

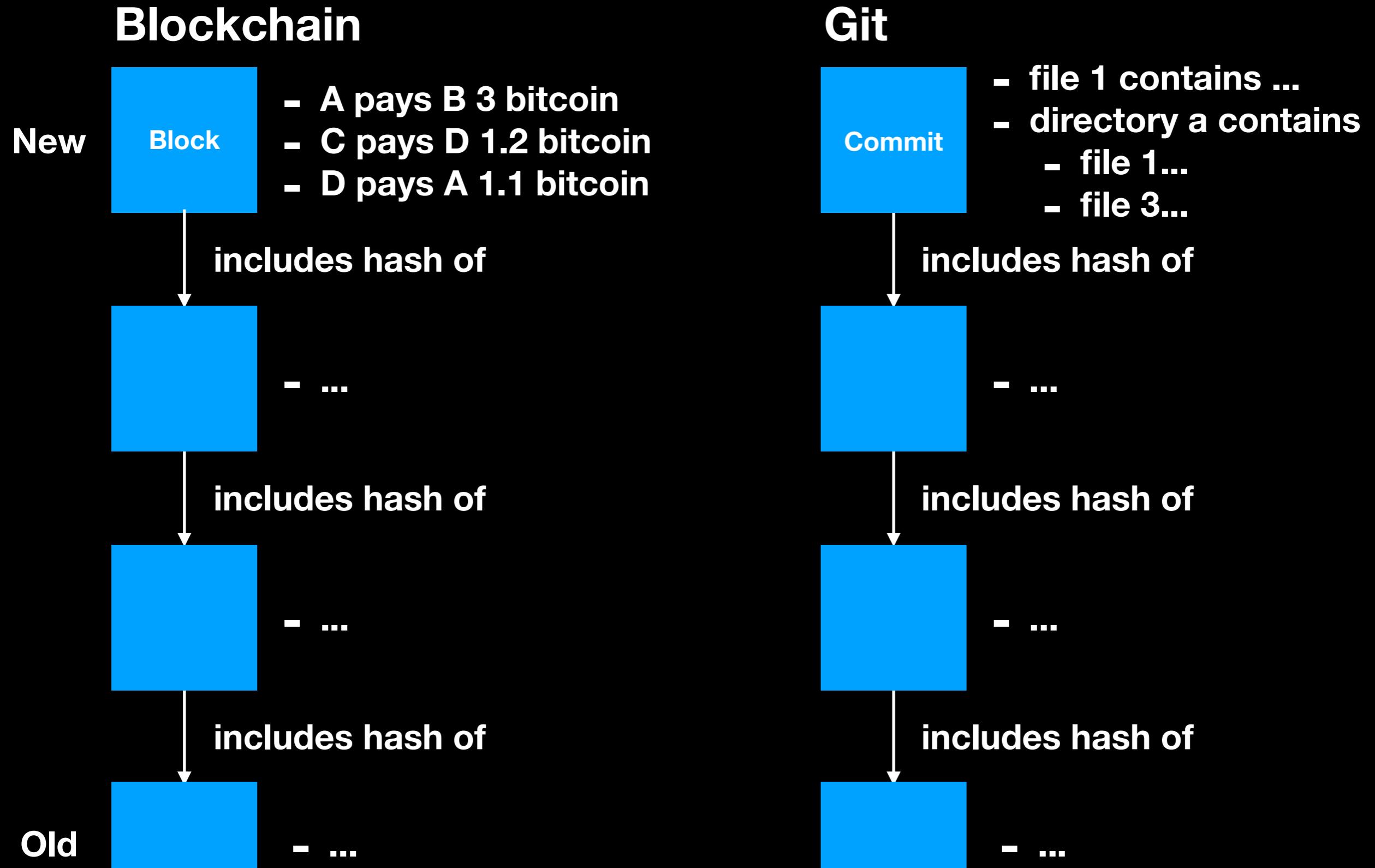
# how to git gut at git

Jelle Besseling





