your items to fit the size of the container.

You can also create grids within grids.

CSS Resets, Devtools

GIT (and github)

↓

version control system.

Git works on local machine, while github is a remote storage facility on the web.

Setting up 'git' :-

→ git config --global user.name "Your Name"
→ git config --global user.email "Yourname@example.com"
To enable colorful output with 'git', type-
→ git config --global color.ui auto

To verify things are working properly enter the below commands
→ git config --get user.name
→ git config --get user.email

Creating an SSH Key

An SSH key is a cryptographically secure identifier. Its a very long password used to identify your machine.

Github uses SSH keys to allow you to upload to your repository without having to type in your username & p/w anytime.

First, we need to see if you have an SSH key already installed.

For that type,

→ ls ~/.ssh/id_rsa.pub

If it doesnot exist then create a new one like below:-

→ ssh-keygen -C <your email>

Link your generated SSH key to your Github.

To even check your SSH key type

→ cat ~/.ssh/id_rsa.pub

Git Command Line fundamentals:-

→ Need help with any 'git' command type:

$ git help <verb> Ex:- git help config.

or

$ git <verb> --help

→ Initialize a repository from ~~existing code~~ within the directory in terminal.

$ git init

where you want the git local repo to be

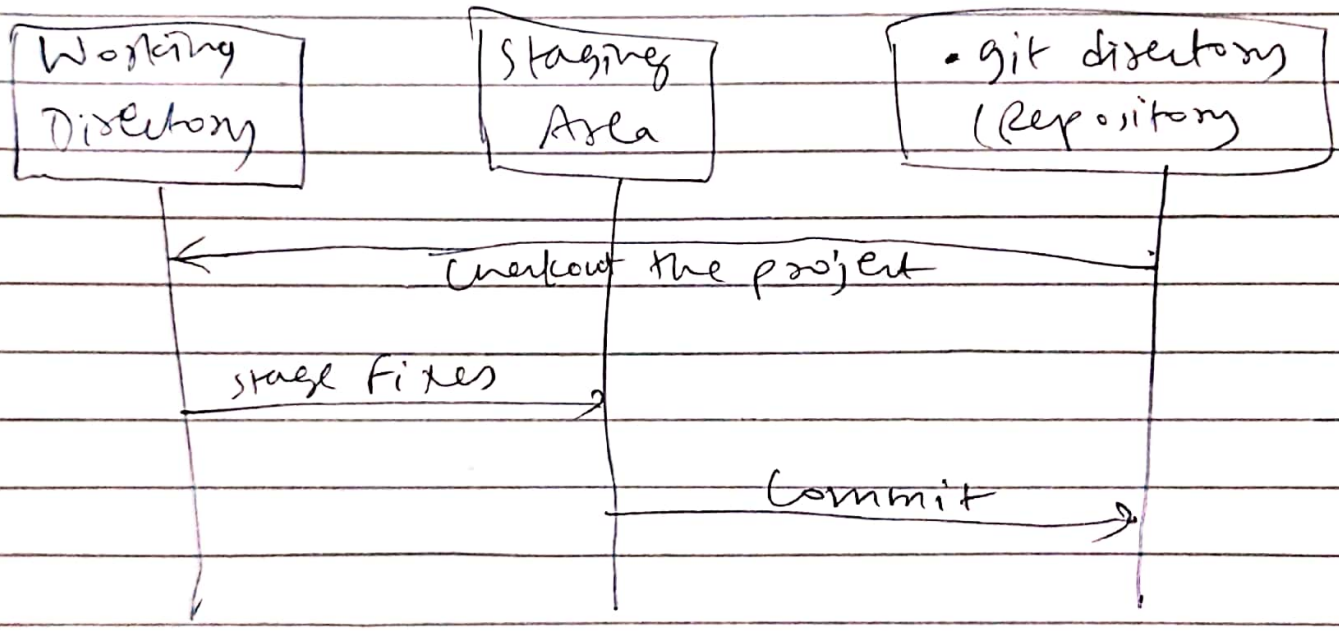
$ git status

To check the status of files, tracked or untracked, ready to commit or not i.e; the state of files.

For ignoring files from git

$ touch .gitignore

create a .gitignore file and add filenames to ignore those files. you can also add *.pyc like extensions to ignore all files of those extensions.



Add files to staging area

To add all files

$ git add -A

To add individual files

$ git add filename.

Remove files from staging area

To remove all files

$ git reset

To remove individual files

$ git reset filename.

$git commit -m "<message>"

will commit the changes with a message

$git log

Through this we can see the history of all the commits we made along with hash numbers of all commits, along with the author name and message of the commit.

Cloning a Remote Repo

$git clone <url> <where to clone>

Ex:-

$git clone ../remote_repo.git .

url can also be a website. ↓ where to clone.

$git clone https://github.com/<username>/<reponame> .

Viewing info about the remote repo

$git remote -v

gives paths

$git branch -a

lists all branches local and remote.

To see changes made to the code

$git diff

Pushing the changes to remote repo

```
$git pull origin master
$git push origin master
```

people often forget git pull but when there are multiple developers people keep pushing code to that remote repo while we're working on our features. So git pull pulls any changes that have been made since the last time we pulled from that repository.

'origin' is the name of remote repo
'master' is the branch we wanna push to

Branching

Its not a good practice to work on master branch hence we create side branches and later merge them back.

```
$git branch <branch-name>   (to create)
$git checkout <branchname>  (to use)
```

```
$git branch
```

lists all the local branches

Ex:
```
$git branch
  calc-divide
* master.
```

the star shows the current branch working on.

After committing to local repo if you want to push branch to remote

$ git push -u origin <branch name>

-u : It tells git that we wanna associate local calc-divide branch with the remote calc-divide branch

<branch name> : branch we wanna push to

This is required for the first time we are pushing (after, (to branch)

we can just say $ git pull and $ git push and it knows that those 2 branches are associated to each other.

Once the companies run their unit tests and all on the sidebranch and are ready to merge with master then,

Merge a Branch

$ git checkout master
$ git pull origin master
$ git branch --merged (tells all the branches that are merged nothing more)
$ git merge <branchname> (merges side branch with master)

$ git push origin master.

You can again run $ git branch --merged to check if the branch got merged or not.

Deleting a branch

Once merged, if you want to delete the side branch

then,

$ git branch -d <branchname> (deleted locally)

$ git push origin --delete <branchname> (deleted from remote repo)