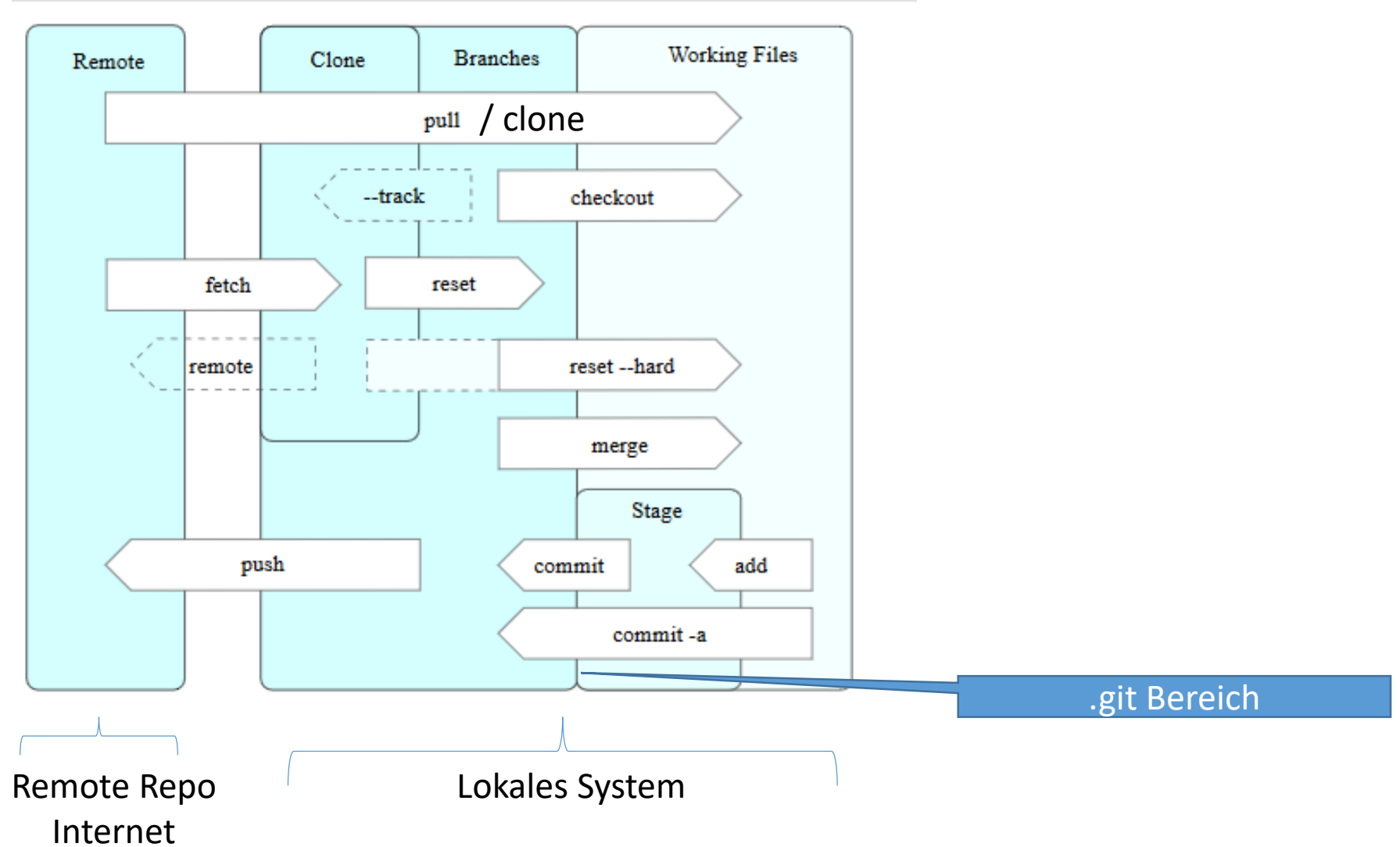


(Software-) Versionsverwaltung mit git, GitHub und PyCharm

Versionsverwaltung

- Verwaltung von Textdokumenten (z.B. Software)
 - Nachvollziehbarkeit von Änderungen
 - Versionierung
 - Zusammenarbeit von vielen Projektpartnern
 - Möglichkeit von Entwicklungszweigen (z.B. feature) - Branches
 - Offlinefähigkeit
-
- GIT - der zweite geniale Wurf von Linus Torvalds (2005)
<https://de.wikipedia.org/wiki/Git>

