

Announcement

Are we what we say/think, or what we do...

- Maybe both
- As engineers, we are makers and go-getters

- Next week, we have an exam:)

Writing programs & source control tools

Editor programs

- Vim, nano/pico, gedit, notepad

IDE softwares

Version control

- Git, svn, etc

Next week: Documentation

Text editor

A computer program that edits plain text (data only represented by characters).
Some simple some more complex..

Windows

- Notepad, Emacs

Linux

- Pico, vi, Gedit, Emacs, etc.

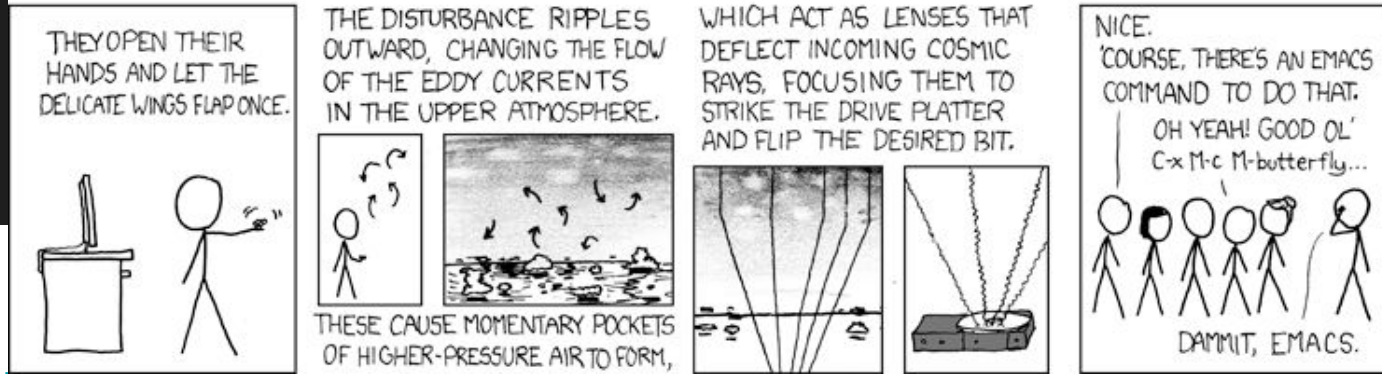
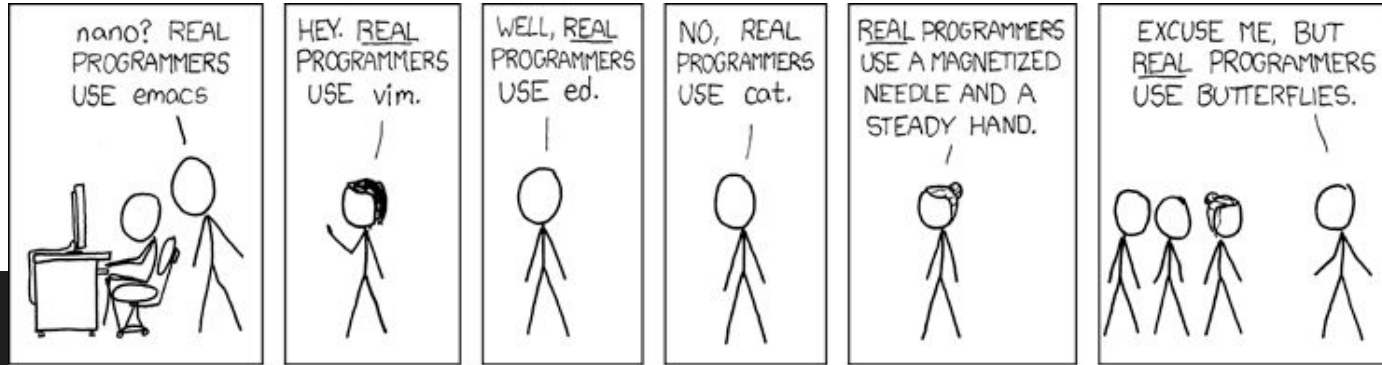
MacOS

