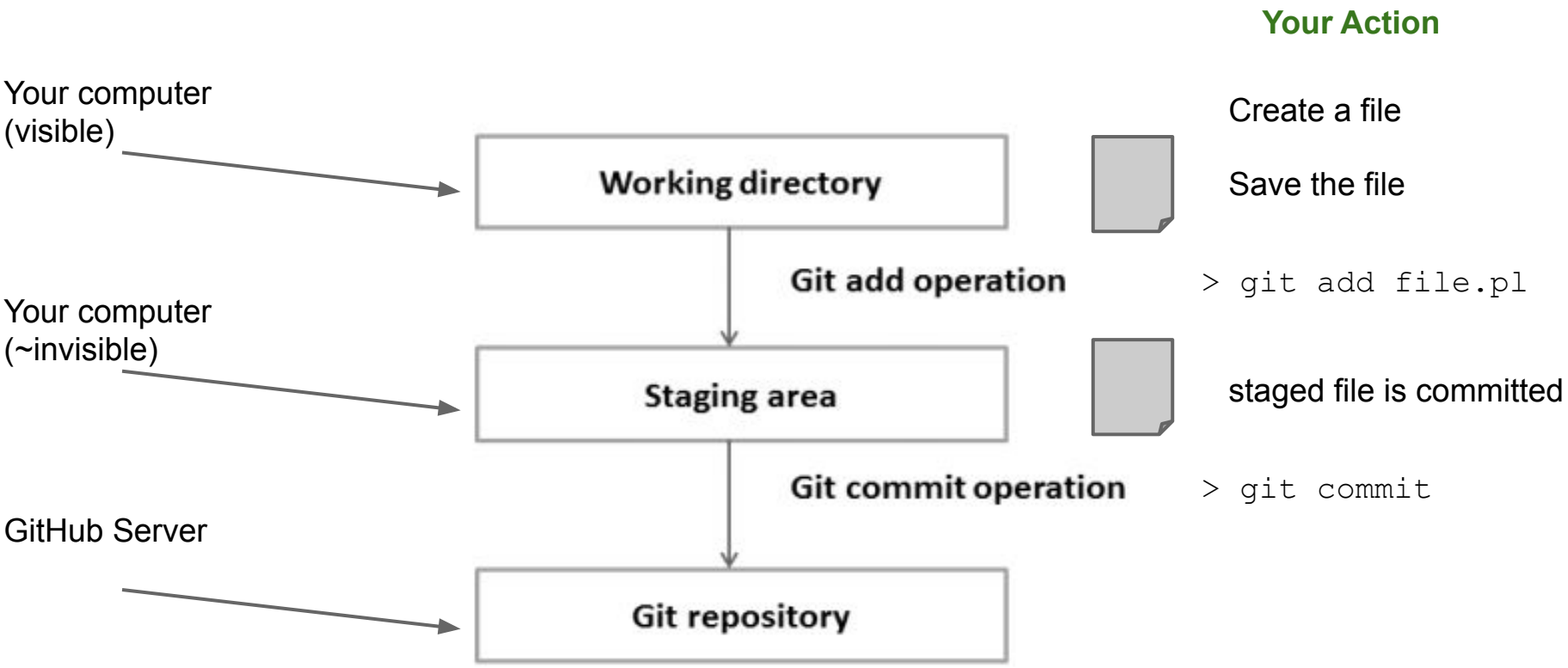
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



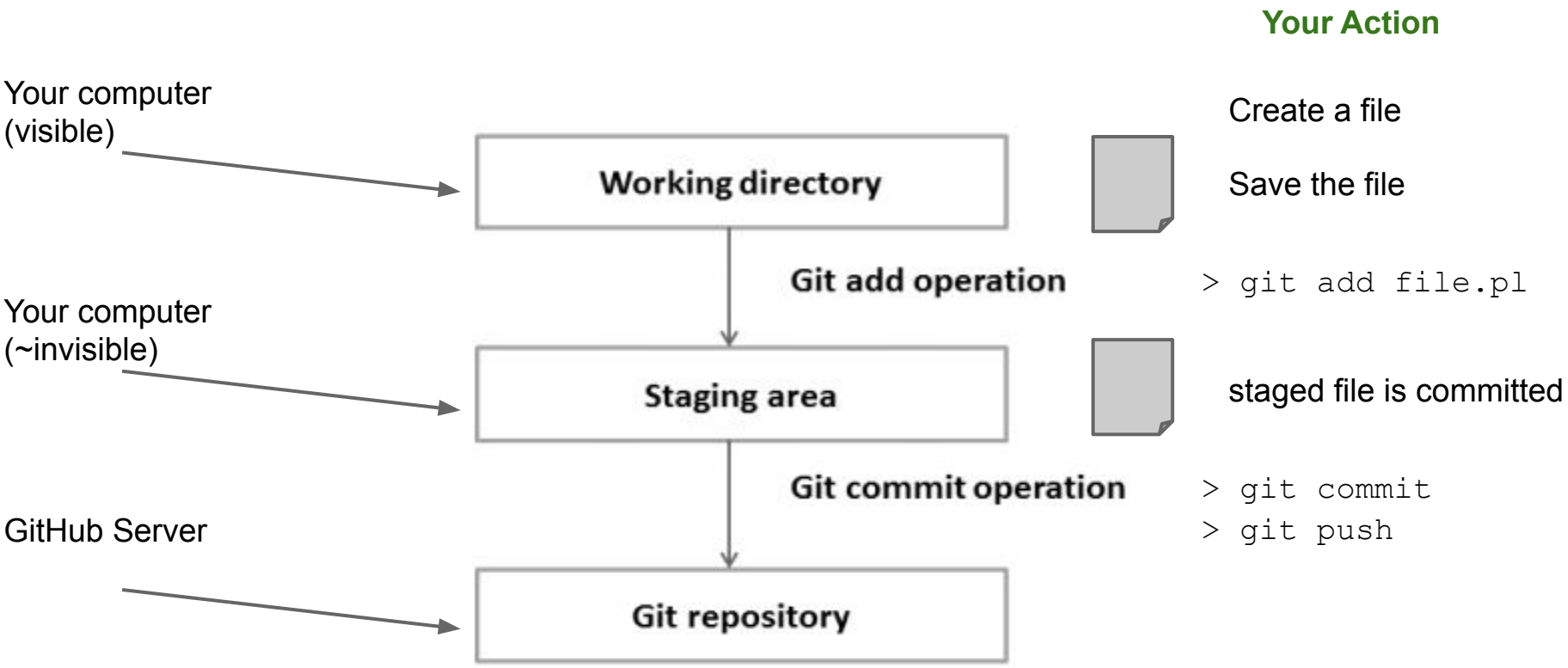
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



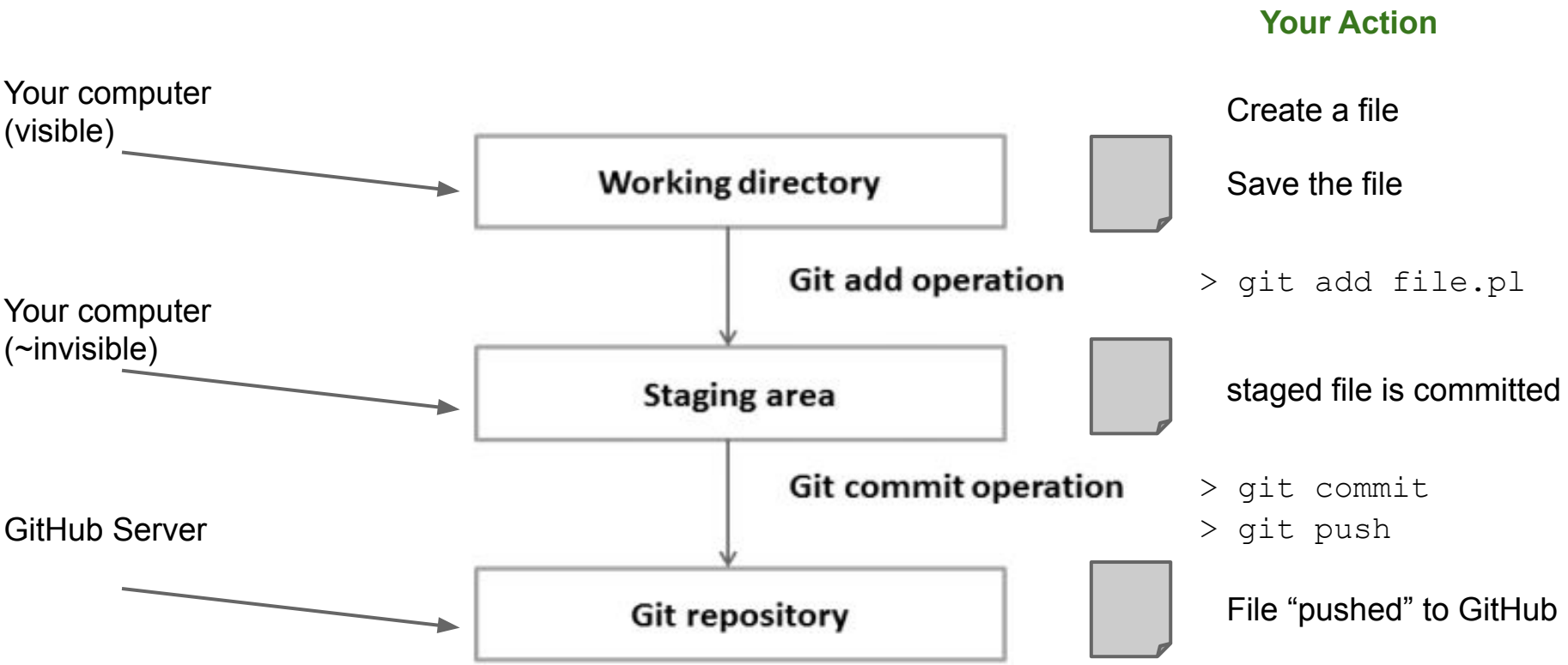
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



