**GIT**

**What is GIT?**

* It is a distributed version control system.
* Every user has a local version.
* It is designed to handle everything from small to very large project with speed and efficiency.
* Helps creating something called src code.
* It allows different people to work on the same file at the same time on the same project.
* Example src code: <head>hello</head>

<body></body>

**VERSION CONTROL**

* Version control is a software that tracks and manage changes to files over time.
* Version control generally allow users to revisit earlier versions of the file, compare changes between versions and undo changes etc.

**BENEFIT OF GIT.**

* It allows you to roll back code.
* To track changes cross multiple files.
* Integrity

**LOCAL GIT REPO.**

* Is any directory that you create.
* Any directory with a GIT file becomes a GIT repo (git init)

**FOURS AREAS IN GIT.**

* Working Area (create or edit)
* Staging, for tracking index files use=> (git add), to track just the file (git add “file").
* Local repo, commit message (git commit-m)
* Remote repo is master version for git push (GitHub).

TWO WAYS OF COPYING CODE FROM SOMEONE ELSE

* **Fork**.
* Is a copy from one remote repo to another remote repo or to your remote repo.
* **Clone**
* Coping code from one remote repo to your local remote repo.

Note: Whenever someone pushes something or commit something it shows exactly what they are doing.

* **CONFIG COMMAND.**

Once GIT is installed, we need to use GIT Config Command.

* Configure GIT => git config--global user.name"name".
* Configure GIT=> got config-- global user. email "email"
* Verify config => git config -- list.
* **Authenticate**.

Then later is Authenticate to GitHub.

* Get personal Access token => GitHub > dev settings>personal Access token>create and save token.
* Save it somewhere.

**BRANCHING**.

* Is an isolated line of code.git branch = list all branched.
* Create and switch branch = git checkout -b <branchname>
* (Switch only) git checkout<branchName>

**GIT COMMANDS**

* repo -> repository
* clone -> bring a repo down from the internet (remote repository like GitHub) to your local machine
* add -> track your files and changes with Git
* commit -> save your changes into Git
* push -> push your changes to your remote repo on GitHub (or another website)
* pull -> pull changes down from the remote repo to your local machine
* status -> check to see which files are being tracked or need to be committed
* init -> use this command inside of your project to turn it into a Git repository and start using Git with that codebase