Version control & git

An indroduction

LISCO Lab Meeting. October 2, 2019

What is version control?

- Manage versions of files projects etc...

What can be versioned?

- source code
- images
- music
- etc...

→ anything

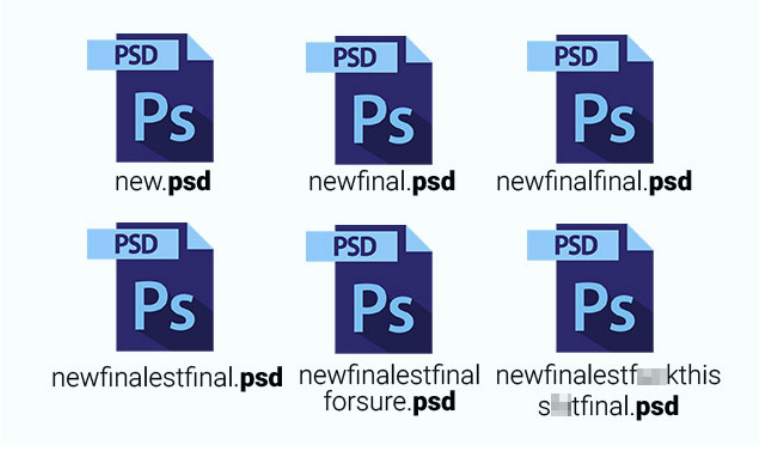
Why version control?

- Traceability
- "Jump" between different versions
- Team work and collaboration

Changes are saved with:

- Author
- Date and time
- Change note

Why version control?



Three kinds of version control

Maybe remove this slide? Maybe reduce?

Local

- Only on local machine
- for working alone on a project
- No data security

Centralized

- Repository is on remote server
- working copy on client
- any change on repository happens online

Distributed

- Complete clone on client AND server
- only necessary network traffic
- high data security through redundancy

Three kinds of version control

Maybe remove this slide? Maybe reduce?

Local

- Only on local machine
- for working alone on a project
- No data security

Centralized

- Repository is on remote server
- working copy on client
- any change on repository happens online

Distributed

- Complete clone on client AND server
- only necessary network traffic
- high data security through redundancy

Three kinds of version control

Maybe remove this slide? Maybe reduce?

Local

- Only on local machine
- for working alone on a project
- No data security

Centralized

- Repository is on remote server
- working copy on client
- any change on repository happens online

Distributed

- Complete clone on client AND server
- only necessary network traffic
- high data security through redundancy

Three kinds of version control

Maybe remove this slide? Maybe reduce?

Local

- Only on local machine
- for working alone on a project
- No data security

Centralized

- Repository is on remote server
- working copy on client
- any change on repository happens online

Distributed

- Complete clone on client AND server
- only necessary network traffic
- high data security through redundancy

Configure git

```
$ git config (--global) user.name "Felix Nonnengiesser"

$ git config (--global) user.email s8819853@stud.uni-frankfurt.de
```

Initialize a repository

Create a project folder:

```
$ mkdir website-with-git
```

```
$ cd website-with-git
```

Initialize repository:

```
$ git init
Initialized empty Git repository in
C:/Users/Felix/website-with-git/.git/
```

Initialize a repository

Create a project folder:

```
$ mkdir website-with-git
```

```
$ cd website-with-git
```

Initialize repository:

```
$ git init
Initialized empty Git repository in
C:/Users/Felix/website-with-git/.git/
```

Initialize a repository

```
$ ls -l .git

total 7
-rw-r--r-- 1 Felix 197121 130 Aug 27 12:02 config
-rw-r--r-- 1 Felix 197121 73 Aug 27 12:02 description
-rw-r--r-- 1 Felix 197121 23 Aug 27 12:02 HEAD
drwxr-xr-x 1 Felix 197121 0 Aug 27 12:02 hooks/
drwxr-xr-x 1 Felix 197121 0 Aug 27 12:02 info/
drwxr-xr-x 1 Felix 197121 0 Aug 27 12:02 objects/
drwxr-xr-x 1 Felix 197121 0 Aug 27 12:02 refs/
```

git status

```
$ git status

On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

Preperations for our little project

add link to project files. Or clone git repo?

Download template from `http://template.zip`

For Linux users:

