

Versionsverwaltung mit Mercurial für Einsteiger

Chemnitzer Linxstage 2009

Stefan Schwarzer, SSchwarzer.com
info@sschwarzer.com

Chemnitz, Germany, 2009-03-14

Überblick

- Nutzen einer Versionsverwaltung
- Begriffe
- Kommandozeilen-Syntax
- Anlegen eines Repositorys
- Anlegen einer und weiterer Versionen
- Anzeigen von Informationen zu Versionen und Dateien
- Umbenennen einer Datei
- Tipps
- Befehlsübersicht
- Zusammenfassung

Überblick

- Nutzen einer Versionsverwaltung
- Begriffe
- Kommandozeilen-Syntax
- Anlegen eines Repositorys
- Anlegen einer und weiterer Versionen
- Anzeigen von Informationen zu Versionen und Dateien
- Umbenennen einer Datei
- Tipps
- Befehlsübersicht
- Zusammenfassung

Nicht in diesem Vortrag: Arbeiten mehrerer Personen mit einem Repository, Zusammenführen von Änderungen

Nutzen einer Versionsverwaltung

Allgemeines

Verschiedene Versionen (= Momentaufnahmen) eines Projekts

- speichern,
- abrufen/wiederherstellen,
- miteinander vergleichen,
- zusammenführen und
- deren Entwicklungsgeschichte verfolgen

Nutzen einer Versionsverwaltung

Anmerkungen

- Versionsverwaltungen sind nicht nur für Softwareprojekte nützlich
- Auch geeignet für andere Projekte (Beispiel: Studienarbeit)

Nutzen einer Versionsverwaltung

Anmerkungen

- Versionsverwaltungen sind nicht nur für Softwareprojekte nützlich
- Auch geeignet für andere Projekte (Beispiel: Studienarbeit)
- **Versionsverwaltungen** (= Versionskontrollsysteme), auch Mercurial, **sind normalerweise für Textdateien gedacht**
- Binärdaten lassen sich zwar speichern und abrufen, aber nicht ohne Weiteres vergleichen oder zusammenführen
- Das spricht aber **nicht** dafür, **bei Binärdateien auf die Versionsverwaltung zu verzichten!**

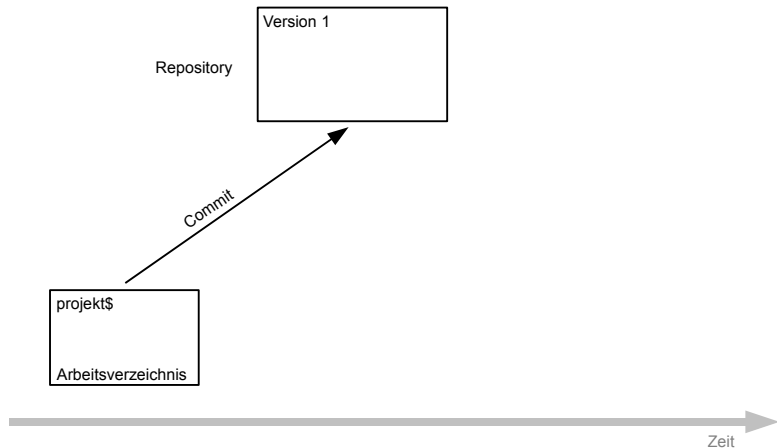
Mercurial

- Verteiltes (= distributed) Versionskontrollsystem
- Open Source
- Erstes Release 2005
- Gut verwendbar für kleine und große Projekte
- Verwendet von Mozilla, Open Solaris, Xine u. v. a.
- Größtenteils in Python geschrieben
- Website: <http://www.selenic.com/mercurial> (englisch)
- Online-Buch: <http://hgbook.red-bean.com> (englisch)
- Deutschsprachige Dokumentation?