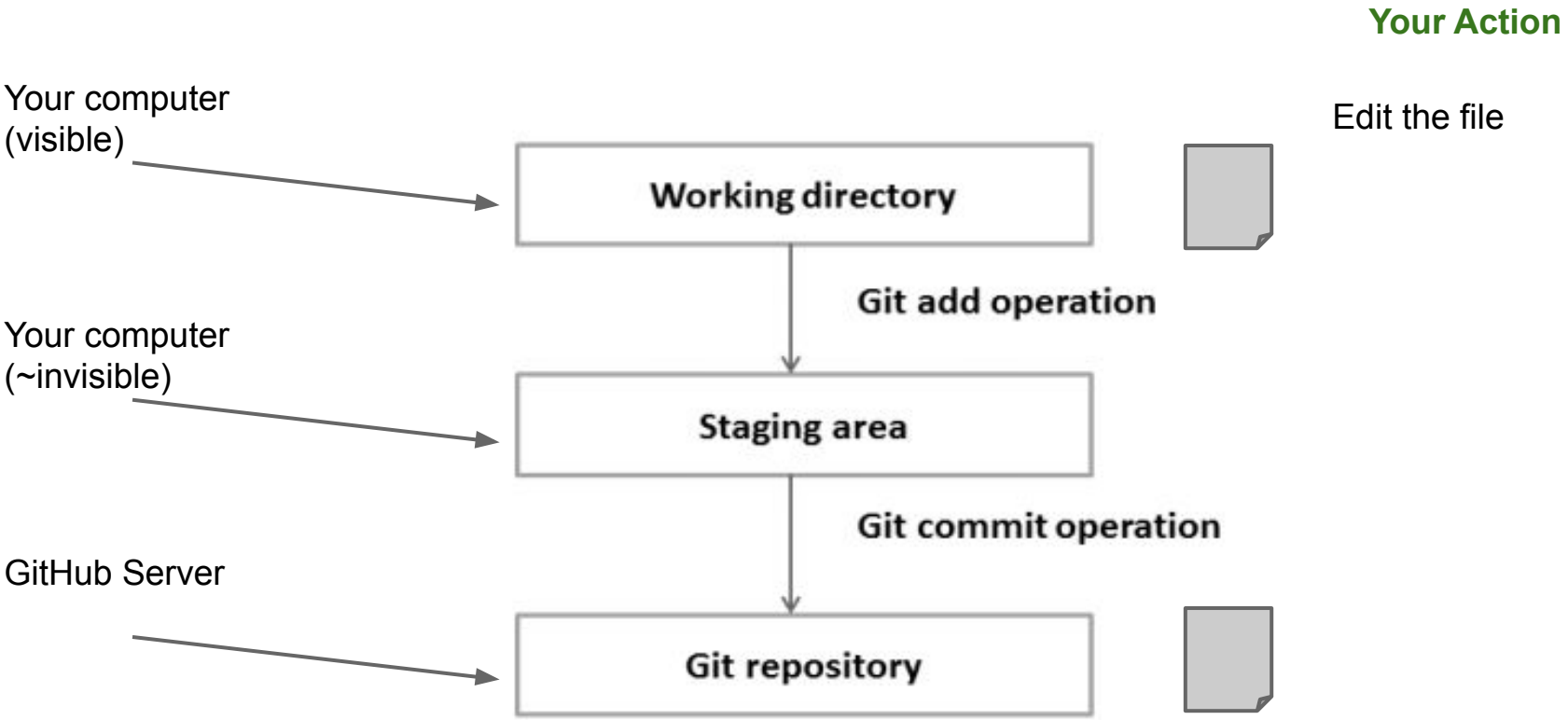
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



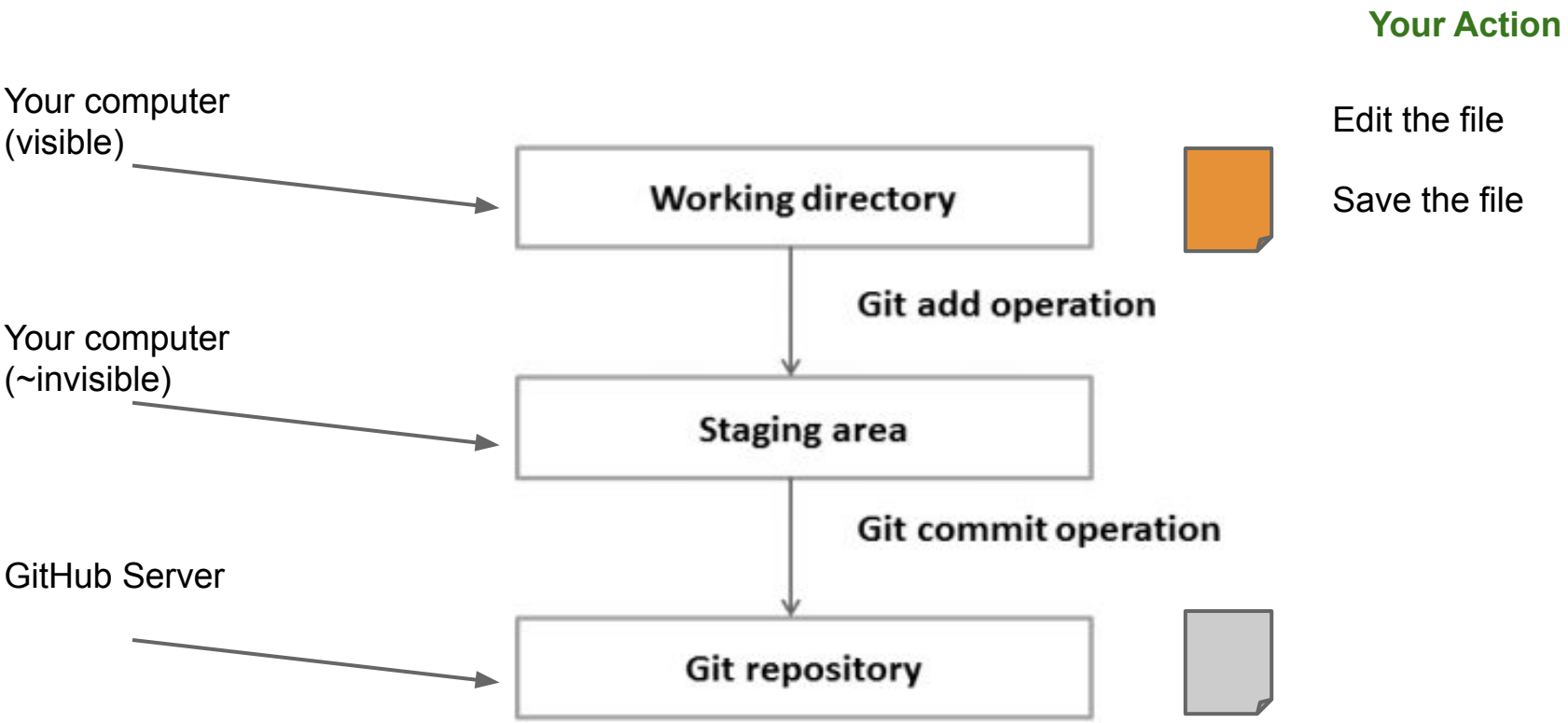
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



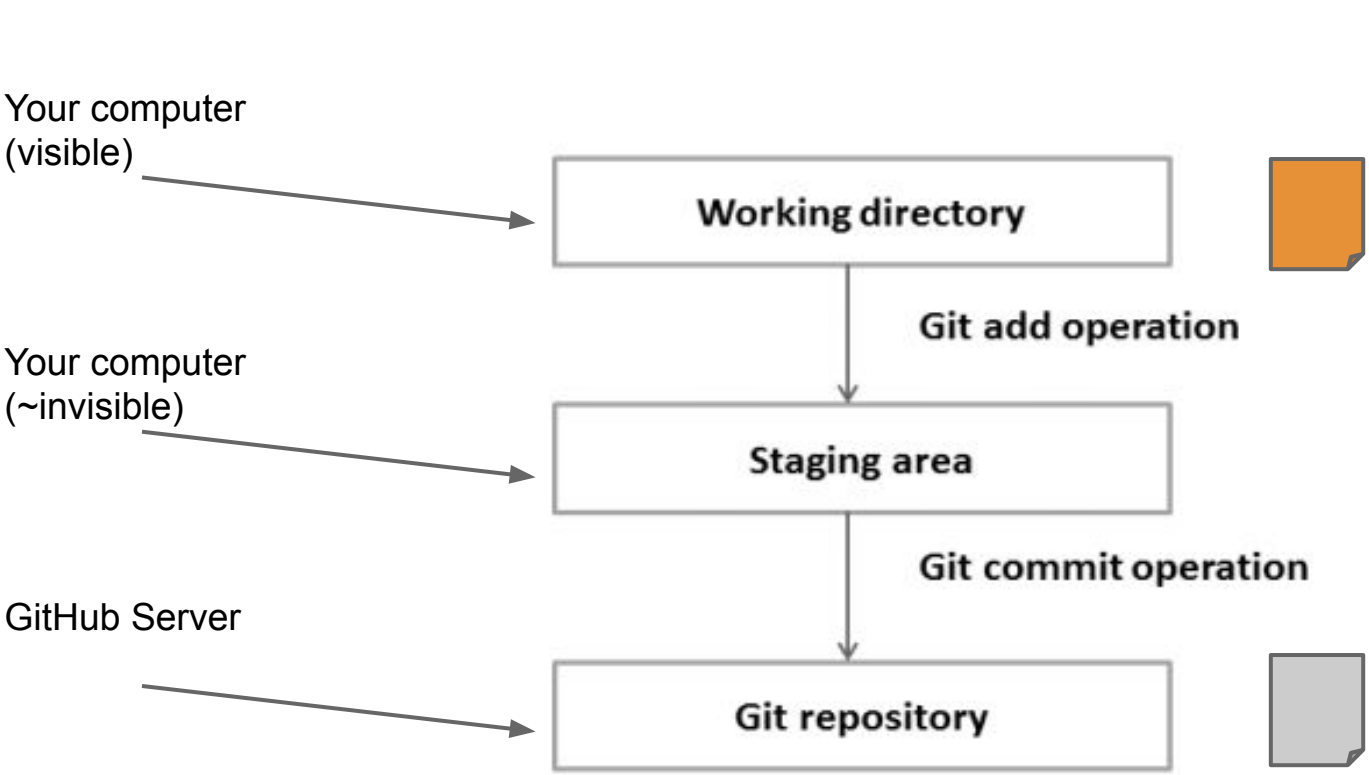
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