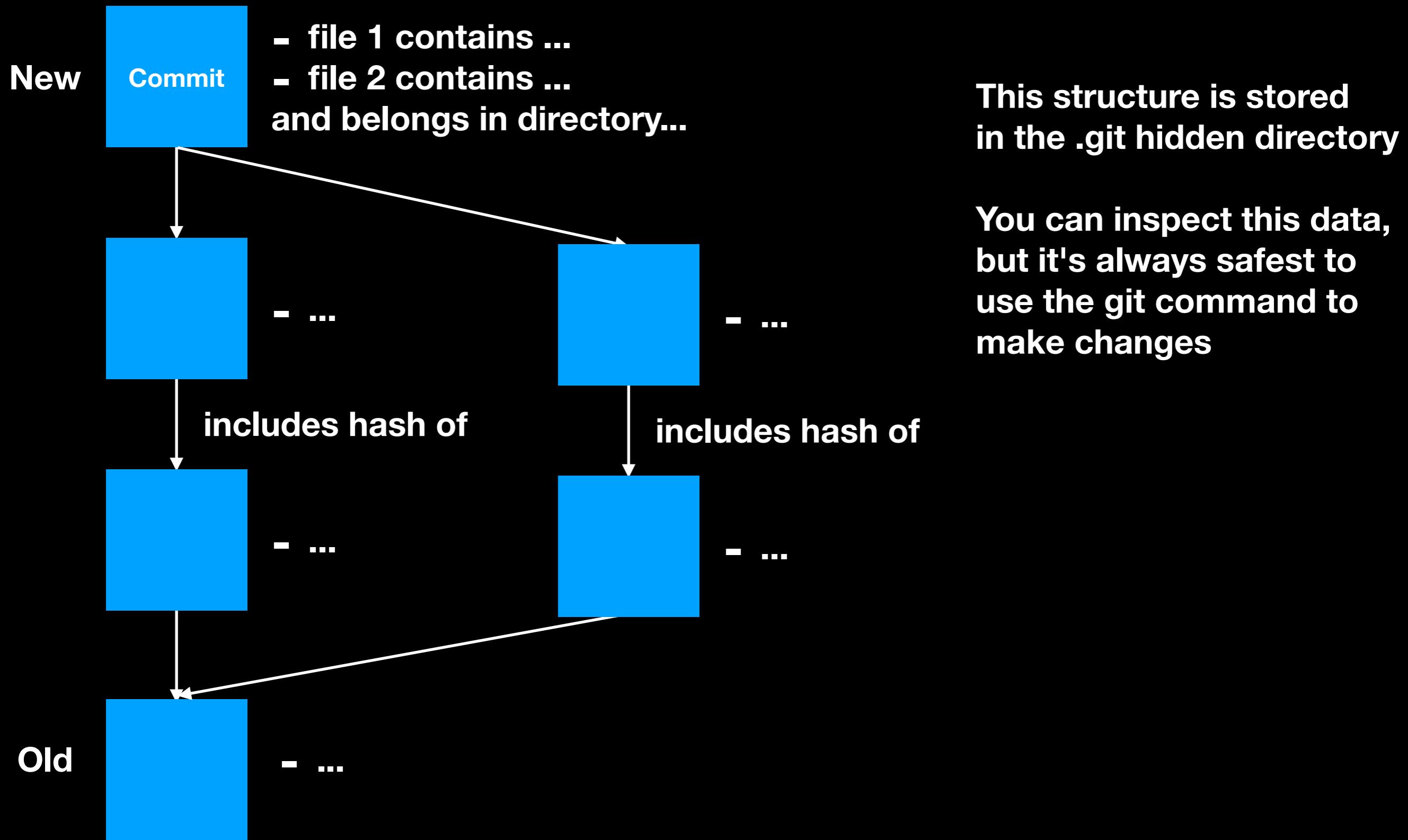
# git is a blockchain?