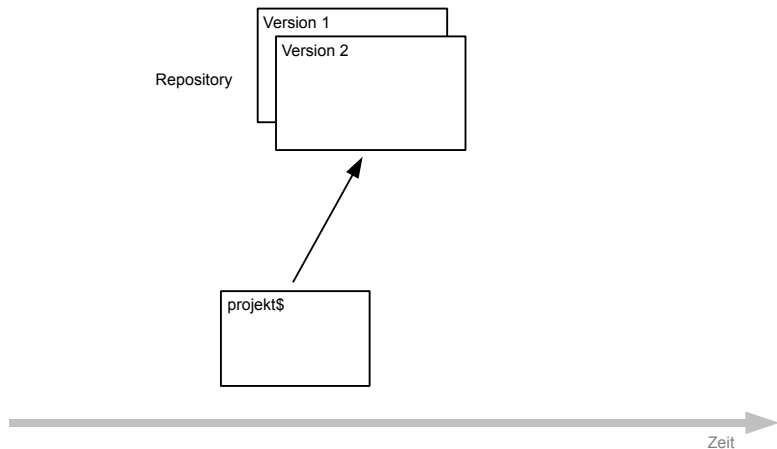
Begriffe

- **Arbeitsverzeichnis** (working directory): das Verzeichnis im Dateisystem, in dem ein Projekt bearbeitet wird
- **Version, Revision** (version, revision): eine Momentaufnahme (Schnappschuss) eines Arbeitsverzeichnisses zu einer bestimmten Zeit; normalerweise **nicht** das Gleiche wie eine Veröffentlichungsversion („Version 1.0“)
- **Repository**: das Verzeichnis, das **alle** Revisionen enthält
- **Commit**: Anlegen einer neuen Version; dabei Eingabe eines Änderungshinweises, der im Repository gespeichert wird; auch synonym für das Ergebnis des Commits (= Version)
- **Versionsgeschichte** (revision log): Auflistung der Änderungshinweise mehrerer Versionen
- **Changeset, Diff, Patch**: Unterschiede zwischen zwei Versionen

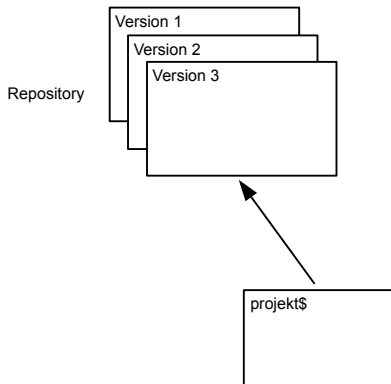
Anlegen von Versionen im Repository



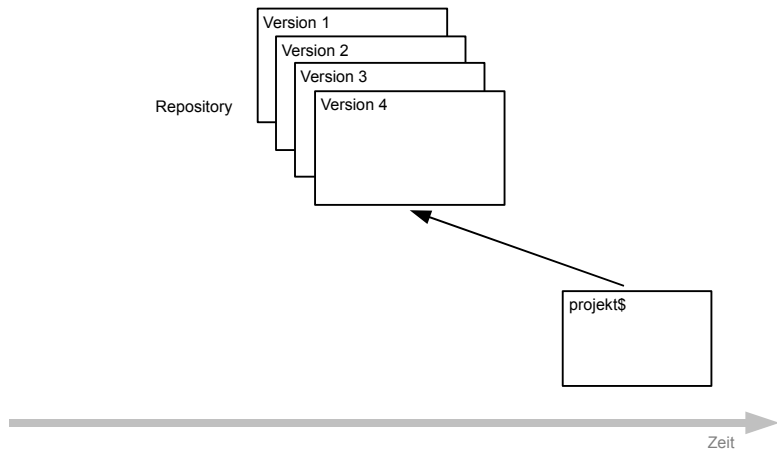
Anlegen von Versionen im Repository



Anlegen von Versionen im Repository



Anlegen von Versionen im Repository



Allgemeines zur Kommandozeilen-Syntax

- hg [globale_optionen] befehl [optionen] [liste]

Allgemeines zur Kommandozeilen-Syntax

- hg [**globale_optionen**] befehl [optionen] [liste]
- **globale_optionen** sind „übergreifende“ Optionen
- Am wichtigsten davon ist wohl `-v` (verbose)

