

INSIDE MASTER BRANCH AGAIN !!!!!!

Topics

1. Setup commands

- Configure
 - Cloning
 - Creating New

2. Frequently used ones

- Get current remote repo (pull and branch)
- Update changes to local repo (add and commit)
- Sending changes to remote repo (merge and push)

3. Rest Misc Commands

Setup

Configure (git config)

- Username:
`git config user.name "Homo Sepian"`
- Email:
`git config user.email "abcd@xyz.com"`

This name and email is used to track who has committed which changes.

Use `git config --global <rest code>` to avoid configuring these for each repository

Cloning

`git clone <link to repository>`

To change the name of directory-name of cloned repository, add directory-name at the end of command: `git clone <link to remote repository> <directory name>`

Creating

Go inside the project folder then:

`git init`
`git remote add origin <link to remote repository>`

Frequently Used Commands

Pushing to remote repository

```
git push <link to remote repository>
```

To not remember the link to remote repository, we create a kind of variable to store the remote repository link using:

```
git remote add <name of variable> <link to remote repository>
```

Generally <name of variable> is *origin*.

We can also specify the branch we want to push by adding the branch name at the end of these commands. Eg: `git push origin master`

Miscellaneous Commands

Create branch

```
git branch <branch name>
```

To override existing branch with current one, use:

```
git branch -f <branch name>
```

Switch branch

```
git checkout <branch name>
```

To create and checkout the branch, use

To not specify origin each time, we can use -u option after git push to make it default remote repository to push to. After doing that, from next time, command for pushing just remains:
`git push`

```
git checkout -b <branch name>
```

Go to just previous commit

```
git checkout -f
```

View commits

```
git log
```

To view commits with differences

```
git log -p
```

To view 5 commits, use

```
git log -p -5
```

Delete a branch

```
git branch -d <branch name>
```

View status

```
git status
```

To view status in a compact form, use:

```
git status -s
```

View differences in different branches

```
git diff <other branch name>
```

staged means changes that are added but not committed. To view difference between last commit and staged files, use:

```
git diff --staged
```

Merging branches