GIT – Datenbestände und Abläufe



GitHub – Account und Repository anlegen

https://github.com

The image shows a browser window with the GitHub homepage. The address bar displays 'https://github.com'. The navigation bar includes links for 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. The main content area is split into two panels. The left panel is the 'Sign up for GitHub' form, and the right panel is the 'Create a new repository' form.

Sign up for GitHub

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: **mue-htl** / Repository name: **4bhel-aap13** ✓

Great repository names are short and memorable. Need inspiration? How about [friendly-lamp](#)?

Description (optional): **Demo Repository**

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

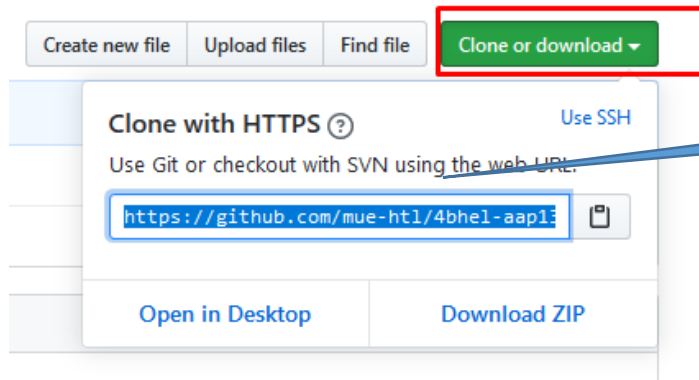
Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

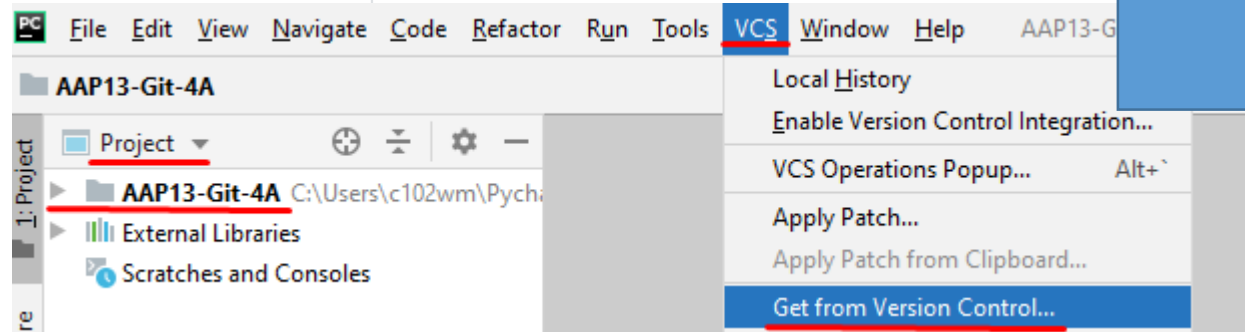
Add .gitignore: **Python** | Add a license: **None** ⓘ