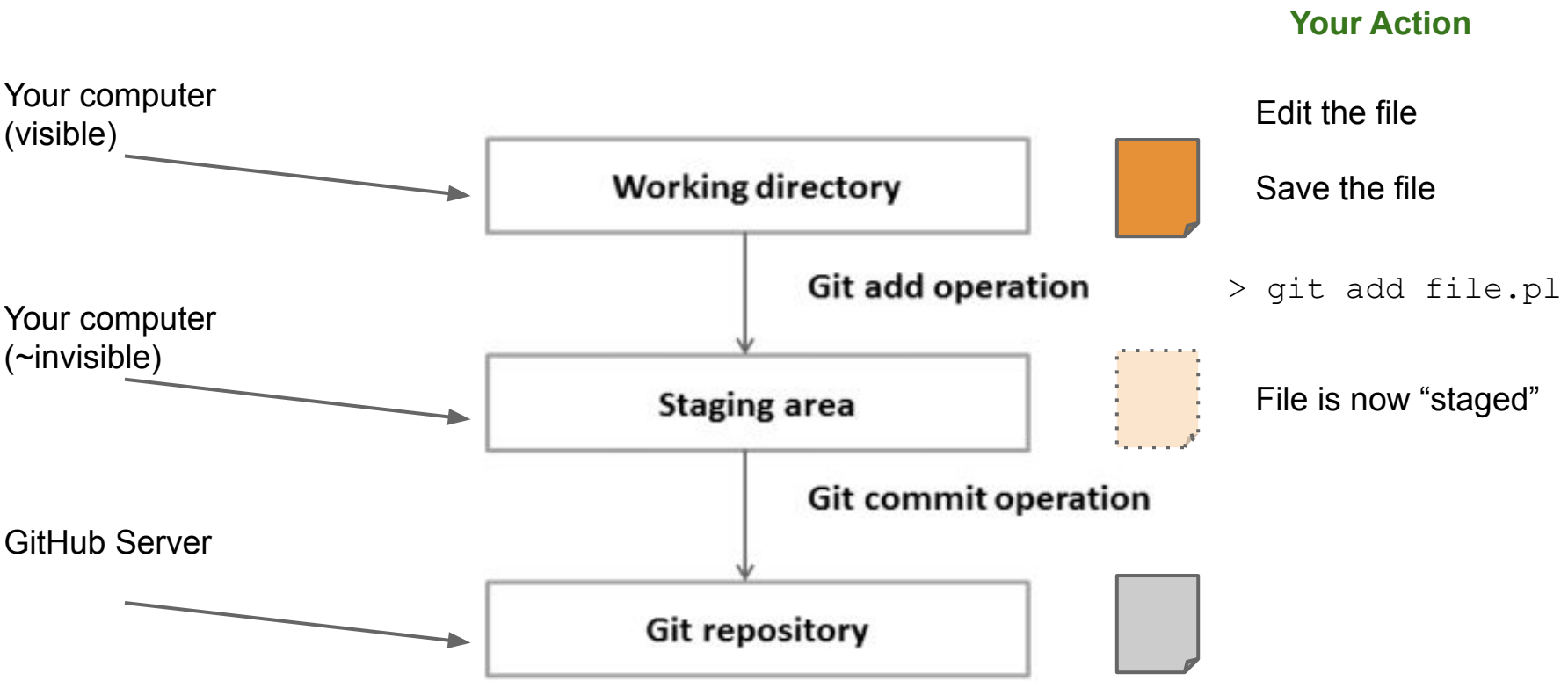
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



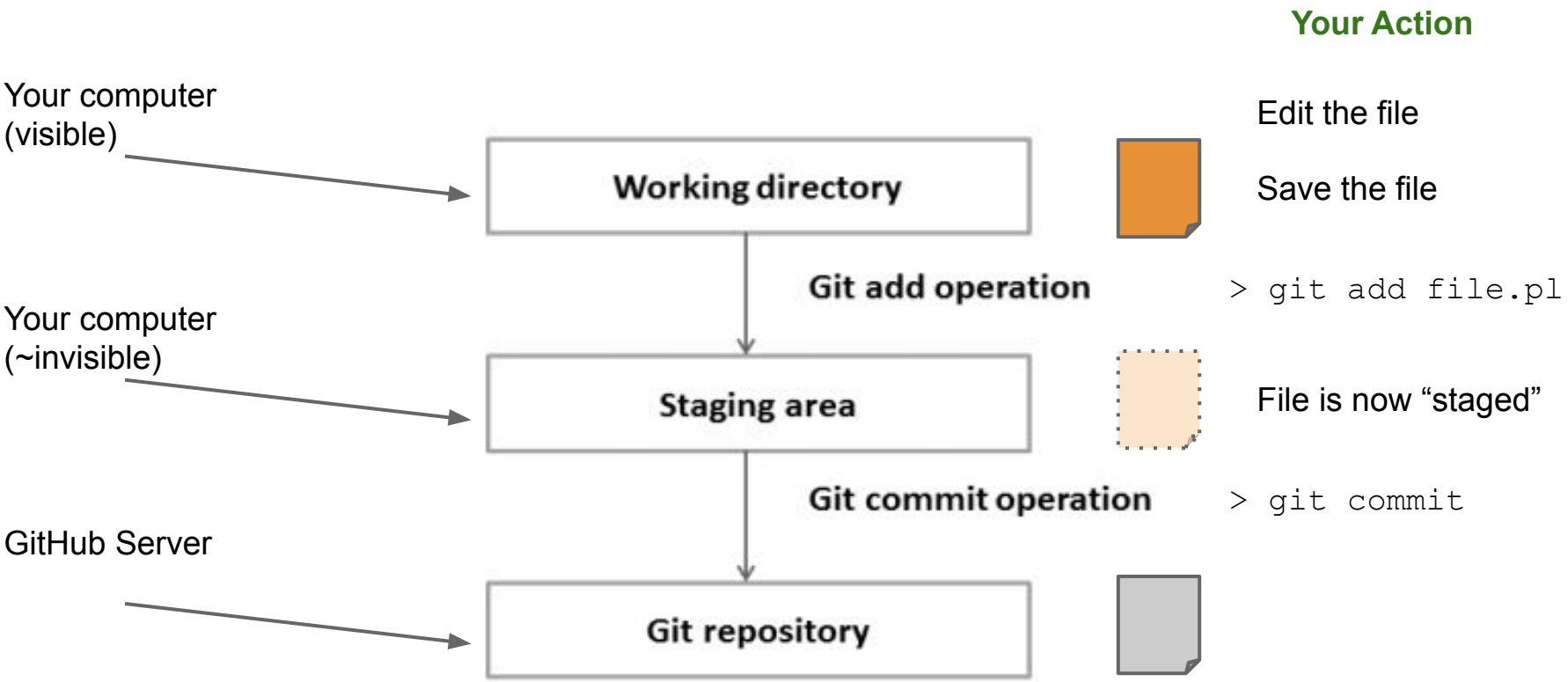
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



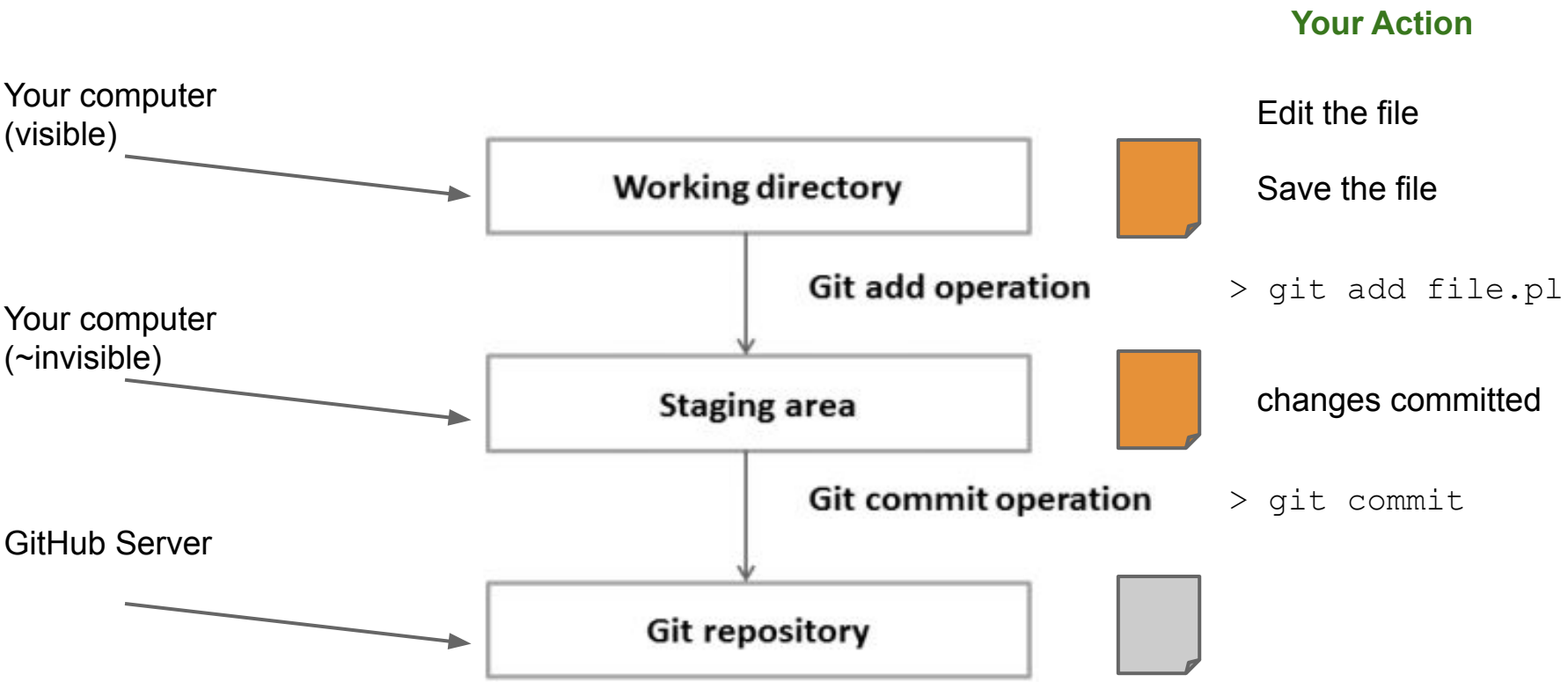
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



