Introduction to Git

Distributed Version Control

AW Academy

Agenda

- What is Version Control? What's Git?
- Install Git
- First steps: Working locally
- Working collaboratively and distributed
- Git in VSCode

AW Academy

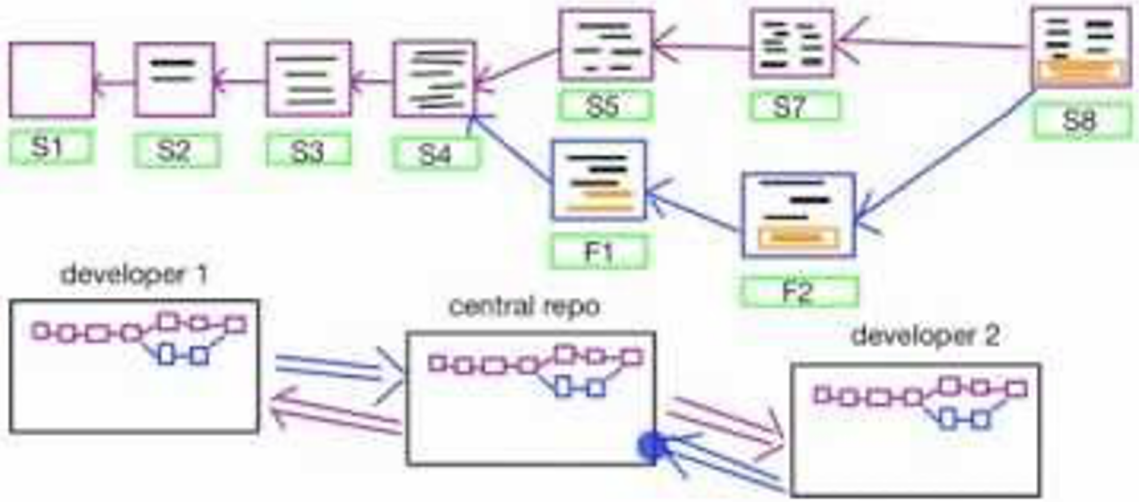
Why Version Control?

Let's say you need to work in groups on a coding project.

How would you work on the same files, sharing results with low friction?

Introduction to Git

Version Control System



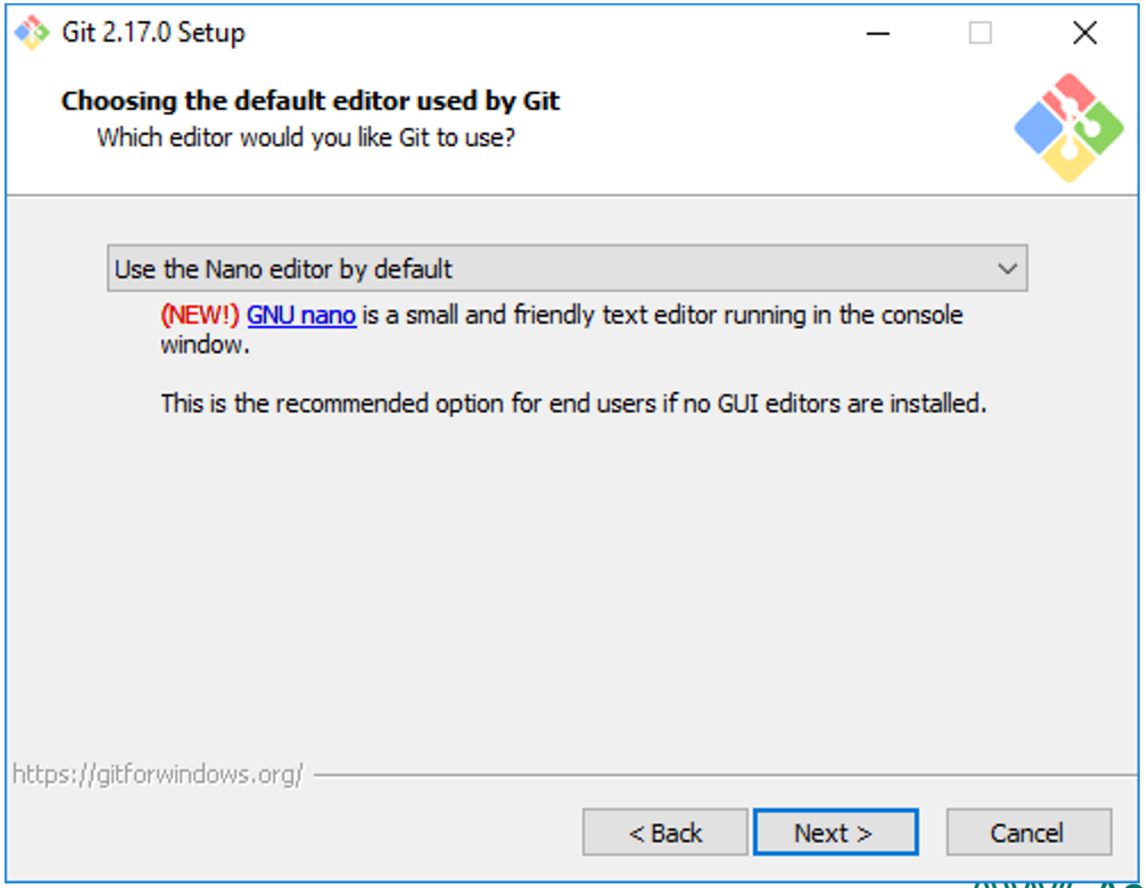
Advantages of Verson Control

- Save-Points (“Commits”)
- Multiple lines of development (“Branches”)
- Collaborate

AW Academy

Installing Git

https://git-scm.com



AW Academy

GitBash

Creating and entering folders

```
c:\ dir

c:\ cd MyName

c:\ md git-test
c:\ mkdir git-test

c:\ cd git-test
```

AW Academy

Creating a Git repository

```
c:\ git init

Initialized empty Git repository in
c:\Temp\Academy\git-test>/.git/
```

AW Academy

Git status

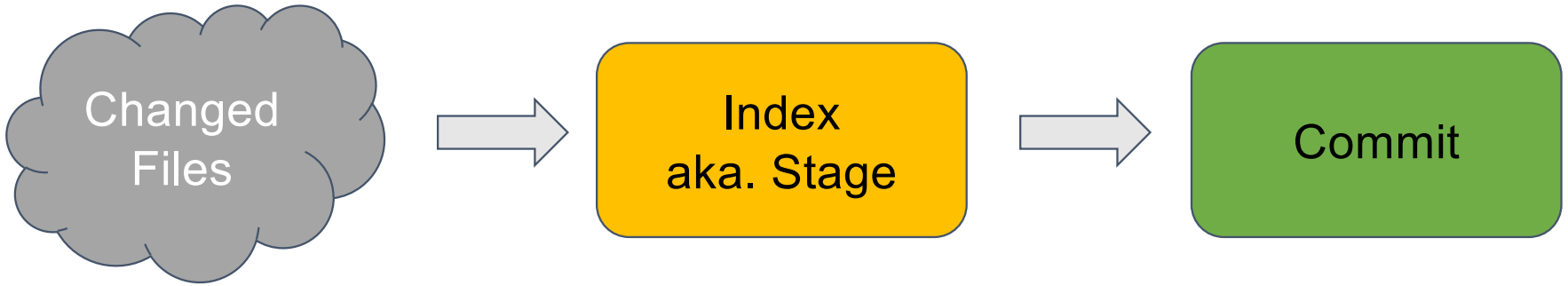
```
c:\Temp\Academy\git-test> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git
add" to track)
```

AW Academy

Commit

- Snapshot of the files
- Message / Description
- Author
- Date and Time

Commit in two steps



AW Academy

Your First Commit

Create a new file: readme.txt

```
c:\Temp\Academy\git-test> git status
On branch master

No commits yet

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

        readme.txt

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

AW Academy

Your First Commit

```
c:\Temp\Academy\git-test> git add readme.txt

c:\Temp\Academy\git-test> git commit

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Philip Frank@LAPTOP-
HH3BJN67.(none)')
```

AW Academy

Your First Commit

```
c:\Temp\Academy\git-test> git commit
```

It is possible to use -m to add a commit message

AW Academy

More Git Commands

```
git log
```

List the latest commits (q to exit)

```
git diff
```

Show changes not on Stage

```
git diff --staged
```

Show changes on Stage

```
git show <commit-id>
```

Show the contents of a commit

More:
https://git-scm.com/docs

AW Academy

Commit-Id

```
c:\Temp\Academy\git-test> git log
commit e30e7f6a77c2fa3f30135bcc6443201a9ff61206
Author: Foo Fooa <mail@foo.se>
Date:   Thu May 24 13:52:48 2018 +0200

    My first Commit
```

Uniquely defined by the commit contents

AW Academy

The opposite of `add` is...

