



Versionsverwaltung

Um Änderungen an Dateien nachvollziehbar zu machen, ist *Versionsverwaltung* sinnvoll. Dafür werden Änderungen in ein *Versionsverwaltungssystem*, wie beispielsweise Git oder Subversion (SVN), eingepflegt. Dieses merkt sich die Änderungen und in den meisten Fällen zugehörige Informationen, wie den Autor, den Zeitpunkt der Änderung oder zusätzliche Erklärungen. Mithilfe eines Versionsverwaltungssystems ist es so auch möglich, zwischen verschiedenen Versionen der Dateien hin und her zu wechseln.



Wichtige Begriffe

Begriff	Bedeutung
Versionsverwaltung	Eine Versionsverwaltung ist eine Software, die es ermöglicht, die Änderungen an Dateien nachzuvollziehen und zwischen den einzelnen Versionen der Dateien zu wechseln.
Git	Git ist eine Software für Versionsverwaltung, die gerne für die Versionsverwaltung von Programmcode verwendet wird.
GitHub	GitHub ist eine auf Git basierende Plattform, auf der Dateien verwaltet, ausgetauscht und veröffentlicht werden können.
Repository	Ein Repository, kurz Repo, ist ein Verzeichnis, also eine Sammlung von Dateien, das mit einem Versionsverwaltungssystem verwaltet wird.
Commit	Ein Commit ist eine Buchung in ein Versionsverwaltungssystem. Dabei werden die Änderungen und evtl. eine passende Erklärung abgespeichert.
Terminal	Das Terminal, die Shell, die Konsole oder die Eingabeaufforderung - hinter allem versteckt sich ein Terminal, das es ermöglicht, schriftlich mit dem Computer zu kommunizieren.
Upstream	Als Upstream wird ein entferntes Repository bezeichnet, das mit dem eigenen lokalen Repo synchronisiert wird.

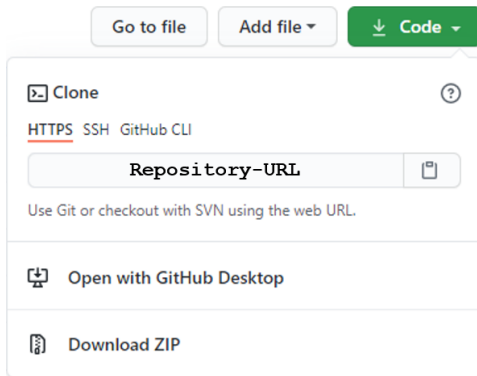


Erklärvideo

Hier findest du eine ausführliche Erklärung rund um die Versionsverwaltung mit Git und GitHub.



Arbeiten mit GitHub



GitHub bietet eine grafische Nutzeroberfläche, mit der man die Repositories verwalten kann.

Über die Schaltfläche *Code* kann man die Repository-URL kopieren oder das ganze Repository als ZIP-Datei herunterladen (*Download ZIP*). Die Repository-URL wird benötigt, wenn man mit dem Terminal arbeiten möchte.

Über die Schaltfläche *Add file* kann man neue Dateien hochladen (*Upload files*). Dabei wird ein neuer Commit erzeugt.

Arbeiten mit dem Terminal

Um Git* im Terminal verwenden zu können, muss Git auf dem PC installiert sein. Das Terminal findet man am besten über die Suchfunktion des PCs.

Code	Beschreibung
<code>cd Verzeichnis</code>	Wechseln in Verzeichnis
<code>cd ..</code>	In der Verzeichnisstruktur eine Ebene nach oben
<code>git init</code>	Aktuelles Verzeichnis als Repository verwenden
<code>git remote add origin R-URL</code>	Entferntes Repository über die Repo-URL als Upstream hinzufügen
<code>git clone R-URL</code>	Repo initialisieren, entferntes Repo klonen und als Upstream verwenden
<code>git status</code>	Aktuellen Status des Repos anzeigen
<code>git add Datei</code>	Eine Datei für den nächsten Commit vormerken
<code>git add .</code>	Alle geänderten Dateien für den nächsten Commit vormerken
<code>git commit -m "Nachricht"</code>	Neuen Commit anlegen, der die vorgemerkten Änderungen und die Nachricht enthält
<code>git pull</code>	Änderungen aus Upstream-Repo in eigenes Repo einpflegen
<code>git push</code>	Eigene Änderungen in Upstream-Repo einpflegen

Laufende Prozesse können über `Strg` + `C` beendet werden. Mit `↑` kann man vorherige Befehle wieder anzeigen.

*<https://git-scm.com/downloads>