# Actually, git is a VCS

- Version Control System
- Meant for text files (including code)
- Allows you to look back in history
- Easy collaboration (with github/gitlab)

## Merge changes from another branch:



# Go back in time

```
2. ~/src/concrexit (less)
*   6a86c28a (HEAD -> master, origin/master, origin/HEAD) Merge branch 'replace-links' into 'master'
|\ \
| * bf2078c0 Replace Google Drive with Nextcloud links in styleguide
|/
*   bb84dc09 (origin/777-automatic-notification-after-uploading-a-new-photo-album) Merge branch 'fix/registrations-api-without-right-permissions' into 'master'
|\ \
| * 6df7ad37 Add permissions to admin views
| * 1a93780a Restrict changing registration to users with change registration perms
* |   5ef05612 Merge branch 'fix/merchandise-thumbnails' into 'master'
|\ \
| * | d6b3b1fc Thumbnail the merchandise images
* | |   3579225e Merge branch 'tc/wiki-nextcloud-menu' into 'master'
|\ \
| * | | 16fb206e Replace the wiki menu item by Nextcloud
| | |
* | | | 4e40f151 Merge branch 'fix/members-tabs' into 'master'
|\ \
| * | | 1abf5bee Fix member tabs active state
* | | |   f6d512dd Merge branch 'tc/better-createfixtures' into 'master'
|\ \
| * | | | fc735536 Add createfixtures to README
| * | | | ccbbe9f7 Improve createfixtures
|\ \
* | | | | 8a48fe8f Merge branch '416-board-membership-cannot-have-end-dates' into 'master'
|\ \
| * | | | 9dd5de2b Allow board membership to contain end date
|\ \
* | | | | 53ff95d4 Merge branch 'use-poetry' into 'master'
|\ \
* | | | | 5ff00200 (origin/use-poetry, use-poetry) Migrate to poetry
```

```
2. ~/src/myrepo (zsh)
~/src λ mkdir myrepo
                           21:24:31
~/src λ cd myrepo
                           21:24:34
src/myrepo λ git init
                           21:24:36
Initialized empty Git repository in /Users/jelle/src/myrepo/.git/
src/myrepo λ echo "My first commit" > hello
                           21:24:39
src/myrepo λ ls
                           21:24:50
hello
src/myrepo λ git add hello
                           21:24:51
src/myrepo λ git commit -m "Initial commit"
                           21:25:01
[master (root-commit) 1b87a92] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello
src/myrepo (master) λ
```

```
src/myrepo (master)λ git log 21:25:48
commit 1b87a921faf4acfe93dfd7a63598f45d9ce4e3ed (HEAD -> master)
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:25:08 2019 +0100

    Initial commit
src/myrepo (master)λ 21:25:49
```

```
2. ~/src/myrepo (zsh)          21:36:54
src/myrepo (master)λ git log
commit 6e413ce2a83544509cc8fb99ce363c84e6915437 (HEAD -> master)
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:36:42 2019 +0100

    Change website URL

commit 003e7a060a2d391473f58e94757e674230dc6b83 ←
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:34:21 2019 +0100

    Add python source code

    Add a functin that prints the URL to a nice website

commit 1b87a921faf4acfe93dfd7a63598f45d9ce4e3ed
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:25:08 2019 +0100

    Initial commit
src/myrepo (master)λ 21:37:01
```

```
2. ~/src/myrepo (zsh)          21:37:34
src/myrepo (master)λ git diff 003e7a060a2d391473f58e94757e674230dc6b83
diff --git a/website.py b/website.py
index 2f92e6d..f5301f5 100644
--- a/website.py
+++ b/website.py
@@ -1,4 +1,4 @@
#!/usr/bin/env python3

def nicesite():
-   print("https://thalia.nu")
+   print("https://technicie.nl")
src/myrepo (master)λ 21:37:38
```

