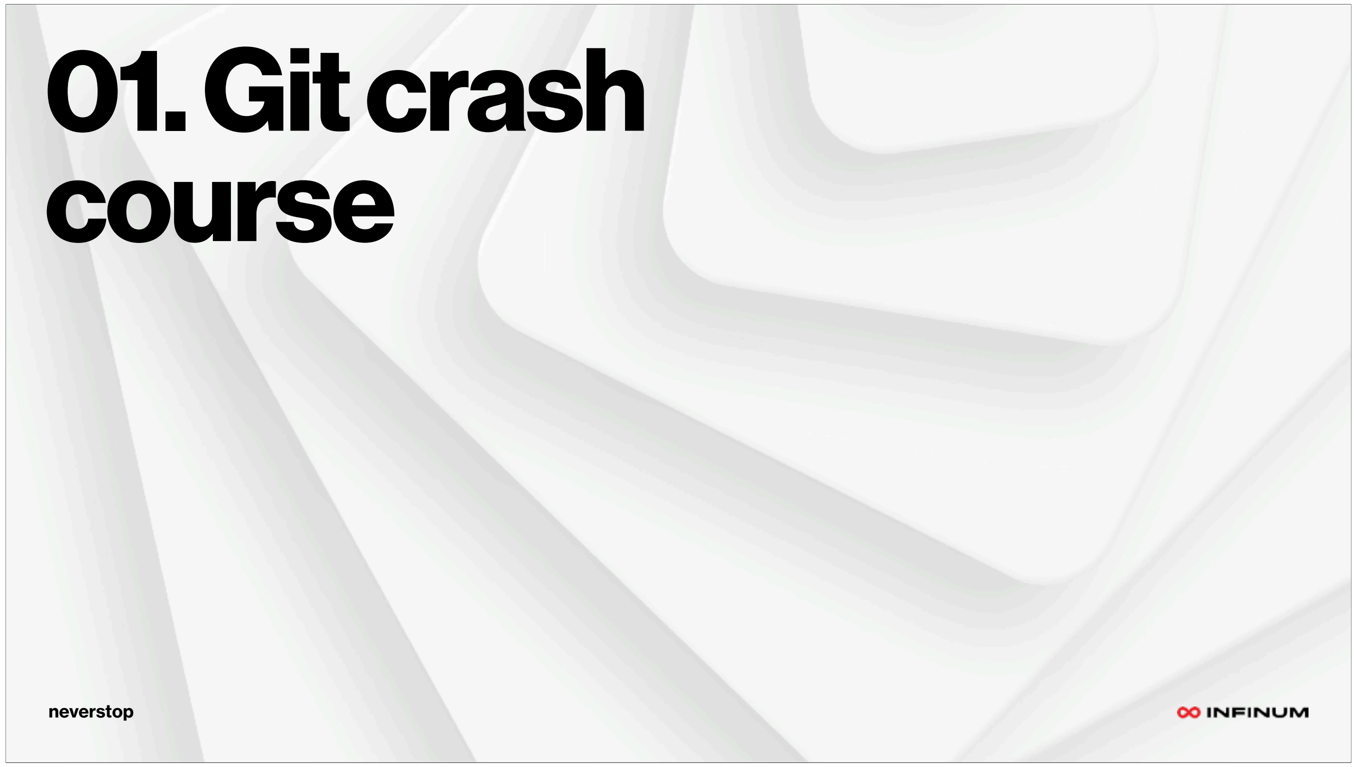


01. Git crash course

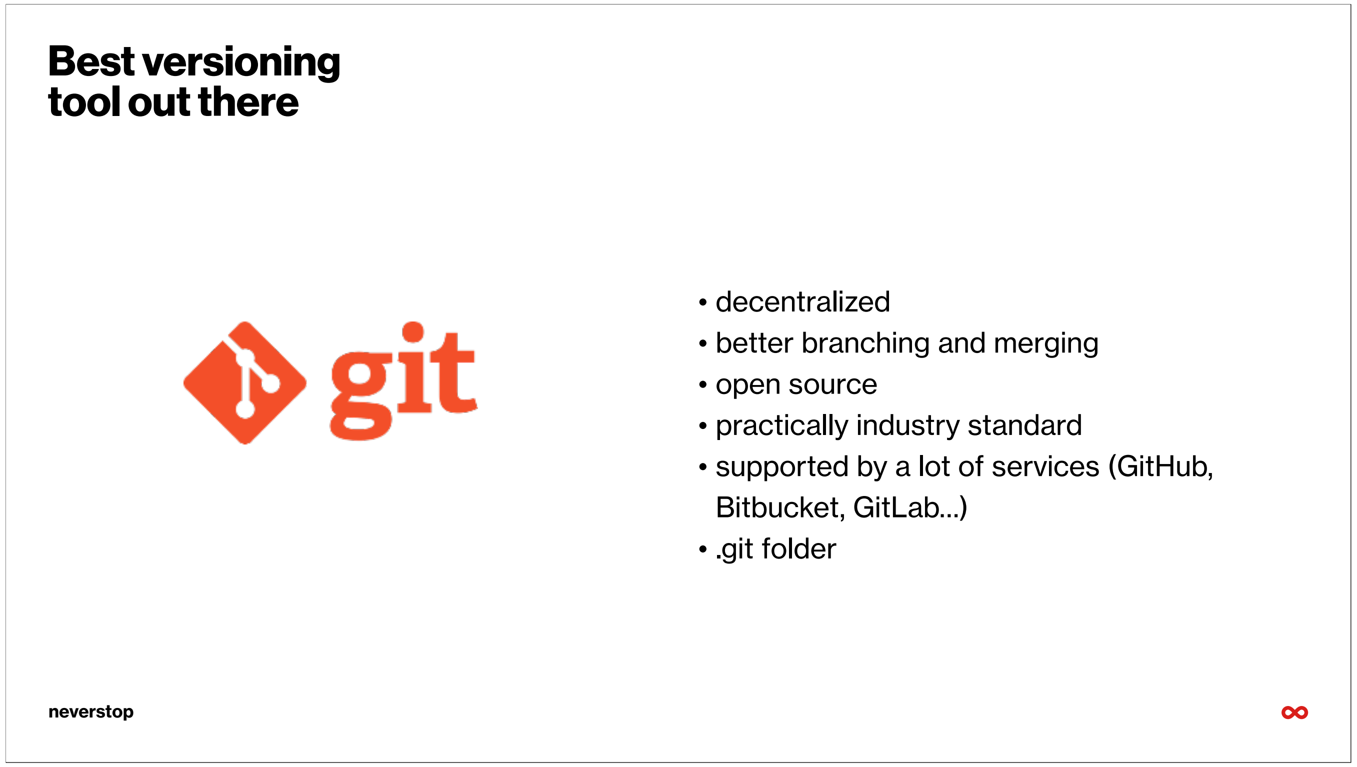
neverstop

INFINUM

- simplified approach to Git through the lecture
- don't fear the git, we'll cover everything in lecture

01

Why Git?



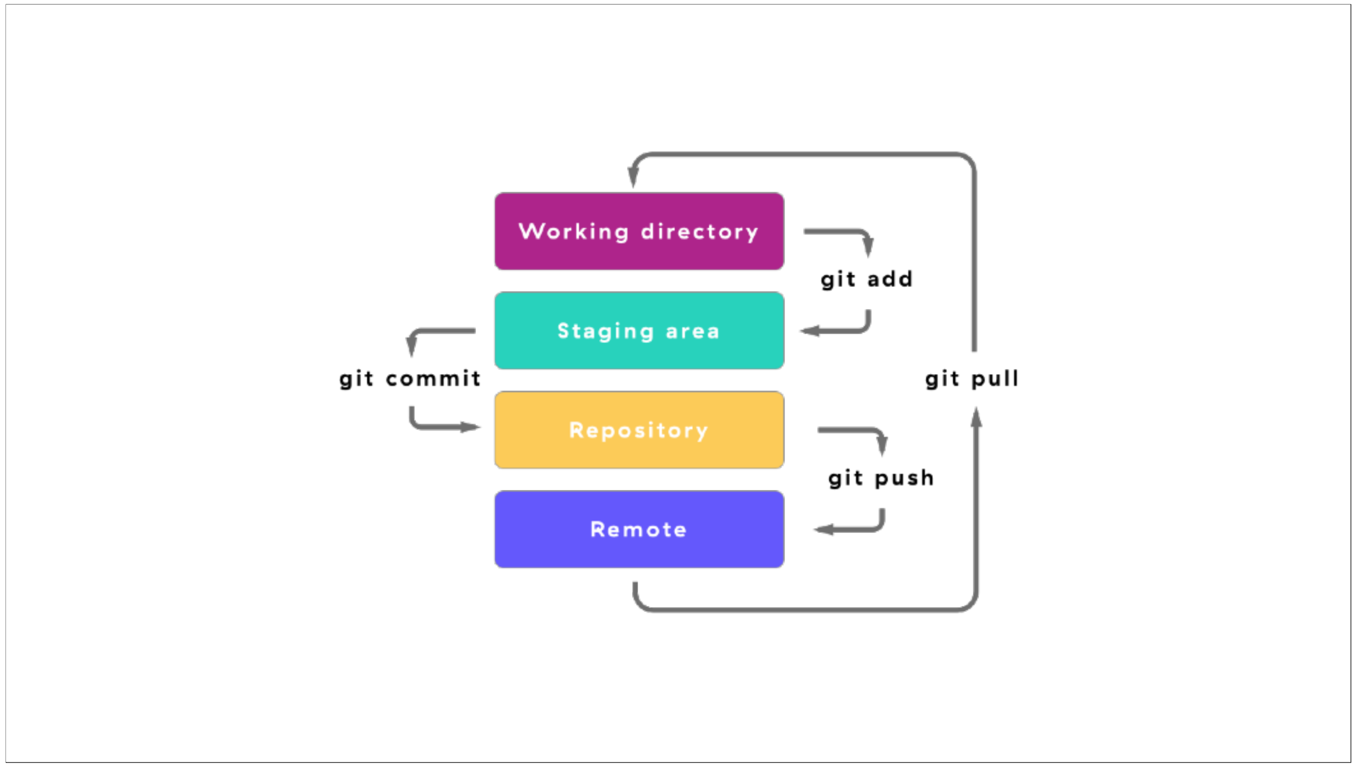
Best versioning tool out there

- decentralized
- better branching and merging
- open source
- practically industry standard
- supported by a lot of services (GitHub, Bitbucket, GitLab...)
- .git folder

- by Linus Torvalds for Linux Kernel
- versioning tool, organize the work in meaningful section aka branches
- code saved locally and remotely
- pull request and merging
- .git folder – technical details
- Git can be visualised as a tree structure, where each commit creates a new node.
- most commands are used to navigate through that

02

How it works?