- Pico, vi, TextEdit

https://en.wikipedia.org/wiki/Text_editor

Vi(m)

\$ vi



```
VIM - Vi Improved
      version 9.0.1499
      by Bram Moolenaar et al.
      Modified by team+vim@tracker.debian.org
      Vim is open source and freely distributable

      Help poor children in Uganda!
      type :help iccf<Enter> for information

      type :q<Enter> to exit
      type :help<Enter> or <F1> for on-line help
      type :help version9<Enter> for version info

      Running in Vi compatible mode
      type :set nocp<Enter> for Vim defaults
      type :help cp-default<Enter> for info on this
```

<https://www.vim.org/about.php>

<https://vimhelp.org/>

<https://vimhelp.org/quickref.txt.html>

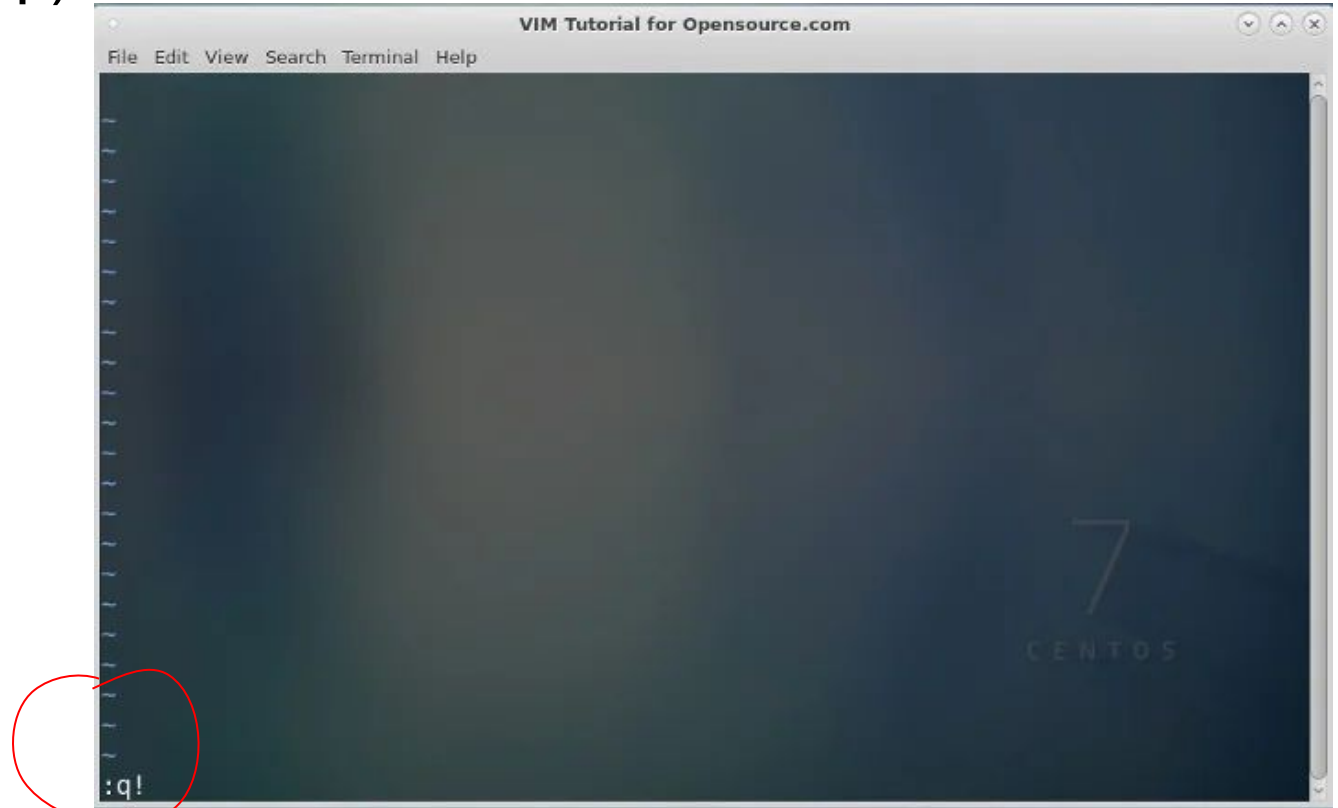
[#quickref](#)