# You can check out old versions

```
2. ~/src/myrepo (zsh)
+++ b/website.py
@@ -1,4 +1,4 @@
#!/usr/bin/env python3

def nicesite():
-   print("https://thalia.nu")
+   print("https://technicie.nl")
src/myrepo (master)λ git checkout 003e7a060a2d391473f58e94757e674230dc6b83          21:37:38
Note: checking out '003e7a060a2d391473f58e94757e674230dc6b83'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD is now at 003e7a0 Add python source code
src/myrepo (HEAD)λ cat website.py                                         20:51:45
#!/usr/bin/env python3

def nicesite():
    print("https://thalia.nu")
src/myrepo (HEAD)λ 
```

```
src/myrepo (HEAD)λ git log master 20:53:37
commit 6e413ce2a83544509cc8fb99ce363c84e6915437 (master)
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:36:42 2019 +0100

    Change website URL

commit 003e7a060a2d391473f58e94757e674230dc6b83 (HEAD)
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:34:21 2019 +0100

    Add python source code

    Add a functin that prints the URL to a nice website

commit 1b87a921faf4acfe93dfd7a63598f45d9ce4e3ed
Author: Jelle Besseling <jelle@pinguin.com>
Date:   Fri Mar 8 21:25:08 2019 +0100

    Initial commit
src/myrepo (HEAD)λ 20:53:53
```

```
src/myrepo (HEAD)λ git status 20:53:53
HEAD detached at 003e7a0
nothing to commit, working tree clean
src/myrepo (HEAD)λ 20:55:22
```

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ touch Hey                                20:56:03
src/myrepo (*master)λ git status                             20:56:07
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Hey

nothing added to commit but untracked files present (use "git add" to track)
src/myrepo (*master)λ █
```

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ rm website.py                            20:57:31
src/myrepo (master)λ git status                             20:57:38
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:    website.py

no changes added to commit (use "git add" and/or "git commit -a")
src/myrepo (master)λ git checkout -- website.py               20:57:44
src/myrepo (master)λ █
```

# Two changes, but commit only one

```
src/tutorial (master)λ git diff
diff --git a/README.md b/README.md
index e45e413..2247df3 100644
--- a/README.md
+++ b/README.md
@@ -2,6 +2,8 @@
 
 Gitlab repo for the git tutorial
+Add something to the first part
+
 You can clone this repo with:
 
 ~~~bash
@@ -24,3 +26,4 @@ For example:
 man git-revert
 ~~~
 
+Add something to the last part
src/tutorial (master)λ
```

```
5. ~/src/tutorial (zsh)          13:59:19
src/tutorial (master)λ git add -p
diff --git a/README.md b/README.md
index e45e413..2247df3 100644
--- a/README.md
+++ b/README.md
@@ -2,6 +2,8 @@
Gitlab repo for the git tutorial

+Add something to the first part
+
You can clone this repo with:

```bash
Stage this hunk [y,n,q,a,d,j,J,g/,e,?]? y ←
@@ -24,3 +26,4 @@ For example:
man git-revert
```

+Add something to the last part
Stage this hunk [y,n,q,a,d,K,g/,e,?]? n ←
src/tutorial (master)λ git status          14:00:06
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
(use "git reset HEAD <file>..." to unstage)

    modified:   README.md

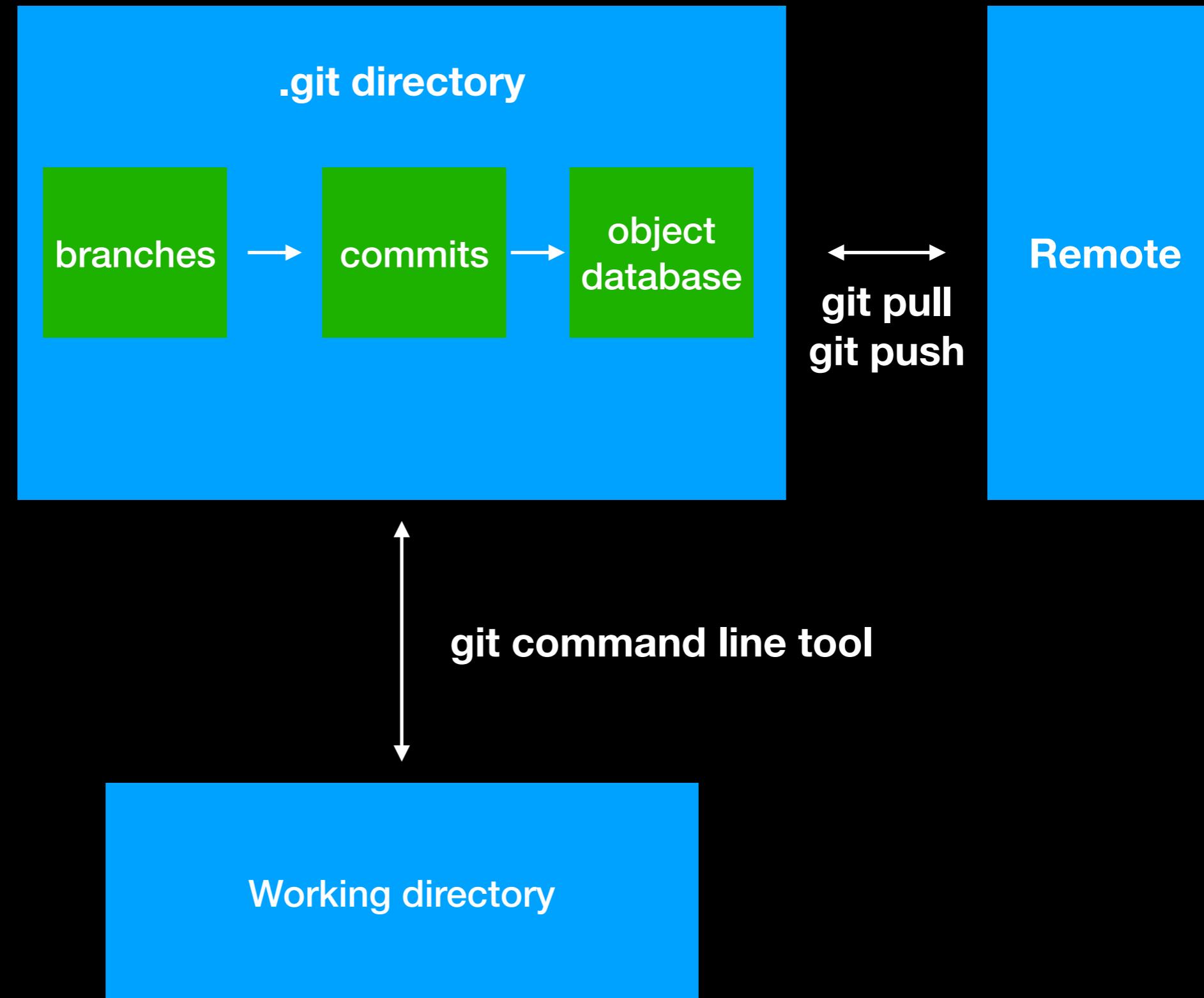
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README.md

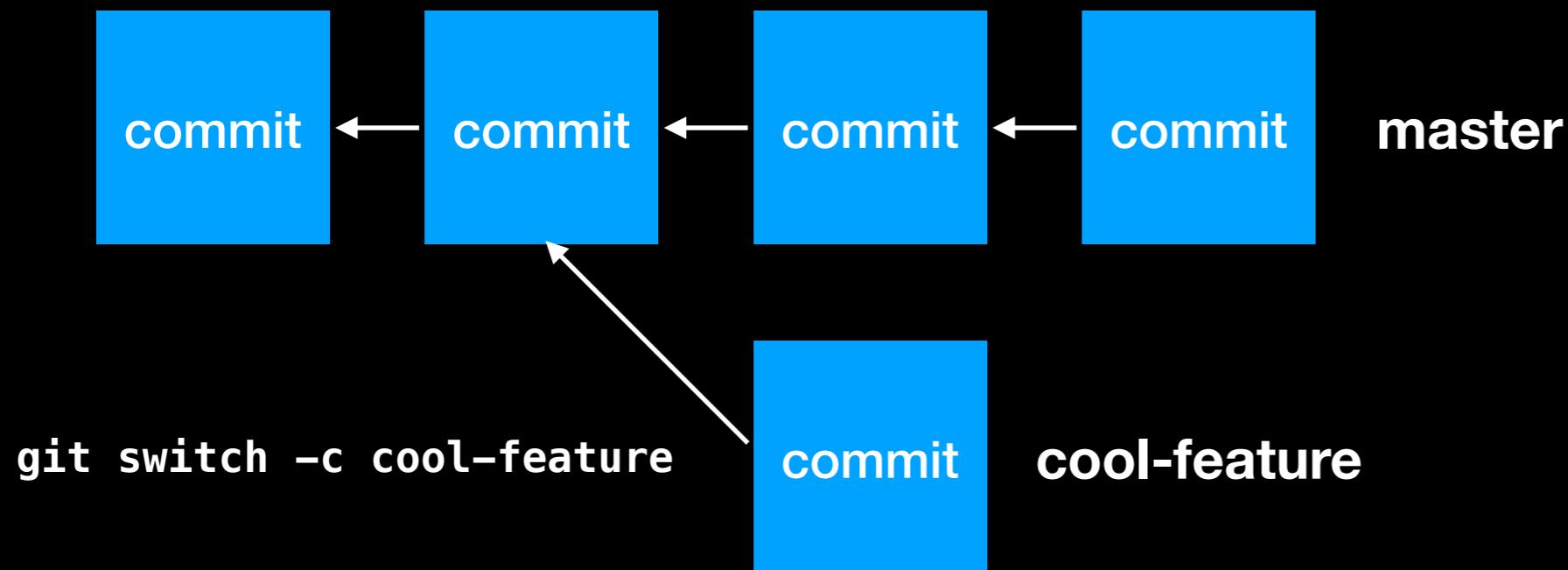
src/tutorial (master)λ
```