Working directory

Staging area

Repository

Remote

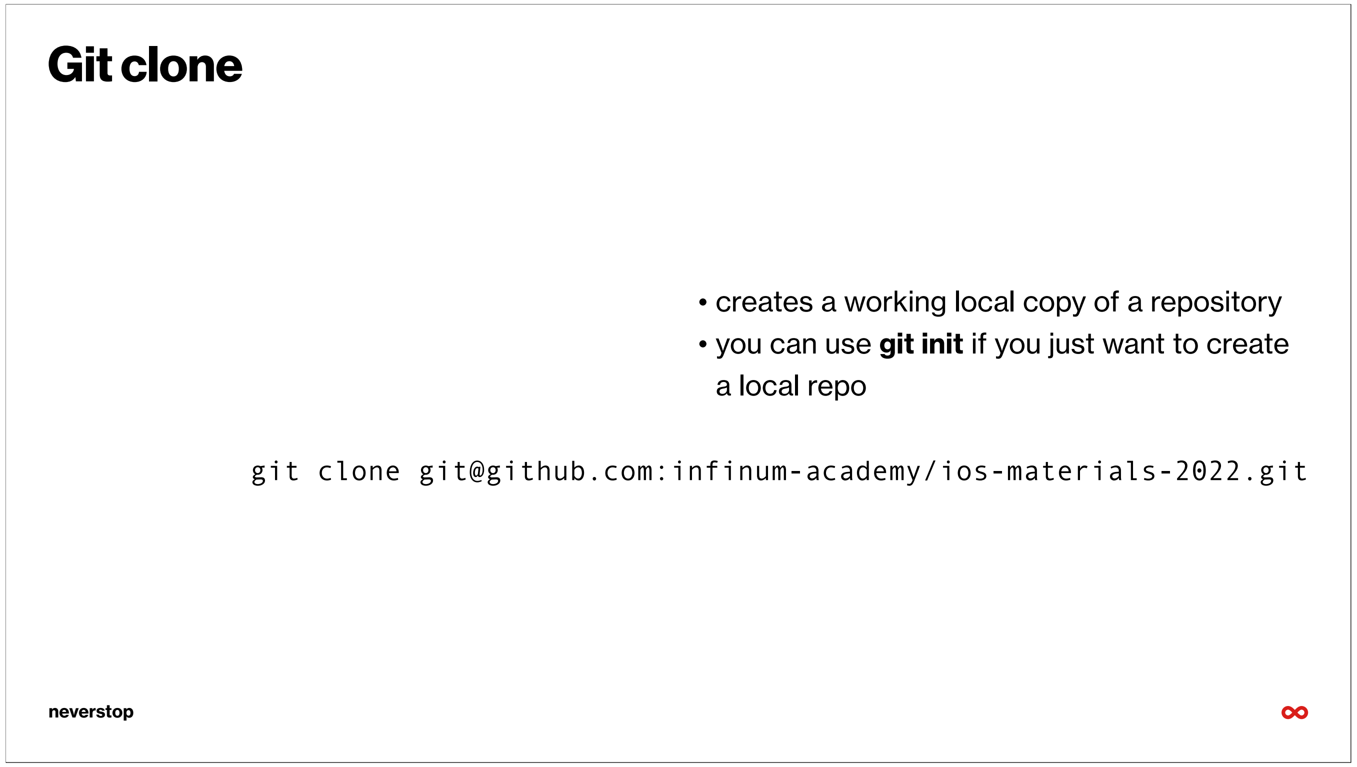
git add

git commit

git push

git pull

- working directory
    - current state of the code on your disk
    - whenever you save the file
- git add groups files in logical entities
- with commit you give it a name
- for example, login -> some backend call, some logic, some UI, will commit multiple files as one commit
- navigate through history through commits
- save on remote repository

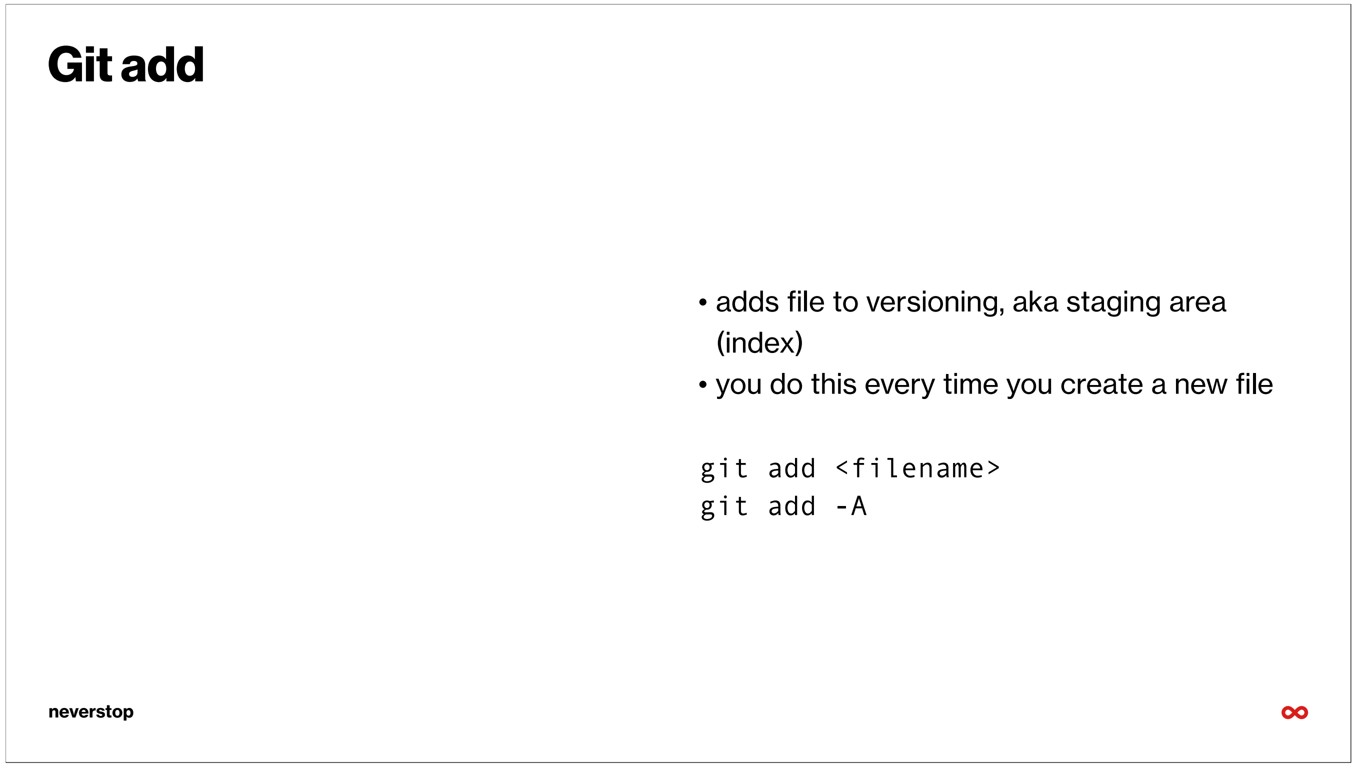


Git clone

- creates a working local copy of a repository
- you can use **git init** if you just want to create a local repo

```
git clone git@github.com:infinum-academy/ios-materials-2022.git
```