Create repository

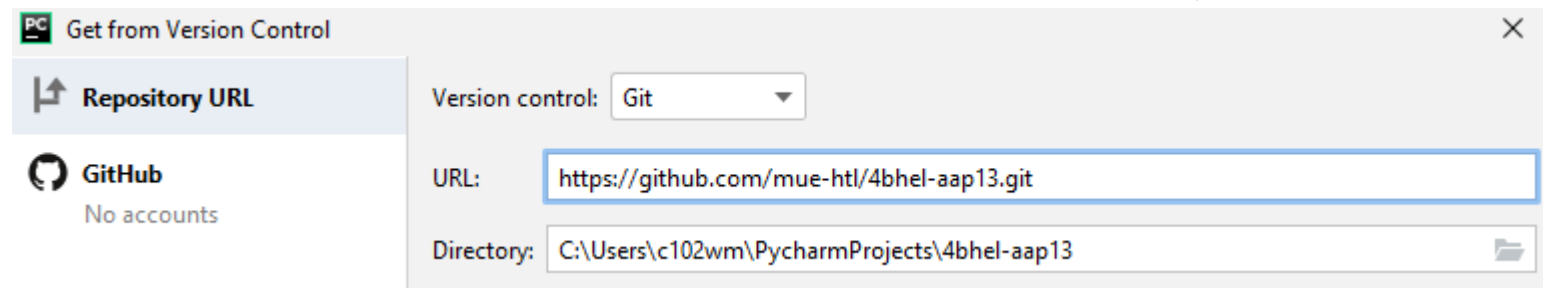
GitHub und PyCharm



Im GitHub Repo URL ermitteln



Im PyCharm mit Repo verknüpfen (git clone)



git add / git commit

The screenshot displays the PyCharm IDE interface with the following components:

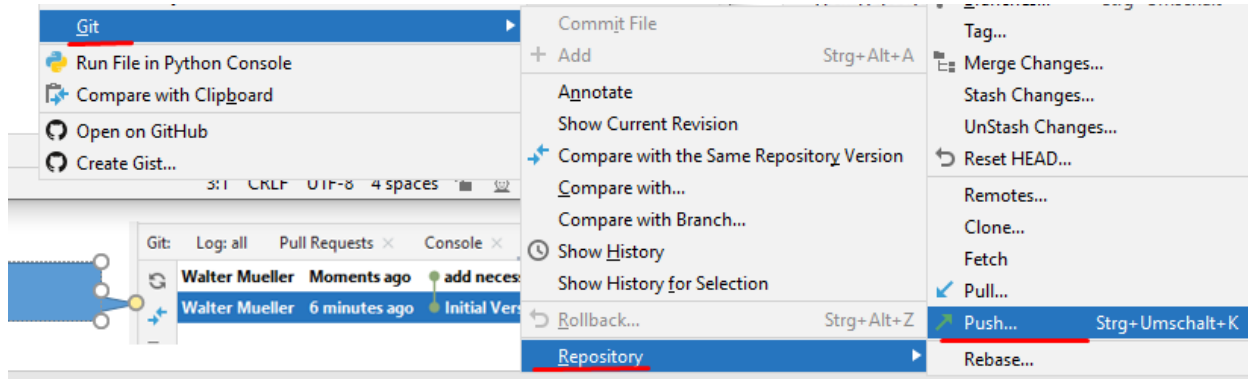
- Local Changes Panel:** Shows the file `helloWalter.py` under the `Default Changelist`. The file path is `C:\Users\c102wm\PycharmProject\helloWalter.py`. Below it, `Unversioned Files` shows 6 files.
- Editor:** Displays the content of `helloWalter.py`:

```
1 # jedes Programm braucht einen Kommentar
2 print("Hello Walter")
3
```
- Local History Panel:** Shows the `Git` history for the file. The `Commit File...` menu is open, showing options: `Add` (Strg+Alt+A), `Annotate`, `Show Current Revision`, `Compare with the Same Repository Version`, `Compare with...`, `Compare with Branch...`, `Show History`, `Show History for Selection`, `Rollback...` (Strg+Alt+Z), and `Repository`.
- Bottom Panel:** Shows the `Commit` button (highlighted with a red box) and the `Amend` button. The commit message field contains `add necessary comment`.
- Git Log:** Shows the commit history for `helloWalter.py`:
 - Walter Mueller Moments ago add necessary comment
 - Walter Mueller 6 minutes ago Initial Version of my hello world program

Annotations:

- A blue callout bubble points to the `git add` and `git commit` buttons in the bottom panel.
- A blue callout bubble points to the `Git / Show History` button in the bottom panel.