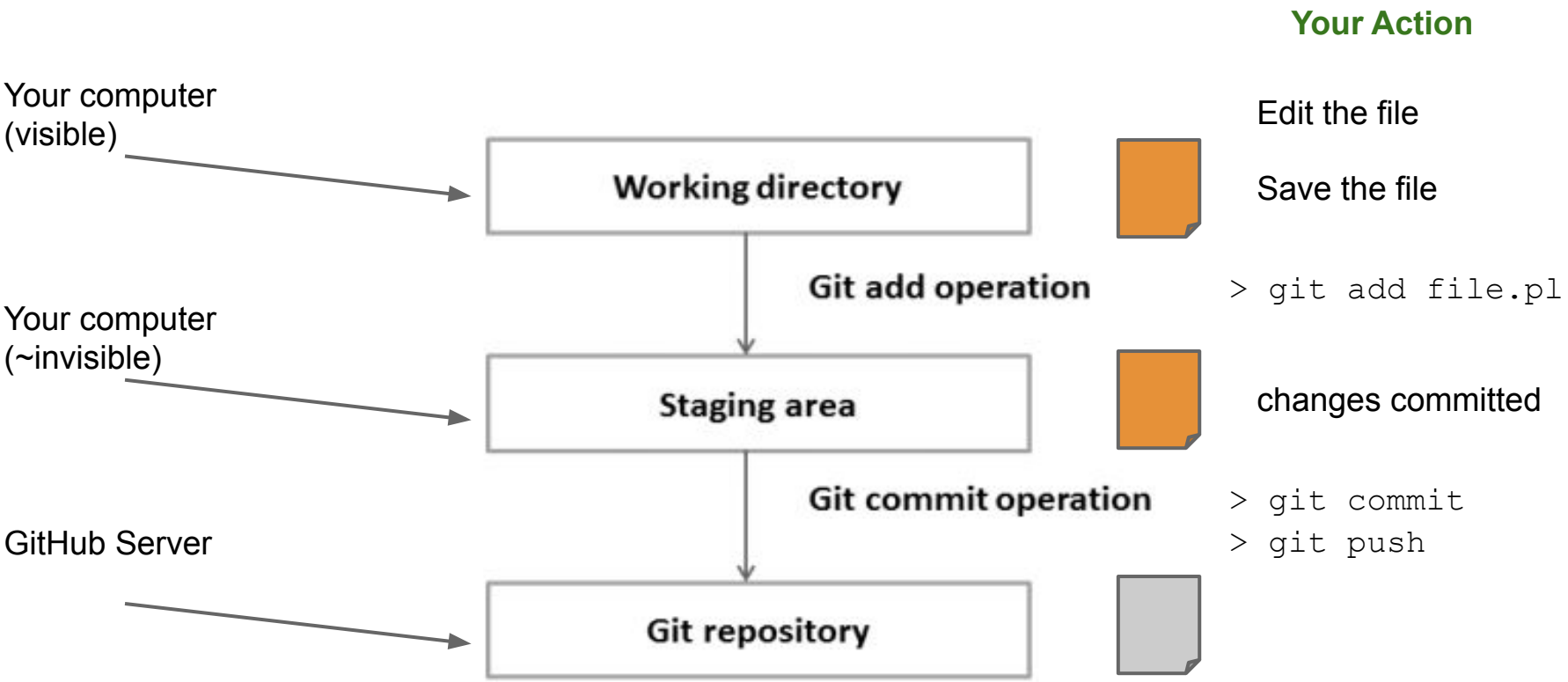
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



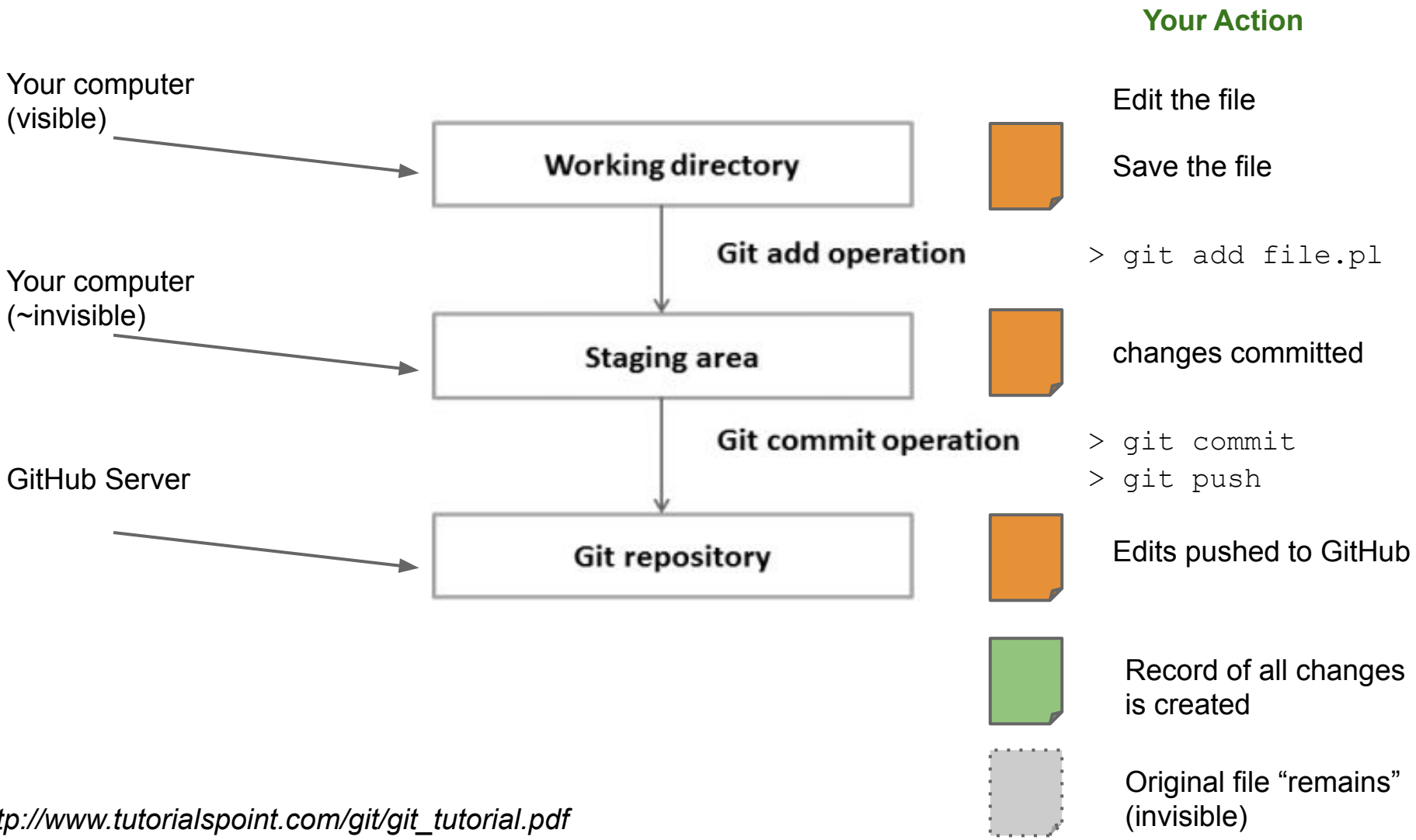
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



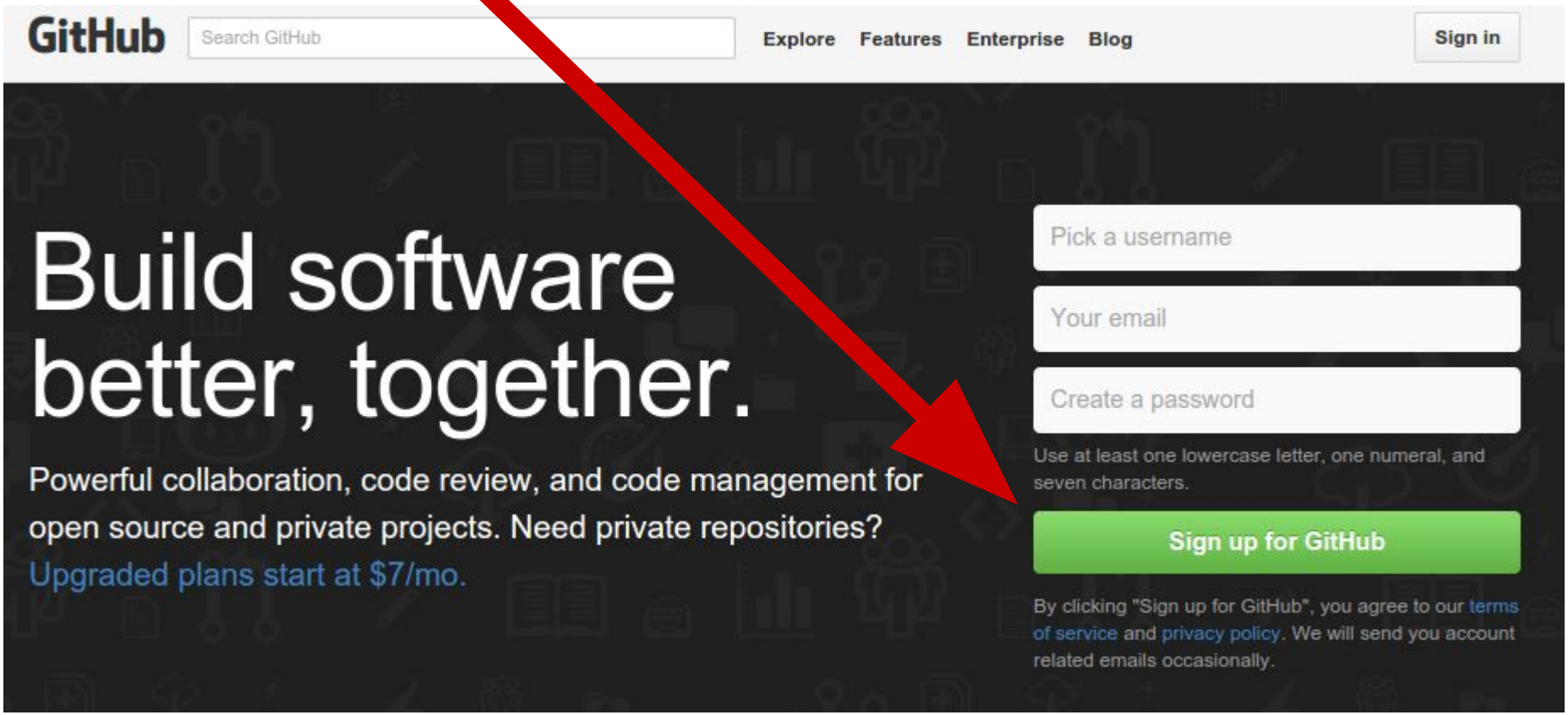
http://www.tutorialspoint.com/git/git_tutorial.pdf