- first git command you'll use on course

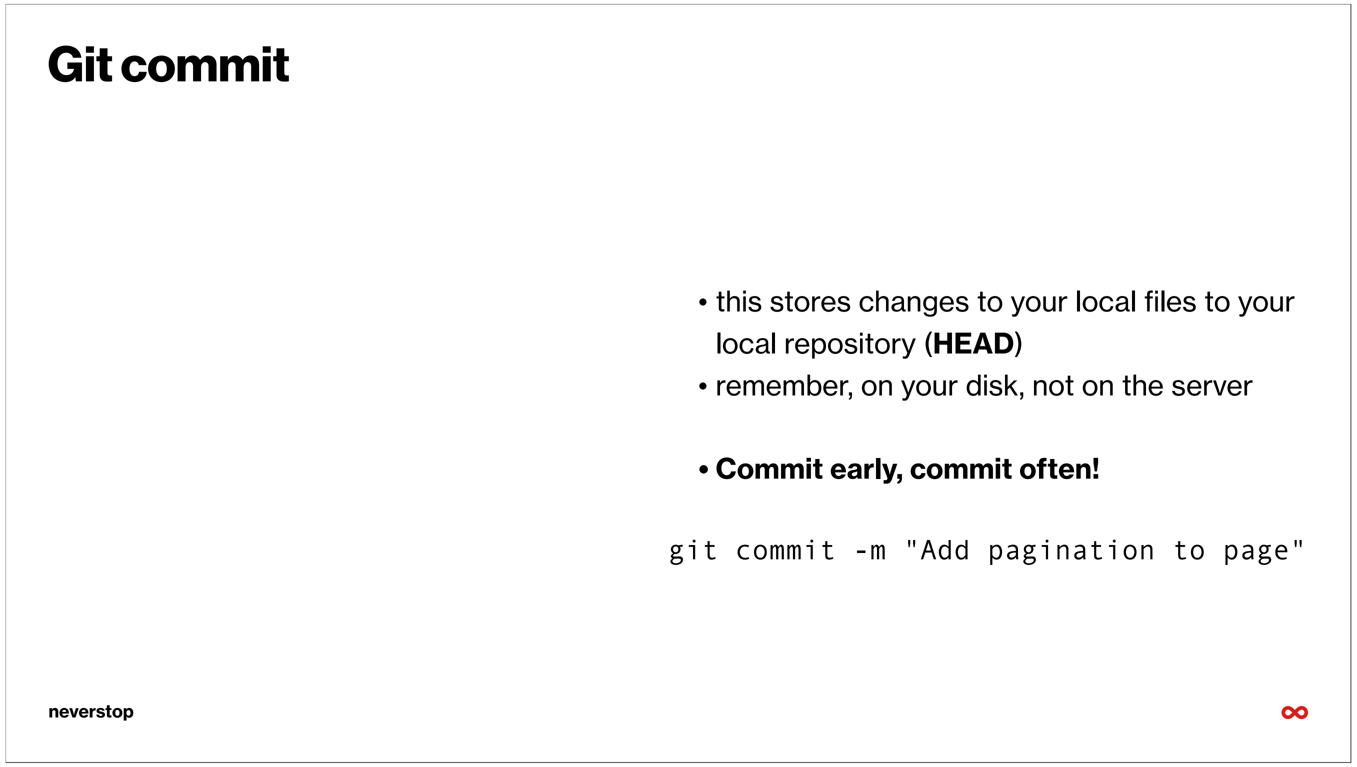


Git add

- adds file to versioning, aka staging area (index)
- you do this every time you create a new file

```
git add <filename>
git add -A
```

- "-A" will add all changes and new files
- a bit more granular approach, so you would use `git add -p`
- in Xcode GUI there is no git add, but git add is part where you can select files and changes before commit
- still not saved - if you change something, you cannot get it back



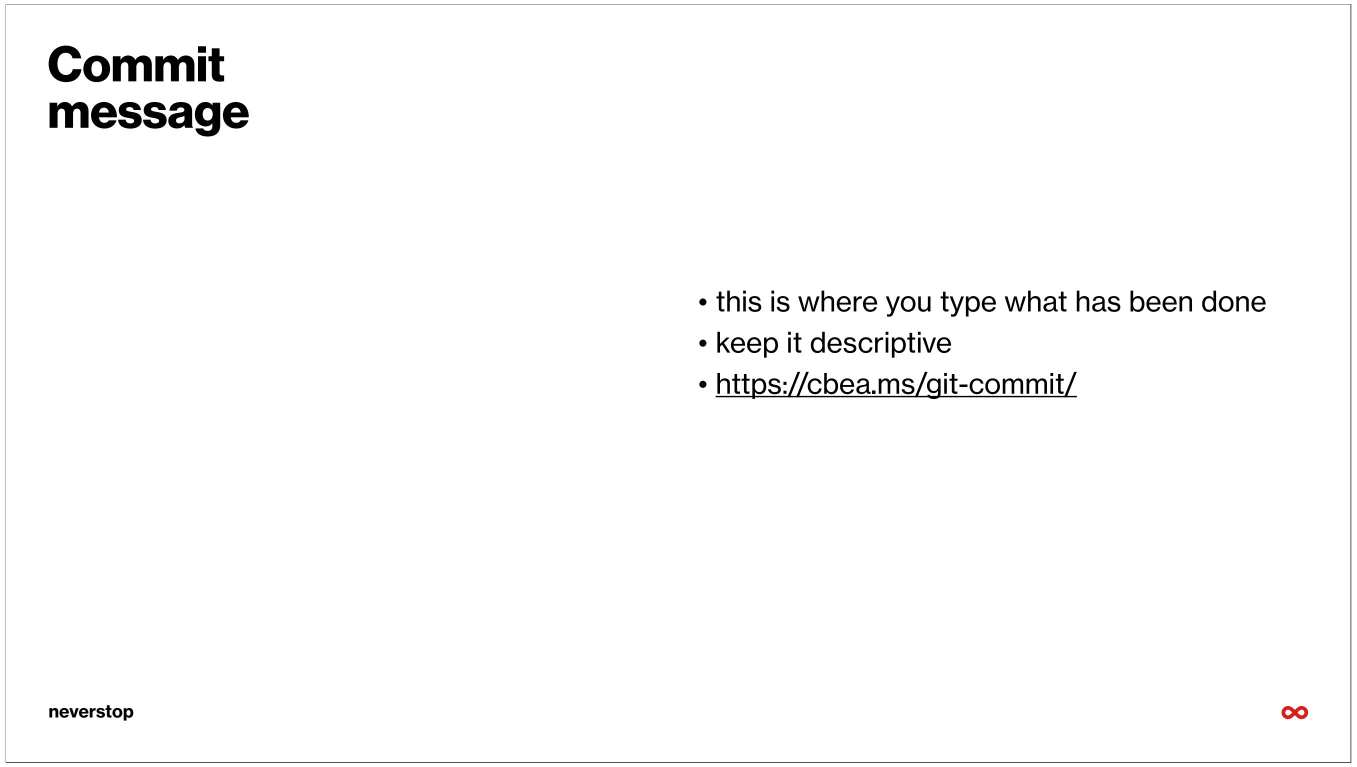
Git commit

- this stores changes to your local files to your local repository (**HEAD**)
- remember, on your disk, not on the server

- **Commit early, commit often!**

```
git commit -m "Add pagination to page"
```

- every commit has a message
- more granular commits
- ability to return in point in history

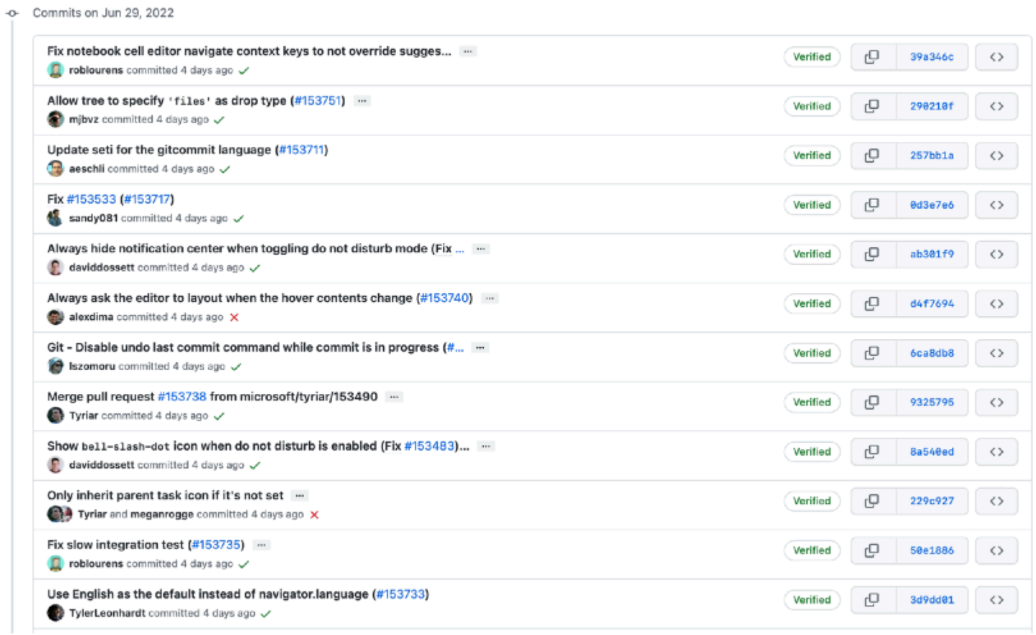


Commit message

- this is where you type what has been done
- keep it descriptive
- https://cbea.ms/git-commit/

neverstop

- commit often and early in development
- Git commit message is the best way to communicate context about a change to fellow developers
- diff will tell you what changed, but only the commit message can properly tell you why



Commits on Jun 29, 2022

Fix notebook cell editor navigate context keys to not override sugges... ···
roblourens committed 4 days ago ✓ — Verified — 39a346c

Allow tree to specify `'files'` as drop type (#153751) ···
mjbvz committed 4 days ago ✓ — Verified — 290210f

Update seti for the gitcommit language (#153711)
aeschli committed 4 days ago ✓ — Verified — 257bb1a

