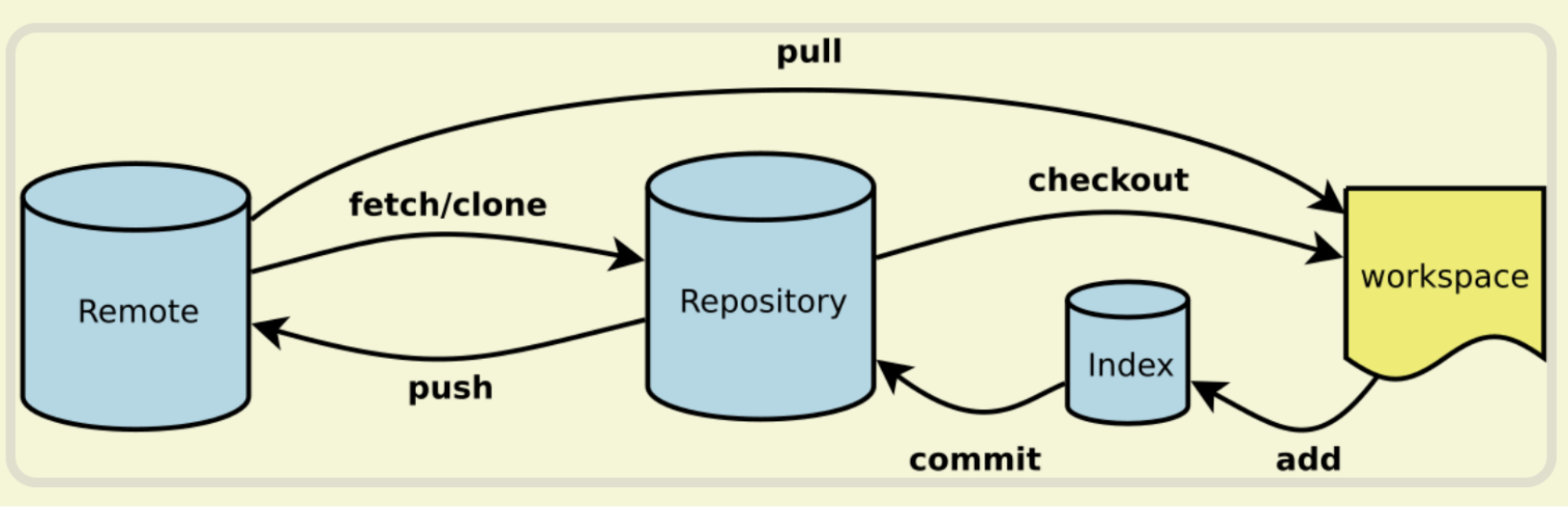
Tutorial to Github

INF - 552

Jan. 18 2019

Introduction



https://classroom.github.com/a/NWuFxKH9

create a new repository

create a new directory, open it and perform a

```
git init
```

to create a new git repository.

checkout a repository

create a working copy of a local repository by running the command

```
git clone /path/to/repository
```

when using a remote server, your command will be

```
git clone username@host:/path/to/repository
```

workflow

your local repository consists of three "trees" maintained by git. the first one is your `Working Directory` which holds the actual files. the second one is the `Index` which acts as a staging area and finally the `HEAD` which points to the last commit you've made.

add & commit

You can propose changes (add it to the **Index**) using

```
git add <filename>
```

```
git add *
```

This is the first step in the basic git workflow. To actually commit these changes use

```
git commit -m "Commit message"
```

Now the file is committed to the **HEAD**, but not in your remote repository yet.

pushing changes

Your changes are now in the **HEAD** of your local working copy. To send those changes to your remote repository, execute

```
git push origin master
```

Change *master* to whatever branch you want to push your changes to.

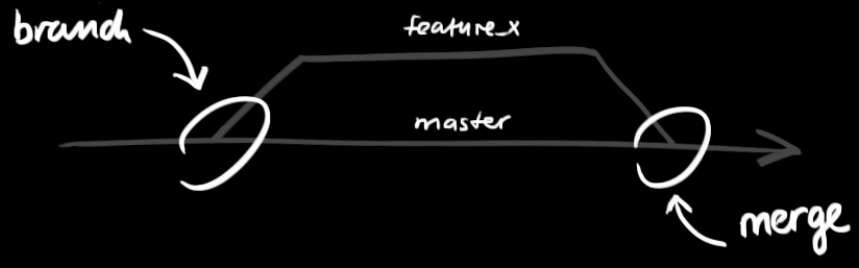
If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

```
git remote add origin <server>
```

Now you are able to push your changes to the selected remote server

branching

Branches are used to develop features isolated from each other. The *master* branch is the "default" branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.



create a new branch named "feature_x" and switch to it using

```
git checkout -b feature_x
```

switch back to master

```
git checkout master
```

and delete the branch again

```
git branch -d feature_x
```

a branch is *not available to others* unless you push the branch to your remote repository

```
git push origin <branch>
```

update & merge

to update your local repository to the newest commit, execute

`git pull`

in your working directory to *fetch* and *merge* remote changes.

to merge another branch into your active branch (e.g. master), use

`git merge <branch>`

in both cases git tries to auto-merge changes. Unfortunately, this is not always possible and results in *conflicts*. You are responsible to merge those *conflicts* manually by editing the files shown by git. After changing, you need to mark them as merged with

`git add <filename>`

before merging changes, you can also preview them by using

`git diff <source_branch> <target_branch>`

git version control

- git init
- git add [name_of_your_file] [–all]
- git commit [-m] [comment]
- git remote add origin [name_of_your_remote]
- git push -u origin master
- git push
- git pull
- git rm

git version control

- git branch [-d]
- git checkout [-b]
- git push origin [name_of_your_new_branch]
- git merge
- ....

`https://git-scm.com/docs`
`https://github.com/GarageGames/Torque2D/wiki/Cloning-the-repo-and-working-with-Git`