```
git reset HEAD my-file
```

… when you added a change to the index but would rather not have it in the next commit

```
git rm --cached my-file
```

… when you want to delete a file (deletion is also a change, will be recorded in the next commit)

Exercise 1

AW Academy

Commits

- Every commit knows its predecessor
  - With this, the history of all files can be reconstructed
- Changes in files are saved linewise in a commit
  - Visible in `git diff` or `git show`

Tips and Tricks

Add all Files (new and changed) to the next commit

```
git add .
```

Inspect your changes while adding them with `-p`

```
git add -p readme.txt
```

Branching and Merging

AW Academy

Branching

Branch off from the last commit

```
git branch my-branch
```

List all (local) Branches

```
git branch
```

Switch to a Branch

```
git checkout my-branch
```

AW Academy

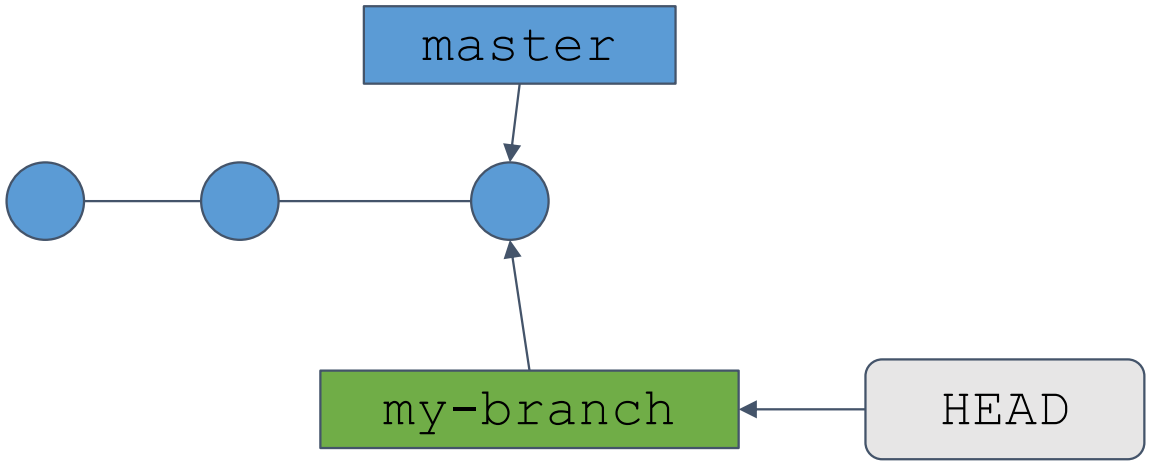
Branching

Create branch and checkout the new branch

```
git checkout -b my-branch
```

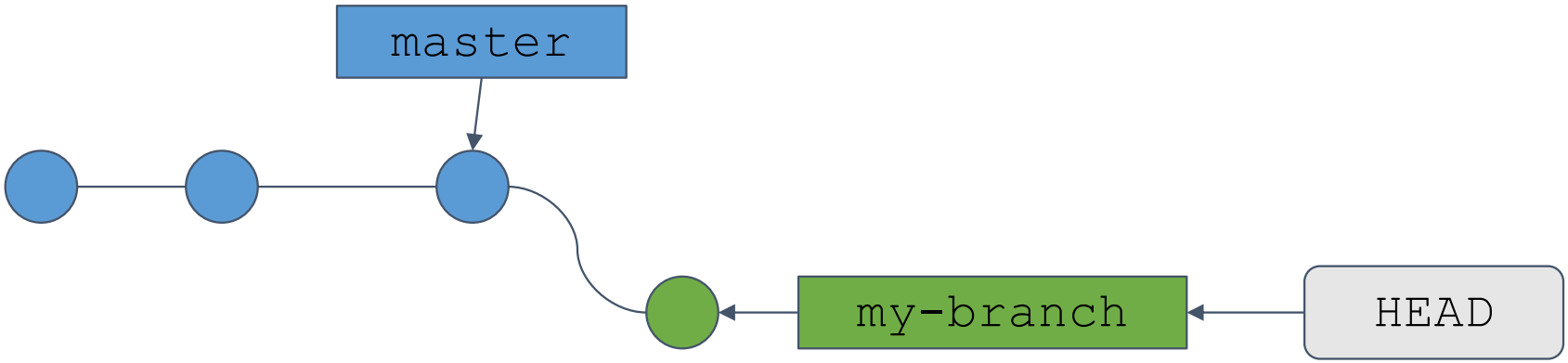
Start on master with three commits

```
$ git branch my-branch
$ git checkout my-branch
```



Continue on my-branch

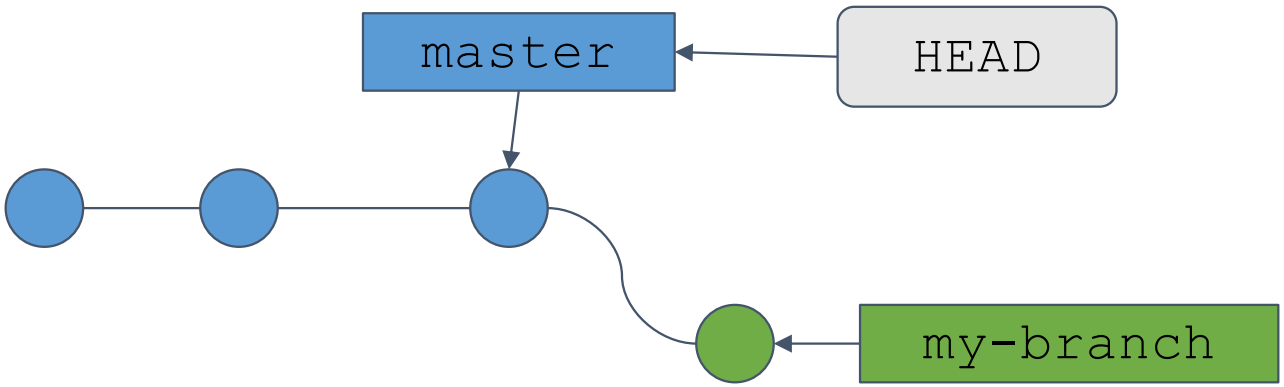
```
$ git add ...
$ git commit
```



AW Academy

Continue on my-branch

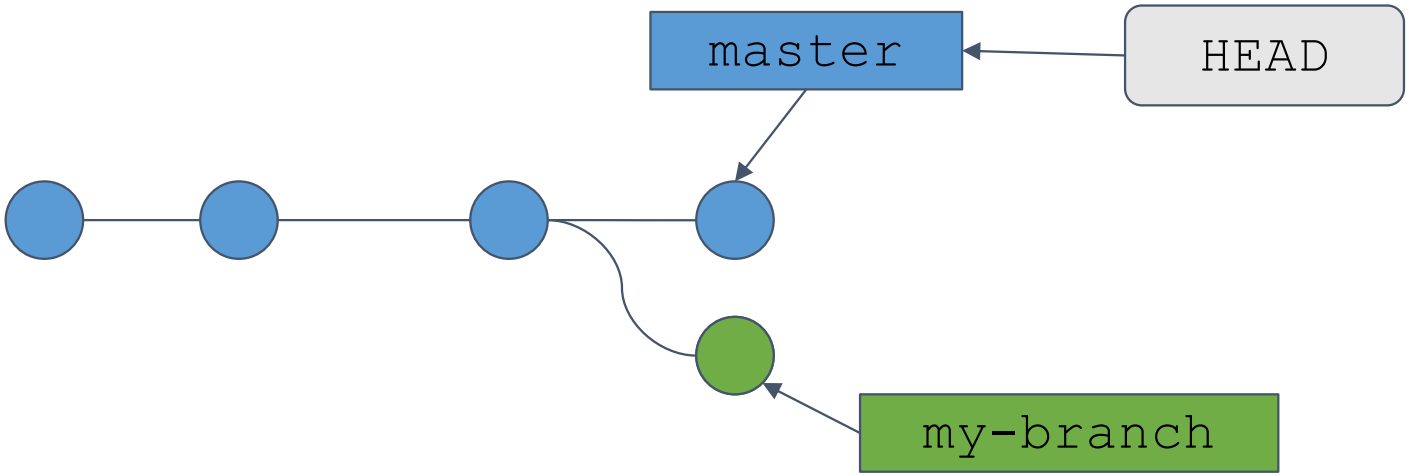
```
$ git checkout master
```



AW Academy

Continue on master