Fix #153533 (#153717)
sandy081 committed 4 days ago ✓ — Verified — 0d3e7e6

Always hide notification center when toggling do not disturb mode (Fix ... ···
daviddossett committed 4 days ago ✓ — Verified — ab301f9

Always ask the editor to layout when the hover contents change (#153740) ···
alexdima committed 4 days ago ✗ — Verified — d4f7694

Git - Disable undo last commit command while commit is in progress (#... ···
lszomoru committed 4 days ago ✓ — Verified — 6ca8db8

Merge pull request #153738 from microsoft/tyriar/153490 ···
Tyriar committed 4 days ago ✓ — Verified — 9325795

Show `bell-slash-dot` icon when do not disturb is enabled (Fix #153483)... ···
daviddossett committed 4 days ago ✓ — Verified — 8a540ed

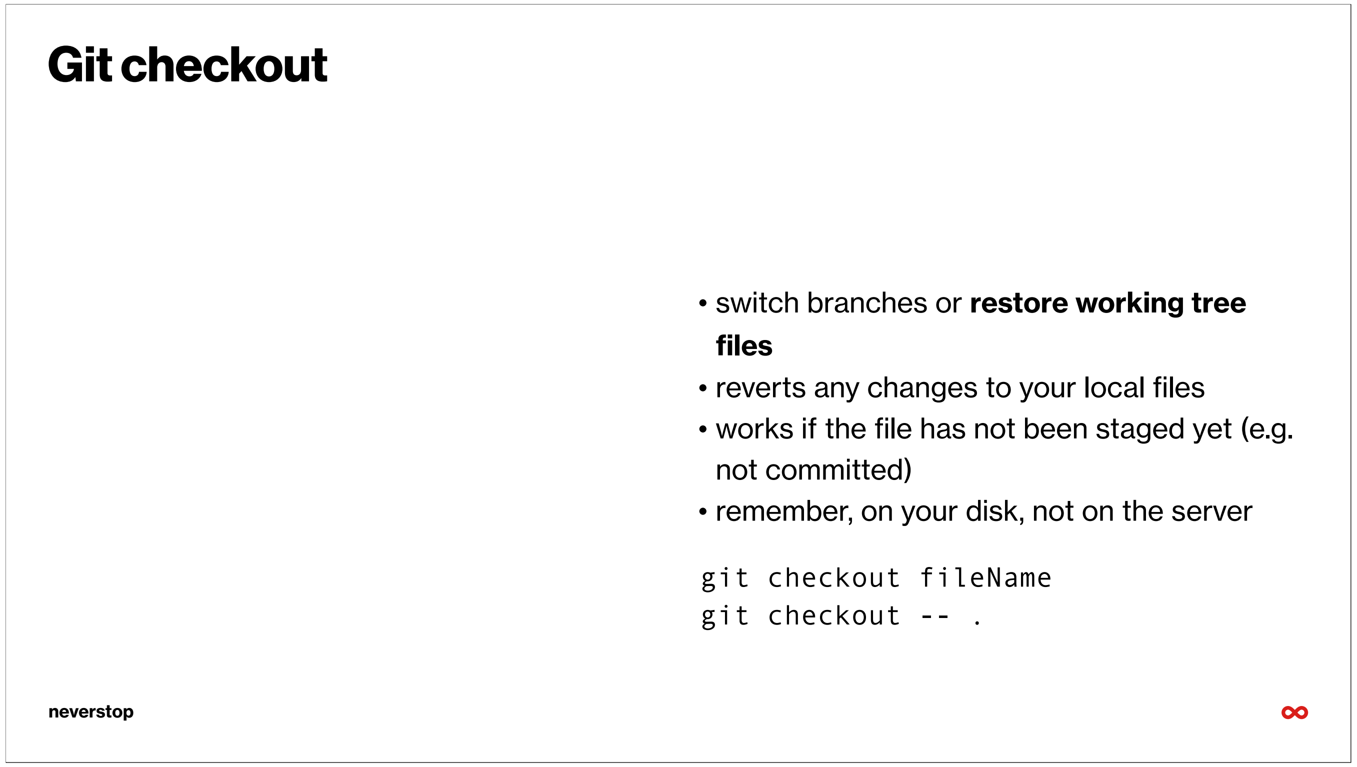
Only inherit parent task icon if it's not set ···
Tyriar and meganrogge committed 4 days ago ✗ — Verified — 229c927

Fix slow integration test (#153735) ···
roblourens committed 4 days ago ✓ — Verified — 50e1886

Use English as the default instead of navigator.language (#153733)
TylerLeonhardt committed 4 days ago ✓ — Verified — 3d9dd01

neverstop

- commits example
-

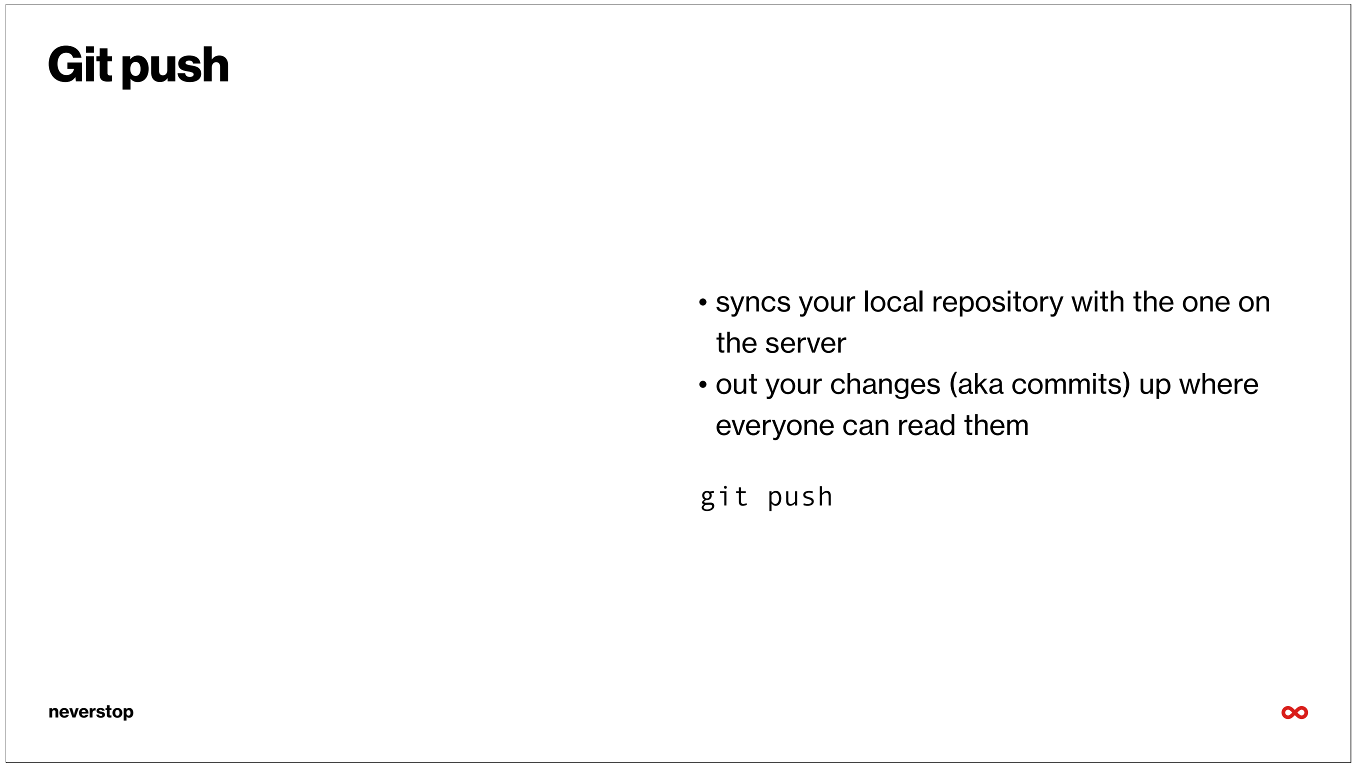


Git checkout

- switch branches or **restore working tree files**
- reverts any changes to your local files
- works if the file has not been staged yet (e.g. not committed)
- remember, on your disk, not on the server

```
git checkout fileName
git checkout -- .
```

- two uses, not so similar
    - one is for discarding changes in working directory, similar to closing the word document without saving the doc
    - other is for switching or creating a new branch



