

**Git – kurz & gut**  
O'Reillys Taschenbibliothek

# **DAS INHALTS- VERZEICHNIS**

» Hier geht's  
direkt  
zum Buch

# Inhalt

<b>Einleitung</b> .....	<b>9</b>
<b>1 Grundlegende Konzepte</b> .....	<b>15</b>
Was ist Versionskontrolle, und warum brauche ich sie? .....	15
Die Grundlagen von Git. ....	17
<b>2 Git installieren und konfigurieren</b> .....	<b>27</b>
Installation .....	27
Git konfigurieren .....	28
<b>3 Arbeiten mit Git</b> .....	<b>31</b>
Hilfe finden .....	31
Das Git-Repository .....	32
Dateien ignorieren und von der Versionierung ausschließen .....	39
Commits erstellen .....	41
Änderungen synchronisieren .....	50
Mit Branches arbeiten .....	55
Änderungen zusammenführen .....	59
Änderungen temporär speichern .....	70
Änderungen nachvollziehen und betrachten .....	73
Änderungen rückgängig machen .....	78
<b>4 Git-Onlinedienste</b> .....	<b>87</b>
Die Relevanz von GitHub .....	88
Ein Repository auf GitHub anlegen .....	88
Pull-Requests .....	92
Issues: Projektmanagement und Bug-Reports .....	99
Git-Onlinedienste in Unternehmen .....	102

<b>5</b>	<b>Typische Git-Workflows</b> .....	<b>109</b>
	Trunk-based: ausschließlich im Haupt-Branch arbeiten .....	109
	Git-Feature-Branch-Workflow .....	112
	Gitflow: der Workflow für versionierte Software .....	113
	Durch den Fork-basierten Workflow zu Open-Source-Projekten beitragen.....	115
<b>6</b>	<b>Good Practices</b> .....	<b>119</b>
	Gute Commits.....	119
	Geschichte nur im Notfall neu schreiben .....	124
	Den Überblick über Branches behalten .....	126
	Regelmäßig aufräumen.....	127
	Entwicklungsstränge häufig zusammenführen und Konflikten nicht aus dem Weg gehen.....	128
	Langlebige Feature-Banches vermeiden .....	129
<b>7</b>	<b>Häufige Fehler und Probleme</b> .....	<b>133</b>
	Lokaler Stand und Remote-Stand weichen voneinander ab .....	133
	Detached-HEAD-Fehlermeldung .....	134
	Du hast im falschen Branch gearbeitet .....	135
	Git-Diff zeigt keine Änderungen an .....	136
	Keine Tracking-Information für den aktuellen Branch vorhanden .....	136
	Git weigert sich, unzusammenhängende Historien zusammenzuführen .....	137
<b>8</b>	<b>Fortgeschrittenere Kommandos, Tipps und Tricks</b> .....	<b>141</b>
	Im Reflog merkt sich Git fast alles .....	141
	Kürzel für Git-Befehle erstellen.....	143
	Pre- und Post-Commit-Hooks.....	143
<b>9</b>	<b>Git als Baustein moderner und agiler Softwareentwicklung</b> .....	<b>145</b>
	DevOps-Kultur.....	145
	Kontinuierlich integrieren und ausliefern mit Git.....	146
	Infrastructure as Code .....	149

<b>10 Git unter der Haube</b> .....	<b>151</b>
Git-Objekte und das Datenmodell.....	153
Packdateien.....	158
Der Index.....	159
Referenzen.....	161
Was sich sonst noch im versteckten .git-Ordner befindet.....	163
Branches und Merges.....	164
Fazit.....	169
<b>11 Die wichtigsten Kommandos zum Nachschlagen</b> .....	<b>171</b>
<b>12 Glossar</b> .....	<b>175</b>
<b>Index</b> .....	<b>179</b>