**Version Control using GIT:**

1. **Implementing versions locally, i.e…. means saving these versions locally using git**

**Initialise Git – empty git repository inside the directory where you are:**

* git init
* ls –a (to see all the hidden files inside a folder)
* .git (is used to track your changes, commit your changes and version control)

**To see what files are inside your staging area?**

* Git status

**\*\* Red files are untracked files**

**To add the files to staging area:**

* Git add “filename”

**To add more than one file name inside a directory to staging area:**

* Git add “filename1” “filename2”
* Git add **.**
* Git status (now you can see the file in green which states this is added to staging area and ready to commit)

**How to remove everything from staging area:**

* Git rm --cached –r **.**

**Commit Command:**

* Git commit –m “commit message”

**To see the commits you have made (history):**

* Git log

**To revert the changes from local repository to working directory:**

1. Check the status of the working directory first to see any files for changes using **“git status”**
2. Then check the differences in the files (Working Dir vs Repository) – “**git diff ‘filename’**”
3. To roll back the changes from repository to working directory – **“git checkout “filename” “**
4. **How you can make a remote repository, so the repository which is hosted on somebody else server by using GitHub.**

**Steps:**

1. Go to “github.com”
2. Login using “vamsheeakula/$Maanvi22062020”
3. Click on New Repository and create a repository
4. “origin” – is simply the name of your remote. You can use any word, but it is recommended as conventions.
5. Use command line instructions to push your code to repository
6. Using the URL in the github, push the existing repository (local) from the command line
7. Create a remote in local repository:

* “Git remote add origin **(paste the remote github repository URL)”**

**Git remote add origin** [**https://github.com/vamsheeakula/Story.git**](https://github.com/vamsheeakula/Story.git)

1. **Remote is created using step 7**
2. **Now, push local repository into remote repository which is called origin**

* **Git push –u origin master**
* **Master is the default branch**

1. **Refresh the github once push is completed and you can see your files.**
2. **You can see your commits in the master branch using:**

* **Insights -> Graphs -> Network**

1. **GitIgnore:**

**How to prevent certain files from committing to you local and remote repositories**

**You can prevent files from adding to staging area (example: secret password files), for this you need to create a “.gitignore” file and add the files list to this “.gitignore” file**

1. **Cloning:**

**Cloning a remote repository to pull it into your local machine:**

**To use any skeleton project from github, we can use closing to download the project from the github**

* **Login to github website**
* **Search for some open source apps which are already committed by others**

<https://github.com/dkhamsing/open-source-ios-apps>

* Select Clone or Download and copy the link
* Now use below link to checkout/clone/download the project to your local
* Git clone <https://github.com/austinzheng/swift-2048.git>

1. **Branching and Merging:**

We can create branches for a specific need (i.e.. if you want to use a specific branch for a specific implementation apart from the main branch.

Let say:

**Commit 1** – This is the main branch with code

**Commit 2** – This is the main branch with code changes

Want to create new feature, so want to create a new branch. Simulatneosly we can update main branch and also can work on experimental branch. Now we have two branches (Main and experimental) and these can be merged with main branch when needed.

Commit 1 🡪 Commit 2 🡪 Commit 3(Main Branch)

Branch (Copy of Commit 1) Experimental Branch

**Command:**

* Git branch “branch-name” --- git branch alien-plot
* Git branch (to check all the branches) --- Highlighted branch is the one which you are on currently

Alien-plot

**. master**

* You can switch to other branch using below command:

Git checkout alien-plot

**Merge Changes from experiment branch to master branch:**

* Go back to your master branch
* While we are on the master branch we are going to merge changes inside the experimental branch to master branch
* Git checkout master
* Git merge alien-plot

1. **Forking and Pull Request:**

**Forking:**

Creating a new remote repository using the existing remote repository is called forking.

Don’t get confuse here and Forking vs cloning are different.

Remote Reposiroty (story1) – Owner (Vamshee Akula)

Now somebody else can create a new remote repository using story1.

Remote Repository (Story2) – Owner (Bhavana Akula)

**Pull Request:**

Now continuing the above example,

Story2 Repository has additional implementation in repository and now if Bhavana Akula wanted to merge Story2 with Story1, then Vamshee Akula needs to accept and approve the merge. Since Vamshee Akula needs to accept and initiate merge, the action here needs to do the PULL, but not the PUSH.

So owner of the repository will verify the new changes in the repository and pulls the code and merges.

**Steps:**

1. There is a Story created by Vamshee Akula
2. Bhavana Akula had forked the Story project
3. Now Bhavana Akula has the copy of remote repository with same name – Story, but this is for Bhavana Akula
4. Now Bhavana Akula, modifies some code and commit to master branch of her story (Please note this will not affect the branch of Vamshee Akula)
5. Now if Bhavana Akula wanted to get her changes merged to the original branch of Vamshee Akula, then she needs to create “PULL REQUEST” with detailed comments for the request.
6. Now Vamshee Akula is notified with the PULL Request with details, then Vamshee Akula will verify the comments and if he is ok, then approves the “PULL request” and “Confirm Merge”.
7. Then the merge will be completed.
8. You can check in the Network tab for the graph. This will be shown in the blue line colour.

**Quick setup — if you’ve done this kind of thing before**

### create a new repository on the command line

**echo "# angularRep" >> README.md**

**git init**

**git add README.md**

**git commit -m "first commit"**

**git branch -M master**

**git remote add origin https://github.com/vamsheeakula/angularRep.git**

**git push -u origin master**

### push an existing repository from the command line

**git remote add origin https://github.com/vamsheeakula/angularRep.git**

**git branch -M master**

**git push -u origin master**