Allgemeines zur Kommandozeilen-Syntax

- `hg [globale_optionen] befehl [optionen] [liste]`
- `globale_optionen` sind „übergreifende“ Optionen
- Am wichtigsten davon ist wohl `-v` (verbose)
- `befehl` ist ein Befehl wie `add`, `commit` oder `diff`
- Viele Befehle haben Abkürzungen, zum Beispiel `di` für `diff`

Allgemeines zur Kommandozeilen-Syntax

- `hg [globale_optionen] befehl [optionen] [liste]`
- `globale_optionen` sind „übergreifende“ Optionen
- Am wichtigsten davon ist wohl `-v` (verbose)
- `befehl` ist ein Befehl wie `add`, `commit` oder `diff`
- Viele Befehle haben Abkürzungen, zum Beispiel `di` für `diff`
- `optionen` sind befehlspezifische Optionen wie `-r` bei `diff` oder `log`

Allgemeines zur Kommandozeilen-Syntax

- `hg [globale_optionen] befehl [optionen] [liste]`
- `globale_optionen` sind „übergreifende“ Optionen
- Am wichtigsten davon ist wohl `-v` (verbose)
- `befehl` ist ein Befehl wie `add`, `commit` oder `diff`
- Viele Befehle haben Abkürzungen, zum Beispiel `di` für `diff`
- `optionen` sind befehlspezifische Optionen wie `-r` bei `diff` oder `log`
- `liste` ist eine Aufzählung von Dateien und/oder Ordnern (durch Leerzeichen getrennt)
- Fehlt `liste`, bezieht sich der Befehl aufs gesamte Arbeitsverzeichnis (anders als bspw. bei Subversion)

Allgemeines zur Kommandozeilen-Syntax

- `hg [globale_optionen] befehl [optionen] [liste]`
- `globale_optionen` sind „übergreifende“ Optionen
- Am wichtigsten davon ist wohl `-v` (verbose)
- `befehl` ist ein Befehl wie `add`, `commit` oder `diff`
- Viele Befehle haben Abkürzungen, zum Beispiel `di` für `diff`
- `optionen` sind befehlspezifische Optionen wie `-r` bei `diff` oder `log`
- `liste` ist eine Aufzählung von Dateien und/oder Ordnern (durch Leerzeichen getrennt)
- Fehlt `liste`, bezieht sich der Befehl aufs gesamte Arbeitsverzeichnis (anders als bspw. bei Subversion)

Ausführliche Informationen liefert der Befehl `help`!

Anlegen eines Repositorys

```
projekt$ █
```

Anlegen eines Repositorys

```
projekt$ ls -a
```

Anlegen eines Repositorys

```
projekt$ ls -a
```

```
. ..
```

```
projekt$ █
```

Anlegen eines Repositorys

```
projekt$ ls -a
```

```
. ..
```

```
projekt$ hg init █
```

Anlegen eines Repositorys

```
projekt$ ls -a
.  ..
projekt$ hg init
projekt$ █
```

Anlegen eines Repositorys

```
projekt$ ls -a  
.  ..  
projekt$ hg init  
projekt$ ls -a
```

Anlegen eines Repositorys

```
projekt$ ls -a
.  ..
projekt$ hg init
projekt$ ls -a
.  ..  .hg
projekt$ █
```

Anlegen eines Repositorys

```
projekt$ ls -a
```

```
. ..
```

```
projekt$ hg init
```

```
projekt$ ls -a
```

```
. .. .hg
```

```
projekt$ ls -a .hg
```

Anlegen eines Repositorys

```
projekt$ ls -a
.  ..
projekt$ hg init
projekt$ ls -a
.  ..  .hg
projekt$ ls -a .hg
.  ..  00changelog.i  requires  store
projekt$ █
```

Anlegen einer Version

```
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
```

Anlegen einer Version

```
projekt$ export EDITOR=vim  
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim  
projekt$ vim mercurial.tex
```


Anlegen einer Version

```
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
~
~
~
~
~
~
~
~
~
~
```

```
mercurial.tex
```

"mercurial.tex" 5 lines, 57 characters written

Anlegen einer Version

```
projekt$ export EDITOR=vim  
projekt$ vim mercurial.tex  
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim  
projekt$ vim mercurial.tex  
projekt$ hg status
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
A mercurial.tex
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
A mercurial.tex
projekt$ hg commit mercurial.tex
```


