

---

---

# Working With a Remote

— Sharing is caring —

---

---

# Agenda

- What is a remote?
- Adding a remote
- Cloning
- Pushing
- Pulling

# What is a remote?

A remote is the short name for a remote repository which is usually an exact clone of our local repository.

There can be multiple remotes not just one (hence distributed VCS).

Remotes are used mostly as single sources of truth where every collaborator syncs their own work.

# Adding a remote

# Flow of Files

```
$ git init
```

```
$ git add
```

```
$ git commit
```

```
$ |
```

Local Repository



Readme.md

Staging Area

Working Directory



Readme.md

# Flow of Files

```
$ git init
```

```
$ git add
```

```
$ git commit
```

```
$ git remote add
```

```
$ |
```

Remote Repository

Local Repository



Readme.md

Staging Area

Working Directory



Readme.md

# Flow of Files

```
$ git init
```

```
$ git add
```

```
$ git commit
```

```
$ git remote add
```

```
$ git push
```

Remote Repository

Local Repository



Readme.md

Staging Area

Working Directory



Readme.md

# Flow of Files

```
$ git init
```

```
$ git add
```

```
$ git commit
```

```
$ git remote add
```

```
$ git push
```

Remote Repository



Readme.md

Local Repository



Readme.md

Staging Area

Working Directory



Readme.md



# git remote

- Adding a remote you need to specify the URL to the remote repo as well as a unique name for your remote

```
git remote add origin git@github.com:PaulNegoescu/gittest.git
```

```
git remote add origin https://github.com/PaulNegoescu/gittest.git
```



name

URL

- List all repos: `git remote -v`
- Delete a repo: `git remote remove origin`

**Warning!**

# Warning!

- You **cannot work** with a remote until you configure your username and email
- You can do this for all projects:

```
git config --global user.name "John Doe"
```

```
git config --global user.email "john@doe.com"
```

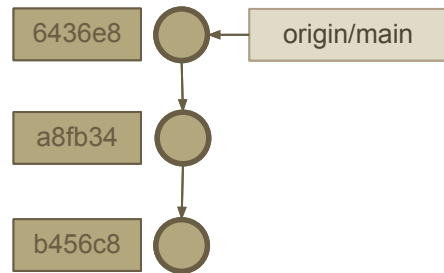
# Cloning – The other way of starting a project

# Cloning

\$ |

Local Repository

Remote Repository



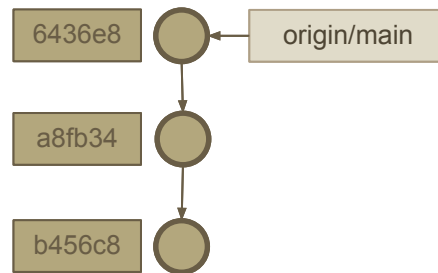
# Cloning

```
$ git clone
```

```
$ |
```

Local Repository

Remote Repository

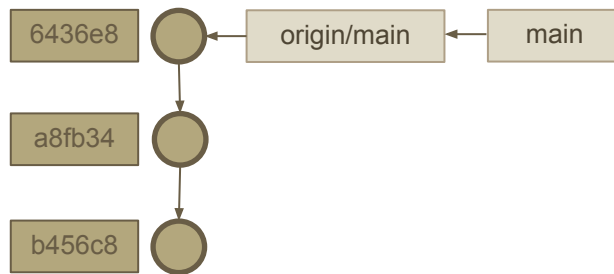


# Cloning

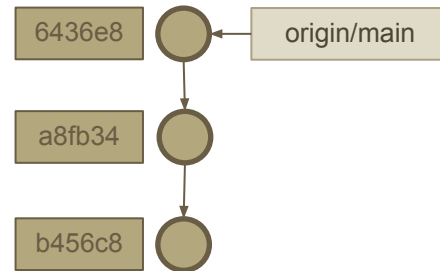
```
$ git clone
```

```
$ |
```

## Local Repository



## Remote Repository



# git clone

- Cloning a remote you need to specify the URL to the remote repo

```
git clone git@github.com:PaulNegoescu/gittest.git
```

```
git clone https://github.com/PaulNegoescu/gittest.git .
```



