



WELCOME

INTRODUCTION TO GIT



Version control is the backbone of modern software development.

Let's dive into Git – a powerful, distributed version control system.

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?






Ctrl	+	Z
Ctrl	+	Z
Ctrl	+	Z
Ctrl	+	Z
Ctrl	+	Z
Ctrl	+	Z
Ctrl	+	Z

What is Git - A Quick Introduction to the Git Version Control...



1 CORE CONCEPTS & HOW GIT WORKS

-  **Version Control:** Tracking changes in source code over time
-  **Distributed System:** Everyone has a full copy of the repository
-  **Snapshots, not diffs:** Git stores a snapshot of the files



GIT'S THREE STATES

1. **Modified** – You've changed a file
2. **Staged** – You've marked it for commit
3. **Committed** – Saved permanently in history



Use `git status` to check the current state!




2 LOCAL REPOSITORIES

- On Windows you might to [install Git](#) first
- Initialize a Git repository:

```
git init
```

- Check current status:

```
git status
```

 The `.git` folder stores the entire version history locally.

3 CORE LOCAL FUNCTIONALITY

+ ADD FILES TO STAGING

Single file:

```
git add <file>
```

All files (new and changed):

```
git add .
```



COMMIT CHANGES

```
git commit -m "Meaningful message"
```

- Every commit knows its predecessor -> the history of all files can be reconstructed
- Changes in files are saved linewise in a commit (visible in `git diff` or `git show`)



OTHER USEFUL COMMANDS

Command

Function

`git log`

List the latest commits

`git diff`

Show changes not on Stage

`git diff --staged`

Show changes on Stage

`git show <commit-id>`

Show the contents of a commit

```
$ git log  
commit e30e7f6a77c2fa3f30135bcc6443201a9ff61206  
Author: Lars Gerhard <mail@largeger.de>  
Date: Thu May 24 13:52:48 2024 +0200  
Meaningful message
```

Every commit gets a unique commit-ID!



THE OPPOSITE OF ADD...

Command

Function

```
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 my-  
file-here
```

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

git commit



git push



git add .





HANDS ON...

Get started with Exercise 1!

4

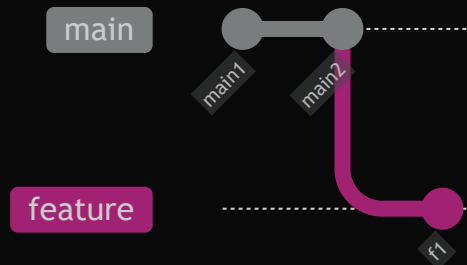
BRANCHING AND MERGING



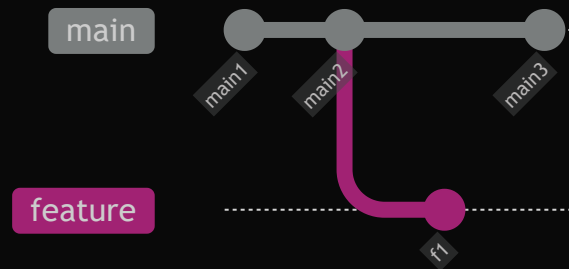
BRANCHING

add-commands are dropped to ease reading!

```
git commit -m "commit on main"
git commit -m "commit on main"
# create a new branch
git branch feature
# switch to the new branch
git checkout feature
git commit -m "commit on feature"
```



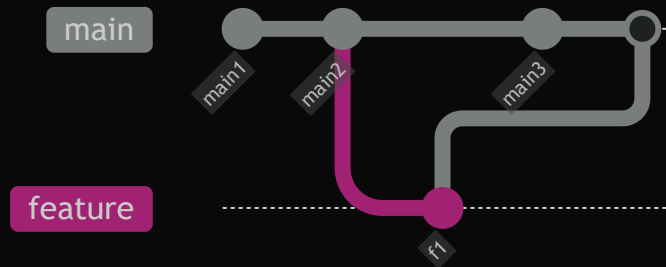

```
# switch back to the main branch  
git checkout main  
git commit -m "commit on main again"
```





MERGING

```
# !make sure to be on the main branch!  
git checkout main  
git merge feature
```



👉 Git tries to auto-merge. Conflicts? You'll fix them manually.