Git push

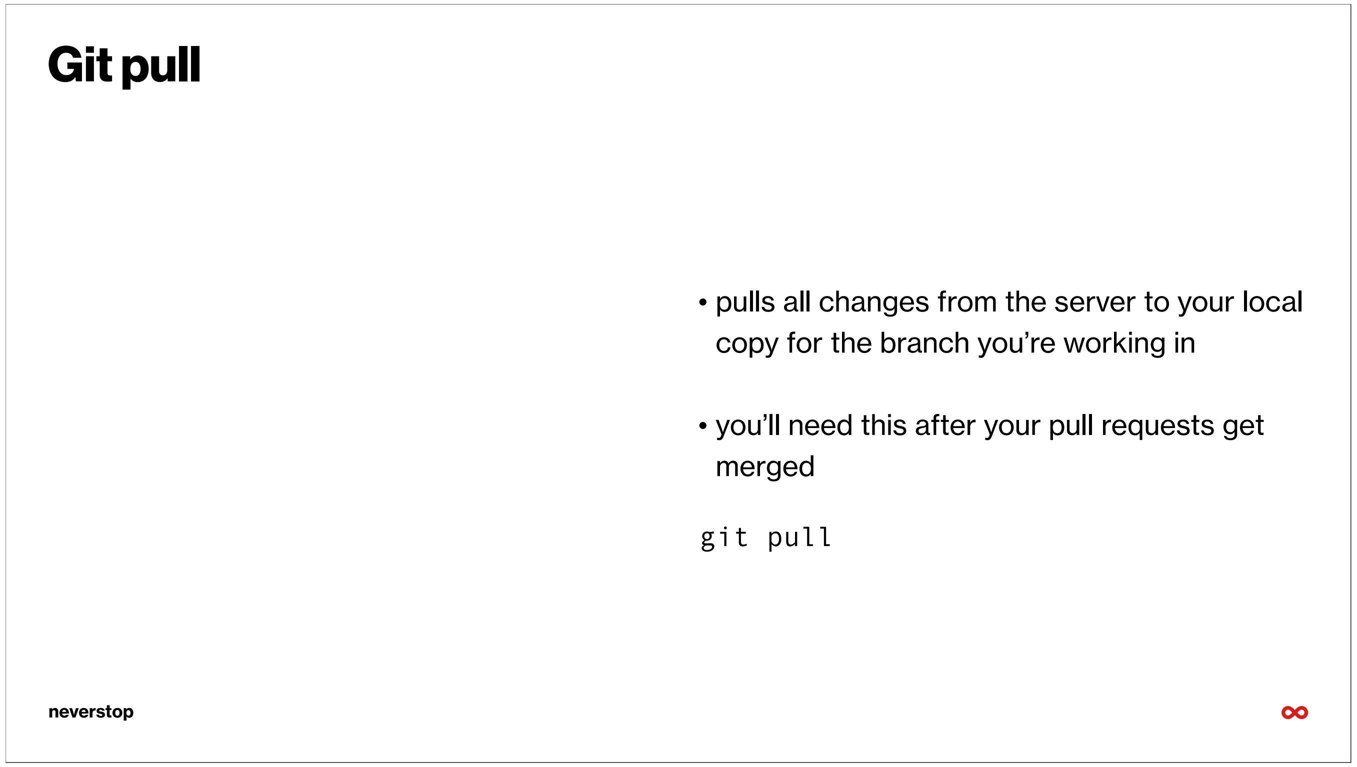
- syncs your local repository with the one on the server
- out your changes (aka commits) up where everyone can read them

```
git push
```

- changing history on the remote is not recommended
- also git will not let you do that, you will need to force push.

In case of fire

1. git commit
2. git push
3. leave building



Git pull

- pulls all changes from the server to your local copy for the branch you’re working in
- you’ll need this after your pull requests get merged

```
git pull
```

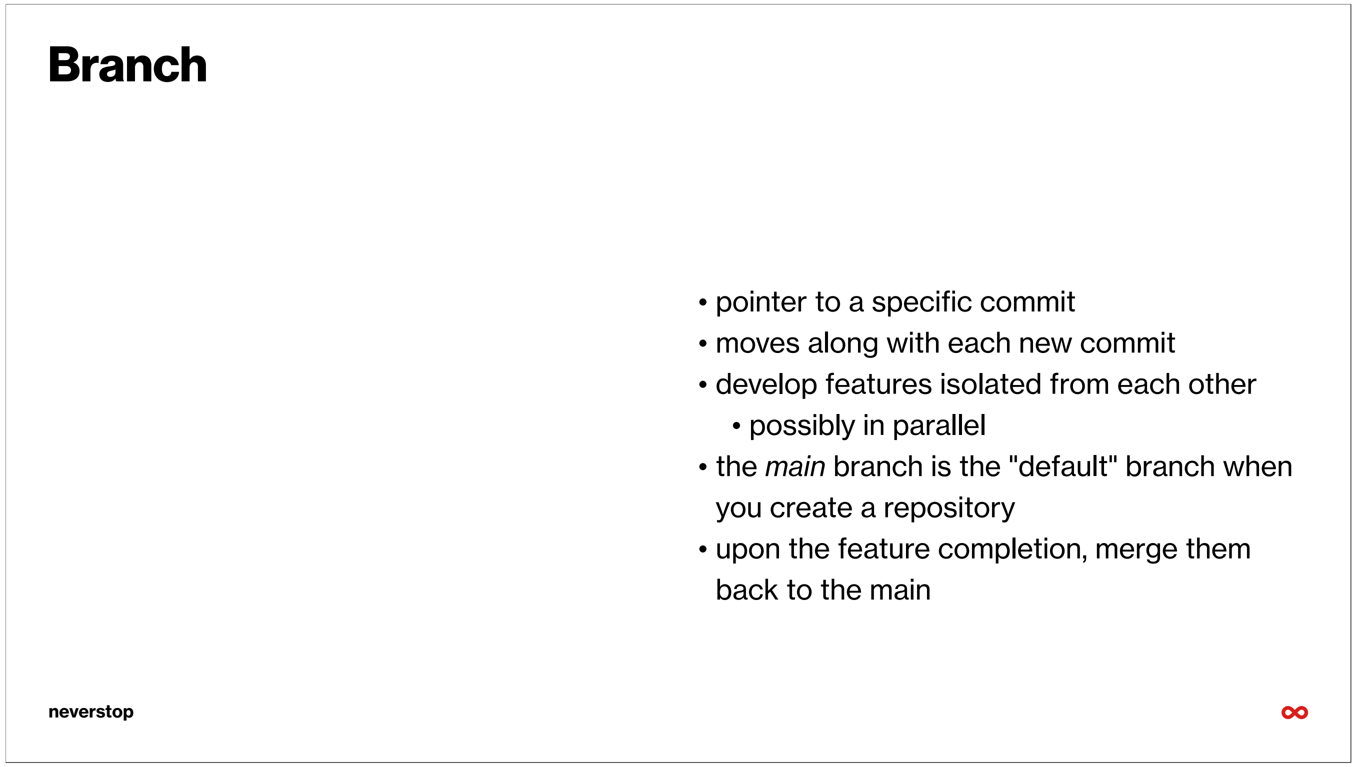
- we'll use it rarely
- mostly when fetching the main branch
- opposite to push

Further reading

- http://rogerdudler.github.io/git-guide/
- https://learngitbranching.js.org/ **<--- interactive tutorial**