# Good commit message style

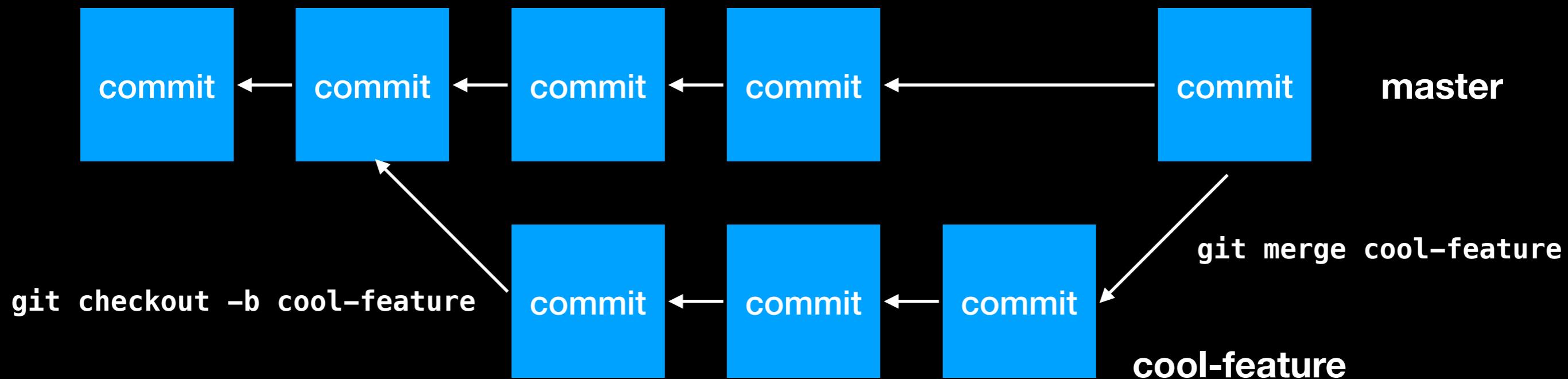
- Use imperative style, so:
- Fix `send_file` function to handle 404 errors
- Not:
- Fixes `send_file` function to handle 404 errors
- Or:
- Will fix `send_file` function to handle 404 errors