Taken from [xkcd: Real Programmers](#)

```
VIM Tutorial for Opensource.com
File Edit View Search Terminal Help
[test@bpms ~]$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[test@bpms ~]$ mkdir Tutorial
[test@bpms ~]$ ls -ltr
total 0
drwxr-xr-x 2 test test 6 Feb 27 13:44 Videos
drwxr-xr-x 2 test test 6 Feb 27 13:44 Templates
drwxr-xr-x 2 test test 6 Feb 27 13:44 Public
drwxr-xr-x 2 test test 6 Feb 27 13:44 Pictures
drwxr-xr-x 2 test test 6 Feb 27 13:44 Music
drwxr-xr-x 2 test test 6 Feb 27 13:44 Downloads
drwxr-xr-x 2 test test 6 Feb 27 13:44 Documents
drwxr-xr-x 2 test test 6 Feb 27 13:44 Desktop
drwxrwxr-x 2 test test 6 Feb 27 13:47 Tutorial
[test@bpms ~]$ cd Tutorial
[test@bpms Tutorial]$
```

<https://opensource.com/article/19/3/getting-started-vim>

To exit (esc, :q!)

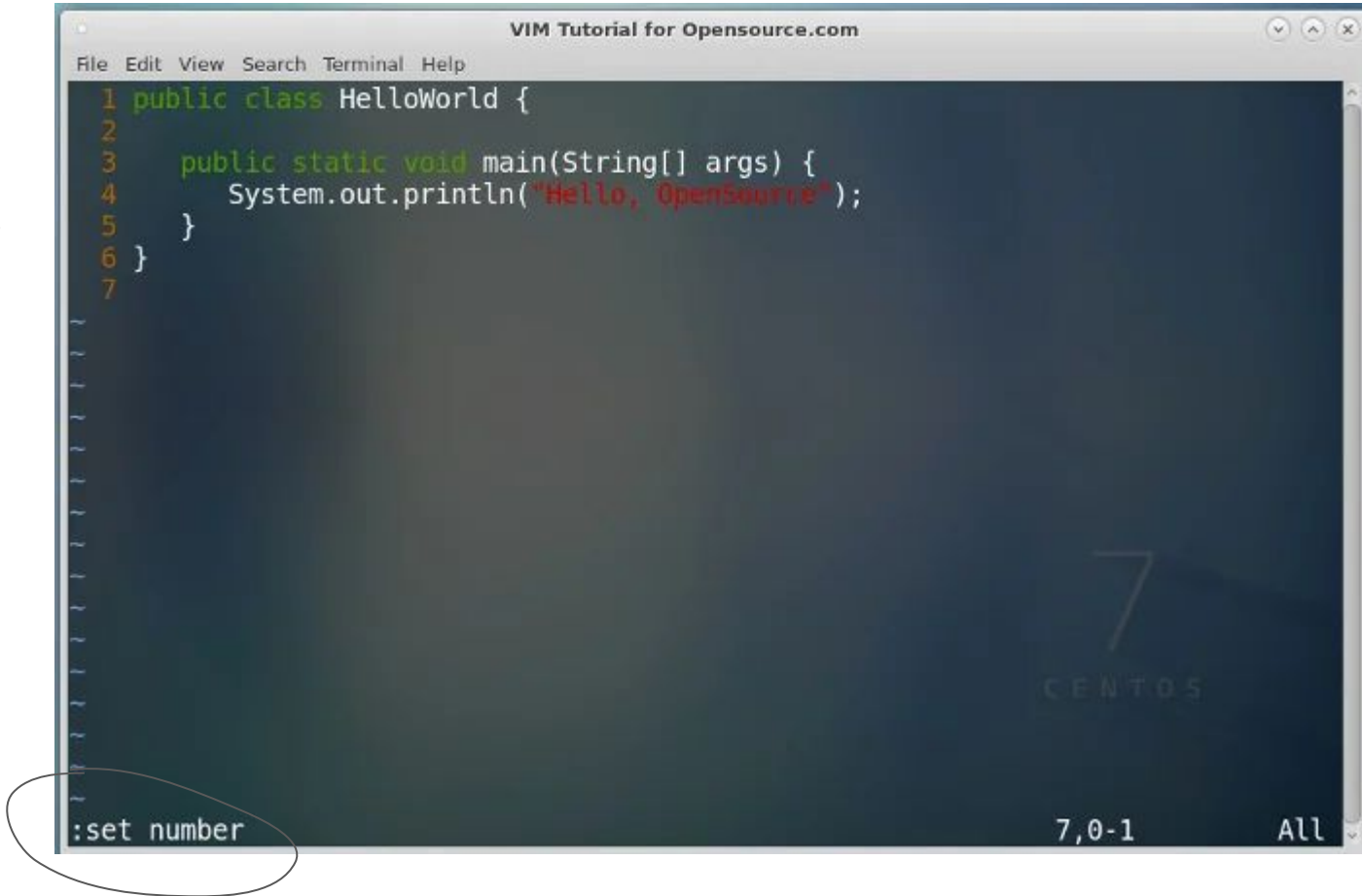


<https://opensource.com/article/19/3/getting-started-vim>

To insert(i)

After inserting, esc
to normal mode

Write line number
to jump



The screenshot shows a Vim editor window titled "VIM Tutorial for Opensource.com". The editor displays a Java code file with the following content:

```
1 public class HelloWorld {  
2  
3     public static void main(String[] args) {  