```
$ git add …
$ git commit
```

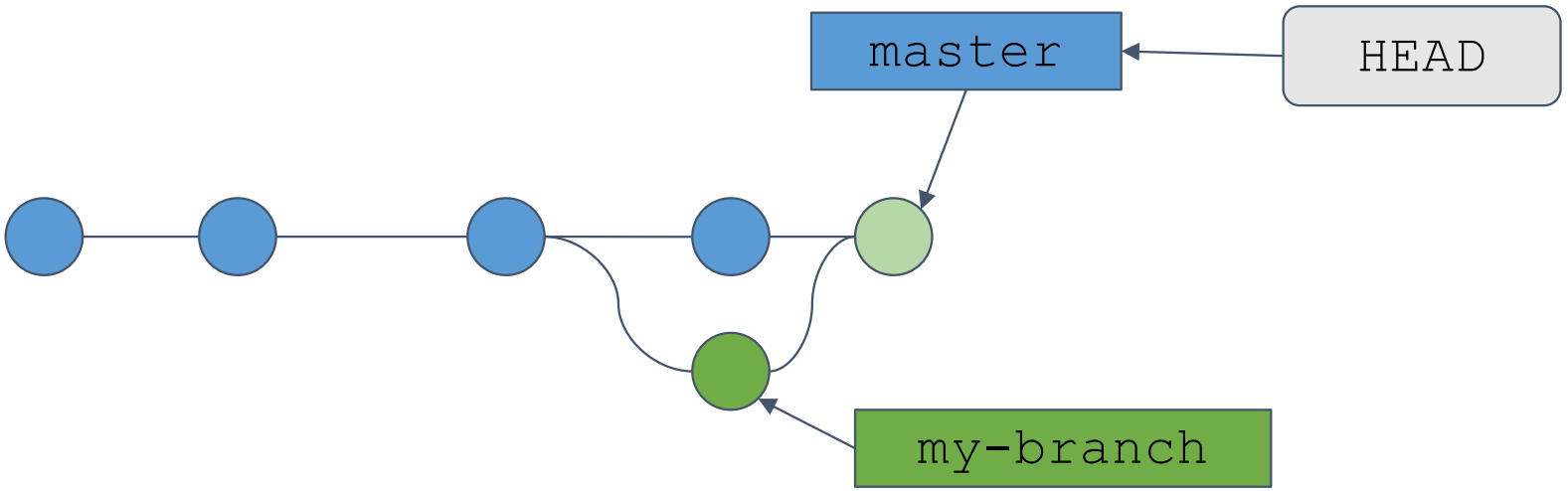


AW Academy

Merge

Continue on master

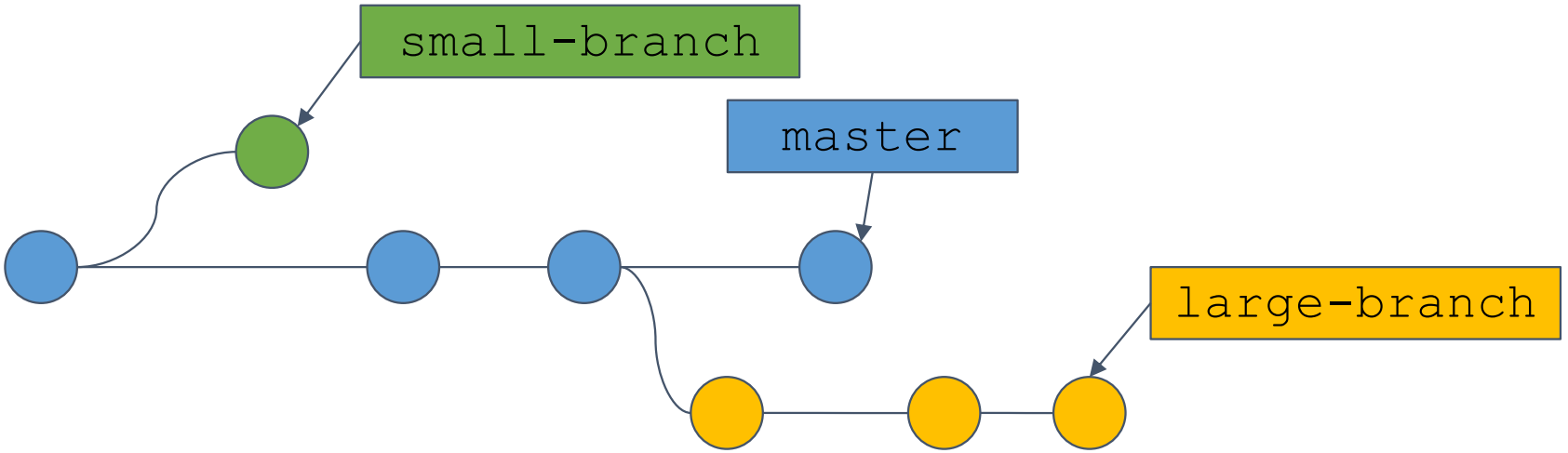
```
$ git merge my-branch
```



AW Academy

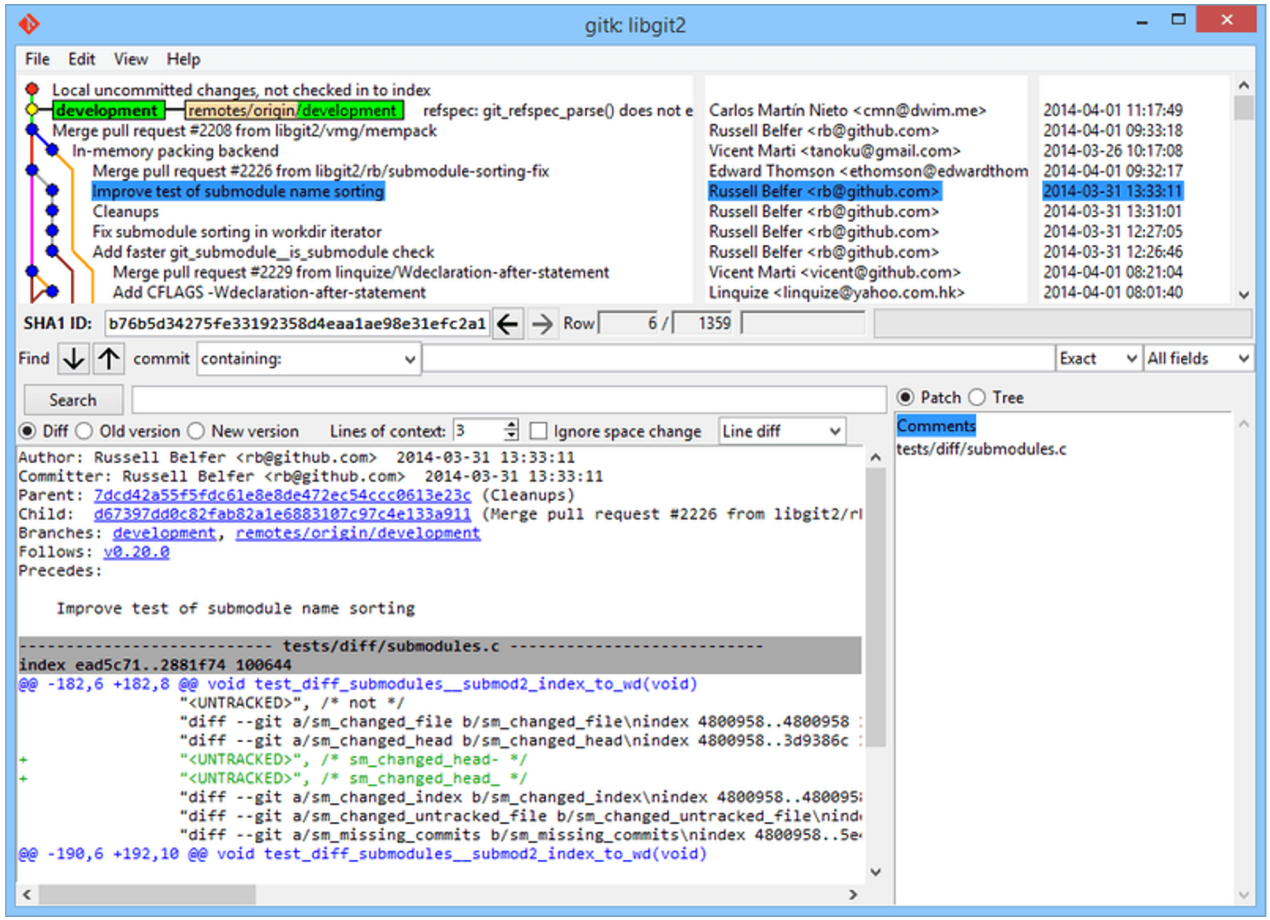
Branches

- Are actually just pointers to commits
- There is always a default Branch `master`
- Directed Acyclic Graph



AW Academy