GIT MERGE



- There are complete strategies for branching and merging professionally!
- You should look at the graph frequently!

```
git log -graph -online
```

gitk libgit2

File Edit View Help

Local uncommitted changes, not checked in to index
development — **remotes/origin/development** refspec: git_refspec_parse() does not e
Merge pull request #2208 from libgit2/vmg/mempack
In-memory packing backend
Merge pull request #2226 from libgit2/rb/submodule-sorting-fix
Improve test of submodule name sorting
Cleanups
Fix submodule sorting in workdir iterator
Add faster git_submodule_is_submodule check
Merge pull request #2229 from linquize/Wdeclaration-after-statement
Add CFLAGS -Wdeclaration-after-statement

SHA1 ID: b76b5d34275fe33192358d4eaa1ae98e31efc2a1 Row 6 / 1359

Find commit containing: Search

Diff Old version New version Lines of context: 3 Ignore space change Line diff

Author: Russell Belfer <rb@github.com> 2014-03-31 13:33:11
Committer: Russell Belfer <rb@github.com> 2014-03-31 13:33:11
Parent: 7dc42a55f5f4c61e8e8de472ec54ccc0613e23c (Cleanups)
Child: d67397dd0c82fab82ale6883107c97c4e133a911 (Merge pull request #2226 from libgit2/rb
Branches: development, remotes/origin/development
Follows: v0.20.0
Precedes:

Improve test of submodule name sorting

tests/diff/submodules.c -----
index ead5c71..2881f74 100644
@@ -182,6 +182,8 @@ void test_diff_submodules__submod2_index_to_wd(void)
"UNTRACKED", /* not */
"diff --git a/sm_changed_file b/sm_changed_file\nindex 4800958..4800958 :
"diff --git a/sm_changed_head b/sm_changed_head\nindex 4800958..3d9386c :
+ "UNTRACKED", /* sm_changed_head- */
+ "UNTRACKED", /* sm_changed_head_ */
"diff --git a/sm_changed_index b/sm_changed_index\nindex 4800958..4800958 :
"diff --git a/sm_changed_untracked_file b/sm_changed_untracked_file\nindex 4800958..4800958 :
"diff --git a/sm_missing_commits b/sm_missing_commits\nindex 4800958..5e
@@ -190,6 +192,10 @@ void test_diff_submodules__submod2_index_to_wd(void)