# Working with branches



# Working with branches



# Add commit to a new branch

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ git checkout -b add-docstring          10:24:5
4
Switched to a new branch 'add-docstring'
src/myrepo (add-docstring)λ vim website.py                10:25:0
1
src/myrepo (add-docstring)λ git diff                      10:25:1
9
diff --git a/website.py b/website.py
index f5301f5..47175de 100644
--- a/website.py
+++ b/website.py
@@ -1,4 +1,5 @@
#!/usr/bin/env python3

def nicesite():
+     """Print the url to a nice website"""
        print("https://technicie.nl")
src/myrepo (add-docstring)λ git commit -am "Add docstring to nicesite function"
"
[add-docstring d4e56f2] Add docstring to nicesite function
 1 file changed, 1 insertion(+)
src/myrepo (add-docstring)λ 10:26:20
```

# Meanwhile, master is edited

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ git diff
diff --git a/website.py b/website.py
index f5301f5..f858238 100644
--- a/website.py
+++ b/website.py
@@ -1,4 +1,5 @@
#!/usr/bin/env python3

-def nicesite():
+def nicesites():
    print("https://technicie.nl")
+   print("https://thalia.nu")
src/myrepo (master)λ git commit -am 'Print two nice sites!'
[master 6bcd739] Print two nice sites!
 1 file changed, 2 insertions(+), 1 deletion(-)
src/myrepo (master)λ 10:33:13
```

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ git log --graph --oneline --all
* 56a31a9 (HEAD -> master) Print two nice sites!
| * d4e56f2 (add-docstring) Add docstring to nicesite function
|/
* 6e413ce Change website URL
* 003e7a0 Add python source code
* 1b87a92 Initial commit
src/myrepo (master)λ 10:29:48
```

# What happens if we merge add-docstring now?

```
2. ~/src/myrepo (zsh)
src/myrepo (master)λ git merge add-docstring
Auto-merging website.py
CONFLICT (content): Merge conflict in website.py
Automatic merge failed; fix conflicts and then commit the result.
src/myrepo (master)λ 10:33:38
```

```
2. ~/src/myrepo (vim)
#!/usr/bin/env python3

<<<<< HEAD
def nicesites():
=====
def nicesite():
    """Print the url to a nice website"""
>>>>> add-docstring
    print("https://technicie.nl")
    print("https://thalia.nu")
~
~
~
~
~
~
"website.py" 10L, 207C
```

2. ~/src/myrepo (vim) 🎙

```
#!/usr/bin/env python3

def nicesites():
    """Print two urls to nice websites"""
    print("https://technicie.nl")
    print("https://thalia.nu")
~
```

2. ~/src/myrepo (zsh)

```
src/myrepo (master) λ git merge add-docstring 10:33:30
Auto-merging website.py
CONFLICT (content): Merge conflict in website.py
Automatic merge failed; fix conflicts and then commit the result.
src/myrepo (master) λ vim website.py 10:34:22
src/myrepo (master) λ git status 10:36:09
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

    both modified:  website.py

no changes added to commit (use "git add" and/or "git commit -a")
src/myrepo (master) λ git add website.py 10:36:16
src/myrepo (master) λ git commit 10:36:44
```

2. ~/src/myrepo (vim)

```
Merge branch 'add-docstring'

# Conflicts:
#       website.py
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#       .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# All conflicts fixed but you are still merging.
#
# Changes to be committed:
"~/src/myrepo/.git/COMMIT_EDITMSG" 20L, 452C
```

2. ~/src/myrepo (zsh)

```
src/myrepo (master)λ git add website.py 10:36:16
src/myrepo (master)λ git commit 10:36:44
[master 00e41c6] Merge branch 'add-docstring'
src/myrepo (master)λ git log --graph --oneline --all 10:37:17
*   00e41c6 (HEAD -> master) Merge branch 'add-docstring'
|\ 
| * d4e56f2 (add-docstring) Add docstring to nicesite function
* | 6bcd739 Print two nice sites!
|/
* 6e413ce Change website URL
* 003e7a0 Add python source code
* 1b87a92 Initial commit
src/myrepo (master)λ cat website.py 10:37:24
#!/usr/bin/env python3

def nicesites():
    """Print two urls to nice websites"""
    print("https://technicie.nl")
    print("https://thalia.nu")
src/myrepo (master)λ 10:38:08
```

# Upstream repo

- In git these are called "remotes"
- Github/Gitlab or just a server you can reach with ssh
- You can "clone" a repository locally, work on it, then commit changes back to upstream
- [gitlab.science.ru.nl](https://gitlab.science.ru.nl)

# Clone upstream repo



4. ~/src/tutorial (zsh)