3 “layers” of Git



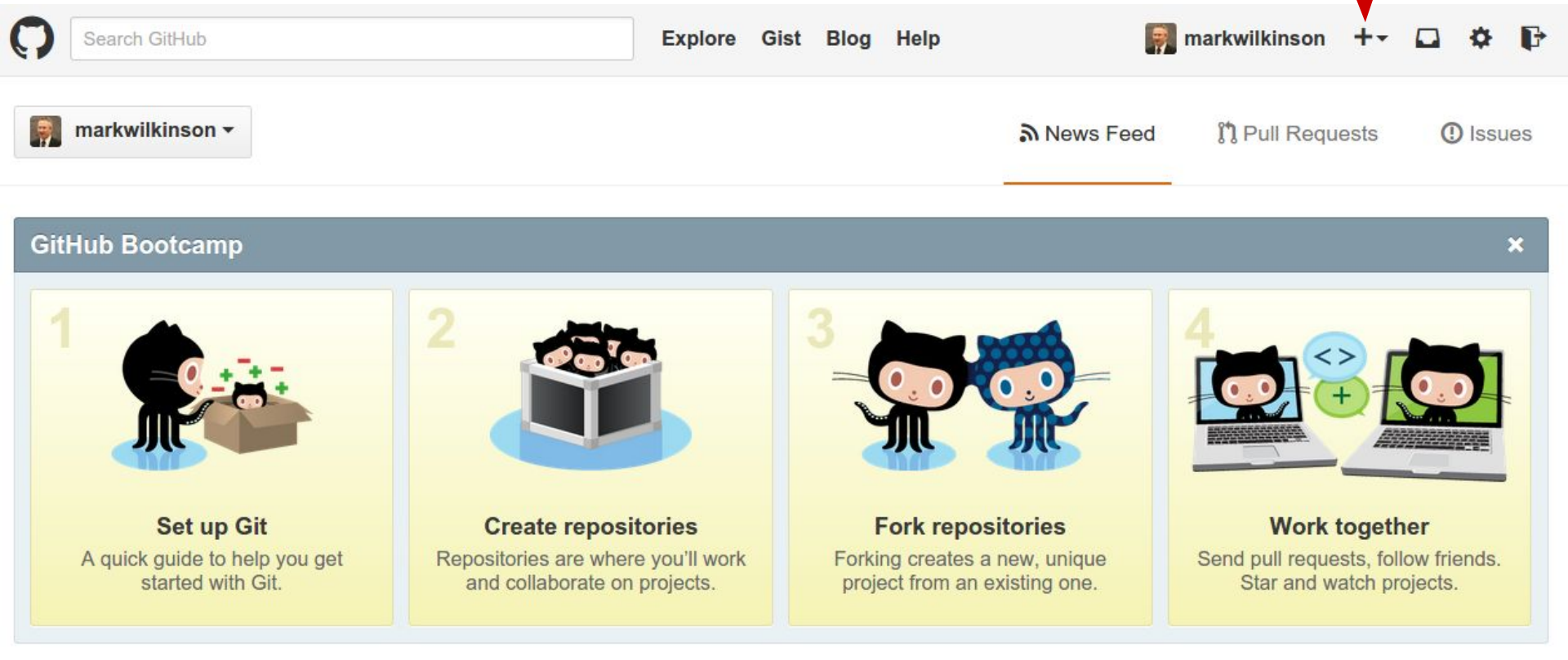
http://www.tutorialspoint.com/git/git_tutorial.pdf

Create an account for yourself
http://github.com

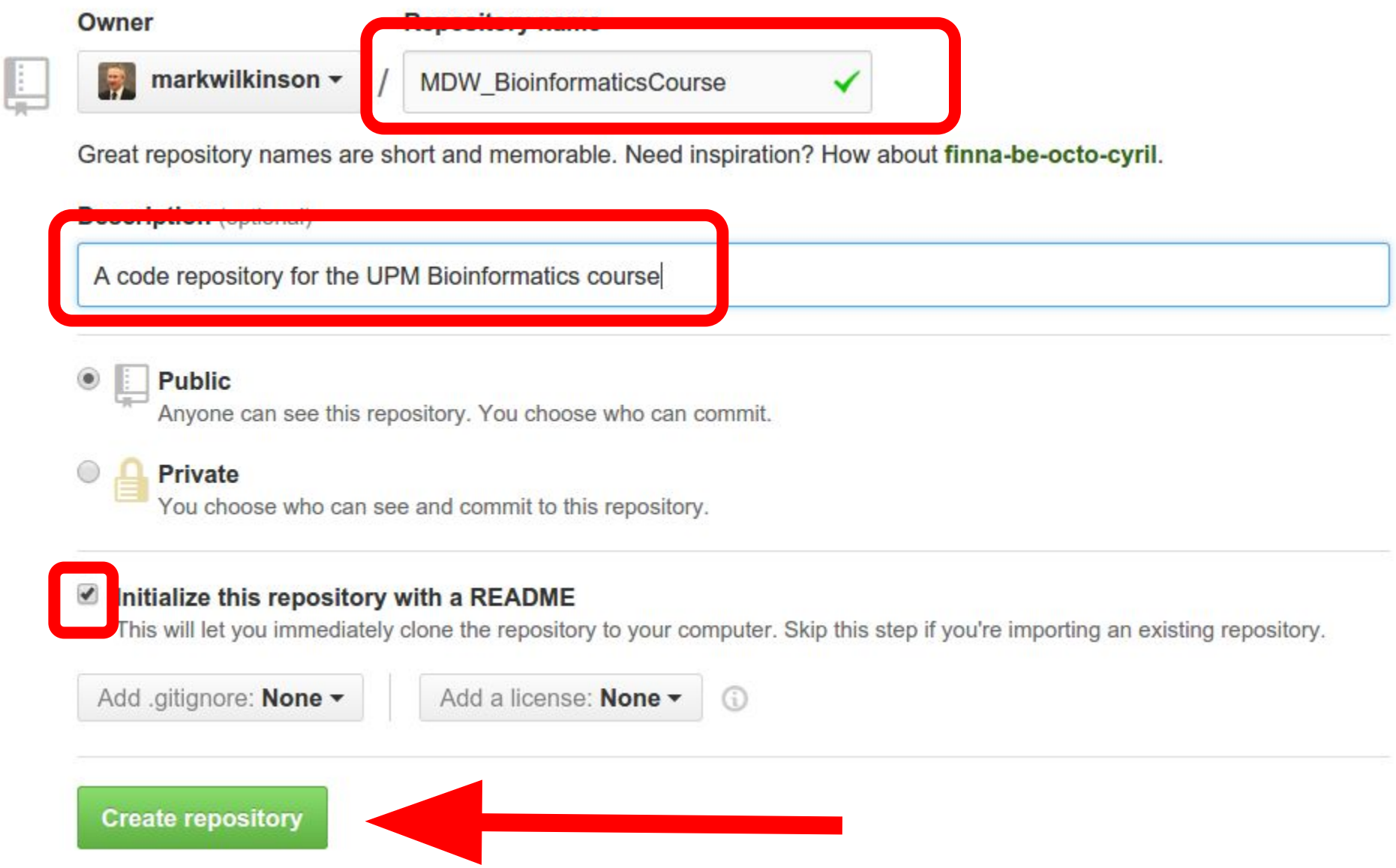


Create a new repository...

"create new"



Please name it using your name or initials so I can see that it is yours



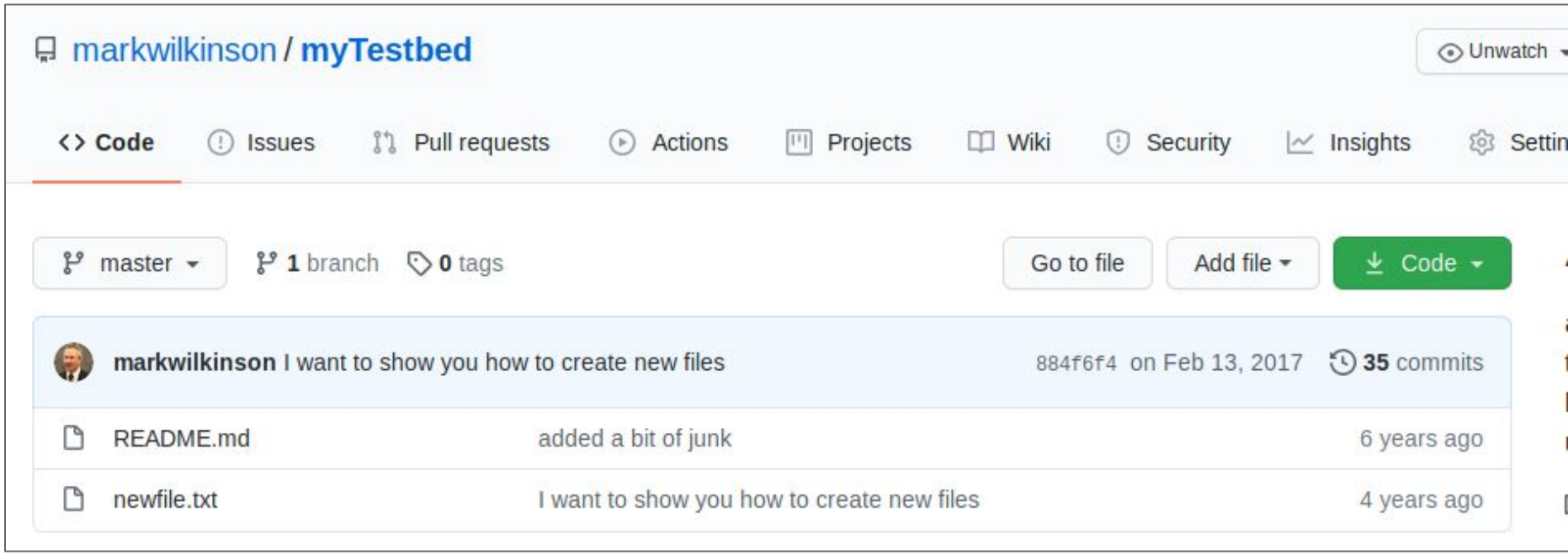
Your first repository!

Keep this page open on your browser, you will need it in 5 minutes...