03

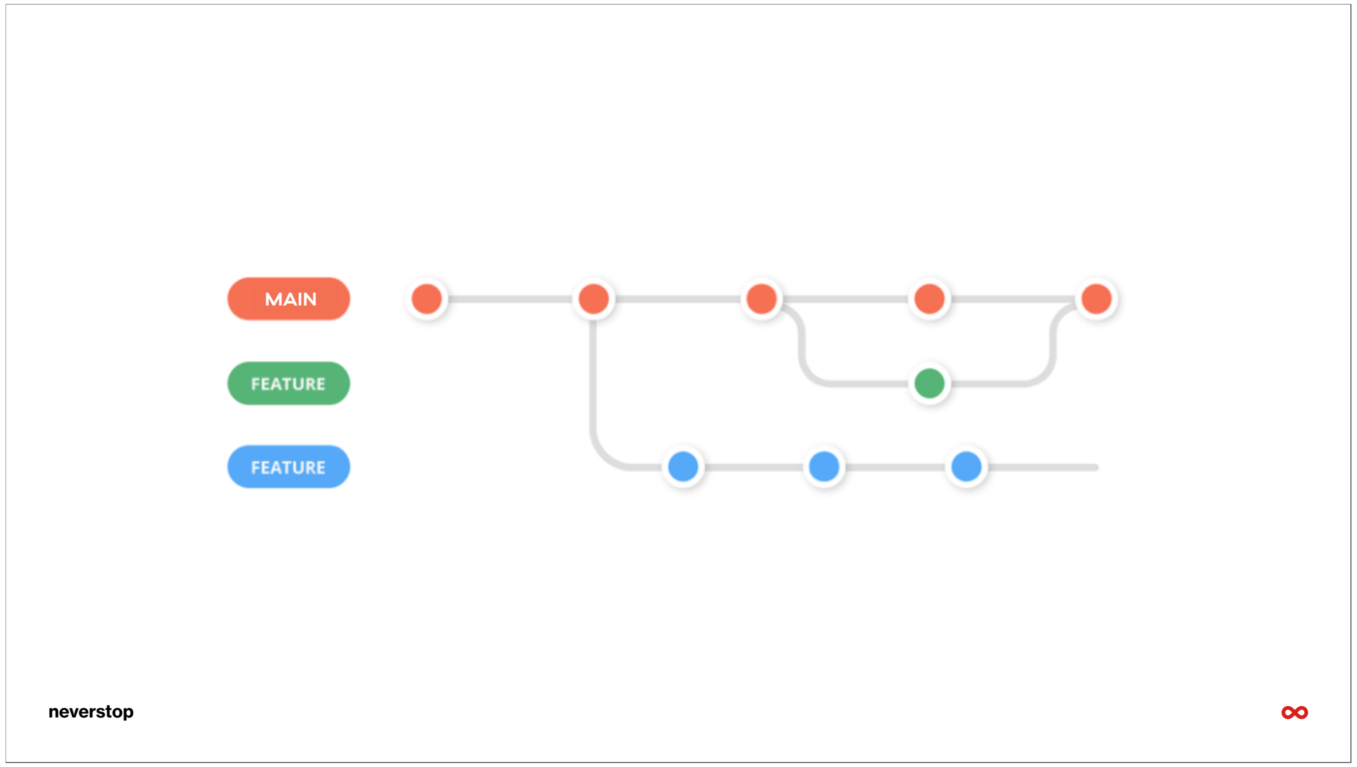
Branching



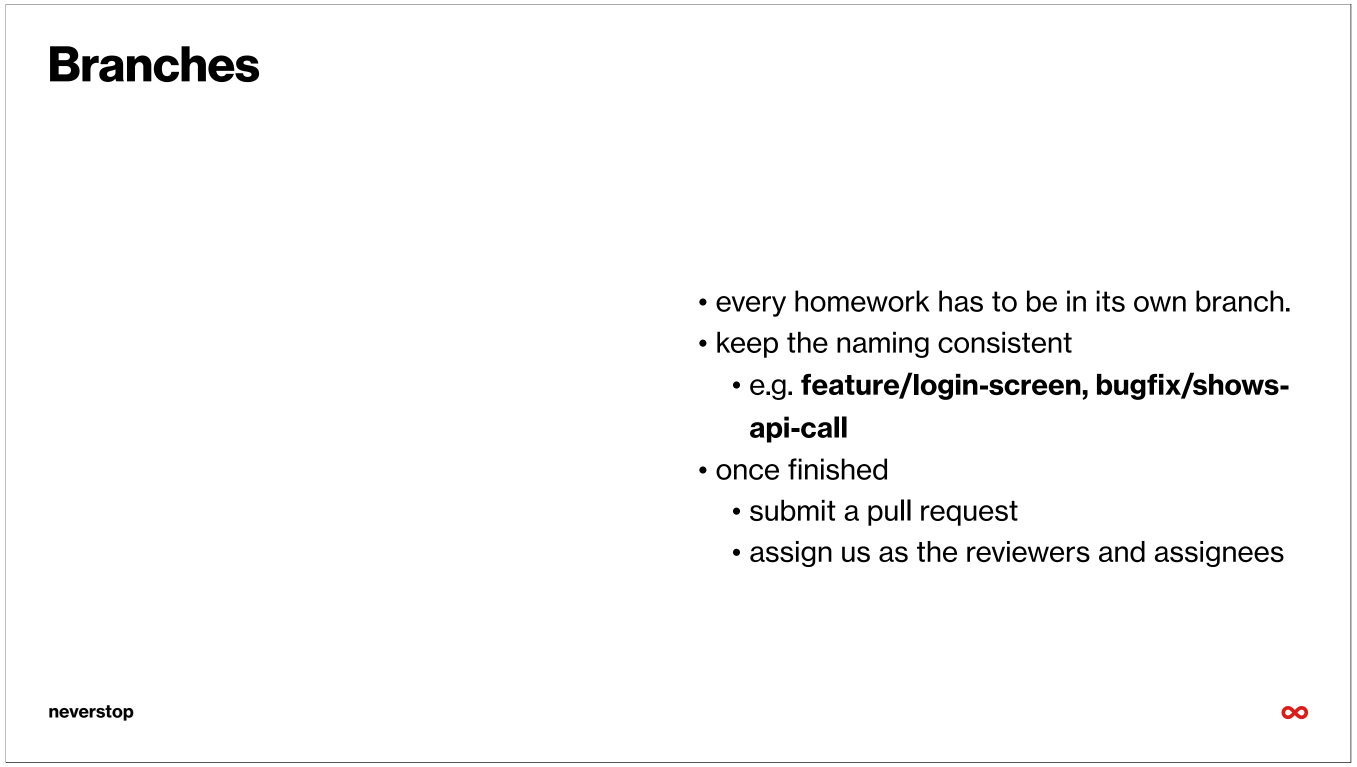
Branch

- pointer to a specific commit
- moves along with each new commit
- develop features isolated from each other
  - possibly in parallel
- the *main* branch is the "default" branch when you create a repository
- upon the feature completion, merge them back to the main

- git works with branches
- commit is the smallest node in git, while branch is named pointer to some commit
- branch pointer moves along with each new commit you make



- basic Git workflow, it doesn’t have to be like that
- main branch must be buildable
- from feature to main with pull request
- review

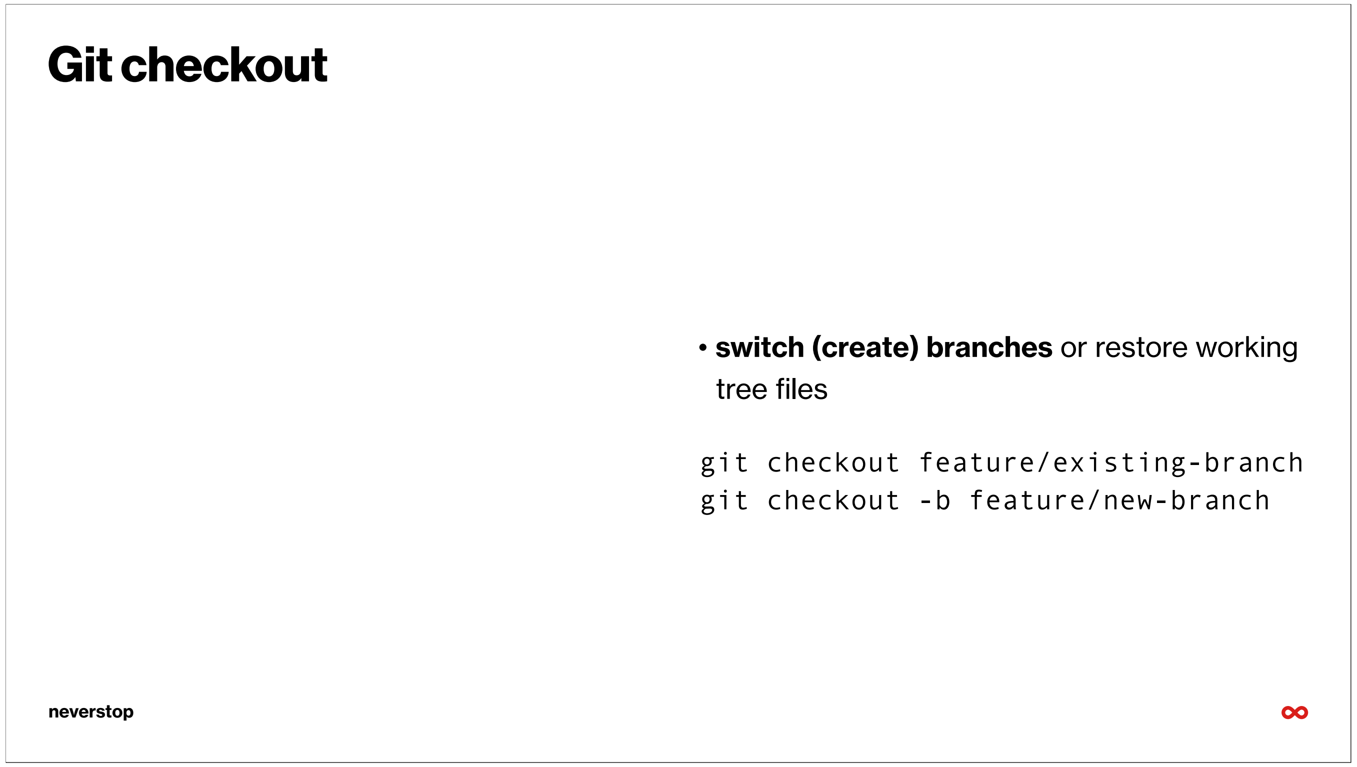


Branches

- every homework has to be in its own branch.
- keep the naming consistent
  - e.g. **feature/login-screen, bugfix/shows-api-call**
- once finished
  - submit a pull request
  - assign us as the reviewers and assignees

neverstop

- how we'll use branches during the course
- per each homework
- for example in case of bugfix but merged, open a new branch