4         System.out.println("Hello, OpenSource");  
5     }  
6 }  
7
```

The line numbers 1 through 7 are visible on the left side of the editor. At the bottom of the editor, the command prompt shows the command `:set number` being entered, which is circled in red. The status bar at the bottom right displays "7,0-1" and "All".

Vim cheat sheet

h - move cursor left

j - move cursor down

k - move cursor up

l - move cursor right

dd-delete

u-undo

v-with left-right arrows to select/deselect

[Vim Cheat Sheet](#)

:/keyword - search keyword

:g!/{pattern}/d - delete all lines not containing pattern

:%s/old/new/g - replace all old with new throughout file

:w - write (save) the file, but don't exit

:w !sudo tee % - write out the current file using sudo

:wq or :x or ZZ - write (save) and quit

:q - quit (fails if there are unsaved changes)

:q! or ZQ - quit and throw away unsaved changes

```
:!gcc -o main.c % && ./a.out
```


Integrated development environment (IDE)

Combines source editing, compiling, building, testing, etc in one application.

- [Geany](#)
- [Atom](#)
- [Visual Studio Code](#)
- [PyCharm](#)
- [IntelliJ IDEA](#)
- [Eclipse](#)
- [Code::Blocks](#)
- [Komodo](#)
- [Xcode IDE](#)
- [Apache NetBeans](#)