```
$ gitk
```



AW Academy

Conflicts

Changes in master

```
 public static void main(String[] args) {
     int x = 4;
+    x = x + 3;
     System.out.println(x);
 }
```

Changes in my-branch

```
 public static void main(String[] args) {
     int x = 4;
+    System.out.println("Number x is:");
     System.out.println(x);
 }
```

AW Academy

Conflicts

```
$ git merge my-branch
Auto-merging src/Test.java
CONFLICT (content): Merge conflict in src/Test.java
Automatic merge failed; fix conflicts and then commit
the result.
```

Solving Conflicts

Test.java

```
public static void main(String[] args) {
    int x = 4;
<<<<<<< HEAD
    x = x + 3;
=======
    System.out.println("Number x is:");
>>>>>>> my-branch
    System.out.println(x);
}
```

Solving Conflicts

Test.java

```
public static void main(String[] args) {
    int x = 4;
    x = x + 3;
    System.out.println("Number x ist:");
    System.out.println(x);
}
```

```
$ git add src/Test.java
$ git commit
```

AW Academy

Exercise 2

AW Academy

Tips and Tricks

Set all changes aside (to the "Stash")

```
$ git stash
```

Bring back changes from stash

```
$ git stash pop
```

https://git-scm.com/docs/git-stash

AW Academy

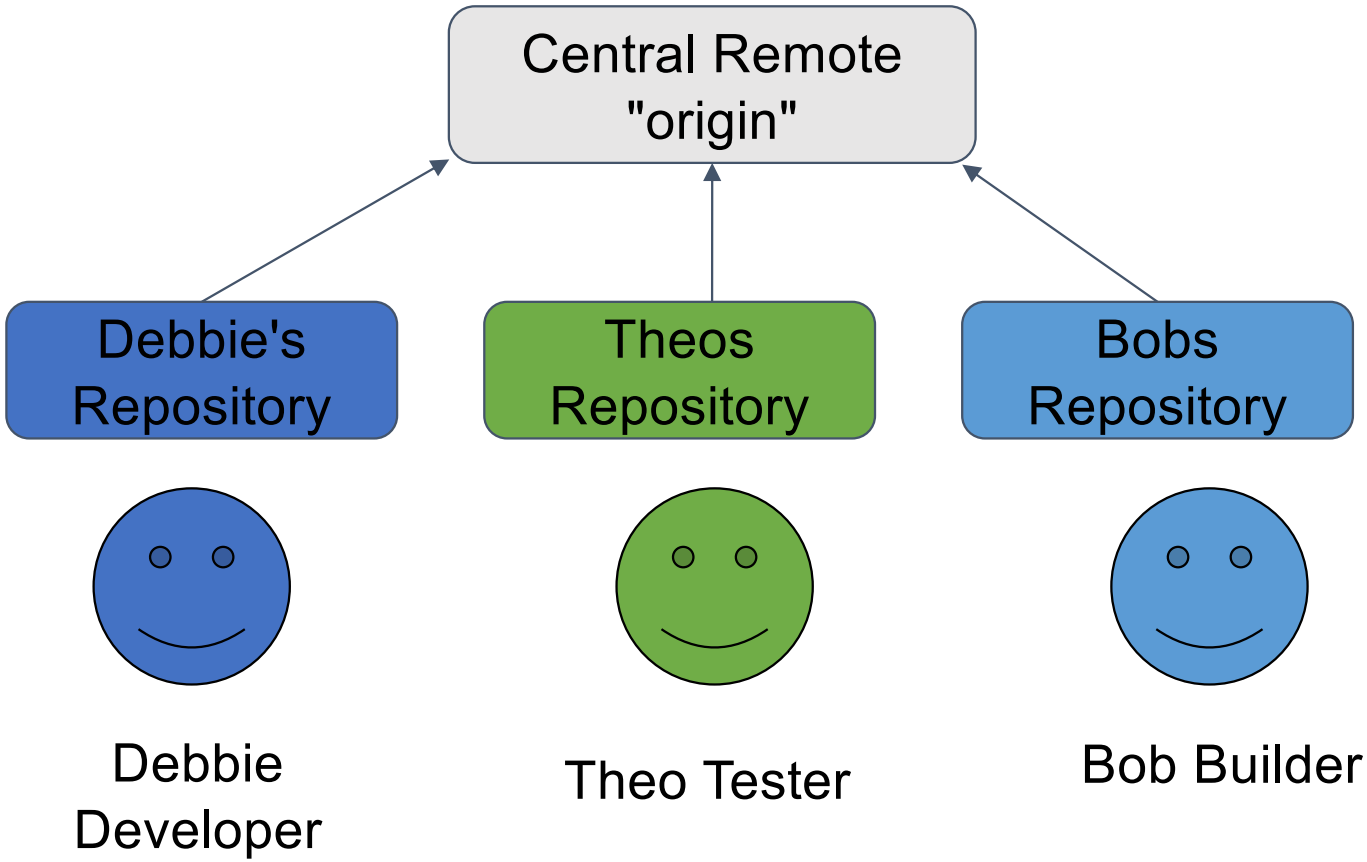
Multiplayer Mode

AW Academy

Remote

- Another copy of the repository
- There might be commits we don't have
- There might be commits missing from us
- There should always be a common ancestor-commit

In most cases: A central repository



AW Academy

Git-Repository as a Service

- There are many Providers of central Repositories
- GitHub is the biggest, provides public repositories for free (private repositories with some limitations)
- GitLab and BitBucket are alternatives, they also provide unlimited free private repositories
- Often additional features are provided, like Issue-Tracking
- With some effort you can also host a git repository on your own server
- We will now use **GitLab**, but everything we learn also applies to other repositories, public or private

AW Academy

Create a GitHub Account!

1. Register
2. Generate an SSH key on your computer
3. Add the SSK public key to your GitHub Account

https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh

AW Academy

Your GitHub Profile

It's your Software Developer Business Card

Publish as much of your work there

AW Academy

Adding a remote

```
$ git remote add origin git@github.com:<user>/<repo>.git
```

Name of the remote: "origin"

Upload your commits

```
$ git push -u origin master
```

Push all commits from the current branch to Remote `origin`, there on branch `master`
`-u` assignes the Remote-master as "upstream" of our master, to use it by default in the future

Do this only for a new, empty remote repository

AW Academy

Join the work on an existing repository

```
$ git clone git@github.com:<user>/<repo>.git
```

You get a copy of the repository

Branch `master` will be connected to the upstream branch.

AW Academy

Push und Pull

Get Commits from Upstream and merge

```
$ git pull
```

Send my Commits to Upstream

```
$ git push
```

AW Academy

Pull in detail

Remote Branches

```
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

AW Academy

Pull in detail

What `pull` actually dues

```
$ git fetch
```

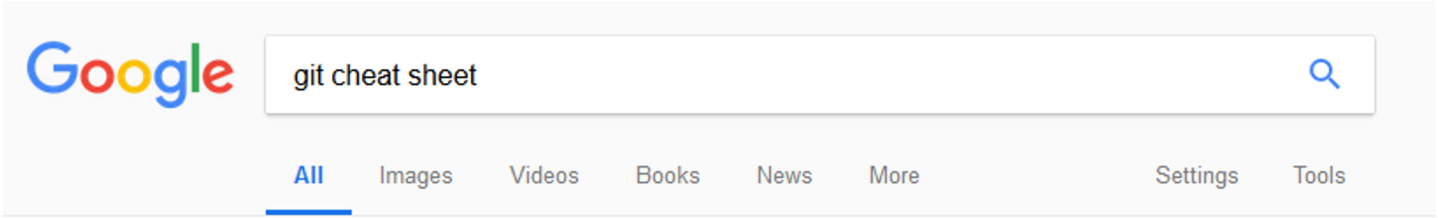
Updates the Remote Branches
Afterwards:

```
$ git merge origin/master
```

Watch out

- Pull can cause merge conflicts
- Push does not work if your branch is not up to date
  - Pull first

AW Academy


Google
git cheat sheet
All
Images
Videos
Books
News
More
Settings
Tools
About 2.540.000 results (0,38 seconds)
Git Cheat Sheet - Git Tower
https://www.git-tower.com/blog/git-cheat-sheet
Git Cheat Sheet. Our Git cheat sheet saves you from learning all the commands by heart. Download it for free. Even with a GUI application like Tower at hand ...
[PDF] Git Cheat Sheet - GitHub Professional Services
https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. This cheat sheet summarizes commonly used ...
[PDF] Git Cheat Sheet - GitHub Education
https://education.github.com/git-cheat-sheet-education.pdf
GIT CHEAT SHEET. STAGE & SNAPSHOT. Working with snapshots and the Git staging area git status show modified files in working directory, staged for your ...
Git cheat sheet | Atlassian Git Tutorial
https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet
Git cheat sheet that serves as a quick reference for basic Git commands to help you learn Git. Git branches, remote repositories, undoing changes, and more.
NDP Software :: Git Cheatsheet
https://ndpsoftware.com/git-cheatsheet.html
Interactive Git Cheatsheet, categorizing commands based on what they affect.

