Git

Stup

set a name that is identifiable for credit when review version history

```
git config --global user.name "[firstname lastname]"
```

set your email

```
git config --global user.email "[valid-email]"
```

set automatic command line coloring for Git

```
git config --global color.ui auto
```

Start

1.already have files in your local machine

```
cd <file_path>
# Initialize a new Git repository
git init
# Add all files to staging
git add .
# Commit the changes
git commit -m "Initial commit"
```

Create a new repository on GitHub

```
# Add the remote origin
git remote add origin <repository URL>
# Push changes to the remote repository
git push -u origin master
```

2.Creates a local copy of an existing repository.

```
git clone <repository URL>
```

HEAD

HEAD: reference that points to the current branch's latest commit.

To compare the latest commit (HEAD) with the commit before it:

```
git diff HEAD~1 HEAD
```

committing changes.

show modified files in working directory, staged for your next commit

```
git status
```

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

```
git add [file]
```

unstage a file while retaining the changes in working directory

```
git reset [file]
```

diff of what is changed but not staged

```
git diff
```

diff of what is staged but not yet commited

```
git diff --staged
```

commit your staged content as a new commit snapshot

```
git commit -m "descriptive messag"
```

Ctrl+Z Changes

Undo **Uncommitted** Changes:

Discard working directory changes:

```
git checkout -- [file]
```

Unstage a file:

```
git reset HEAD [file]
```

Undo **Committed** Changes:

see different commit:

```
git log
```

Create a new commit that undoes the changes of a specified commit:

```
git revert [commit]
```

stage the changes without creating a new commit:

```
git -n revert [commit]
```

permanently delete your changes and commits after that

```
git reset --hard [commit]
```

Branch

see exciting branches:

```
git branch
```

Create a new branch:

```
git branch [branch-name]
git push -u origin [branch-name]
```

Switch Branches:

```
git checkout [branch-name]
```

Merge a branch into the current branch:

```
git merge [branch-name]
```

Delete a branch:

```
git branch -d [branch-name]
```

Tag

simple reference to a specific commit,

see exciting tag:

```
git tag
```

Create a new tag:

```
git tag [tag-name]
git push origin --tags
```

conflict

we try to Push Changes:

```
git push origin master #we get error about conflict
```

Pull Latest Changes:

```
git pull origin master
```

Resolve conflicts by editing files to remove conflict markers, this command will open merge tool:

```
git mergetool
```

Commit the Merge:

```
git add .
git commit -m "Resolved merge conflicts"
```

Push Resolved Changes:

```
git push origin master
```