Version control

Revision control, source control,
source code management

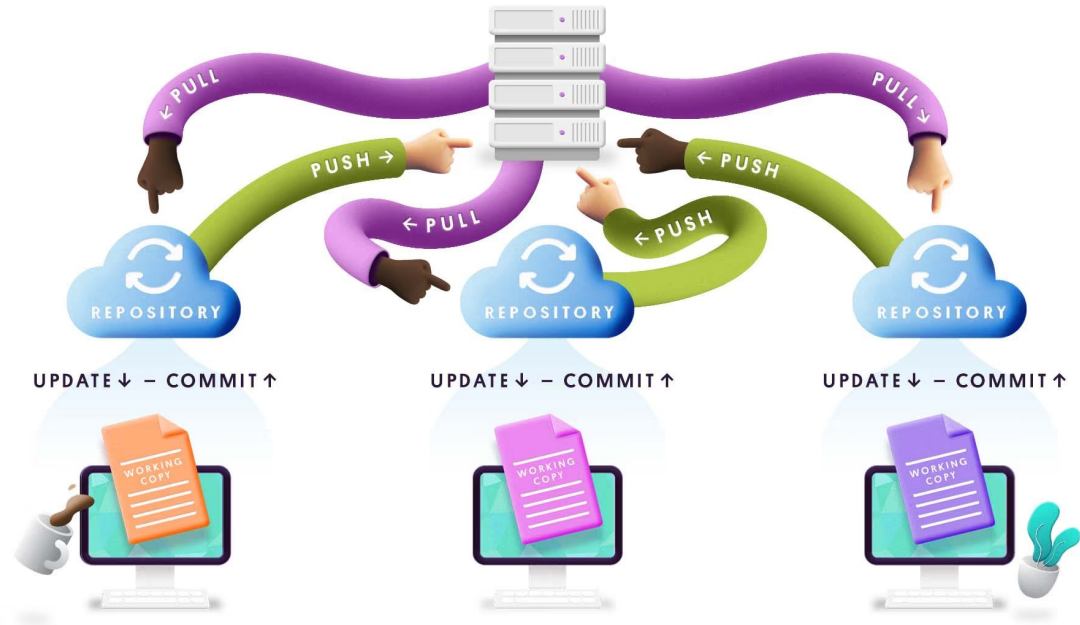
Benefits

Tools

- Git
- svn

Some benefits

- Modification Tracking
- Keep a history
- File comparison
- File sharing
- Troubleshooting
- Code together
 - Distributed development environment
- Automate tasks



Types of version control systems

- Distributed
 - Checkin, branch, merge workflow
 - Git <https://git-scm.com/>
 - Mercurial SCM
<https://www.mercurial-scm.org/>
 - GNU Bazaar
- Centralized
 - Checkin, push workflow
 - Subversion (SVN)
 - Apache Subversion
<https://subversion.apache.org/>
 - CVS
- Local
 - For solo developers

Git intro

Github/gitlab/bitbucket
uses **git** version control
system

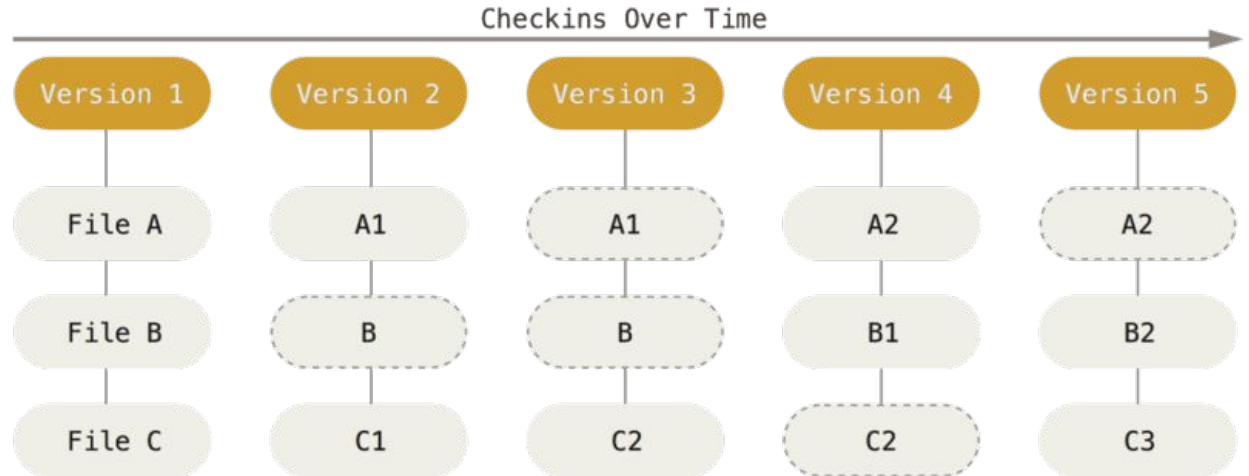
You can also self-host

[Gitea](#)