Here's mine. I called it "myTestbed". Open a new browser window and go to:

https://github.com/markwilkinson/myTestbed

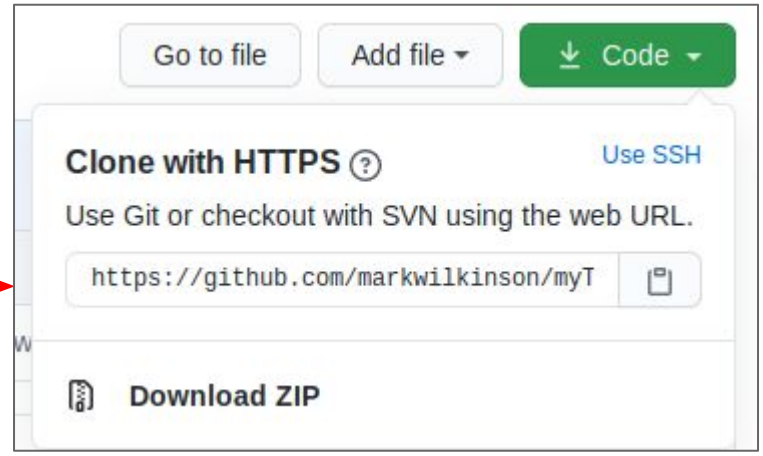
We are going to "clone" my repository



Prepare to make a local clone of myTestBed Repository from GitHub

Look at the right-hand side of the page

Ctrl-C the “HTTPS clone URL”



FIRST:
You need to configure your local Git software to know about you

```
$ git config --global user.email "you@example.com"
$ git config --global user.name "Your Name"
```

```
> cd BioinformaticsCourseGit

> git clone https://github.com/markwilkinson/myTestbed.git
```

```
> cd BioinformaticsCourseGit

> git clone https://github.com/markwilkinson/myTestbed.git
Cloning into 'myTestbed'...
WARNING: gnome-keyring:: couldn't connect to:
/tmp/keyring-rFHr0v/pkcs11: No such file or directory
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 1), reused 5 (delta 1)
Unpacking objects: 100% (7/7), done.
>
> ls
myTestbed
```

```
> cd BioinformaticsCourseGit

> git clone https://github.com/markwilkinson/myTestbed.git
Cloning into 'myTestbed'...
WARNING: gnome-keyring:: couldn't connect to:
/tmp/keyring-rFHr0v/pkcs11: No such file or directory
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 1), reused 5 (delta 1)
Unpacking objects: 100% (7/7), done.
>
> ls
myTestbed
> cd myTestbed
> ls
README.md
>
```

Success! This is your repository cloned onto your local computer

Now, do exactly the same thing with the repository that YOU just created (the one that uses your name/initials).

1. Go to the web page of YOUR repository
2. Find the URL of your repository
3. Git Clone it

Now add something to your repository...

- ‘cd’ into the new folder created when you cloned your repository

  for me, it is:

  ```
  ./BioinformaticsCourseGit/myTestbed
  ```

- Open your text editor and create a file (call it whatever you want, e.g. junk.txt)

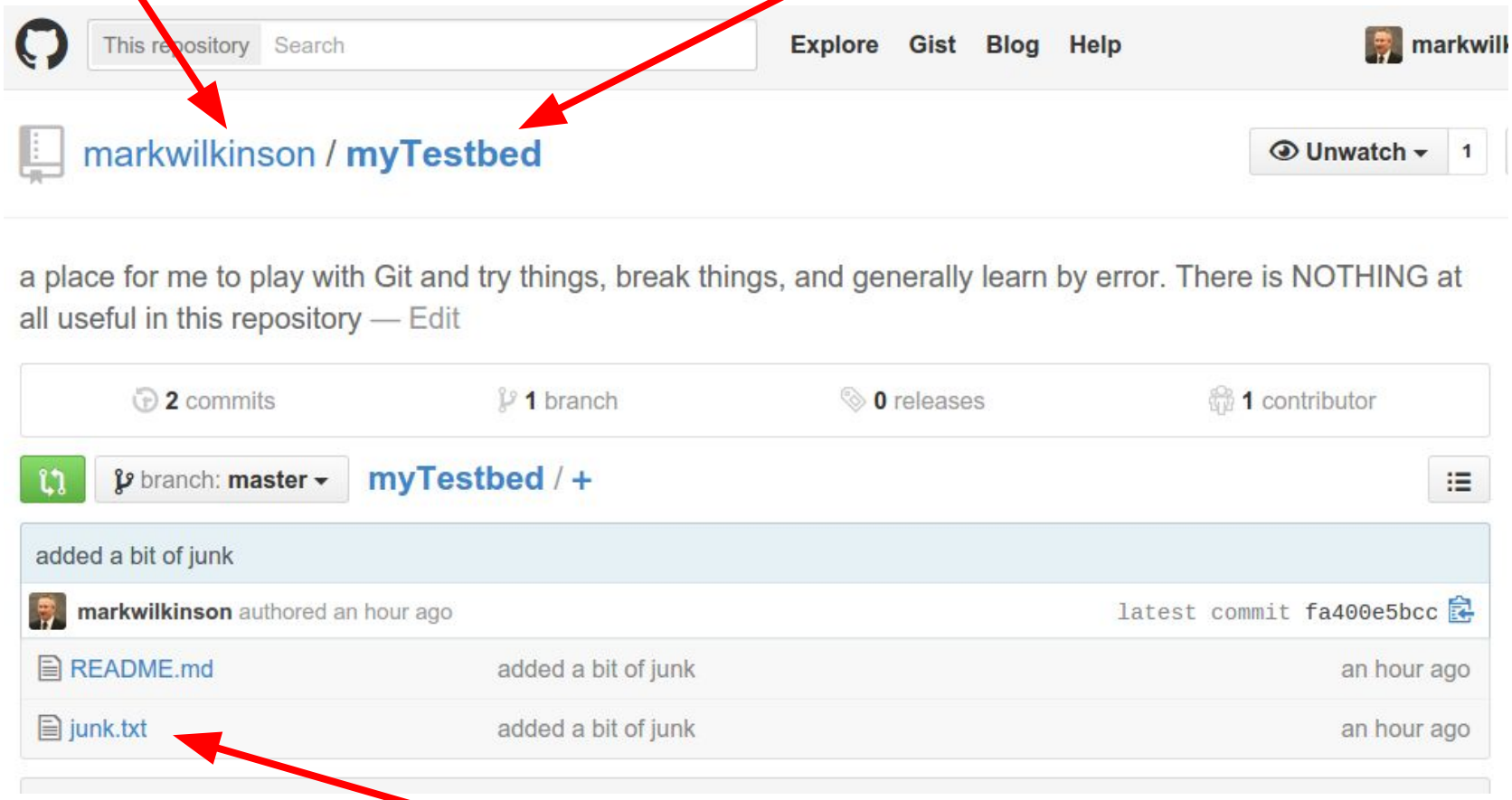
- Save that file

Now add something to your repository...

```
> ls
junk.txt  README.md
> git add junk.txt
> git commit -m "learning how to add files to Git"
[master a415c1b] learning how to add files to Git
 1 file changed, 1 insertion(+), 1 deletion(-)
> git push
WARNING: gnome-keyring:: couldn't connect to:
/tmp/keyring-rFHr0v/pkcs11: No such file or directory
Username for 'https://github.com': markwilkinson
Password for 'https://markwilkinson@github.com':
```

Your Git username

Refresh your browser window (or click here - on YOUR repository)



your file has been added

Now go back to your local repository, and edit

Open your text editor and edit your junk.txt file, then save it again

```
> ls
junk.txt  README.md

> git add junk.txt

> git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#  modified:   junk.txt
#
> git commit -m "modified content of junk file"
[master 61953a9] modified content of junk file
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Now go back to your local repository, and edit

To see the difference between your local commit, and the GitHub repository

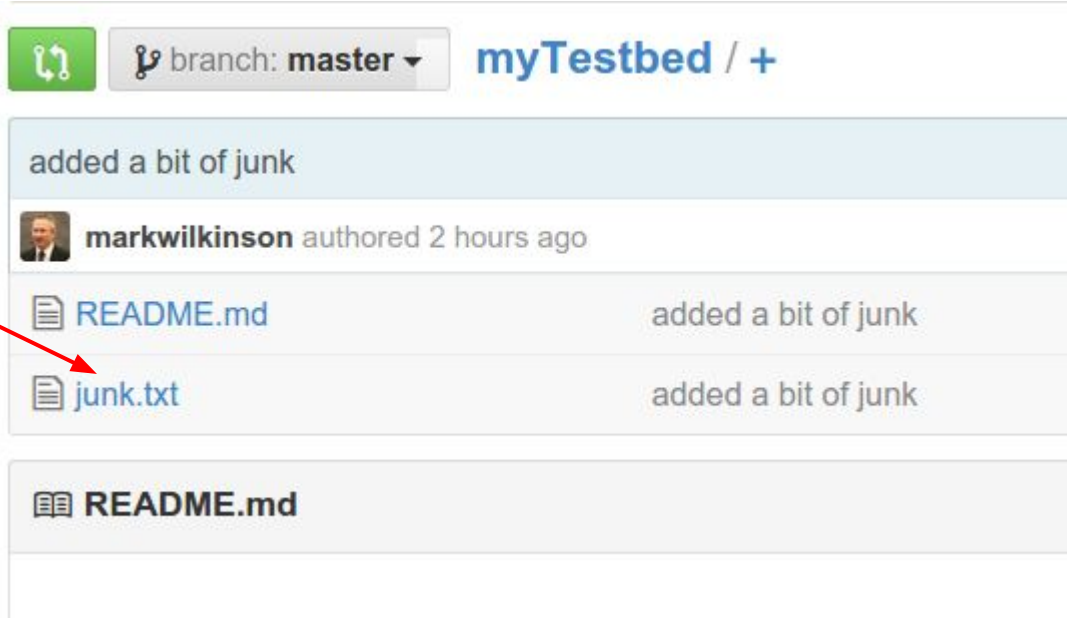
```
> git diff origin

