GIT : getting started (part 1)

linux commands

- `pwd` - present working directory
- `cd <directory-name>`
- `ls -la` - everything including hidden folders

git configuration

system configurations

- System-wide configurations when you install git software in your machine.
- Typically, on linux it lies under `/etc/gitconfig`.
- On windows it could be `c:\\program-files\` or something similar (you can check).

global configurations

- This is again System-wide configuration.
- Typically, on linux it lies under `~/gitconfig`.
- On windows, it could be `C:\\Users\\$USER` (you can check).

We will configure our user name and email using commands -

```
git config --global user.name "prashant"
git config --global user.email prashant@gmail.com
```

local configurations

- When you run `git init` command, a hidden directory with name `.git` should be created.
- This is the directory that contains all your git configurations particular to your project (locally).

If you want to remove git tracking from your project, you can delete this `.git` folder completely.

Verify git config

```
git config --list
```

STEPS

```
# First navigate to your project directory / folder
# And then run following commands

# STEP 1 : Open terminal (`cd` to project root)
git init

# STEP 2: check of status of git repository
git status

# STEP 3: to add files in staging area (to track)
git add <file-name>

# STEP 4: create a file `.gitignore`

# STEP 5: run following command *only if* un-necessary files are not reflecting
# in `git status` command
git add --all

# STEP 6: whenever you run git commit all files under staging area are commited and
# certain unique "commit-id" is generated (`SHA1`).
git commit -m "<commit-message>"

# STEP 7: run git log to see commit id and commit message history
```

```
git log

# STEP 7: copy https URL from github and paste
git remote add origin <https://github.com/python10sep/mini_xkcd.git>

# STEP 8:
git push -u origin master
```

`git` documentation

- what is `staging area`?

> The staging area can be described as a preview of your next commit. When you create a git commit, Git takes changes that are in the staging area and make them as a new commit. You are allowed to add and remove changes from the staging area. The staging area can be considered as a real area where git stores the changes.

- `git init` - initializes project to git. Basically this command will create local git configurations. It tells `git` that this is the code repository we need to "track" changes in.
- `git add` - adds the files into `staging area`. It basically prepares files for to be committed. Meaning, its part of preparation for versioning.
- `git status` - feel free to run this command a ton times!
  This command is always a good idea. This command shows you what branch you're on,
  what files are in the staging area, what files git is tracking and any other important information.
- `git commit` - Records file version permanently in version history.
- `git commit -m "descriptive message"` - elaborated version of `git commit` command where you can provide your own "custom" message against a particular commit.
- `git log` - Browse history of changes and inspect the evolution of project files.

- `git remote add origin <https-url-copied-from-github>`
  We can add associated remote repository against label called `origin`.
  We can create multiple such labels like `origin`.
  `origin` is just a popular label used to represent remote URL.

- **NOTE** -
  Don't be confused.
  Here `remote repository` means the repository we created on `github.com`

  Refer additional notes (will be provided separately).