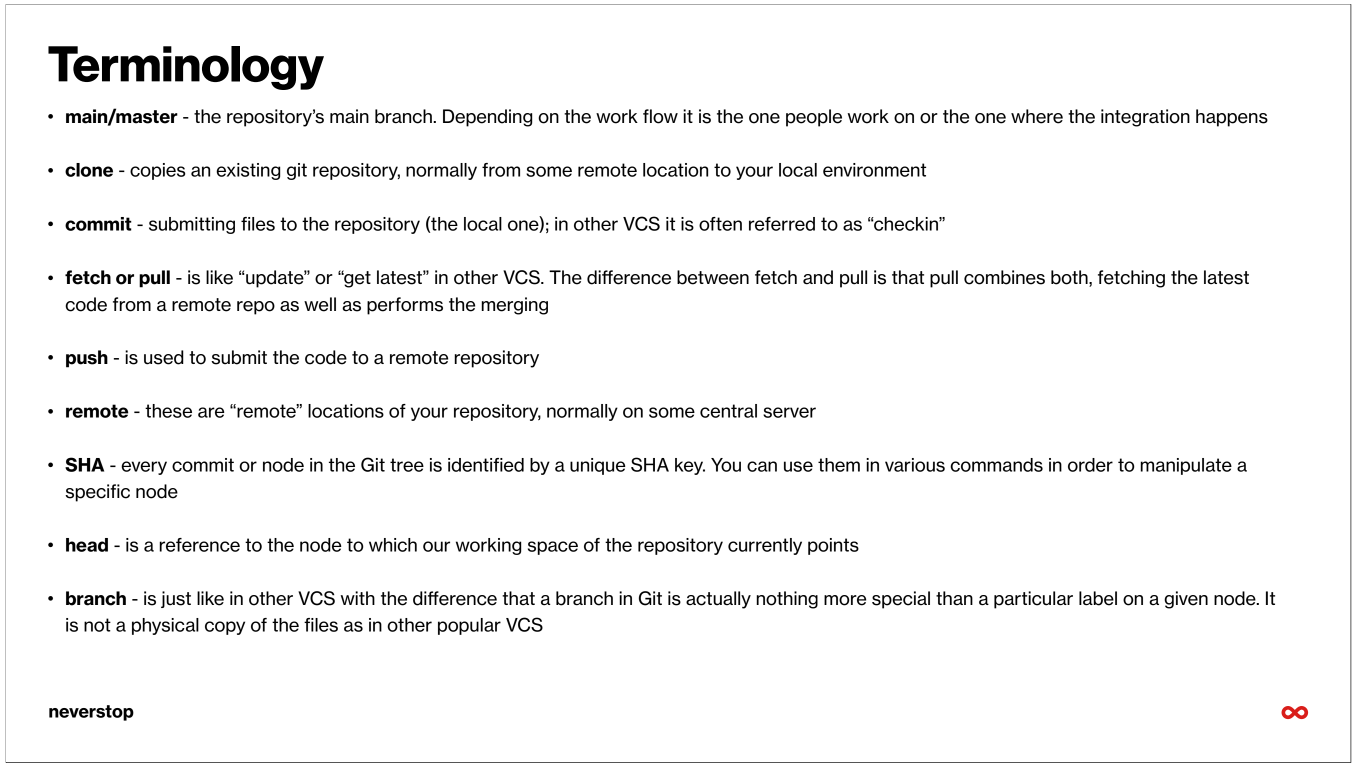
Git checkout

- **switch (create) branches** or restore working tree files

```
git checkout feature/existing-branch
git checkout -b feature/new-branch
```

neverstop

- another purpose of git checkout
- create branch, and switch branch



Terminology

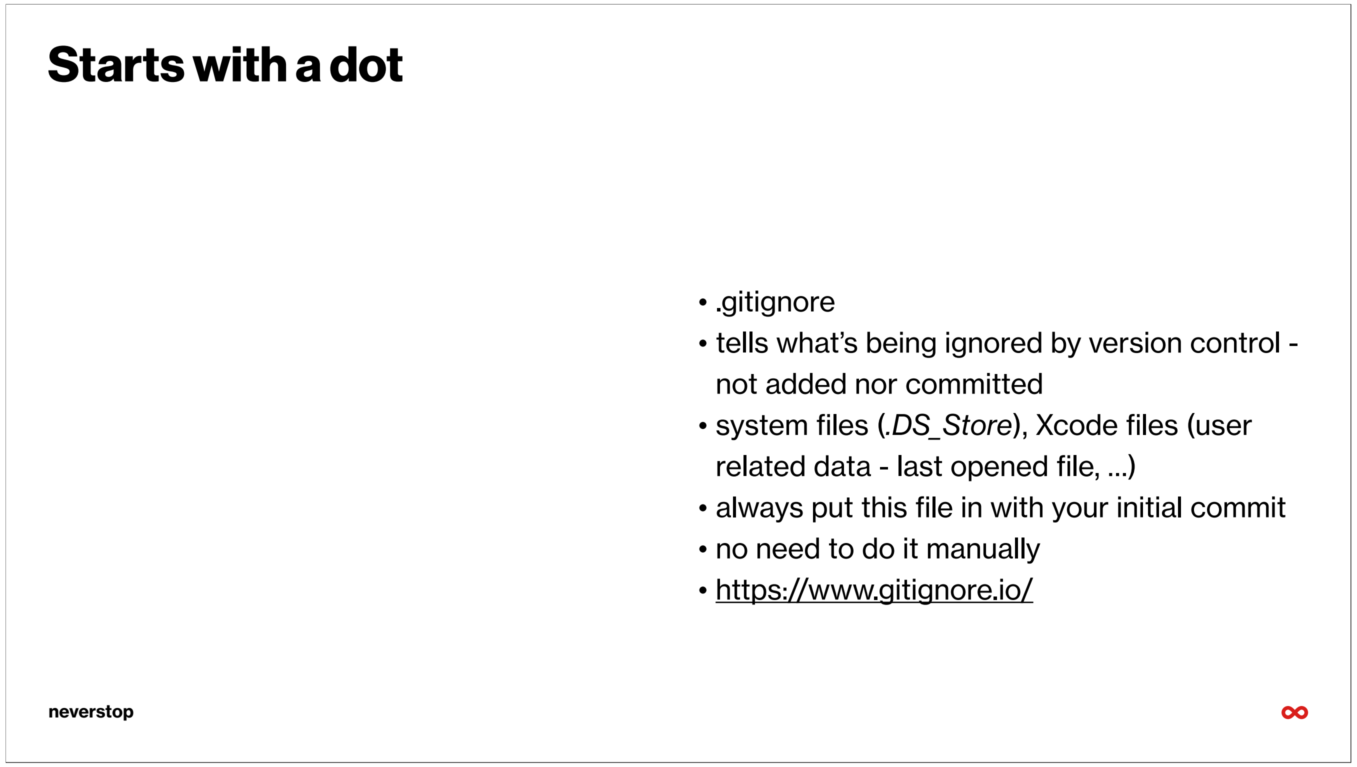
- **main/master** - the repository’s main branch. Depending on the work flow it is the one people work on or the one where the integration happens
- **clone** - copies an existing git repository, normally from some remote location to your local environment
- **commit** - submitting files to the repository (the local one); in other VCS it is often referred to as “checkin”
- **fetch or pull** - is like “update” or “get latest” in other VCS. The difference between fetch and pull is that pull combines both, fetching the latest code from a remote repo as well as performs the merging
- **push** - is used to submit the code to a remote repository
- **remote** - these are “remote” locations of your repository, normally on some central server
- **SHA** - every commit or node in the Git tree is identified by a unique SHA key. You can use them in various commands in order to manipulate a specific node
- **head** - is a reference to the node to which our working space of the repository currently points
- **branch** - is just like in other VCS with the difference that a branch in Git is actually nothing more special than a particular label on a given node. It is not a physical copy of the files as in other popular VCS

neverstop

- terminology
- every commit has has, you can navigate through history

04

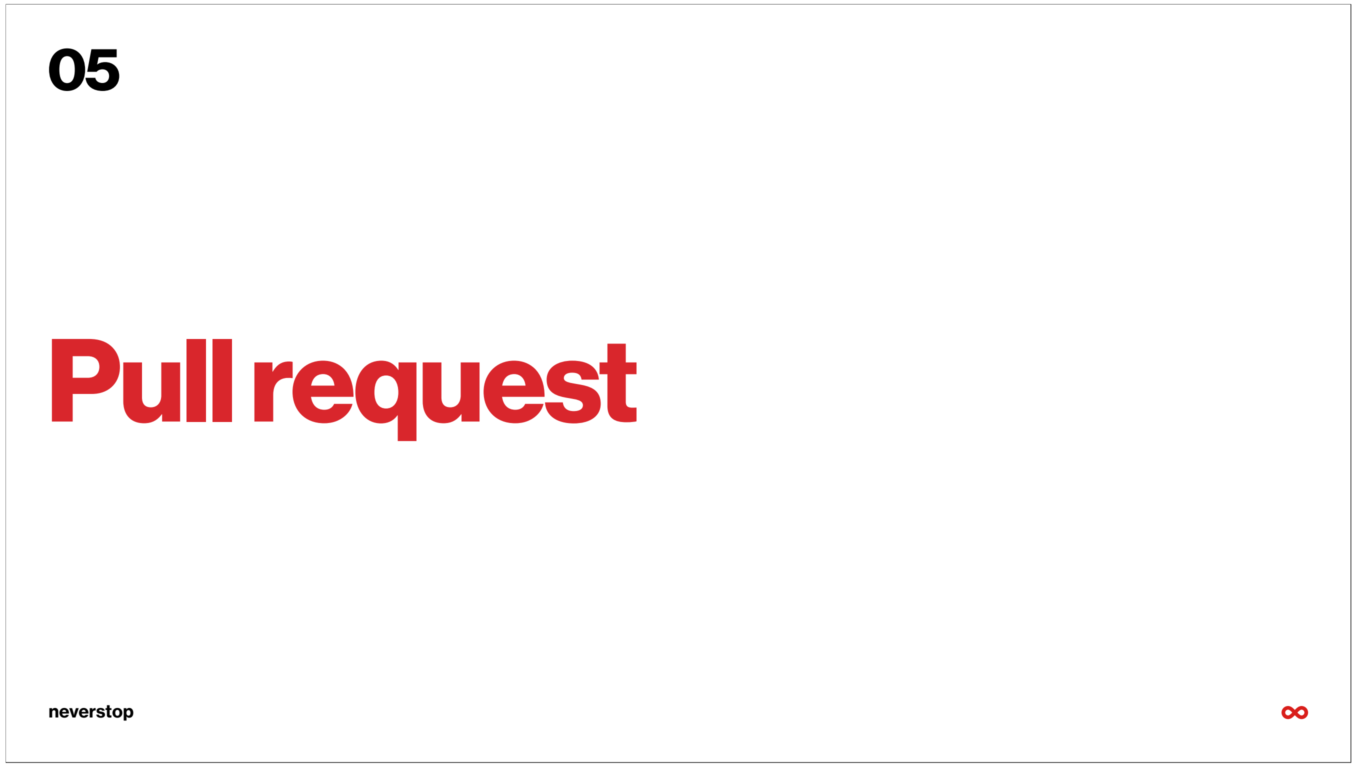
.gitignore



Starts with a dot

- .gitignore
- tells what’s being ignored by version control - not added nor committed
- system files (*.DS_Store*), Xcode files (user related data - last opened file, ...)
- always put this file in with your initial commit
- no need to do it manually
- https://www.gitignore.io/

- main purpose – tells which files to ignore
- important for binary files – conflict resolution not possible with git
- dot files – hidden on Unix based systems
- you usually don’t want to publish user defined settings, build artefacts, secrets…
- we'll create one on the next lecture
- how to decide what to ignore?