URL

Path



# Pushing your changes to the remote

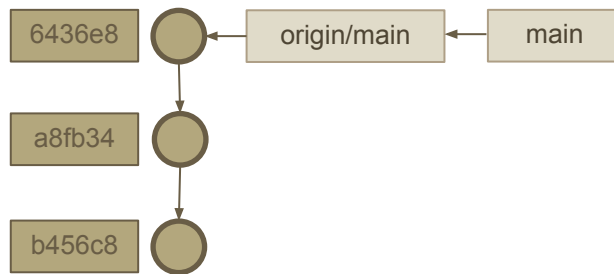
# Pushing

```
$ git clone
```

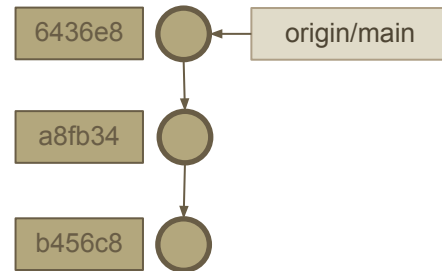
```
$ git commit
```

```
$ |
```

## Local Repository



## Remote Repository



# Pushing

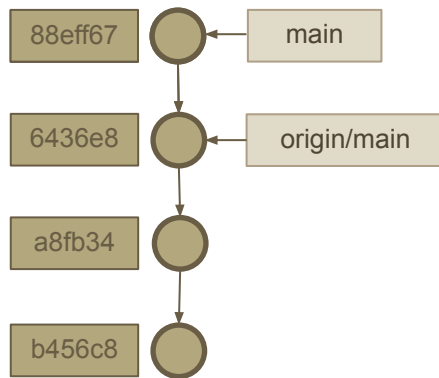
```
$ git clone
```

```
$ git commit
```

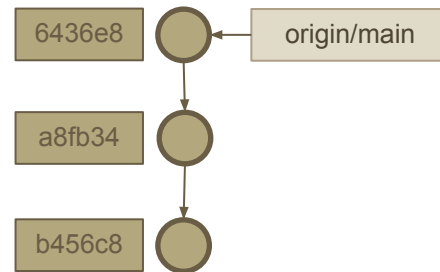
```
$ git push
```

```
$ |
```

## Local Repository



## Remote Repository



# Pushing

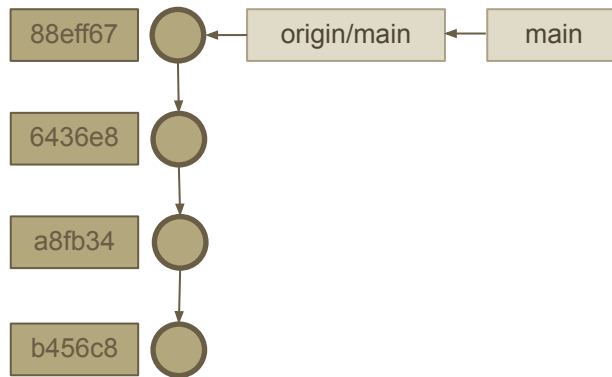
```
$ git clone
```

```
$ git commit
```

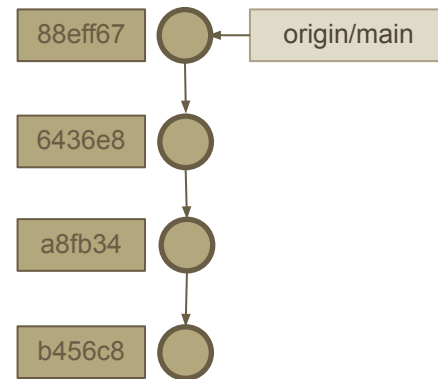
```
$ git push
```

```
$ |
```

## Local Repository



## Remote Repository



# git push

- git push send all your commits on the current branch to the remote if the current branch is tracking a remote branch or if a remote branch is specified

```
git push
```

```
git push origin main
```

- You can set up tracking when pushing:

```
git push -u origin main
```

```
git push --set-upstream origin main
```