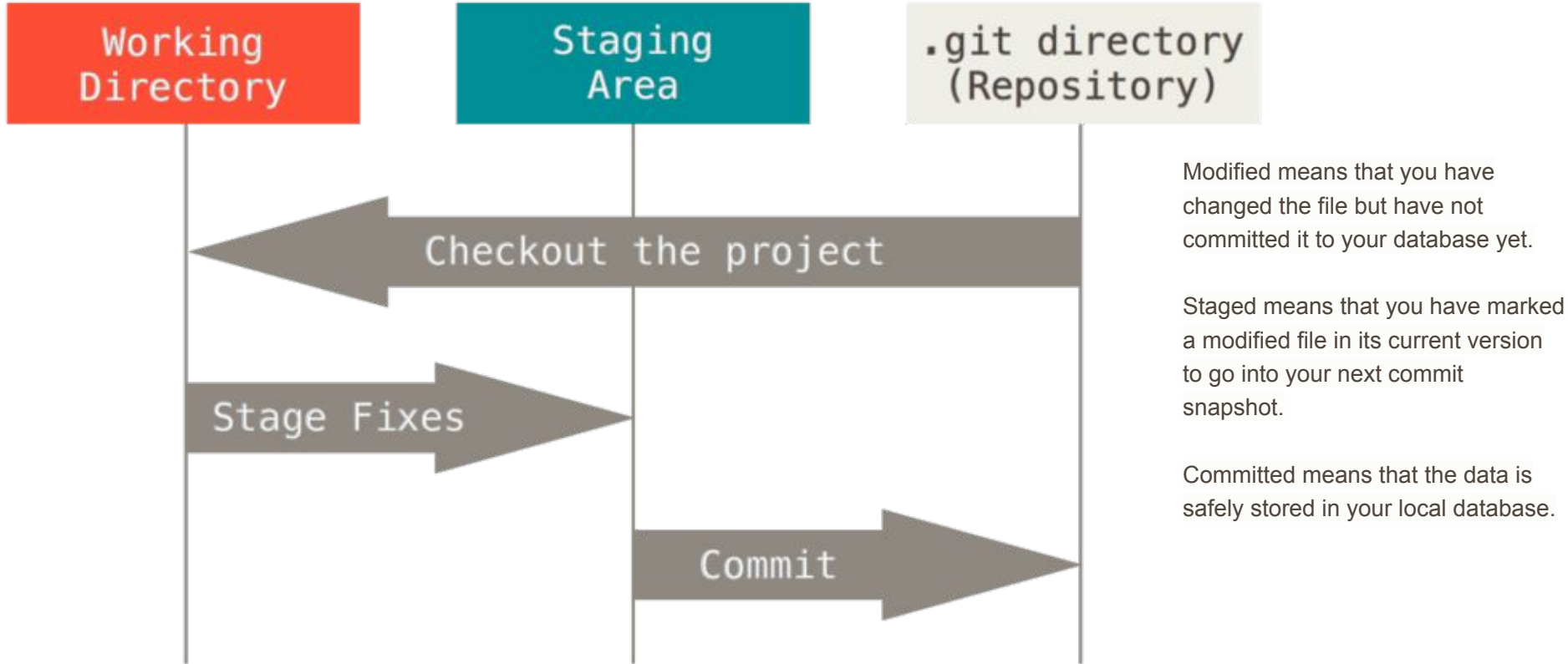
So what is git?

A version control system

- It stores an entire directory-called **repository**



Git states (modified, staged, and committed)



Installing git

Linux

```
sudo dnf install git-all
```

MacOS

Enable xcode, then it is already installed.