```
$ wget http://template.zip
```

git status

update outputs with new project files

```
$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)

      myprofile.html

nothing added to commit but untracked files present (use "git add" to
track)
```

git add

```
$ git add myprofile.html
```

```
$ git status
On branch master
Changes to be committed:
(use "git reset HEAD <file>..." to unstage)

      new file:  myprofile.html
```

git commit

```
$ git commit -m "added profile page"
[master c18ea18] added profile page
1 file changed, 36 insertions(+)
create mode 100644 myprofile.html
```

git status

update outputs with new project files

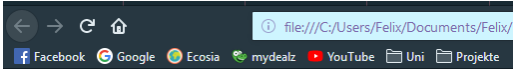
```
$ git status
On branch master
nothing to commit, working tree clean
```

git log

```
$ git log
Author:  Felix <nonni1997@hotmail.de>
Date:    Tue Aug 27 16:04:27 2019 +0200

     added profile page
```

Change content

- Open the myprofile.html file with a text editor of your choice. e.g. notepad++
- Fill in the gaps



This is the profile of Felix:

Full name: Felix Nonnengiesser
Date of birth: 05.08.1997
Favourite animal: flying bison

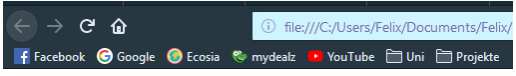
About me:

Hi I am Felix and I like ice cream!

```
<html>

<head>
    <title>My profile</title>
</head>

<body>
    <h1>This is the profile of *name*:</h1>

    <table>
        <tr>
            <td>Full name:</td>
            <td>*name goes here*</td>
        </tr>

        <tr>
            <td>Date of birth:</td>
            <td>*birthday goes here*</td>
        </tr>

        <tr>
            <td>Favourite animal:</td>
            <td>*animal goes here*</td>
        </tr>
    </table>

    <hr>

    <br>
    About me:
    <p>
        *This is some text about me*
    </p>
</body>

</html>
```

Change content

- Open the myprofile.html file with a text editor of your choice. e.g. notepad++
- Fill in the gaps



This is the profile of Felix:

Full name: Felix Nonnengiesser
Date of birth: 05.08.1997
Favourite animal: flying bison

About me:

Hi I am Felix and I like ice cream!

```
<html>

<head>
    <title>My profile</title>
</head>

<body>
    <h1>This is the profile of *name*:</h1>

    <table>
        <tr>
            <td>Full name:</td>
            <td>*name goes here*</td>
        </tr>

        <tr>
            <td>Date of birth:</td>
            <td>*birthday goes here*</td>
        </tr>

        <tr>
            <td>Favourite animal:</td>
            <td>*animal goes here*</td>
        </tr>
    </table>

    <hr>

    <br>
    About me:
    <p>
        *This is some text about me*
    </p>
</body>

</html>
```

git add & git commit

```
$ git add myprofile.html

$ git commit -m "updated profile information"
[master 7de6662] updated profile information
1 file changed, 5 insertions(+), 5 deletions(-)
```

git log

```
$ git log
commit 7de6662d0efd82b3e469594be85d38561c3fc536 (HEAD -> master)
Author:  Felix <nonni1997@hotmail.de>
Date:  Tue Aug 27 16:44:47 2019 +0200

updated profile information

commit c18ea185996a38679a0f3a623a9c10459045dccc
Author:  Felix <nonni1997@hotmail.de>
Date:  Tue Aug 27 16:04:27 2019 +0200

added profile page
```

Change content

git status

git diff

Branching

Figure: Git logo

Why branching

- stable
- development
- newAwesomeFeature
- featureXY

Add a new Feature

Let's add an image of ourselves to the profile page!

- we want to add a new feature
- we don't want to have unfinished features in our main/master branch
- we want to tinker around without breaking the master

$\rightarrow$ We create a new branch!

Show existing branches