# **Pulling changes from the remote**

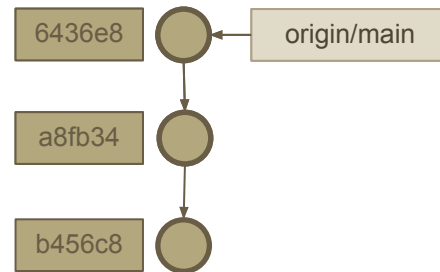
# Pulling

\$ |

Local Repository



Remote Repository

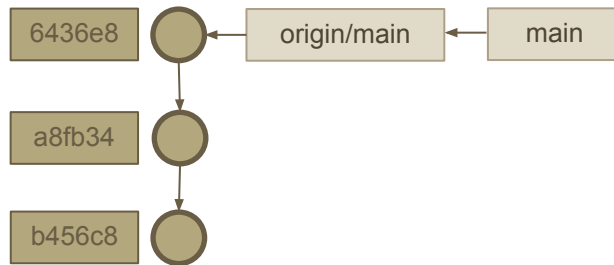


# Pulling

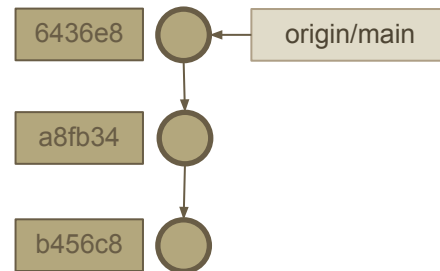
```
$ git pull
```

```
$ |
```

## Local Repository



## Remote Repository





# git pull

- git pull grabs all the changes from the remote branch and brings them into the local repo

```
git pull
```

- git pull is actually a shorthand command for two other commands which are played out in order:

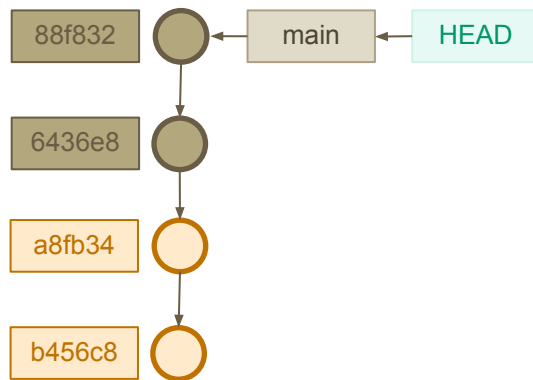
```
git fetch
```

```
git merge origin/main
```