Comments
tests/diff/submodules.c



RESOLVING CONFLICTS

Test.java with changes on main branch

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

Test.java with changes on feature branch

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

Auto-merging? 🤔



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

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

now edit this file manually to get to your favorized
result! 🤪





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

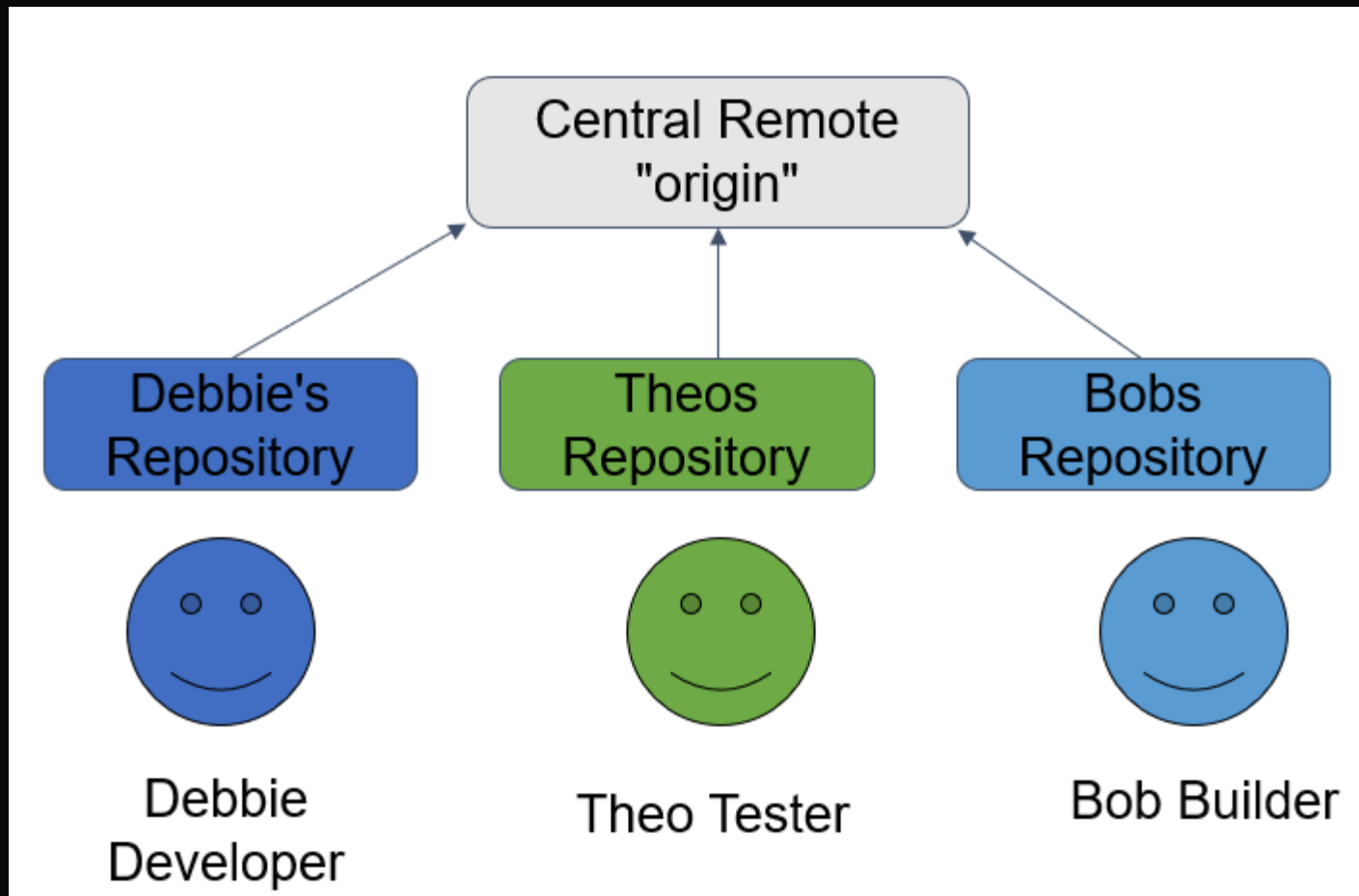

 **HANDS ON...**

Get started with Exercise 2!

5 REMOTE REPOSITORIES

WHAT IS A 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 



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

CREATE A GITHUB ACCOUNT!

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

YOUR GITHUB PROFILE

- It's your Software Developer Business Card
- Publish as much of your work there

CONNECT TO A REMOTE (E.G. GITHUB, GITLAB):

```
# Either:  
git remote add origin https://github.com/user/repo.git  
# Or:  
git remote add origin git@github.com:<user>/<repo>.git
```

PUSH LOCAL COMMITS:

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

Do this only for a new, empty remote repository

```
git push -u origin main
```


PULL REMOTE CHANGES (& MERGE):

```
git pull
```

JOIN THE WORK ON AN EXISTING REPOSITORY

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

WATCH OUT

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

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.
- This is called **Merge Request / Pull Request**.

.GITIGNORE

List of files & folders that shall be ignored by git.

```
out/  
.idea/workspace.xml
```

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



HANDS ON...

Get started with Exercise 3!

6 IDE INTEGRATION

VISUAL STUDIO CODE

- Built-in Git panel
- Stage/commit via GUI
- Extensions: GitLens for history & blame



INTELIJ IDEA

- VCS Integration
- Smart diff tools & conflict resolution
- Git log visualizer



SUMMARY

- Git helps track and manage code changes efficiently
- Work locally, sync remotely
- Use branches to isolate work
- IDEs make Git usage smoother