```
$ git branch
* master
```

Create a new branch

```
$ git branch picture
```

```
$ git branch
* master
  picture
```

Change branch

```
$ git checkout picture
Switched to branch 'picture'

$ git branch
  master
* picture
```

Implement our new feature

tidy up

- open th
- remove
  add lin

  XX
- replace

```
<img src='
```

change title of page in this branch to provoke merge conflict later

Commit changes

```
$ git add myprofile.html

$ git commit -m "add picture to profile page"
```

add outputs?

Change branches

- Let's go back to the master branch
  `$ git checkout master`
- When loading our profile page, our picture is gone

→ master branch is not affected by changes in other branches.
We will find out later how to add these changes to the master branch.

Another feature

Our page looks a bit boring. Let's beautify it!

add output

- New feature → new branch
  `$ git branch style`
- Change to new branch
  `$ git checkout style`

Adding a new file to our repository

- We add a "stylesheet" to our website to make it look better.
- Let's add the stylesheet to our git repository

```
$ git add styles.css
```

add output

- Link the stylesheet to your html page:
  - open the html file in the editor
  - remove the comment tags `<!--` and `-->` in line

    add line number

    XX

Have a look at the new version of the profile page!

change title of page in this branch to provoke merge conflict later

add commit...

The updated page

THIS IS THE PROFILE OF FELIX:

Full name:	Felix Nonnengiesser
Date of birth:	05.08.1997
Favourite animal:	Flying Bison

ABOUT ME:

I like ice cream

Our current status

- We do have three different versions of our website
- None of them contains all the features we want
- But we want our master branch combining all these features!

Merge branches

add output

- to merge changes into the master branch we have to be inside the master branch
  `$ git checkout master`

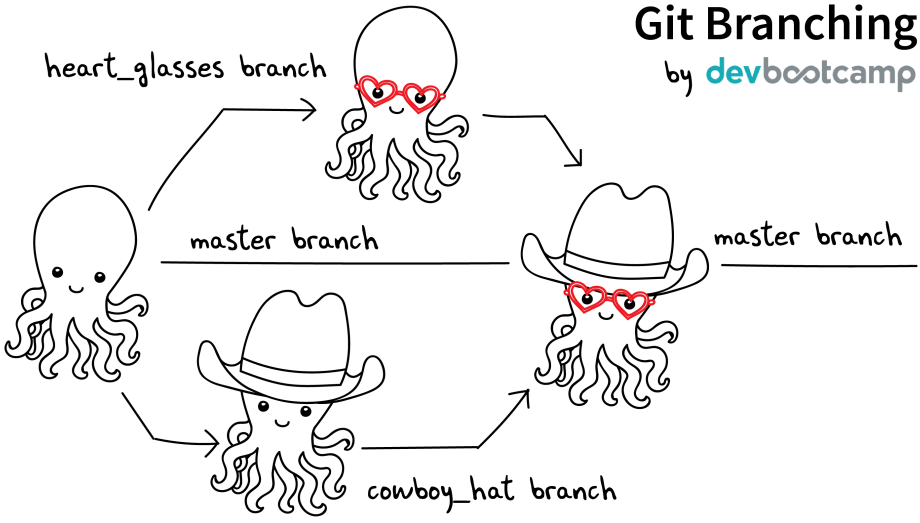
- Merge the picture branch into the master:
  `$ git merge picture`
- Merge the style branch into the master:
  `$ git merge style`

  handle merge conflict

Recap: What did we do?

add tikz graph with version history

Recap: What did we do?



Working together

Working together

- Git repos can be hosted on a server
- $\rightarrow$ other people can access the repo and collaborate

Services providing such hosting:

- GitHub
- GitLab $\leftarrow$ what we use in our lab
- Bitbucket
- ...

GitLab

- create and modify repositories online
- admins can manage repos, give access to users...

GitLab

mini gitlab tut here

GitLab

stuff like git clone, git push, git pull here?

Change branches