Windows

<https://git-scm.com/downloads/win>

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Configuration

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.com
```

```
$ git config --list
```

```
user.name=John Doe
```

```
user.email=johndoe@example.com
```

```
color.status=auto
```

```
color.branch=auto
```

```
color.interactive=auto
```

```
color.diff=auto
```

```
...
```

<https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

Initializing a repository in an existing directory

for Linux:

```
$ cd /home/user/my_project
```

for macOS:

```
$ cd /Users/user/my_project
```

for Windows:

```
$ cd C:/Users/user/my_project
```

and type:

```
$ git init
```

This creates a new subdirectory named `.git` that contains all of your necessary repository files

Begin tracking existing files

```
$ git add *.c
```

```
$ git add LICENSE
```

do an initial commit

```
$ git commit -m 'Initial project version'
```

Untrack and .gitignore

Untrack a file already checked-in

```
git rm --cached FILENAME
```

```
.gitignore file
# Compiled source #
#####
*.com
*.class
*.dll
*.exe
*.o
*.so

# Logs and databases #
#####
*.log
*.sql
*.sqlite

# OS generated files #
#####
####
.DS_Store
.DS_Store?
.*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db
```

[Ignoring files - GitHub Docs](https://docs.github.com/en/committing-changes-to-your-project/ignoring-files)

<https://gist.github.com/octocat/9257657>

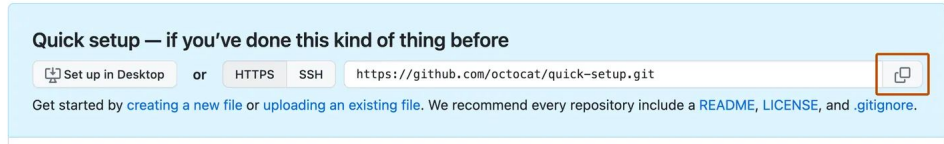
To add this to github

```
$git init -b main
```

```
$ git add .
```

```
$ git commit -m "First commit"
```

Create new repo on github.com



```
$ git remote add origin REMOTE-URL
```

to check if url is correct

```
$ git remote -v
```

Push changes to github.com

```
$ git push origin main
```

- You have upload the files to the server

See for windows, macos, and linux [Adding locally hosted code to GitHub](#)

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```

```
$ cd YOUR-REPOSITORY
```

Make some changes or add files

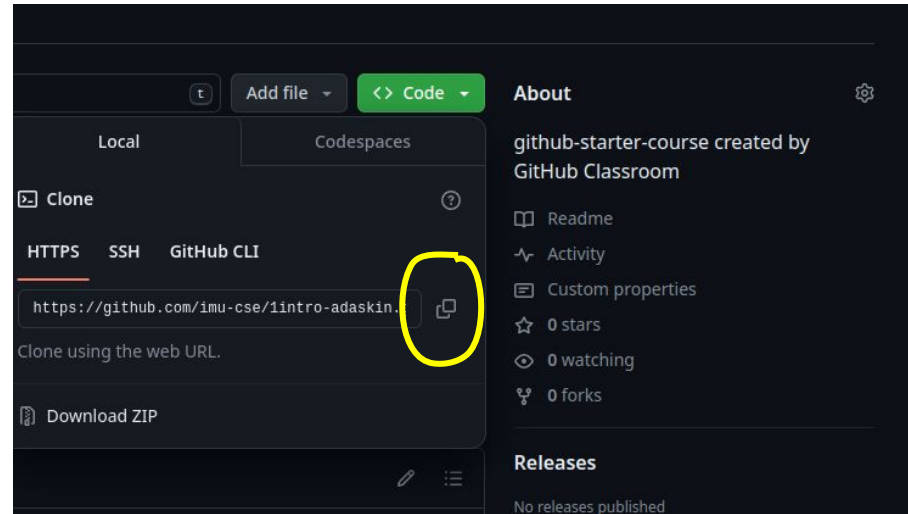
```
$ echo "int main(){}" > newfile.c
```

```
$ git add newfile.c
```

```
$ git commit -m "new file added!"
```

```
$ git push origin main
```

- You have pushed the changes to remote repo on github.com and submitted your assignment



For secure connection

Create ssh-key

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

And use gpg-key for signing commits

<https://docs.github.com/en/authentication/managing-commit-signature-verification/adding-a-gpg-key-to-your-github-account>

Next

Documentation and 2d graphs

- Source documentation
- Word, Excel etc
- Markdown
- Latex
- Jupyter notebook
 - Google Colab
- Python matplotlib
 - 2D plots
 - 3D plots