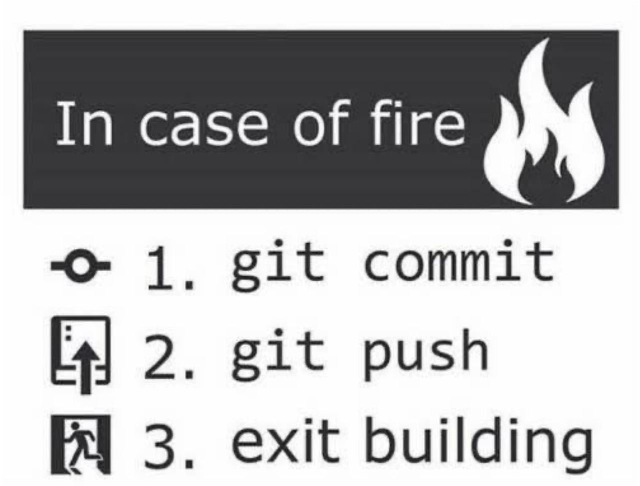
UNIVERSITÄT DES SAARLANDES

Basic Tools for NLP

Rishu Kumar, Iuliia Zaitova, Chris Hyek

Saarland University

25.05.2023



In case of fire

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

Git > Your life

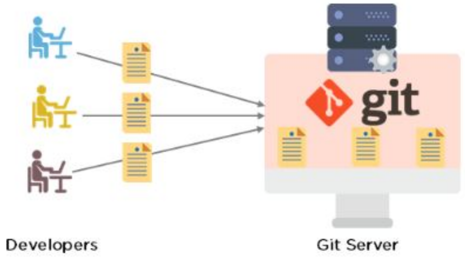
Git

- Type `git version` in your terminal
- Does it work?

Git

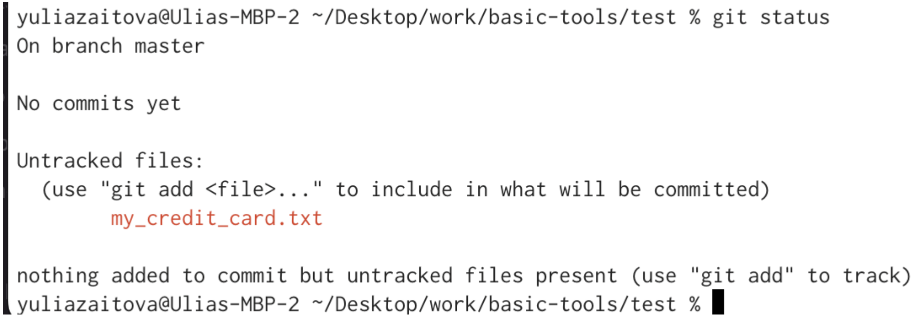
- Tool for your code project management
- Used to tracking changes, enabling multiple developers to work together on non-linear development

Git



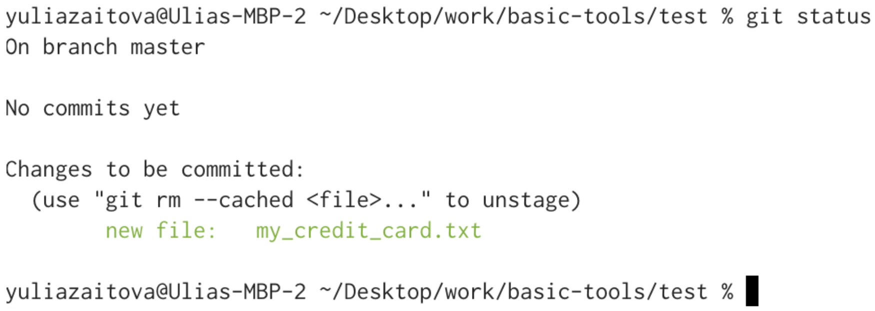
Initiating a git instance

- Create a repo `mkdir test`
- `cd test`
- Initialize a git instance `git init`
- `git status`
- `echo '1111' >> my_credit_card.txt`
- `git status`



```
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        my_credit_card.txt

nothing added to commit but untracked files present (use "git add" to track)
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test %
```

- `git add my_credit_card.txt`
- `git status`

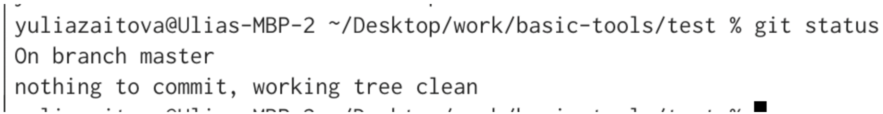


```
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   my_credit_card.txt

yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test %
```

- touch `my_password.txt`
- `git status`
- `git add .` or `git add *`

Git commit

- `git commit`
- `git commit -m 'Add credit card <name>'`
- `git status`



```
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git status
On branch master
nothing to commit, working tree clean
```

- But what about CVV?