Anlegen einer Version

Grundgerüst für Vortragsdatei angelegt.

```
HG: Enter commit message.
```

```
HG: Lines beginning with 'HG:' are removed.
```

```
HG: --
```

```
HG: user: Stefan Schwarzer <sschwarzer@sschwarzer.net>
```

```
HG: branch 'default'
```

```
HG: changed mercurial.tex
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
/tmp/hg-editor-6tF_v.txt
```

```
"/tmp/hg-editor-6tF_v.txt" 6 lines, 221 characters written
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
A mercurial.tex
projekt$ hg commit mercurial.tex
projekt$ █
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
A mercurial.tex
projekt$ hg commit mercurial.tex
projekt$ hg status█
```

Anlegen einer Version

```
projekt$ export EDITOR=vim
projekt$ vim mercurial.tex
projekt$ hg status
? mercurial.tex
projekt$ hg add mercurial.tex
projekt$ hg status
A mercurial.tex
projekt$ hg commit mercurial.tex
projekt$ hg status
projekt$ █
```

Anlegen einer weiteren Version

```
projekt$ █
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
```

Anlegen einer weiteren Version

```
\documentclass{beamer}

\begin{document}
\begin{frame}
  \frametitle{Überblick}
  ...
\end{frame}
\end{document}
~
~
~
~
~
~
~
```

```
mercurial.tex
```

"mercurial.tex" 8 lines, 118 characters written

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex  
projekt$ █
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex  
projekt$ hg status
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
projekt$ hg status
M mercurial.tex
projekt$ █
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
projekt$ hg status
M mercurial.tex
projekt$ hg commit -m "Erste Seite (Überblick) ergänzt." ■
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
projekt$ hg status
M mercurial.tex
projekt$ hg commit -m "Erste Seite (Überblick) ergänzt."
projekt$ █
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
projekt$ hg status
M mercurial.tex
projekt$ hg commit -m "Erste Seite (Überblick) ergänzt."
projekt$ hg status
```

Anlegen einer weiteren Version

```
projekt$ vim mercurial.tex
projekt$ hg status
M mercurial.tex
projekt$ hg commit -m "Erste Seite (Überblick) ergänzt."
projekt$ hg status
projekt$ █
```

Versionsgeschichte anzeigen

```
projekt$ █
```

Versionsgeschichte anzeigen

```
projekt$ hg log
```

Versionsgeschichte anzeigen

```
projekt$ hg log
changeset: 1:084665da1b92
tag:       tip
user:      Stefan Schwarzer <sschwarzer@sschwarzer.net>
date:      Sun Mar 08 18:38:42 2009 +0100
summary:   Erste Seite (Überblick) ergänzt.

changeset: 0:dd5b4c70cf0b
user:      Stefan Schwarzer <sschwarzer@sschwarzer.net>
date:      Sun Mar 01 20:22:31 2009 +0100
summary:   Grundgerüst für Vortragsdatei angelegt.

projekt$ █
```

Versionsgeschichte anzeigen

```
projekt$ hg log
changeset:  1:084665da1b92
tag:        tip
user:       Stefan Schwarzer <sschwarzer@sschwarzer.net>
date:       Sun Mar 08 18:38:42 2009 +0100
summary:    Erste Seite (Überblick) ergänzt.

changeset:  0:dd5b4c70cf0b
user:       Stefan Schwarzer <sschwarzer@sschwarzer.net>
date:       Sun Mar 01 20:22:31 2009 +0100
summary:    Grundgerüst für Vortragsdatei angelegt.

projekt$ █
```

Unterschiede zwischen zwei Versionen anzeigen

```
projekt$ █
```

Unterschiede zwischen zwei Versionen anzeigen

```
projekt$ hg diff -r 0:1 mercurial.tex
```

Unterschiede zwischen zwei Versionen anzeigen

```
projekt$ hg diff -r 0:1 mercurial.tex
diff a/mercurial.tex b/mercurial.tex
--- a/mercurial.tex
+++ b/mercurial.tex
@@ -1,5 +1,8 @@
 \documentclass{beamer}

 \begin{document}
-% Hier kommen später die "Folien" hin
+\begin{frame}
+  \frametitle{Überblick}
+  ...
+\end{frame}
 \end{document}
projekt$ █
```

Unterschiede zwischen zwei Versionen anzeigen

