

Auf einen Blick

1	Git in zehn Minuten	13
2	Learning by Doing	21
3	Git-Grundlagen	75
4	Datenanalyse im Git-Repository	153
5	GitHub	177
6	GitLab	209
7	Azure DevOps, Bitbucket, Gitea und Gitolite	235
8	Workflows	257
9	Arbeitstechniken	281
10	Git in der Praxis	315
11	Git-Probleme und ihre Lösung	347
12	Kommandoreferenz	369

Inhalt

Vorwort	9
1 Git in zehn Minuten	13
1.1 Was ist Git?	13
1.2 Software von GitHub herunterladen	16
1.3 Programmieren lernen mit Backup und Undo	18
2 Learning by Doing	21
2.1 git-Kommando installieren	21
2.2 GitHub-Account und -Repositorys einrichten	28
2.3 Mit dem Kommando »git« arbeiten	32
2.4 Authentifizierung	45
2.5 Git spielerisch lernen (Githug)	55
2.6 Entwicklungsumgebungen und Editoren	56
2.7 An einem fremden GitHub-Projekt mitarbeiten	70
2.8 Synchronisation und Backups	72
3 Git-Grundlagen	75
3.1 Nomenklatur	75
3.2 Die Git-Datenbank	80
3.3 Commits	84
3.4 Commit-Undo	92
3.5 Branches	100
3.6 Merge	105
3.7 Stashing	113
3.8 Remote Repositorys	115
3.9 Merge-Konflikte lösen	126
3.10 Rebasing	133
3.11 Tags	139
3.12 Referenzen auf Commits	144
3.13 Git-Interna	149

4	Datenanalyse im Git-Repository	153
4.1	Commits durchsuchen (git log)	153
4.2	Dateien durchsuchen	164
4.3	Fehler suchen (git bisect)	169
4.4	Statistik und Visualisierung	171
5	GitHub	177
5.1	Pull-Requests	178
5.2	Actions	183
5.3	Paketmanager (GitHub Packages)	191
5.4	Automatische Sicherheits-Scans	194
5.5	Weitere GitHub-Funktionen	198
5.6	GitHub CLI	204
6	GitLab	209
6.1	On Premises versus Cloud	210
6.2	Installation	211
6.3	Das erste Projekt	218
6.4	Pipelines	220
6.5	Merge-Requests	231
6.6	Web-IDE	233
7	Azure DevOps, Bitbucket, Gitea und Gitolite	235
7.1	Azure DevOps	235
7.2	Bitbucket	240
7.3	Gitea	242
7.4	Gitolite	252
8	Workflows	257
8.1	Anweisungen für das Team	257
8.2	Solo-Entwicklung	258
8.3	Feature-Branches für Teams	260
8.4	Merge/Pull-Requests	267
8.5	Long-Running Branches – Gitflow	271

8.6	Trunk-based Development	276
8.7	Welcher Workflow ist der Richtige?	279
9	Arbeitstechniken	281
9.1	Hooks	281
9.2	Prägnante Commit-Messages	287
9.3	Submodule und Subtrees	294
9.4	Mehr Komfort in Bash und Zsh	304
9.5	Zwei-Faktor-Authentifizierung	307
10	Git in der Praxis	315
10.1	Etckeeper	316
10.2	Dotfiles mit Git verwalten	319
10.3	Zugriff auf Subversion mit git-svn	326
10.4	Von SVN zu Git migrieren	330
10.5	Ein Blog mit Git und Hugo	335
11	Git-Probleme und ihre Lösung	347
11.1	Git-Fehlermeldungen (Ursache und Lösung)	347
11.2	Merge für eine einzelne Datei	354
11.3	Dateien permanent aus Git löschen	355
11.4	Ein Projekt aufteilen	363
11.5	Commits in einen anderen Branch verschieben	364
12	Kommandoreferenz	369
12.1	git-Kommando	369
12.2	Revisionssyntax	401
12.3	git-Konfiguration	402
Index	409