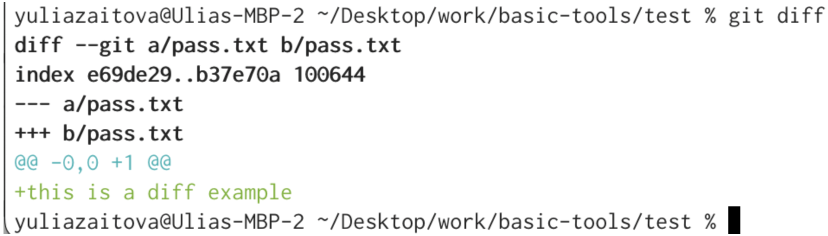
Amend commit

- `echo '222' >> cvv.txt`
- `git log`
- `git add .`
- `git commit --amend -m 'Add credit card <name>'`
- `git log`

```
git add .
git commit -m "changes" --no-verify
git push origin/master --force
```

Comparing changes

- `echo '321' >> cvv.txt`
- `git diff`

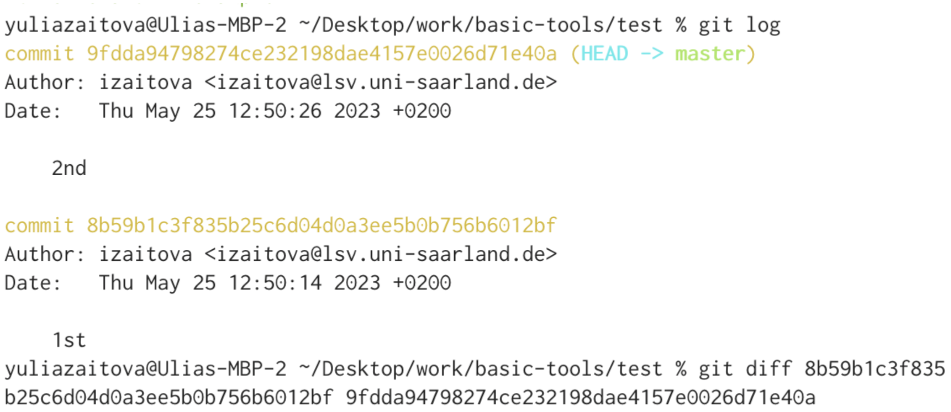


```
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git diff
diff --git a/pass.txt b/pass.txt
index e69de29..b37e70a 100644
--- a/pass.txt
+++ b/pass.txt
@@ -0,0 +1 @@
+this is a diff example
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test %
```

- `git diff HEAD <file>`

Git diff commits

- `echo '321' >> cvv.txt`
- `git log`
- `git diff <commit> <commit>` (can put any input)



```
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git log
commit 9fdda94798274ce232198dae4157e0026d71e40a (HEAD -> master)
Author: izaitova <izaitova@lsv.uni-saarland.de>
Date:   Thu May 25 12:50:26 2023 +0200

    2nd

commit 8b59b1c3f835b25c6d04d0a3ee5b0b756b6012bf
Author: izaitova <izaitova@lsv.uni-saarland.de>
Date:   Thu May 25 12:50:14 2023 +0200

    1st
yuliazaitova@Ulias-MBP-2 ~/Desktop/work/basic-tools/test % git diff 8b59b1c3f835
b25c6d04d0a3ee5b0b756b6012bf 9fdda94798274ce232198dae4157e0026d71e40a
```

Git stash

Temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else

- `touch diary.txt`
- `git stash`
- `touch notebook.txt`
- `git stash drop`
- `touch book.txt`
- `git stash pop` Diary is back!

.gitignore

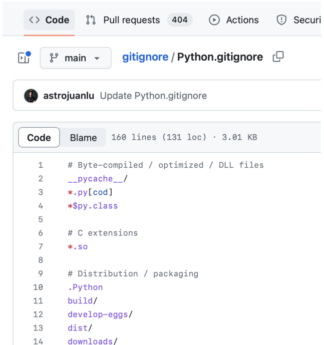
Files in git:

- tracked - a file which has been previously staged or committed;
- untracked - a file which has not been staged or committed; or
- ignored - a file which Git has been explicitly told to ignore.

What can we ignore?

- hidden system files, such as `.DS_Store` or `Thumbs.db`
- large data files (more than 50MB)[1]
- history files
- anything you do not want to track such as credential files etc

You can find some `.gitignore` templates online :)


Code
Pull requests 404
Actions
Securi
main
gitignore / Python.gitignore
astrojuanlu Update Python.gitignore
Code
Blame
160 lines (131 loc) · 3.01 KB

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
```

Branching

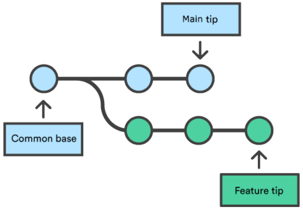
a branch is a new/separate version of the main repository (current status!)
Why? – keeps the main branch free from questionable code

- `git branch` – list all branches
- `git branch branch` – create a new branch called 'branch'
- `git branch -d branch` – delete the branch called 'branch'
- `git checkout branch` – switch to the branch called 'branch'

Quicker way – `git checkout -b new-branch`

Merging

- Create some files and write something in them
- Add and commit `git add .`, `git commit -m 'Add files'`
- Switch to the master branch `git checkout master`
- Merge with the new branch `git merge branch`



Pasting random git commands from stackoverflow instead of reading the docs
Fucking Git

Please don't do this.