05

Pull request

neverstop

- PR a polite way to ask…
- Staple of open source
- example: you found an issue on some public repository
- example: in team, you want someone to review your code
- example: static checks, tests

A pull request

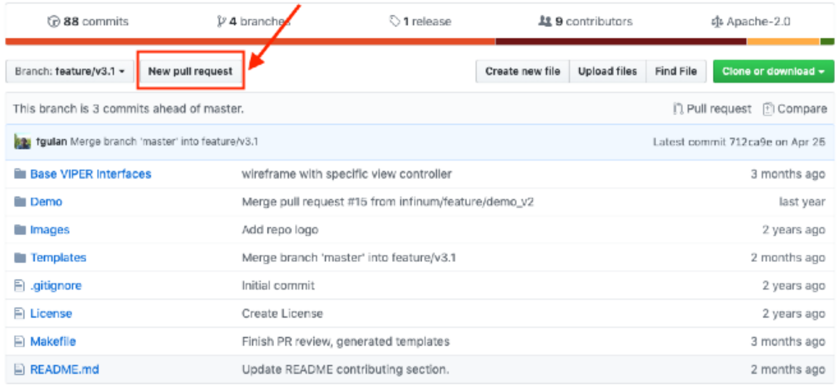
- a polite way to ask for your changes to be merged in to the main codebase
- **the staple of Open Source!**
- pull requests are easy to review
  - keep them clean
  - keep each homework in its own branch
- open up a pull request only, and only if your code is working

Pull request cookbook

For GitHub

neverstop

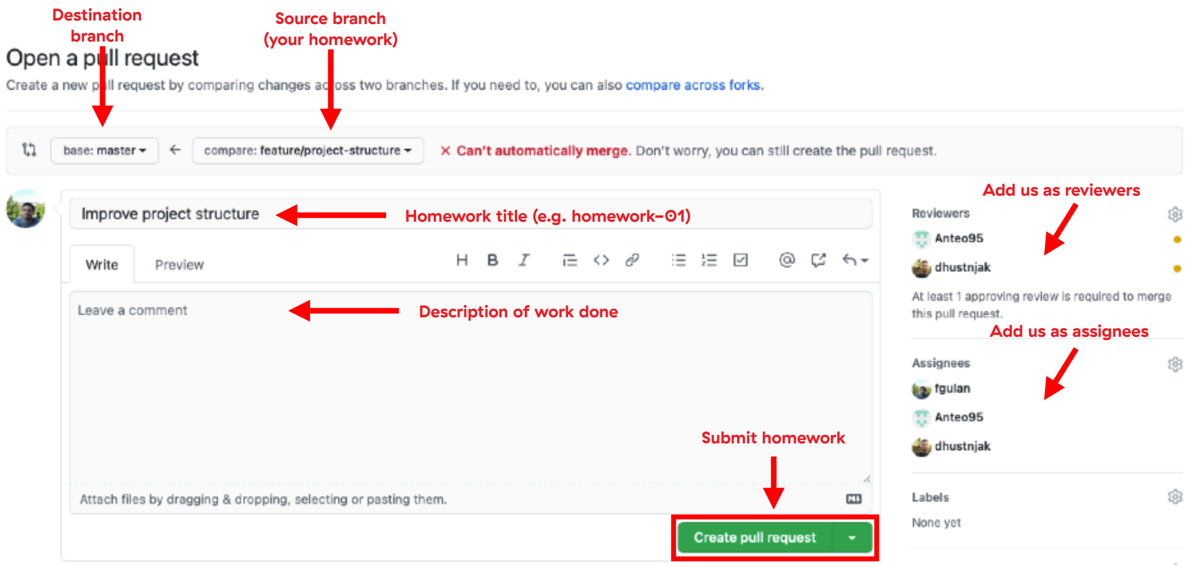
- we won't go over this today but you'll find it useful when opening the PR
-



This branch is 3 commits ahead of master.

fgulan Merge branch 'master' into feature/v3.1 Latest commit 712ca9e on Apr 25

Base VIPER Interfaces	wireframe with specific view controller	3 months ago
Demo	Merge pull request #15 from infinum/feature/demo_v2	last year
Images	Add repo logo	2 years ago
Templates	Merge branch 'master' into feature/v3.1	2 months ago
.gitignore	Initial commit	2 years ago
License	Create License	2 years ago
Makefile	Finish PR review, generated templates	3 months ago
README.md	Update README contributing section.	2 months ago



Destination branch

Source branch (your homework)

Homework title (e.g. homework-01)

Description of work done

Add us as reviewers

Add us as assignees

Submit homework

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ← compare: feature/project-structure

✕ Can't automatically merge. Don't worry, you can still create the pull request.

Improve project structure

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Reviewers

Anteo95 ●

dhustnjak ●

At least 1 approving review is required to merge this pull request.

Assignees

fgulan

Anteo95

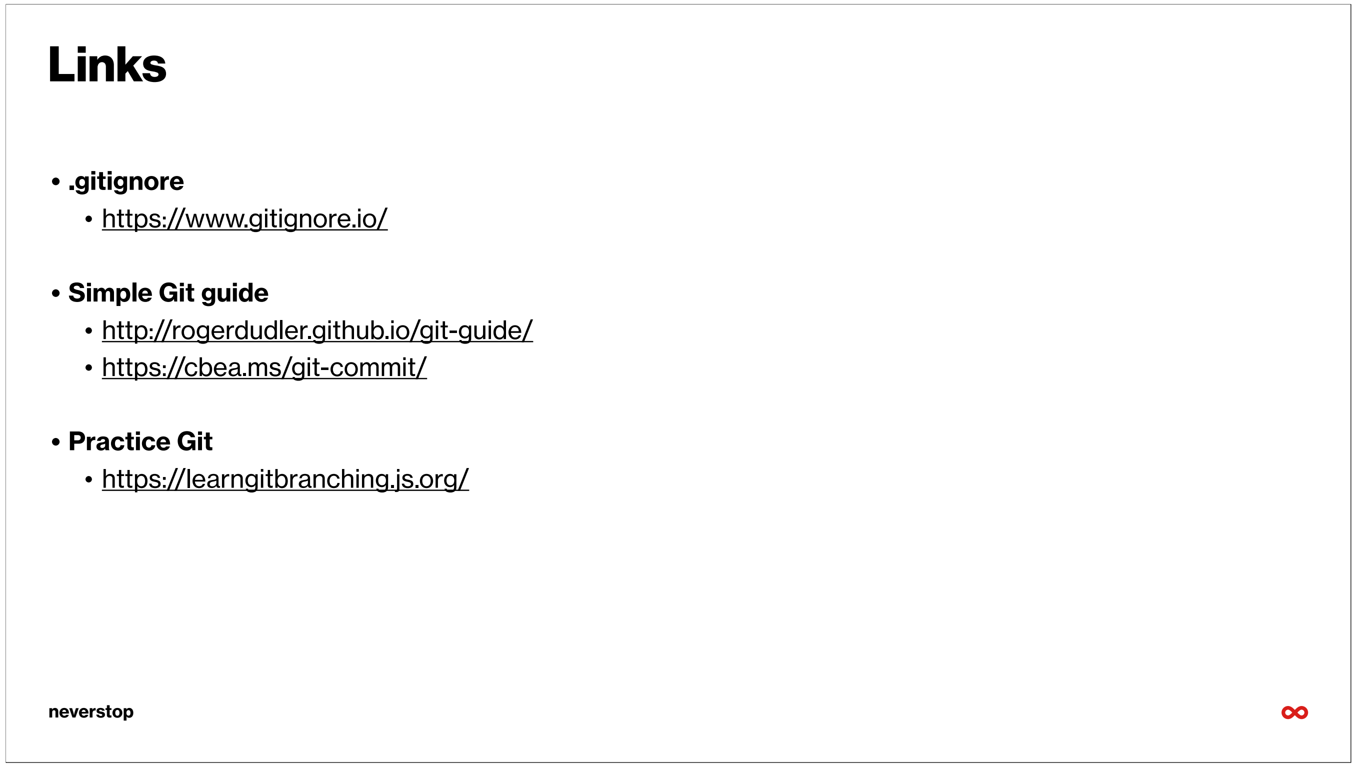
dhustnjak

Labels

None yet

06

Appendix



Links

- **.gitignore**
  - https://www.gitignore.io/

- **Simple Git guide**
  - http://rogerdudler.github.io/git-guide/
  - https://cbea.ms/git-commit/

- **Practice Git**
  - https://learngitbranching.js.org/

- if you don't have much experience with Git:

Thanks!