```
projekt$ hg diff -r 0:1 mercurial.tex
diff a/mercurial.tex b/mercurial.tex
--- a/mercurial.tex
+++ b/mercurial.tex
@@ -1,5 +1,8 @@
 \documentclass{beamer}

 \begin{document}
-% Hier kommen später die "Folien" hin
+\begin{frame}
+  \frametitle{Überblick}
+  ...
+\end{frame}
 \end{document}
projekt$ █
```

Versionsgeschichte innerhalb einer Datei

```
projekt$ █
```

Versionsgeschichte innerhalb einer Datei

```
projekt$ hg annotate mercurial.tex
```

Versionsgeschichte innerhalb einer Datei

```
projekt$ hg annotate mercurial.tex
0: \documentclass{beamer}
0:
0: \begin{document}
1: \begin{frame}
1:   \frametitle{Überblick}
1:   ...
1: \end{frame}
0: \end{document}
projekt$ █
```

Datei aus einer bestimmten Version anzeigen

```
projekt$ █
```

Datei aus einer bestimmten Version anzeigen

```
projekt$ hg cat -r 0 mercurial.tex
```

Datei aus einer bestimmten Version anzeigen

```
projekt$ hg cat -r 0 mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ █
```

Umschalten auf eine bestimmte Version

```
projekt$ █
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0  
1 files updated, 0 files merged, 0 files removed, 0 files unreso  
projekt$ █
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0  
1 files updated, 0 files merged, 0 files removed, 0 files unreso  
projekt$ cat mercurial.tex
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ █
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ hg update
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ hg update
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ █
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ hg update
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
```

Umschalten auf eine bestimmte Version

```
projekt$ hg update -r 0
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
% Hier kommen später die "Folien" hin
\end{document}
projekt$ hg update
1 files updated, 0 files merged, 0 files removed, 0 files unreso
projekt$ cat mercurial.tex
\documentclass{beamer}

\begin{document}
\begin{frame}
...
```

Umbenennen einer Datei

```
projekt$ █
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex  
projekt$ █
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex  
projekt$ hg status
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
projekt$ hg status
A mercurial_vortrag.tex
R mercurial.tex
projekt$ █
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
projekt$ hg status
A mercurial_vortrag.tex
R mercurial.tex
projekt$ hg commit
```

Umbenennen einer Datei

```
HG: Enter commit message.  
HG: Lines beginning with 'HG:' are removed.  
HG: --  
HG: user: Stefan Schwarzer <sschwarzer@sschwarzer.net>  
HG: branch 'default'  
HG: changed mercurial_vortrag.tex  
HG: removed mercurial.tex  
~  
~  
~  
~  
~  
~  
~
```

```
/tmp/hg-editor-Wks_B.txt
```

```
"/tmp/hg-editor-Wks_B.txt"
```

Umbenennen einer Datei

```
mercurial.tex in mercurial_vortrag.tex umbenannt.
```

```
HG: Enter commit message.
```

```
HG: Lines beginning with 'HG:' are removed.
```

```
HG: --
```

```
HG: user: Stefan Schwarzer <sschwarzer@sschwarzer.net>
```

```
HG: branch 'default'
```

```
HG: changed mercurial_vortrag.tex
```

```
HG: removed mercurial.tex
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
/tmp/hg-editor-Wks_B.txt
```

```
"/tmp/hg-editor-Wks_B.txt" 7 lines, 260 characters written
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
projekt$ hg status
A mercurial_vortrag.tex
R mercurial.tex
projekt$ hg commit
projekt$ █
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
projekt$ hg status
A mercurial_vortrag.tex
R mercurial.tex
projekt$ hg commit
projekt$ hg log -f mercurial_vortrag.tex
```

Umbenennen einer Datei

