

**CHEAT SHEAT  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction**

**What is Version Control?**

Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time.a

**2 Types of version control system?**

* Centralized
* Distributed

**What is Git?**

**Git** is the most commonly used version control system.

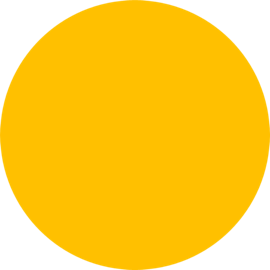
**Git** tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to.

**Git** also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

**Why Git?**

* **Free**
* **Open source**
* **Super-fast**
* **Scalable**
* **Cheap branching/ Merging**

**Git workflow**

****

****

**Directory Staging Area Repository**

**Setting up an account**

git config --global user.name “<username>”

git config --global user.email “<github email>”

git config user.name # Display name

git config user.email # Display email

**Creating Snapshots**

**Helpful Message**

git –help # Display help in full version

git -h # Display help in shorter version

**Initializing a repository**

git init

**Staging files**

git add sample.html # Stages a single file

git add sample.html sample2.html # Stages multiple files

git add \*.html # Stages with a pattern

git add . # Stages all the content of your current dir

**Viewing the Status**

git status # View the full status

git status -s # View the short status

**Committing the staged files**

git commit -m “Your Message” # Commits with one line of message

git commit # Open the default editor to type long message

**Skipping the staging area**

git commit -am “Message”

**Upload your local repo to Github**

1. git remote add origin <”Github Repo Link”>
2. git push -u <name of remote> <name of branch>

ex. git push -u origin main

*once you set an upstream, you can only use “git push” only*

**Download from Github repo to your local repo**

git pull -u <name of remote> <name of branch>

*once you set an upstream, you can only use “git pull” only*

**Creating Branch**

**Show all git branches**

git branch

**Creating a new branch**

git checkout -b <name of new branch>

**Go to a specific branch**

git checkout <name of a branch>

**Checking the difference of your current branch to other branch**

git diff <other branch>

**Merging Branch**

**Merge your current branch to different branch**

git merge <other branch>

* If there’s a merge conflict. (fix it manually, then commit)

**Remove file from Staging area**

git reset

**Remove commit**

git reset --hard HEAD~1 # remove 1 commit

**View all commits**

**View all commits**

git log

**Go back in Specific commits**

git checkout <hash code of commit>

* + To undo and go back on the current commit  
     git switch –
  + To create a new branch with that commit  
     git switch -c <new Branch name>