WEEK 01

TEAM: I´m on the number 2 team, and they all speak English. We´ll have our meetings on Mondays and Wednesdays.

NOTES:

Resources I have free access to:

- JavaScript: Novice to Ninja (SitePoint)
- Eloquent JavaScript (EJS)
- You Don't Know JS Yet
- Exploring ES6
- MDN Web Docs

Software I´ll be working with:

- VS Code
- Sitepoint
- Trello
- Node/NPM – latest version
- Git and Github
- Microsoft Teams

GIT IT TOGETHER

Git is a version control system. Version control is a method of recording changes made to a file or a set of files. Recording changes allows us to come back to specific versions as we carry on coding.

Git is a version control tool

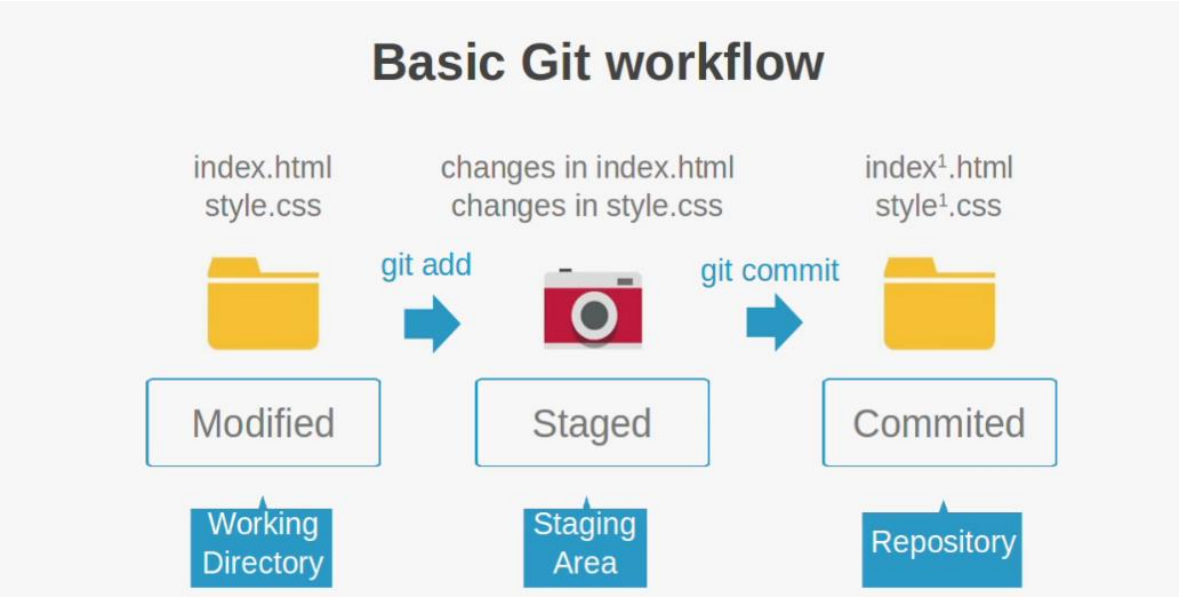
GitHub is a code sharing and publishing website that hosts git repositories.

3 Main States:

Modified: State of files in the working directory that you are currently working on.

Staged: State of files in the staged area that you have modified and added to the staging area for Git to take a snapshot of.

Committed: State of files when Git has taken a snapshot of the current state and are stored in the repo now.



Git commands:

git add git commit git push git pull

Short way to show changes in detail / Log:

git log –pretty=oneline

Undoing a staged change:

Use git reset HEAD [name of file I want to reset]

```
kauress@kauress-yolopad ~/Desktop/theme $ git reset HEAD index.html
Unstaged changes after reset:
M       index.html
kauress@kauress-yolopad ~/Desktop/theme $ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Undoing a commited change:

Use git reset –soft HEAD~1

```
kauress@kauress-yolopad ~/Desktop/theme $ git reset --soft HEAD~1
kauress@kauress-yolopad ~/Desktop/theme $
```

Undoing a pushed change:

Use git revert HEAD

```
kauress@kauress-yolopad ~/Desktop/theme $ git revert HEAD
[master a9d3686] Revert "Committing our third mistake, we left the other two in
 place"
 1 file changed, 3 insertions(+), 3 deletions(-)
```

More commands

Git diff: This displays the differences between the working directory and the index.

Git diff <file>: This displays file changes so that you can go over what has been changed the previous version and the newer changed version of your file.

Git fetch and git merge: To get changes made by others.

```
kauress@kauress-yolopad ~ $ cd Desktop
kauress@kauress-yolopad ~/Desktop $ cd theme
kauress@kauress-yolopad ~/Desktop/theme $ git status
On branch master
nothing to commit, working directory clean
kauress@kauress-yolopad ~/Desktop/theme $ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/LearnableGitCourse/Learnable-Git-Course
   a9d3686..9dfb217  master     -> origin/master
kauress@kauress-yolopad ~/Desktop/theme $ git merge origin/master
Updating a9d3686..9dfb217
Fast-forward
 README.md | 27 +-------------------------
 1 file changed, 1 insertion(+), 26 deletions(-)
kauress@kauress-yolopad ~/Desktop/theme $
```

Git pull: Basically both fetch and merge at the same time.

Branches

_A git branch represents and independent line of development

_Each brand is associated with its own unique changes

_**Master** is the main branch. HEAD refers to the tip of the branch I am on (most recent commit)

git branch -a: List all the current branches.

git branch [name of the branch]: To create a new branch.

git checkout [name of the branch]: To change to the branch I want to work on.

git checkout -b [name of a new branch]: To create and switch to the new branch.

git branch -m [name of an existing branch] [new name of that branch]: To change the name of a branch.

To delete a branch we need to switch to another branch and use:

git branch -d [name of the branch I want to delete]

Merge

git merge: Will merge independent branches.

git merge [branch name]: Merge the branch.

FAST FORWARD MERGE

git checkout master

git merge [name of the branch we want merged into master]

NO FAST FORWARD MERGE: A separated merge commit is generated

git checkout master

git merge –no-ff [name of the branch we want merged into master]

Add a commit message when asked

GIT REBASE

Moves a branch up to a new base commit. Creates a linear branching history. Recommended to local changes and not repositories that are shared with others.

GITHUB PAGES

_Webpages hosted by Github

_Two types: User and Projects

_Great way to showcase projects to potential employers and customers

Putting a project on a GitHub Page:

```
kauress@kauress-yolopad ~ $ cd Desktop
kauress@kauress-yolopad ~/Desktop $ cd theme
kauress@kauress-yolopad ~/Desktop/theme $ git checkout -b gh-pages
Switched to a new branch 'gh-pages'
kauress@kauress-yolopad ~/Desktop/theme $ git push origin gh-pages
Username for 'https://github.com': LearnableGitCourse
Password for 'https://LearnableGitCourse@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/LearnableGitCourse/Learnable-Git-Course.git
 * [new branch]      gh-pages -> gh-pages
```

Git Clone

Copy the https URL from a repo

Use git clone [paste de URL here]

Forking

To have a copy of any repo on our own GitHub account and use it.

If we wanted it locally, we can clone it.

PULL REQUESTS

Useful to contribute to open-source projects. It helps us propose changes to someone else´s project and that person can decide to merge it or reject the proposed changes.

EXTRA RESOURCES I USED

https://www.youtube.com/watch?v=M8H-mT4oeAg

https://www.youtube.com/watch?v=Uszj_k0DGsg