```
projekt$ hg move mercurial.tex mercurial_vortrag.tex
projekt$ hg status
A mercurial_vortrag.tex
R mercurial.tex
projekt$ hg commit
projekt$ hg log -f mercurial_vortrag.tex
...
summary: mercurial.tex nach mercurial_vortrag.tex umbenannt.
...
summary: Erste Seite (Überblick) ergänzt.
...
summary: Grundgerüst für Vortragsdatei angelegt.
projekt$ █
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

projekt\$ █

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex  
projekt$ █
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex  
projekt$ hg diff
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ █
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ hg commit # falls alles ok
```

Tipp: hg diff vor hg commit

Zwischen dem Vornehmen einer Änderung und dem Anlegen einer neuen Version mit `hg commit` empfiehlt es sich, die Änderungen mit `hg diff` zu prüfen

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ hg commit # falls alles ok
projekt$ █
```

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

```
projekt$ █
```

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

```
projekt$ vim mercurial.tex
```

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

```
projekt$ vim mercurial.tex  
projekt$ █
```

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

```
projekt$ vim mercurial.tex  
projekt$ hg diff
```

Tipp: Änderungen mit `hg revert` rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit `hg revert`:

```
projekt$ vim mercurial.tex
```

```
projekt$ hg diff
```

```
...
```

```
projekt$ █
```

Tipp: Änderungen mit hg revert rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit hg revert:

```
projekt$ vim mercurial.tex
```

```
projekt$ hg diff
```

```
...
```

```
projekt$ hg revert mercurial.tex # falls nicht ok
```

Tipp: Änderungen mit hg revert rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit hg revert:

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ hg revert mercurial.tex # falls nicht ok
projekt$ █
```

Tipp: Änderungen mit hg revert rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit hg revert:

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ hg revert mercurial.tex # falls nicht ok
projekt$ hg status
```

Tipp: Änderungen mit hg revert rückgängig machen

Falls die Änderungen an einer oder mehreren Dateien verworfen werden sollen, geht das mit hg revert:

```
projekt$ vim mercurial.tex
projekt$ hg diff
...
projekt$ hg revert mercurial.tex # falls nicht ok
projekt$ hg status
projekt$ █
```

Übersicht wichtiger Befehle

Befehl	Kürzel	Bedeutung
<code>init</code>		Repository anlegen (initialisieren)

Übersicht wichtiger Befehle

Befehl	Kürzel	Bedeutung
<code>init</code>		Repository anlegen (initialisieren)
<code>add</code>		Verzeichnis/Datei unter Versionsverwaltung stellen
<code>remove</code>	<code>rm</code>	Verzeichnis/Datei nicht mehr versionieren
<code>move</code>	<code>mv</code>	Verzeichnis/Datei einschl. Versionsgeschichte verschieben/umbenennen
<code>copy</code>	<code>cp</code>	Verzeichnis/Datei einschl. Versionsgeschichte kopieren

Übersicht wichtiger Befehle

Befehl	Kürzel	Bedeutung
<code>init</code>		Repository anlegen (initialisieren)
<code>add</code>		Verzeichnis/Datei unter Versionsverwaltung stellen
<code>remove</code>	<code>rm</code>	Verzeichnis/Datei nicht mehr versionieren
<code>move</code>	<code>mv</code>	Verzeichnis/Datei einschl. Versionsgeschichte verschieben/umbenennen
<code>copy</code>	<code>cp</code>	Verzeichnis/Datei einschl. Versionsgeschichte kopieren
<code>commit</code>	<code>ci</code>	Neue Version mit aktuellen Änderungen erzeugen
<code>update</code>	<code>up</code>	Version des Arbeitsverzeichnisses ändern

Übersicht wichtiger Befehle

Befehl	Kürzel	Bedeutung
<code>init</code>		Repository anlegen (initialisieren)
<code>add</code>		Verzeichnis/Datei unter Versionsverwaltung stellen
<code>remove</code>	<code>rm</code>	Verzeichnis/Datei nicht mehr versionieren
<code>move</code>	<code>mv</code>	Verzeichnis/Datei einschl. Versionsgeschichte verschieben/umbenennen
<code>copy</code>	<code>cp</code>	Verzeichnis/Datei einschl. Versionsgeschichte kopieren
<code>commit</code>	<code>ci</code>	Neue Version mit aktuellen Änderungen erzeugen
<code>update</code>	<code>up</code>	Version des Arbeitsverzeichnisses ändern
<code>status</code>	<code>st</code>	Status der unversionierten Änderungen zeigen
<code>log</code>		Versionsgeschichte anzeigen
<code>diff</code>	<code>di</code>	Unterschiede zweier Versionen oder einer Version und des Arbeitsverzeichnisses zeigen
<code>cat</code>		Datei in einer bestimmten Version zeigen
<code>annotate</code>	<code>ann</code>	Datei mit Versionen anzeigen