```
src/tutorial (master)λ git checkout -b new-feature                                         11:15:32
Switched to a new branch 'new-feature'
src/tutorial (new-feature)λ echo "This is code for my new feature" > feature.py          11:15:35
src/tutorial (*new-feature)λ git add feature.py                                         11:15:51
src/tutorial (new-feature)λ git commit -m "Add feature"                                11:15:56
[new-feature a24960f] Add feature
 1 file changed, 1 insertion(+)
  create mode 100644 feature.py
src/tutorial (new-feature)λ git push --set-upstream origin new-feature                  11:16:15
Username for 'https://gitlab.science.ru.nl': ^C
src/tutorial (new-feature)λ git push --set-upstream origin new-feature                  11:16:27
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1008 bytes | 1008.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for new-feature, visit:
remote:   https://gitlab.science.ru.nl/jbesseling/tutorial/merge_requests/new?merge_request%5Bsource_branch%5D=new-feature
remote:
To gitlab.science.ru.nl:jbesseling/tutorial.git
 * [new branch]      new-feature -> new-feature
Branch 'new-feature' set up to track remote branch 'new-feature' from 'origin'.
src/tutorial (new-feature)λ echo "extend new feature" >> feature.py                11:17:03
src/tutorial (new-feature)λ git add feature.py                                         11:17:15
src/tutorial (new-feature)λ git commit -m "Extend new feature"                         11:17:19
[new-feature bf06152] Extend new feature
 1 file changed, 1 insertion(+)
src/tutorial (new-feature)λ git push                                              11:17:24
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1018 bytes | 1018.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for new-feature, visit:
remote:   https://gitlab.science.ru.nl/jbesseling/tutorial/merge_requests/new?merge_request%5Bsource_branch%5D=new-feature
remote:
To gitlab.science.ru.nl:jbesseling/tutorial.git
 a24960f..bf06152  new-feature -> new-feature
src/tutorial (new-feature)λ
```

# Someone else changed your branch? Use git pull

```
4. ~/src/tutorial (zsh)          11:17:31
src/tutorial (new-feature)λ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitlab.science.ru.nl:jbesseling/tutorial
  bf06152..1f3742c  new-feature -> origin/new-feature
Updating bf06152..1f3742c
Fast-forward
 feature.py | 1 +
 1 file changed, 1 insertion(+)
src/tutorial (new-feature)λ git diff HEAD^          11:20:03
diff --git a/feature.py b/feature.py
index 174371f..6655344 100644
--- a/feature.py
+++ b/feature.py
@@ -1,2 +1,3 @@
 This is code for my new feature
 extend new feature
+This is edited by someone else
\ No newline at end of file
src/tutorial (new-feature)λ █          11:20:11
```

- Get latest changes from master back in your feature branch with
  - `git checkout master`
  - `git pull`
- `git checkout my-feature`
- `git merge master`

- Completely reset your branch to the version that is on origin (upstream branch)
- You will lose any changes in your own master branch!  
This hard to recover!
- `git fetch`
- `git reset --hard origin/master`

# Practice branching

- <https://learngitbranching.js.org>

Go to my repo for the  
resources

<https://git.io/JtV34>

<https://github.com/pingiun/workshop>

# Resources

- <https://learnxinyminutes.com/docs/git/>
- Git man pages, access these with e.g. `man git-commit`
- Git cheatsheet: <https://about.gitlab.com/images/press/git-cheat-sheet.pdf>
- How does git internally work? <https://www.git-scm.com/book/en/v2/Git-Internals-Git-Objects>
- Learn branching: <https://learngitbranching.js.org>
- Commit message style: <https://chris.beams.io/posts/git-commit/>
- Glossary: <https://help.github.com/en/articles/github-glossary>