AW Academy

Exercise 3

AW Academy

Tips and Tricks

"Oh no, I forgot something in my commit!"

```
$ git add …
$ git commit --amend
```

Only allowed, if you have not pushed the last commit yet!

Merge Requests

Often you should not (or can not) push directly to the master-Branch of a project.

Instead you work on your own branch or in a separate remote repository ("fork") and request your changes to be merged after review.

Also called "Pull Request"

.gitignore

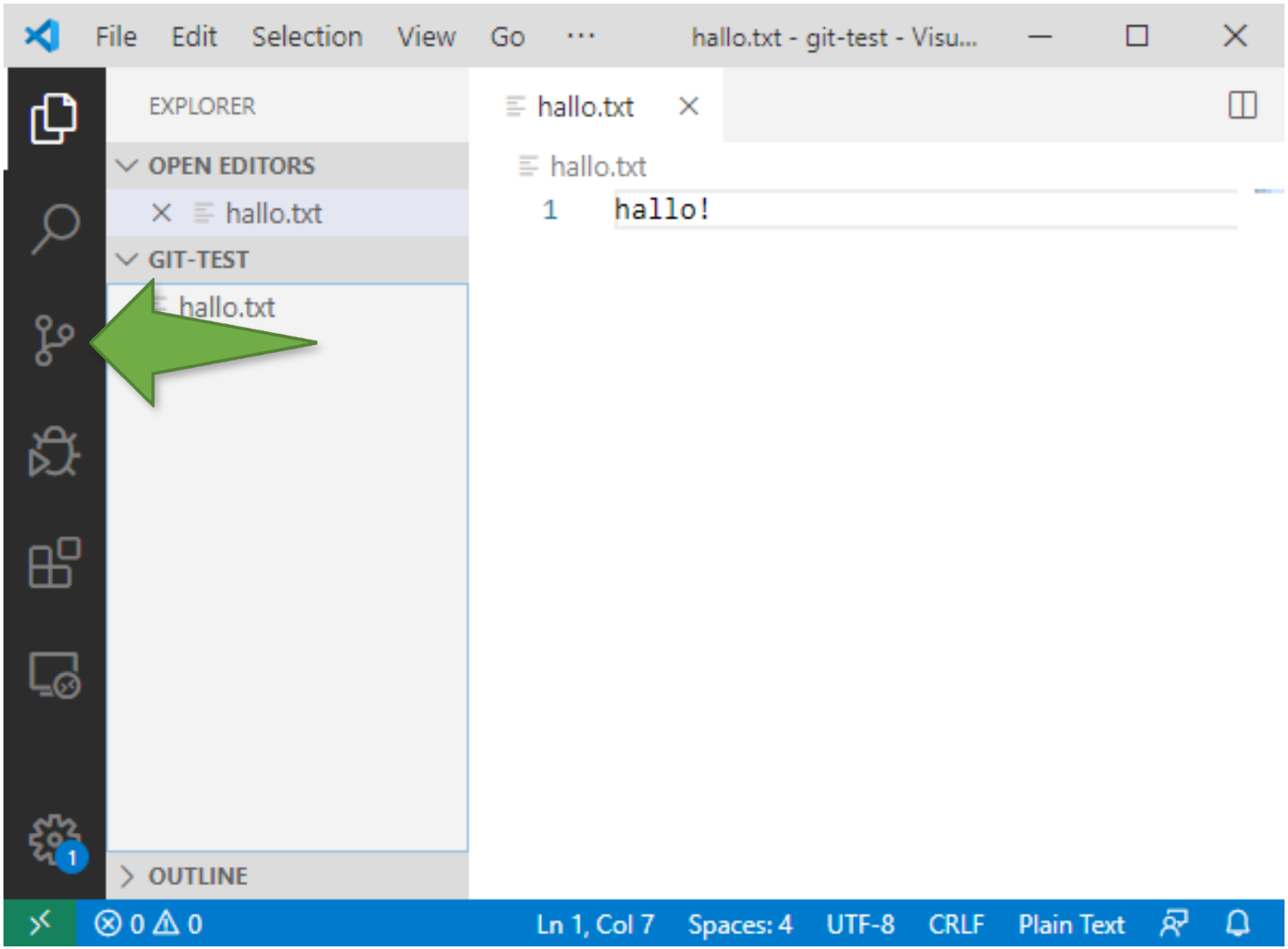
A file listing all files to be ignore by git, e.g.:

```
node_modules
```

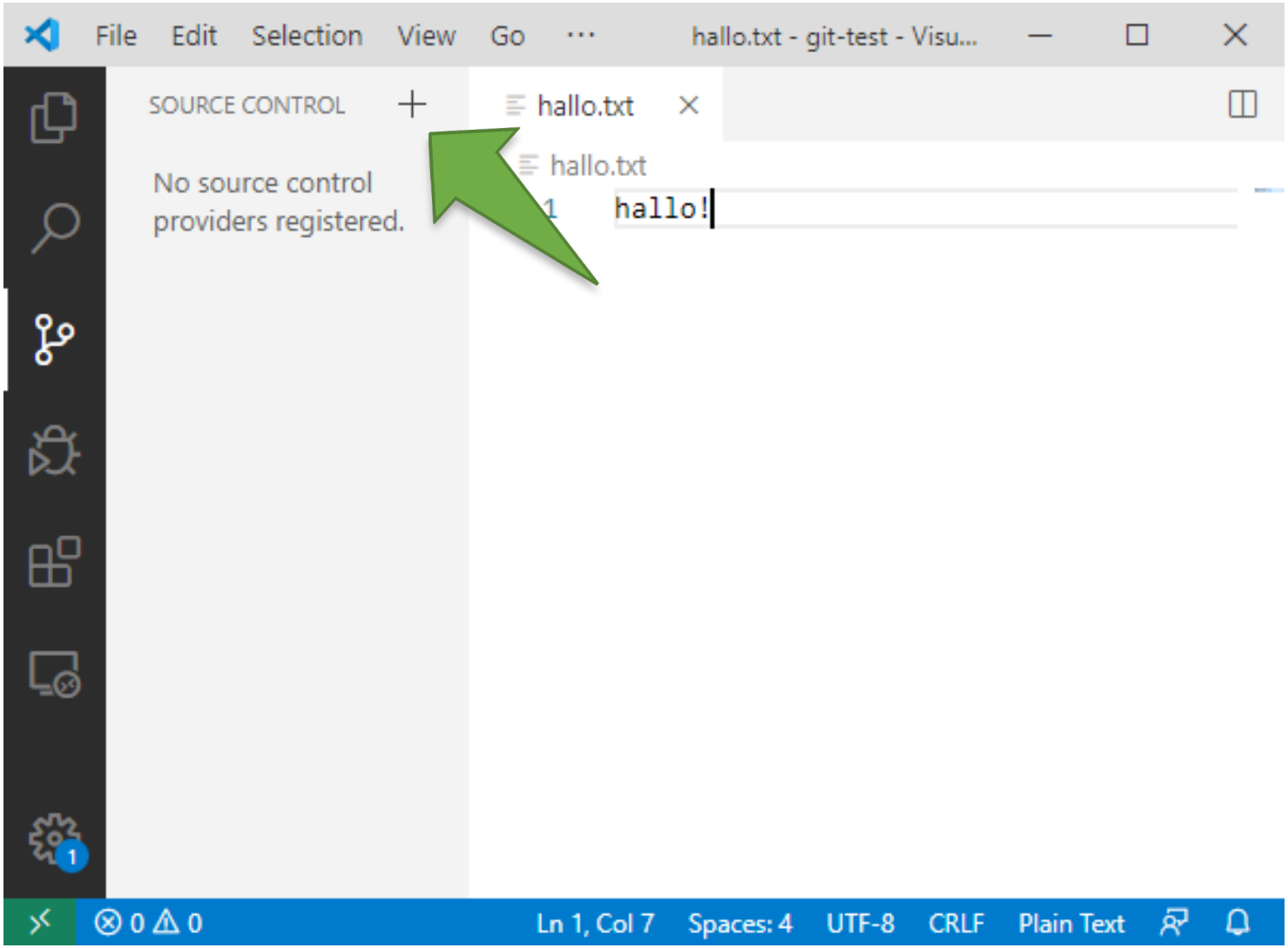
```
$ git add .gitignore
$ git commit
```

