**GITHUB**

1. **git remote add origin URL** specifies that you are adding a remote repository, with the specified URL, as an origin to your local Git repo.
2. pull is a combination of 2 different commands:  
   -fetch  
   -merge
3. **fetch** gets all the change history of a tracked branch/repo.
4. **merge** combines the current branch, with a specified branch.
5. **pull** is a combination of fetch and merge. It is used to pull all changes from a remote repository into the branch you are working on.

# **GITHUB FLOW**

1. The GitHub flow is a workflow designed to work well with Git and GitHub.
2. It focuses on branching and makes it possible for teams to experiment freely, and, make deployments regularly.
3. The GitHub flow works like this:  
   -Create a new Branch  
   -Make changes and add Commits  
   -Open a Pull Request  
   -Review  
   -Deploy  
   -Merge

# **BRANCHING**

1. Branching is the key concept in Git. And it works around the rule that the master branch is ALWAYS deployable.
2. That means, if you want to try something new or experiment, you create a new branch! Branching gives you an environment where you can make changes without affecting the main branch.
3. When your new branch is ready, it can be reviewed, discussed, and merged with the main branch when ready.
4. When you make a new branch, you will (almost always) want to make it from the master branch.

# **COMMITS**

1. After the new branch is created, it is time to get to work. Make changes by adding, editing, and deleting files. Whenever you reach a small milestone, add the changes to your branch by commit.
2. Adding commits keeps track of your work. Each commit should have a message explaining what has changed and why. Each commit becomes a part of the history of the branch, and a point you can revert back to if you need to.

# **PULL REQUEST**

1. Pull requests are a key part of GitHub. A Pull Request notifies people you have changes ready for them to consider or review.
2. You can ask others to review your changes or pull your contribution and merge it into their branch.
3. GitHub shows new commit and feedback in the "unified Pull Request view".

# **REVIEW**

1. When a Pull Request is made, it can be reviewed by whoever has the proper access to the branch. This is where good discussions and review of the changes happen.
2. Pull Requests are designed to allow people to work together easily and produce better results together!
3. If you receive feedback and continue to improve your changes, you can push your changes with new commits, making further reviews possible.

# **DEPLOY**

1. When the pull request has been reviewed and everything looks good, it is time for the final testing. GitHub allows you to deploy from a branch for final testing in production before merging with the master branch.
2. If any issues arise, you can undo the changes by deploying the master branch into production again!
3. Teams often have dedicated testing environments used for deploying branches.

# **MERGE**

1. After exhaustive testing, you can merge the code into the master branch!
2. Pull Requests keep records of changes to your code, and if you commented and named changes well, you can go back and understand why changes and decisions were made.
3. You can add keywords to your pull request for easier searching!