Flutter Basic UI elements cheat Sheet

VCS (Version Control Software)

1. As soon as you start your Development process, a VCS keeps a copy of your work to a cloud storage which prevents file loss.
2. Since a VCS saves your files to a cloud storage, your friend who wants to work with you can easily get a copy of your project leaving the original copy in the cloud storage. And when he/she is done a VCS will do all the code merging for you, It’s nice isn’t it? 😁
3. Finally, for every time you save your file, a VCS creates a new instance of your file and keeps the older version. This instances are called commit. Let’s say you have just finished building your application, but want to add a feature, just like the 3rd problem above, you decided to revert back, instead of spending hours debugging your ‘extra feature ' you can just easily ask a VCS to take you back to when you’re code was working 😁 it’s that simple.

Git/Git Hub

git saves your work locally on your machine just in case you’re in a place with no/slow internet connection and can’t access the cloud version of your work. And when you finally get a stable network, it’ll upload your work the Cloud storage (This is called pushing). And I think another reason is because of the wide range of features the GitHub website offers, such as forking another Dev/organization’s project and make pull requests making it easy to contribute to open source projects and all that.

Flutter Basic UI elements cheat Sheet

VCS (Version Control Software)

1. As soon as you start your Development process, a VCS keeps a copy of your work to a cloud storage which prevents file loss.
2. Since a VCS saves your files to a cloud storage, your friend who wants to work with you can easily get a copy of your project leaving the original copy in the cloud storage. And when he/she is done a VCS will do all the code merging for you, It’s nice isn’t it? 😁
3. Finally, for every time you save your file, a VCS creates a new instance of your file and keeps the older version. This instances are called commit. Let’s say you have just finished building your application, but want to add a feature, just like the 3rd problem above, you decided to revert back, instead of spending hours debugging your ‘extra feature ' you can just easily ask a VCS to take you back to when you’re code was working 😁 it’s that simple.

Git/Git Hub

git saves your work locally on your machine just in case you’re in a place with no/slow internet connection and can’t access the cloud version of your work. And when you finally get a stable network, it’ll upload your work the Cloud storage (This is called pushing). And I think another reason is because of the wide range of features the GitHub website offers, such as forking another Dev/organization’s project and make pull requests making it easy to contribute to open source projects and all that.

Flutter Basic UI elements cheat Sheet

Git Hub

• STAGE & SNAPSHOT

Working with snapshots and the Git staging area

- git status

show modified files in working directory, staged for you next commit .

- git add [file]

add a file as it looks now to your next commit (stage)

- git reset [file]

unstage a file while retaining the changes in working directory

- git diff

diff of what is changed but not staged

- git diff --staged

diff of what is staged but not yet commited

- git commit -m “[descriptive message]”

commit your staged content as a new commit snapshot

• BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

- git branch

list your branches. a * will appear next to the currently active branch

- git branch [branch-name]

create a new branch at the current commit

- git checkout

switch to another branch and check it out into your working directory

- git merge [branch]

merge the specified branch’s history into the current one

Flutter Basic UI elements cheat Sheet

Git Hub

- ### SHARE & UPDATE

Retrieving updates from another repository and updating local repos

- git remote add [alias] [url]

add a git URL as an alias

- git fetch [alias]

fetch down all the branches from that Git remote

- git merge [alias]/[branch]

merge a remote branch into your current branch to bring it up to date

- git push [alias] [branch]

Transmit local branch commits to the remote repository branch

- git pull

fetch and merge any commits from the tracking remote branch

- ### REWRITE HISTORY

Rewriting branches, updating commits and clearing history

- git rebase [branch]

apply any commits of current branch ahead of specified one

- git reset --hard [commit]

clear staging area, rewrite working tree from specified commit