Git with Visual Studio Code

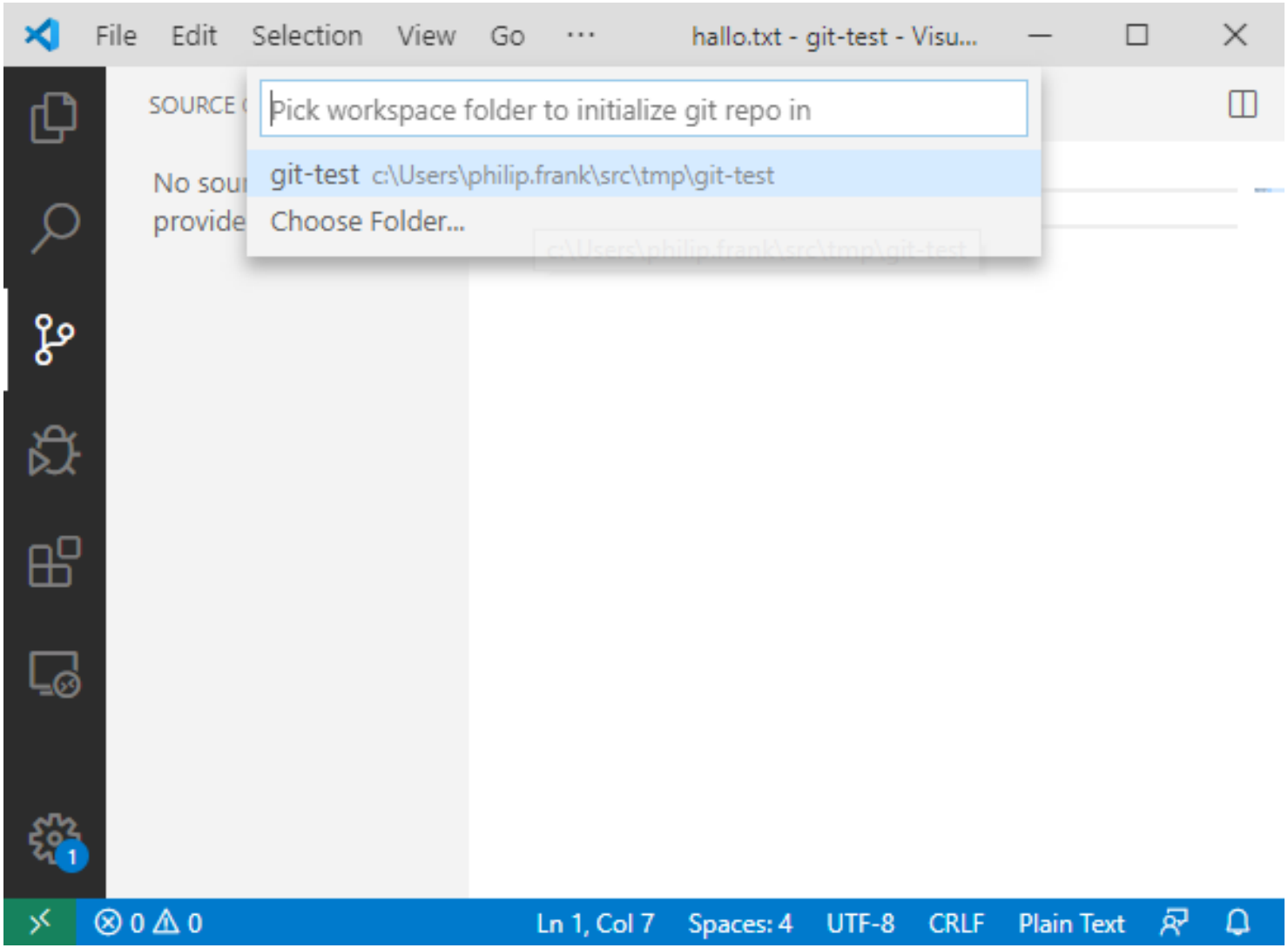
AW Academy


File
Edit
Selection
View
Go
···
hallo.txt - git-test - Visu...
EXPLORER
OPEN EDITORS
hallo.txt
GIT-TEST
hallo.txt
OUTLINE
hallo.txt
hallo.txt
1
hallo!
0 0
Ln 1, Col 7
Spaces: 4
UTF-8
CRLF
Plain Text

AW Academy


File
Edit
Selection
View
Go
hallo.txt - git-test - Visu...
SOURCE CONTROL
No source control providers registered.
hallo.txt
hallo.txt
1 hallo!
0 0
Ln 1, Col 7
Spaces: 4
UTF-8
CRLF
Plain Text