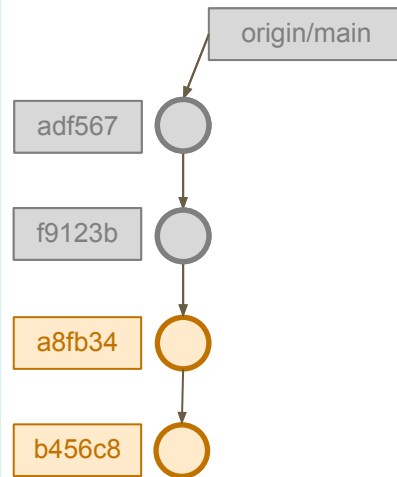
# git pull

```
$ |
```

## Local Repository



## Remote Repository

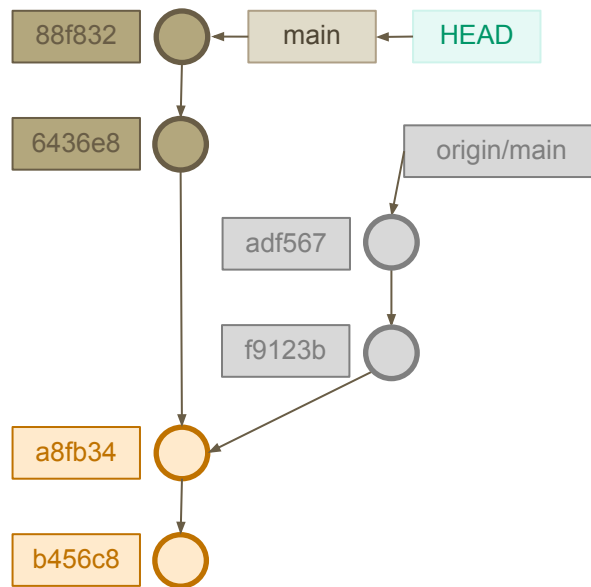


# git pull

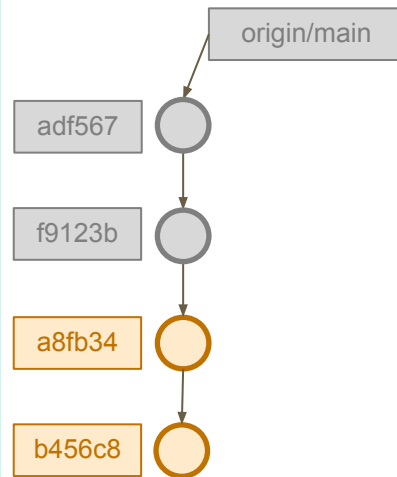
```
$ git pull
```

```
$ |
```

## Local Repository



## Remote Repository

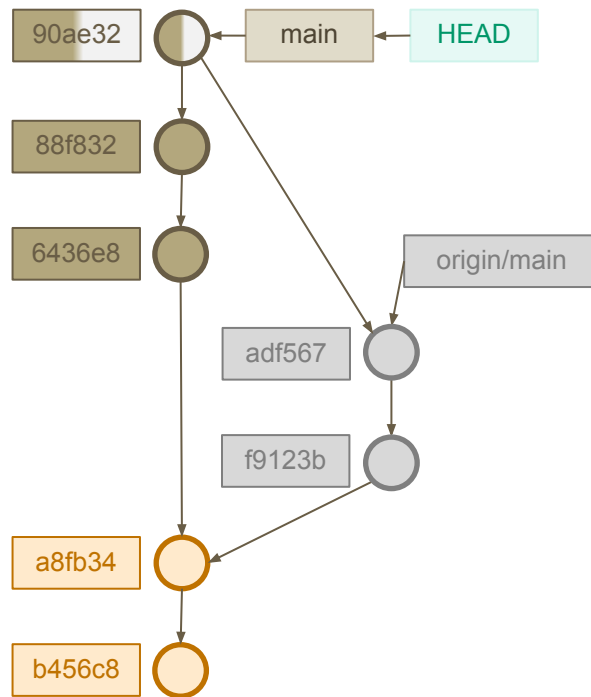


# git pull

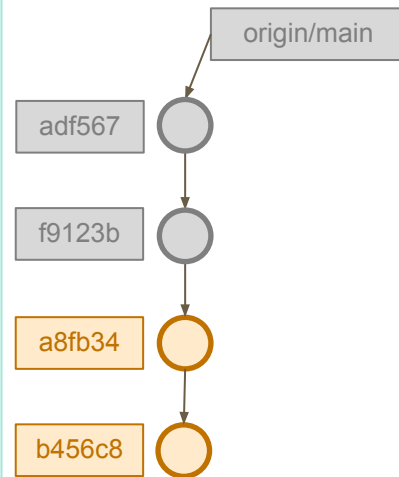
```
$ git pull
```

```
$ |
```

## Local Repository



## Remote Repository



# Cheat Sheet – Usual Workflow

# Cheat Sheet

Right after installing git

```
git config --global user.name "John Doe"  
git config --global user.email "john@doe.com"  
git config --global core.autocrlf false
```

When joining an existing project

```
git clone <repo_url>
```

If beginning a new project from scratch

```
git init
```

# Cheat Sheet

Every time you start working on a task (you will branch off of **develop** or **main**)

```
git switch -c feature/task-name
```

```
git push -u origin feature/task-name
```

As often as possible (at least once every day)

```
git add .
```

```
git commit -m "My very meaningful commit message"
```

```
git switch develop
```

```
git pull
```

```
git switch feature/task-name
```

```
git merge develop
```

```
git push
```

```
git fetch
```

```
git merge origin/develop
```

# Cheat Sheet

When you finish a task

```
git switch develop
```

```
git pull
```

```
git switch feature/task-name
```

```
git merge develop
```

# test that your feature and the project are working, test A LOT!

```
git switch develop
```

```
git merge feature/task-name
```

# you could squash and rebase but that is a discussion for

# another time

```
git push
```

```
git fetch
```

```
git merge origin/develop
```



# Home Assignment

# Home Assignment (~1.5 hrs)

- Go to: <https://learngitbranching.js.org/>
- On the "Main" tab, do the first two sections (Introduction and Ramping Up), all 8 exercises
- On the „Remote" tab do the first 10 exercises (7 + 3)
- Take a screenshot of your **whole screen** on each tab and submit these two screenshots as your homework