diff --git a/junk.txt b/junk.txt
index 4621e09..0bc0c72 100644
--- a/junk.txt
+++ b/junk.txt
@@ -1 +1 @@
-this is still just junk
+This is the edit of my junk.txt file
```

Now go back to your local repository, and edit

now push the changes to GitHub

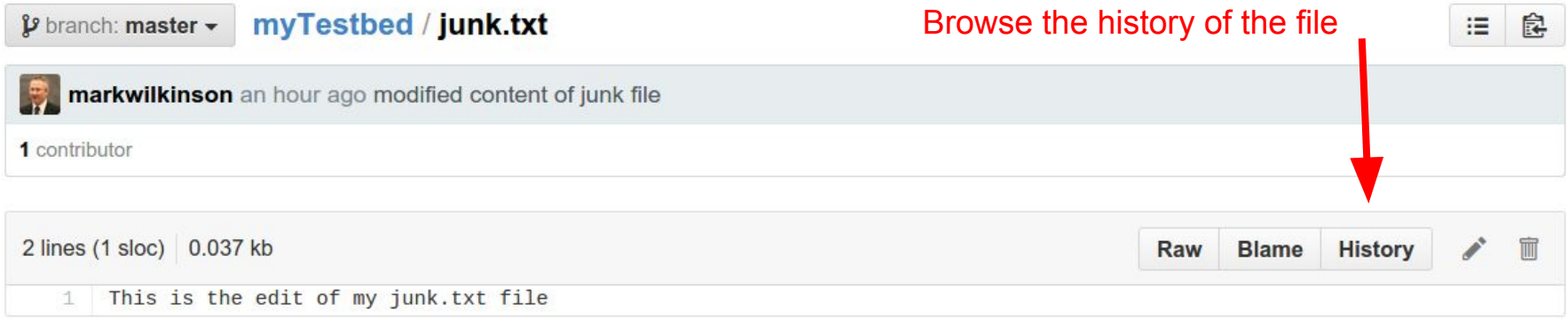
```
> git push
...
...
...
```

click the filename in GitHub



branch: master ▾ myTestbed / junk.txt

Browse the history of the file



A record of everything you (or anyone!) did to the file between this version and the previous

the content of the file at that moment...

Try a few things on your own

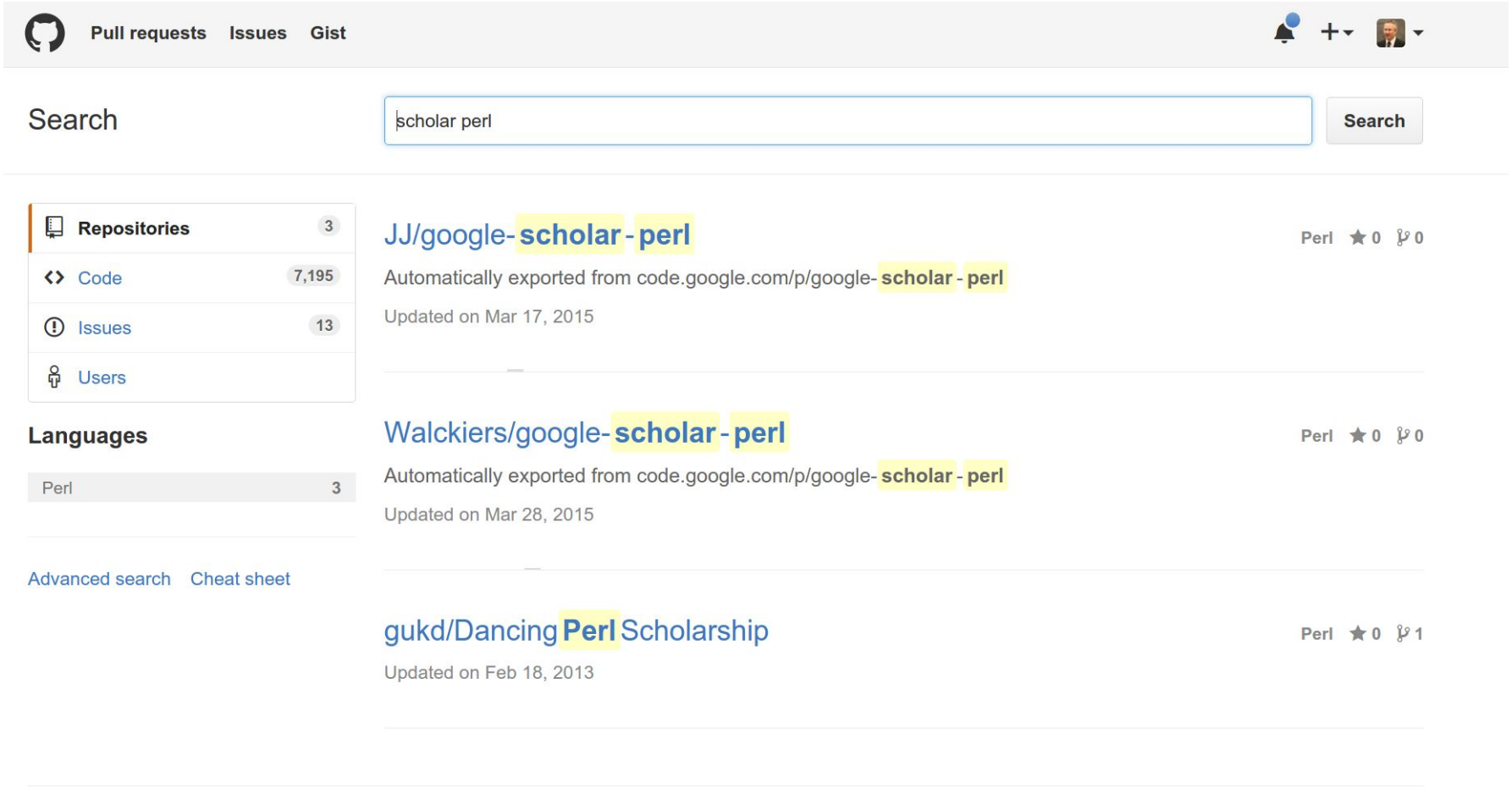
making folders, making files

try deleting something!

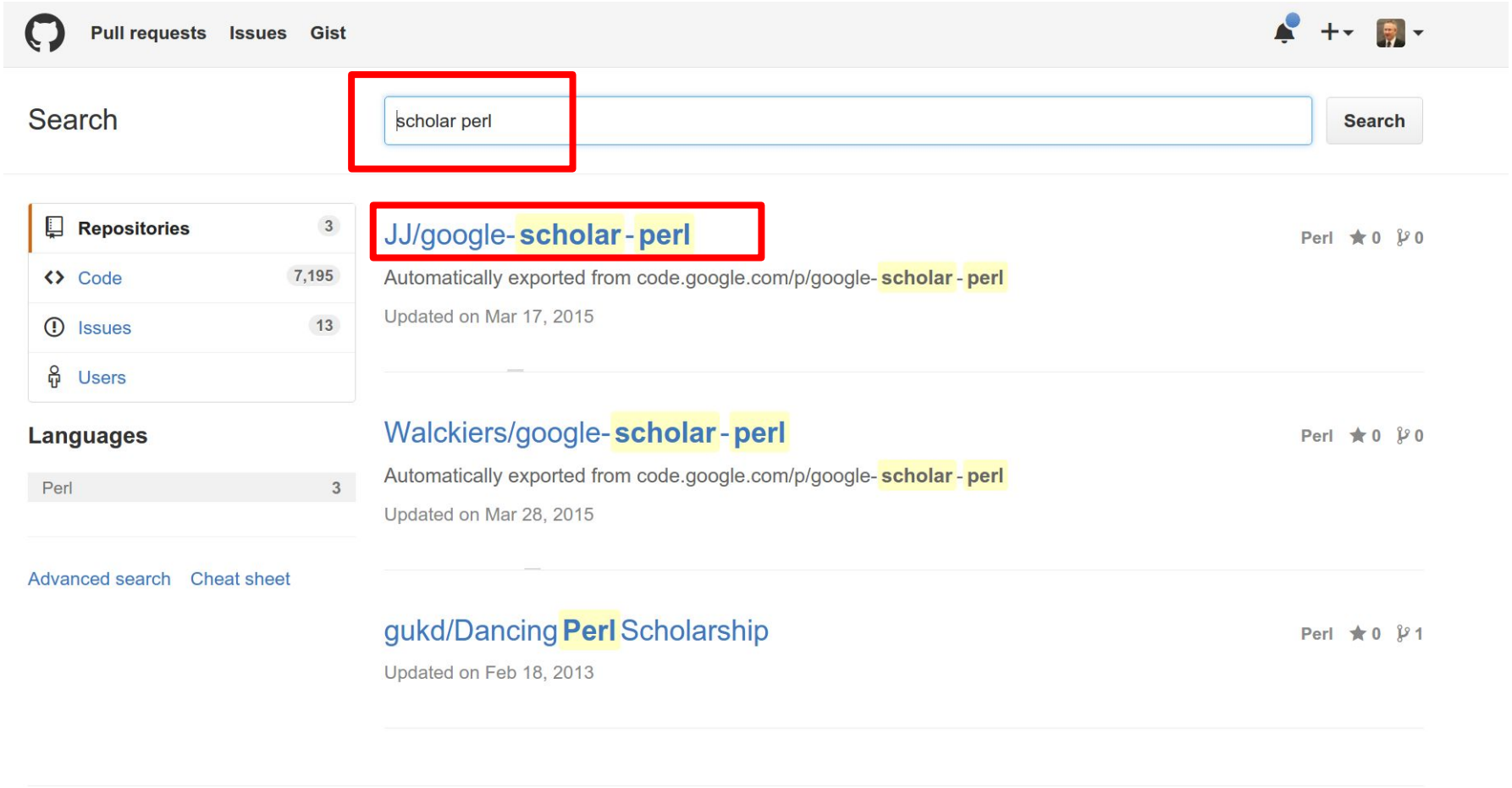
```
> git rm filename.txt
> git commit -m “removed”
> git push
```

then see what happens in GitHub

Forking another person's project



Forking another person's project



JJ / google-scholar-perl

Automatically exported from code.google.com/p/google-scholar-perl

34 commits 2 branches 0 releases 1 contributor

Branch: master New pull request New file Find file HTTPS https://github.com/JJ/g Download ZIP

JJ Small format changes — Latest commit 8f9ed67 on Mar 17, 2015

examples	Cleaning up	10 months ago
lib/My/Google	Cleaning up	10 months ago
t	Cleaning up	10 months ago
Changes	Cleaning up	10 months ago
MANIFEST	Cleaning up	10 months ago
Makefile.PL	Cleaning up	10 months ago
README.md	Small format changes	10 months ago

README.md

My-Google-Scholar

JJ / google-scholar-perl

Watch 3 Star 0 Fork 0

Code Issues 2 Pull requests 0 Pulse Graphs

Automatically exported from code.google.com/p/google-scholar-perl

34 commits 2 branches 0 releases 1 contributor

Branch: master New pull request New file Find file HTTPS https://github.com/JJ/g Download ZIP

JJ Small format changes Latest commit 8f9ed67 on Mar 17, 2015

examples	Cleaning up	10 months ago
lib/My/Google	Cleaning up	10 months ago
t	Cleaning up	10 months ago
Changes	Cleaning up	10 months ago
MANIFEST	Cleaning up	10 months ago
Makefile.PL	Cleaning up	10 months ago
README.md	Small format changes	10 months ago

README.md

My-Google-Scholar

This repository Search Pull requests Issues Gist

markwilkinson / **google-scholar-perl**
forked from JJ/google-scholar-perl

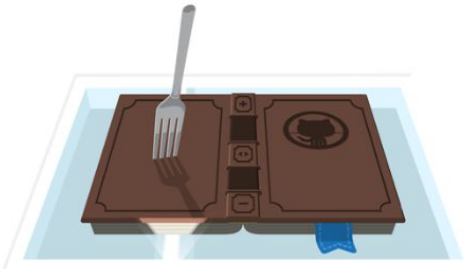
Unwatch 1 Star 0 Fork 1

Code Pull requests 0 Pulse Graphs Settings

Forking JJ/google-scholar-perl

It should only take a few seconds.

Refresh



markwilkinson / google-scholar-perl

forked from JJ/google-scholar-perl

Unwatch 1 Star 0 Fork 1

Code Pull requests 0 Pulse Graphs Settings

Automatically exported from code.google.com/p/google-scholar-perl — Edit

34 commits 2 branches 0 releases 1 contributor

Branch: master New pull request New file Find file HTTPS https://github.com/mark Download ZIP

This branch is even with JJ:master. Pull request Compare

JJ Small format changes Latest commit 8f9ed67 on Mar 17, 2015

examples	Cleaning up	10 months ago
lib/My/Google	Cleaning up	10 months ago
t	Cleaning up	10 months ago
Changes	Cleaning up	10 months ago
MANIFEST	Cleaning up	10 months ago
Makefile.PL	Cleaning up	10 months ago
README.md	Small format changes	10 months ago

README.md

Now, do exactly the same as before

Create a local copy of the forked repository on your computer:

```
> git clone https://github.com/markwilkinson/google-scholar-perl.git
Cloning into 'google-scholar-perl'...
WARNING: gnome-keyring:: couldn't connect to:
/tmp/keyring-d0t0Gv/pkcs11: No such file or directory
remote: Counting objects: 267, done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 267 (delta 96), reused 266 (delta 95), pack-reused 0
Receiving objects: 100% (267/267), 41.48 KiB, done.
Resolving deltas: 100% (96/96), done.
>
> cd google-scholar-perl
> ls
Changes  examples  lib  Makefile.PL  MANIFEST  README.md  t
```

```
$ git remote -v
origin	https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin	https://github.com/markwilkinson/google-scholar-perl.git (push)
```

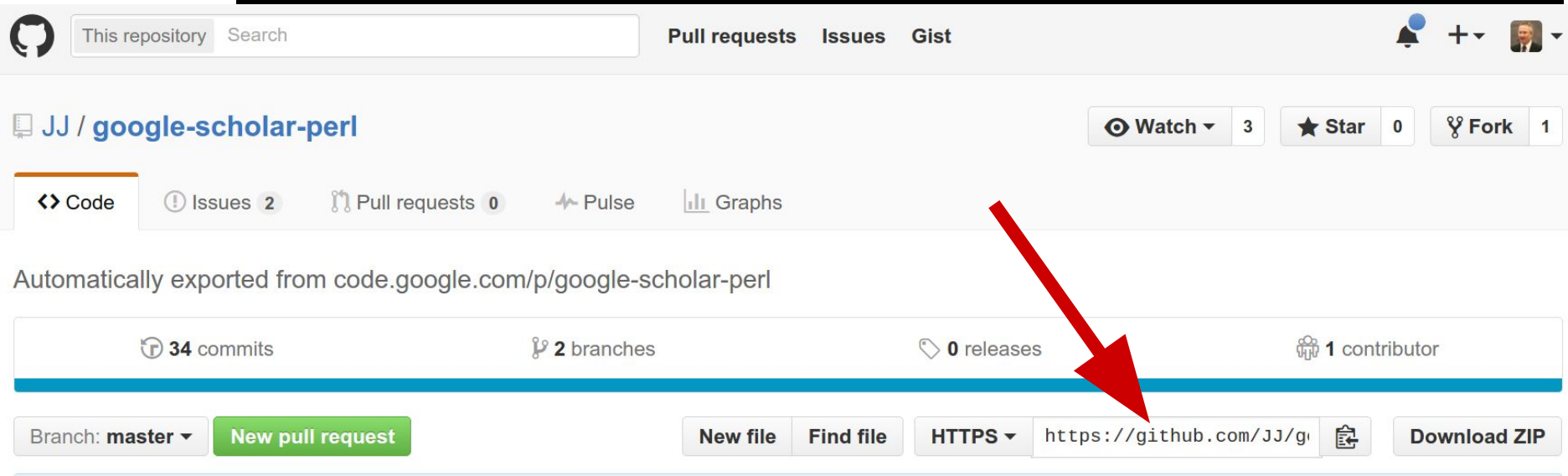
```
$ git remote -v
origin	https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin	https://github.com/markwilkinson/google-scholar-perl.git (push)
```

This is the “designation” for the repository: a short label that Git uses to know which repository you are referring to

note that you are allowed to fetch and push ONLY from your own copy (“origin”)

If you want to collaborate with the original author, then you need to add a reference to THEIR repository

```
$ git remote -v
origin	https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin	https://github.com/markwilkinson/google-scholar-perl.git (push)

$
```



If you want to collaborate with the original author, then you need to add a reference to THEIR repository

```
$ git remote -v
origin    https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin    https://github.com/markwilkinson/google-scholar-perl.git (push)

$ git remote add upstream https://github.com/JJ/google-scholar-perl.git

$ git remote -v
origin    https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin    https://github.com/markwilkinson/google-scholar-perl.git (push)
upstream  https://github.com/JJ/google-scholar-perl.git (fetch)
upstream  https://github.com/JJ/google-scholar-perl.git (push)
```

If you want to collaborate with the original author, then you need to add a reference to THEIR repository

```
$ git remote -v
origin        https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin        https://github.com/markwilkinson/google-scholar-perl.git (push)

$ git remote add upstream https://github.com/JJ/google-scholar-perl.git

$ git remote -v
origin        https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin        https://github.com/markwilkinson/google-scholar-perl.git (push)
upstream      https://github.com/JJ/google-scholar-perl.git (fetch)
upstream      https://github.com/JJ/google-scholar-perl.git (push)
```

git fetch upstream

a way to synchronize with
any changes from the original author

```
$ git remote -v
origin        https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin        https://github.com/markwilkinson/google-scholar-perl.git (push)

$ git remote add upstream https://github.com/JJ/google-scholar-perl.git

$ git remote -v
origin       https://github.com/markwilkinson/google-scholar-perl.git (fetch)
origin       https://github.com/markwilkinson/google-scholar-perl.git (push)
upstream https://github.com/JJ/google-scholar-perl.git (fetch)
upstream https://github.com/JJ/google-scholar-perl.git (push)

$ git fetch upstream
WARNING: gnome-keyring:: couldn't connect to: /tmp/keyring-d0t0Gv/pkcs11: No
such file or directory
From https://github.com/JJ/google-scholar-perl
 * [new branch]      master     -> upstream/master
 * [new branch]      wiki       -> upstream/wiki
```

```
$ git checkout master
Switched to branch ‘master’

$ git merge upstream/master
Already up-to-date.
```

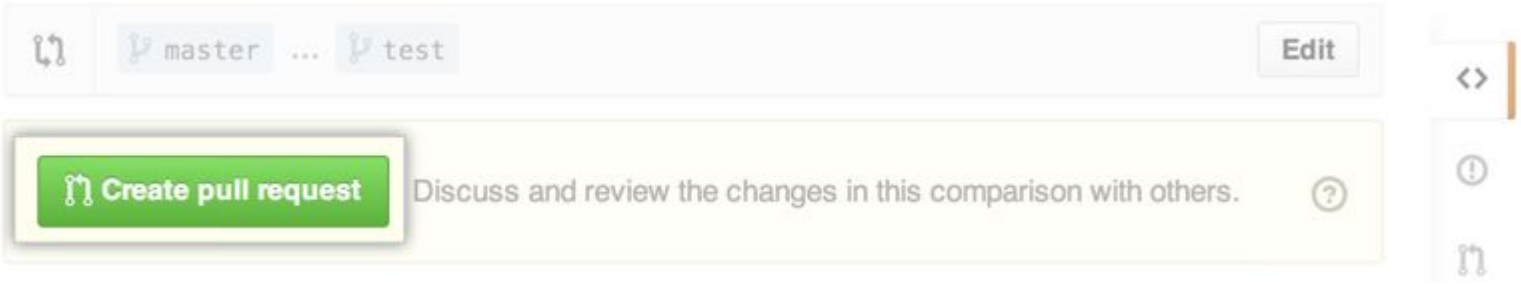
Now you - prove that you understand

The rest of this course is in a different repository:

https://github.com/CBGP-UPM-INIA-PUBLIC/Accelerated_Intro_to_CompBio_Part_2

- Fork that repository into YOUR github account
- Clone it to your local laptop
- Add the reference to the upstream (mine)
- Create a new file (whatever you want to call it)
- Commit and push the changes to your Github Fork

After you have committed to your Fork of myTestBed, go to your Github page on the Web, and refresh the page. You should see:



When you click this button, it begins the process of merging your changes into my original code.

First, it asks you to add a bit of additional information (just a short text message about what the changes are, and why)

Then you submit the request.

On my repository,I am notified that you have requested a code-merge. I can view the changes, and decide to accept, or reject, your changes

Stop your jupyter notebook

(CTRL-C twice)

Now you:

Now in your terminal window, cd into the folder called "Accelerated_Intro_to_CompBio_Part_2"

$ jupyter notebook

You are now looking at the Jupyter Notebooks we will use for the rest of this course.

NOTE:

Git Hub is completely open and public! Your code can be seen by anyone in the world, including the other students in this course.

Therefore, I will be watching very carefully for people who copy each other

You may collaborate with each other and share ideas or even pieces of code - this is normal in bioinformatics
(and in fact, it is the reason that GitHub exists!)

<u>However, there are two rules:</u>

1) ***You must document the code yourself*** - I want to see that you understand how the code works.

2) ***You must **BOTH** say, in your documentation, that you are collaborating***: e.g. “I collaborated with my colleague Juan Nadie to write this portion of the code”

Finally: Prepare for the exam

The exam questions are in another Git folder

https://github.com/CBGP-UPM-INIA-PUBLIC/Accelerated_Intro_WilkinsonExams

- Fork that repository into YOUR github account
  https://github.com/YourUsername/Accelerated_Intro_WilkinsonExams
- Clone it to your local laptop

This is the folder that you will use to submit the answers for your exams

For each exam:

- Create a new Jupyter Notebook
- Put your answers into that Notebook
- Commit your answers
- Push your answers to GitHub so that I can see them.