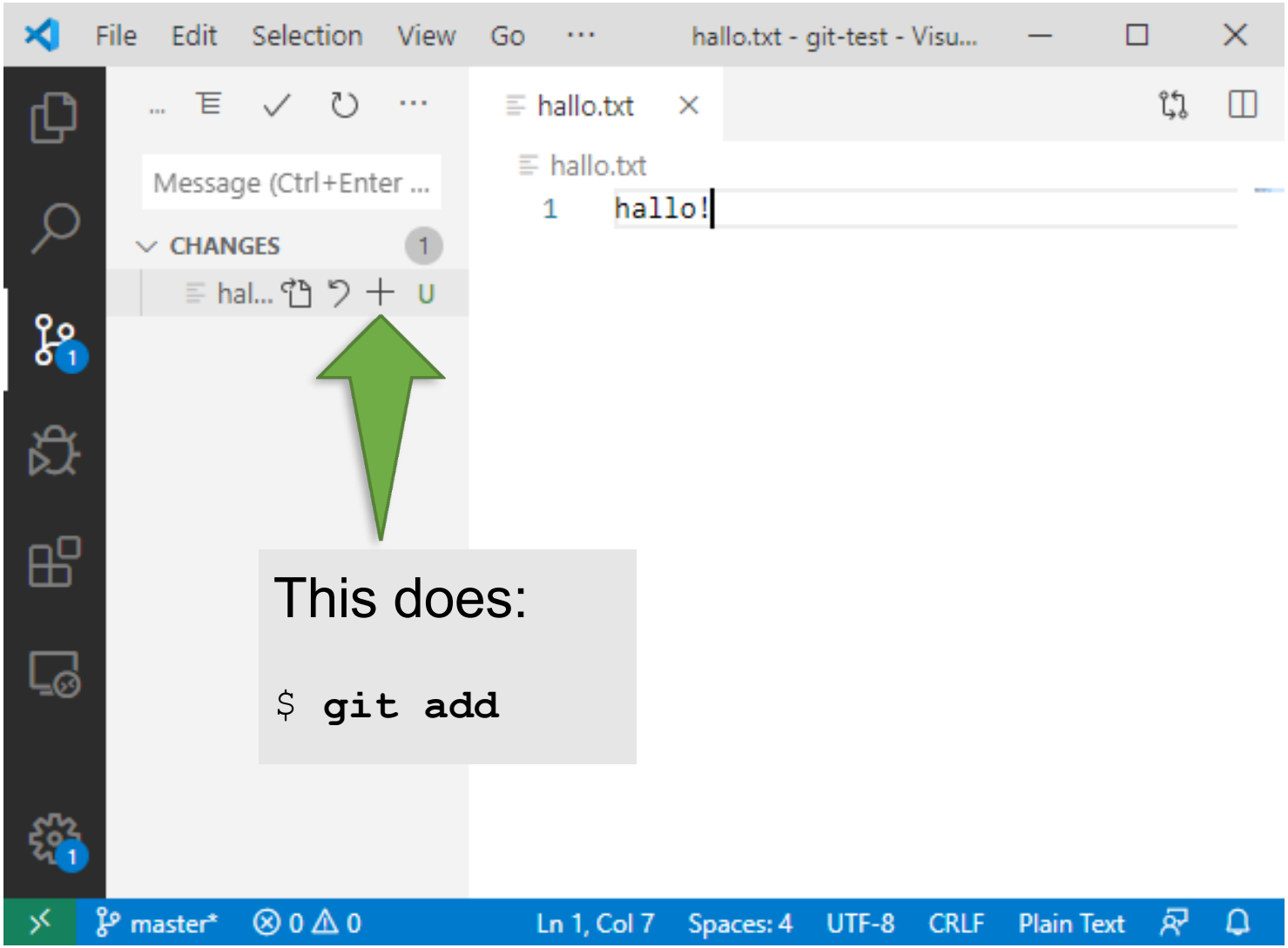
AW Academy



Hier passiert:

```
$ git init
```

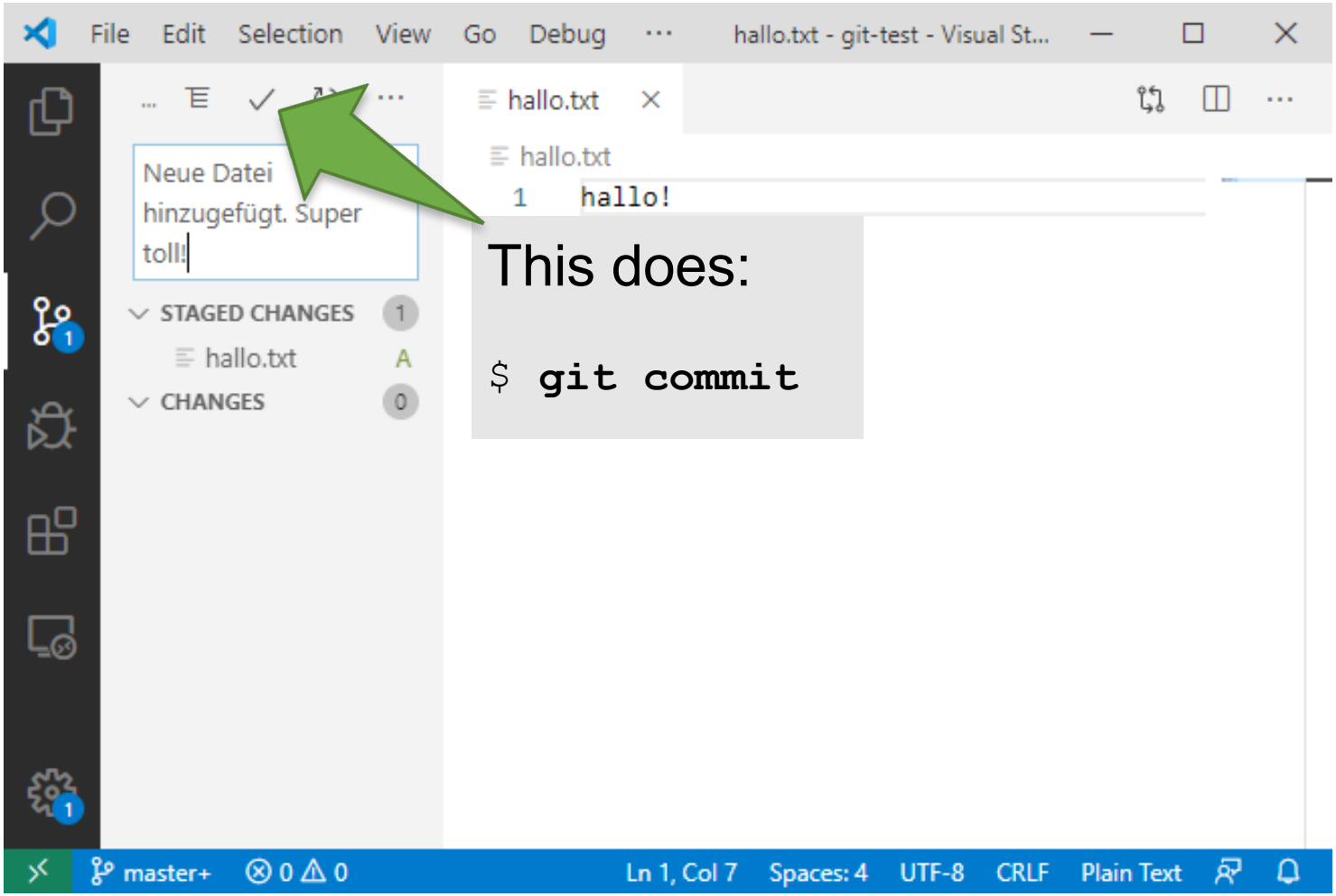
Academy



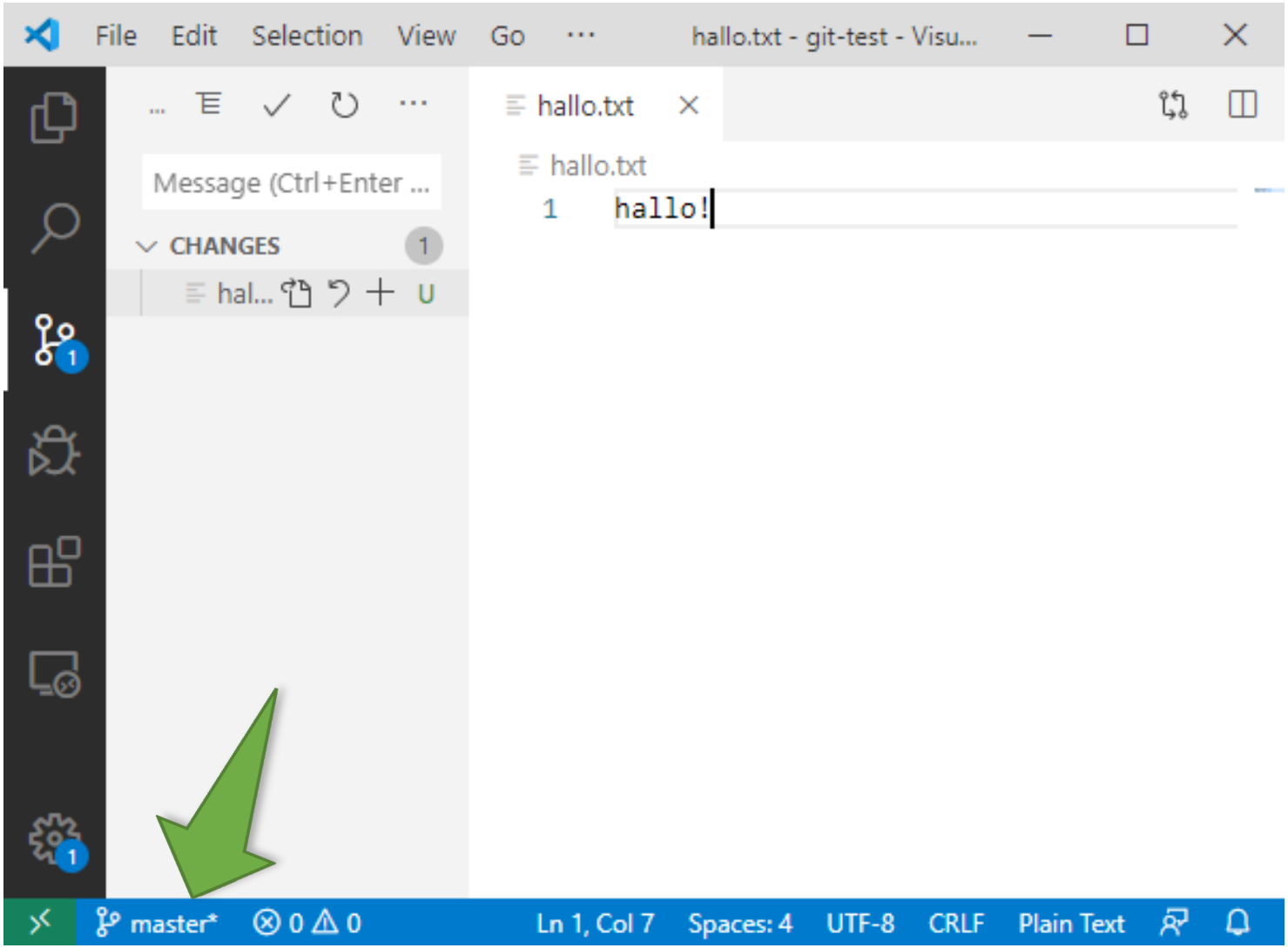
This does:

```
$ git add
```

Academy


File Edit Selection View Go Debug ··· hallo.txt - git-test - Visual St...
hallo.txt
Neue Datei hinzugefügt. Super toll!
STAGED CHANGES 1
hallo.txt A
CHANGES 0
hallo.txt
1 hallo!
This does:
$ git commit
master+ 0 0 Ln 1, Col 7 Spaces: 4 UTF-8 CRLF Plain Text

AW Academy


File
Edit
Selection
View
Go
hallo.txt - git-test - Visu...
hallo.txt
hallo.txt
1 hallo!
Message (Ctrl+Enter ...
CHANGES
1
hal...
U
master*
0 0
Ln 1, Col 7
Spaces: 4
UTF-8
CRLF
Plain Text