git push – mit/zum remote repository abgleichen



Anmeldung mit github User + Passwort
PUSH

Änderungen im Web verfügbar

Demo Repository

Manage topics

Edit

3 commits

1 branch

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



mueller-w-htl add necessary comment

Latest commit fe98e3b 9 minutes ago



.gitignore

Initial commit

33 minutes ago



README.md

Initial commit

33 minutes ago



helloWalter.py

add necessary comment

9 minutes ago

GIT im Dateisystem

(C:) > Benutzer > c102wm > PycharmProjects > 4bhel-aap13 > .git >

Name	Änderungsdatum	Typ	Größe
hooks	04.06.2020 18:24	Dateiordner	
info	04.06.2020 18:24	Dateiordner	
logs	04.06.2020 18:24	Dateiordner	
objects	04.06.2020 18:53	Dateiordner	
refs	04.06.2020 18:24	Dateiordner	
COMMIT_EDITMSG	04.06.2020 18:34	Datei	1 KB
config	04.06.2020 18:24	Datei	1 KB
description	04.06.2020 18:24	Datei	1 KB
FETCH_HEAD	04.06.2020 18:53	Datei	1 KB
HEAD	04.06.2020 18:24	Datei	1 KB
index	04.06.2020 18:53	Datei	1 KB
ORIG_HEAD	04.06.2020 18:53	Datei	1 KB
packed-refs	04.06.2020 18:24	Datei	1 KB

Unter .git

Content unter hash abgelegt

(C:) > Benutzer > c102wm > PycharmProjects > 4bhel-aap13 > .git > objects > 3f >

Name	Änderungsdatum	Typ	Größe
75dd696bb78792e2c99b613e5ae715a0463e59	04.06.2020 18:27	Datei	1 KB

Git for Windows - git in der shell

.git	04.06.2020 19:20	Dateiordner	
.idea	04.06.2020 19:02	Dateiordner	
.gitignore	04.06.2020 19:20	Textdokument	2 KB
helloWalter.py	04.06.2020 18:53	PY-Datei	1 KB
README.md	04.06.2020 18:24	MD-Datei	1 KB

Git for Windows:

<https://git-scm.com/download/win>

rechte Maustaste/
Git bash here

```
MINGW64:/c/Users/c102wm/PycharmProjects/4bhel-aap13

c102wm@c102-mwa2 MINGW64 ~/PycharmProjects/4bhel-aap13 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

no changes added to commit (use "git add" and/or "git commit -a")

c102wm@c102-mwa2 MINGW64 ~/PycharmProjects/4bhel-aap13 (master)
$ git commit -a -m "ignore pycharm config files in .idea"
[master 0eaa4a4] ignore pycharm config files in .idea
1 file changed, 3 insertions(+)

c102wm@c102-mwa2 MINGW64 ~/PycharmProjects/4bhel-aap13 (master)
$ git push
```

repository an neue Stelle clonen

- PyCharm / VCS / Git / Clone
in neuen Pfad
- Dort Änderungen machen,
dann:
git commit, push
- Im ursprünglichen Projekt
Änderungen vom remote
Repo nachziehen
git pull

anderen Zugriff auf Projekt im GitHub geben

The screenshot shows the GitHub interface for a repository named 'mue-htl / 4bhel-aap13'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security' (0), 'Insights', and 'Settings'. The 'Settings' tab is highlighted with a red box. On the left sidebar, under 'Options', the 'Manage access' option is highlighted with a red box. The main content area is titled 'Who has access' and is divided into two sections: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'PUBLIC REPOSITORY' section states 'This repository is public and visible to anyone.' and has a 'Manage' link. The 'DIRECT ACCESS' section states '0 collaborators have access to this repository. Only you can contribute to this repository.' and has a red box around the collaborator icon.

[mue-htl / 4bhel-aap13](#) Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security 0](#) [Insights](#) **Settings**

[Options](#)
[Manage access](#)
[Branches](#)
[Webhooks](#)
[Notifications](#)
[Integrations](#)

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Git

- Git Branches
 - Entwicklungszweige für noch nicht akzeptierte Änderungen
- Git Merges
 - Integrieren von Branches in den Entwicklungsbaum