Übersicht wichtiger Befehle

Befehl	Kürzel	Bedeutung
init		Repository anlegen (initialisieren)
add		Verzeichnis/Datei unter Versionsverwaltung stellen
remove	rm	Verzeichnis/Datei nicht mehr versionieren
move	mv	Verzeichnis/Datei einschl. Versionsgeschichte verschieben/umbenennen
copy	cp	Verzeichnis/Datei einschl. Versionsgeschichte kopieren
commit	ci	Neue Version mit aktuellen Änderungen erzeugen
update	up	Version des Arbeitsverzeichnisses ändern
status	st	Status der unversionierten Änderungen zeigen
log		Versionsgeschichte anzeigen
diff	di	Unterschiede zweier Versionen oder einer Version und des Arbeitsverzeichnisses zeigen
cat		Datei in einer bestimmten Version zeigen
annotate	ann	Datei mit Versionen anzeigen
help		Hilfe zu einem Befehl zeigen

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts
- `hg init` legt ein Repository an; `hg add` und implizit `hg copy` fügen Verzeichnisse oder Dateien hinzu, `hg rm` entfernt sie

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts
- `hg init` legt ein Repository an; `hg add` und implizit `hg copy` fügen Verzeichnisse oder Dateien hinzu, `hg rm` entfernt sie
- `hg status` zeigt noch nicht versionierte Änderungen an

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts
- `hg init` legt ein Repository an; `hg add` und implizit `hg copy` fügen Verzeichnisse oder Dateien hinzu, `hg rm` entfernt sie
- `hg status` zeigt noch nicht versionierte Änderungen an
- Eine neue Version wird jeweils mit `hg commit` erstellt; dies lässt sich durch Angabe der betroffenen Verzeichnisse/Dateien auf einen Teil des Arbeitsverzeichnisses beschränken

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts
- `hg init` legt ein Repository an; `hg add` und implizit `hg copy` fügen Verzeichnisse oder Dateien hinzu, `hg rm` entfernt sie
- `hg status` zeigt noch nicht versionierte Änderungen an
- Eine neue Version wird jeweils mit `hg commit` erstellt; dies lässt sich durch Angabe der betroffenen Verzeichnisse/Dateien auf einen Teil des Arbeitsverzeichnisses beschränken
- `hg log` bzw. `hg diff` zeigen die Versionsgeschichte als Ganzes bzw. Unterschiede zwischen Versionen an

Zusammenfassung

- Ein Versionskontrollsystem erlaubt das Speichern und Abrufen von Momentaufnahmen und Versionsunterschieden eines Projekts
- `hg init` legt ein Repository an; `hg add` und implizit `hg copy` fügen Verzeichnisse oder Dateien hinzu, `hg rm` entfernt sie
- `hg status` zeigt noch nicht versionierte Änderungen an
- Eine neue Version wird jeweils mit `hg commit` erstellt; dies lässt sich durch Angabe der betroffenen Verzeichnisse/Dateien auf einen Teil des Arbeitsverzeichnisses beschränken
- `hg log` bzw. `hg diff` zeigen die Versionsgeschichte als Ganzes bzw. Unterschiede zwischen Versionen an
- `hg update` erlaubt das Einstellen des Arbeitsverzeichnisses auf eine bestimmte Version

Danke für Ihre Aufmerksamkeit! :-)

Fragen?

Danke für Ihre Aufmerksamkeit! :-)

Fragen?

Anmerkungen?

Danke für Ihre Aufmerksamkeit! :-)

Fragen?

Anmerkungen?

Diskussion?