AW Academy

Exercise 1 with VSCode

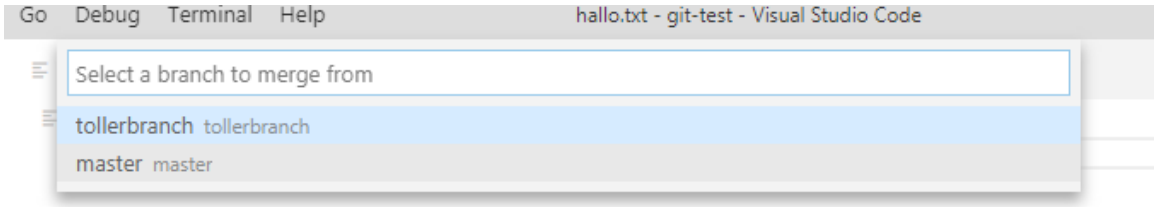
- `status` and `diff` in “Source Control” view
- `log` and `show` are not available in VSCode
  - There is a Plugin
  - After pushing, you can see your commits on GitHub

AW Academy

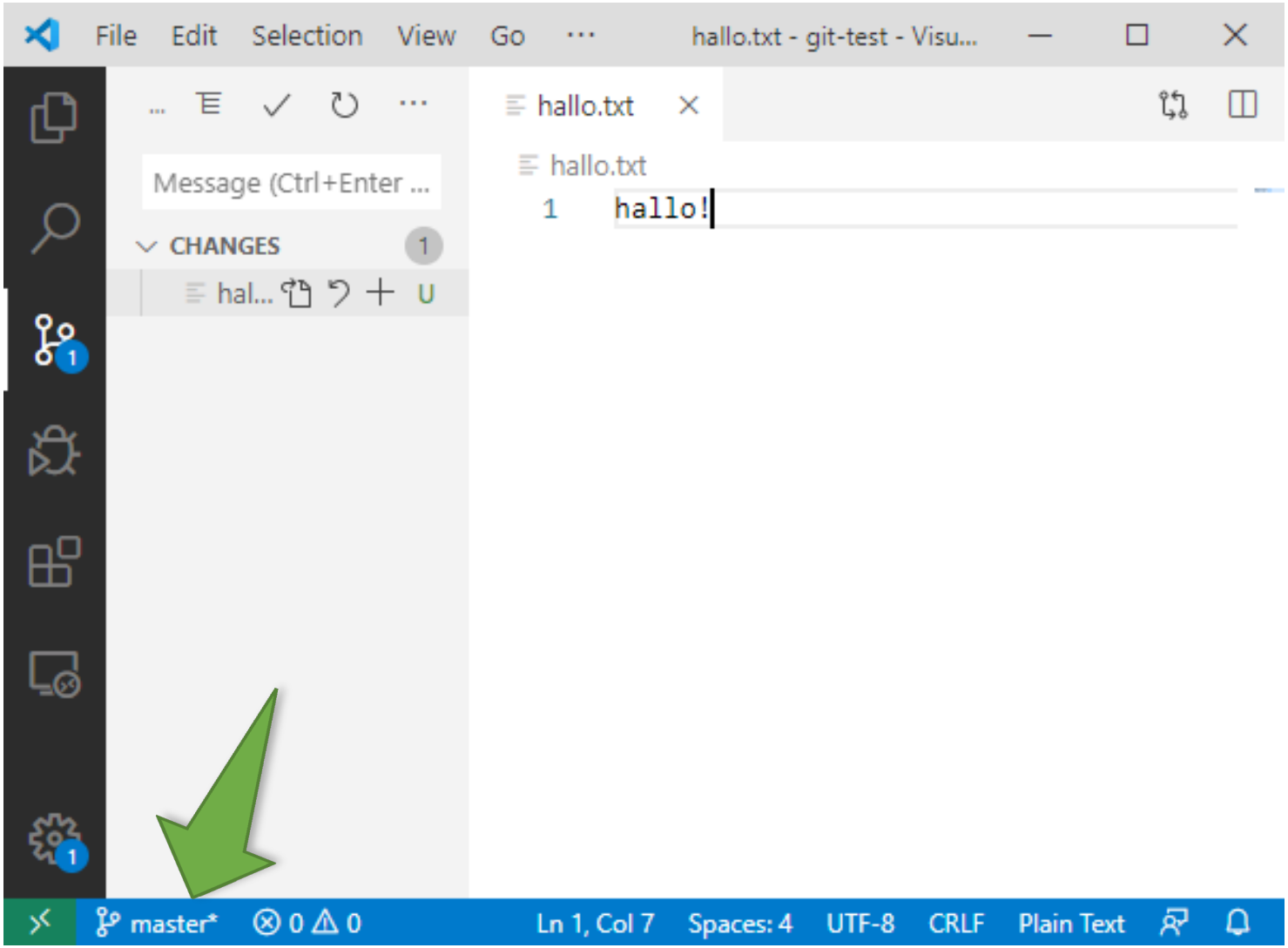
Exercise 2 with VSCode

Merge:

- Open Command Pallete
  - in Menu “View”
  - Or Ctrl+Shift+P
- Select “Git: Merge Branch”

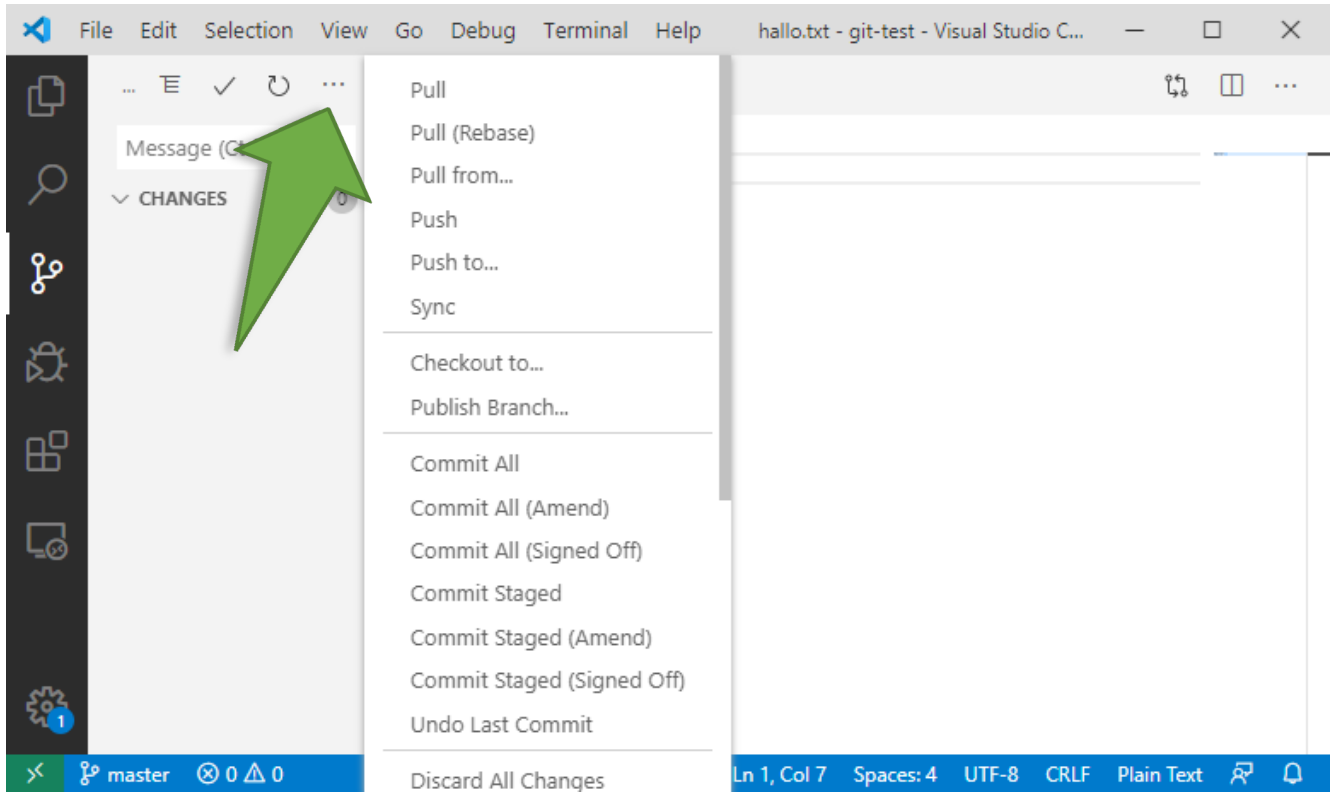


AW Academy


File
Edit
Selection
View
Go
hallo.txt - git-test - Visu...
hallo.txt
hallo.txt
1 hallo!
Message (Ctrl+Enter ...
CHANGES
1
hal...
U
master*
0 0
Ln 1, Col 7
Spaces: 4
UTF-8
CRLF
Plain Text

AW Academy

Exercise 3 with VSCode



AW Academy

Official Documentation

https://code.visualstudio.com/docs/editor/versioncontrol

AW Academy

Bonus: More useful Git-Commands

`git blame`

`git rebase` (Rewrite History)

AW Academy

Bonus: How do I write a useful Commit Message?

AW Academy