

Git

Distributed Version Control System

Besonderheiten

- Dezentralität / Verteilte Repositories:
Kein zentraler Entwicklungsserver
- Nicht-lineare Entwicklung
- Dateisystem mit Versionierungsfunktionen
- Git != Subversion + Magic

Workflows

Konfiguration

```
$ git-config --global user.name „Dennis Bloete“
```

```
$ git-config --global user.email mail@dennisbloete.de
```

```
$ git-config --global color.branch auto
```

```
$ git-config --global color.diff auto
```

```
$ git-config --global color.status auto
```

```
$ git-config --list    # Konfiguration anzeigen
```

Neues Repository

```
$ rails my_new_app
```

```
$ git init          # Repository anlegen
```

```
$ mate .gitignore   # ignorierte Dateien setzen
```

```
$ git status        # Status der Working Copy Inhalte
```

```
$ git add .          # alle Dateien zum Commit hinzufügen
```

```
$ git commit -m „My first commit“
```

Branching & Merging

<code>\$ git checkout -b my_branch</code>	<code># Branch anlegen</code>
<code>-- Änderungen --</code>	
<code>\$ git rebase master</code>	<code># Update des Branches</code>
<code>\$ git checkout master</code>	<code># Wechsel in master-Branch</code>
<code>\$ git diff master my_branch</code>	<code># Diff anzeigen</code>
<code>\$ git merge my_branch</code>	<code># Merge mit Branch</code>
<code>\$ git branch -d my_branch</code>	<code># Branch löschen</code>

Tags

```
$ git tag v1.0.1
```

```
# Tag anlegen
```

```
$ git tag -d v1.0.1
```

```
# Tag löschen
```

Verteilte Entwicklung (I)

```
$ git clone git://github.com/my/project.git
```

-- Änderungen --

```
$ git commit -a -m „My commit“      # alles (-a) committen
```

```
$ git push                          # Commits uploaden
```

```
$ git branch -a -v --color           # alle Branches anzeigen
```

```
$ git fetch    # Objekte und Refs downloaden
```

```
$ git pull     # wie fetch, aber mit anschließendem merge
```


Verteilte Entwicklung (2)

```
$ git clone git://github.com/my/project.git
```

-- Repository wird als „origin“ (default) hinzugefügt --

```
$ git push    # Upload zu origin, weil default
```

```
$ git pull    # Merge mit origin, weil default
```

```
$ git remote add laptop git://dbloete.local:repos/app.git
```

-- Repository wird als „laptop“ hinzugefügt --

```
$ git push laptop    # Upload zu laptop
```

```
$ git pull laptop    # Merge mit laptop
```

```
$ git remote -v      # Remote-Repositories anzeigen
```

Pros & Cons

- Einfaches Branchen und Mergen
- Verteilte Repositories (als Option)
- Im Vergleich sehr effizient und schnell
- Ignorieren von Dateien und externe Quellen
- SVN weiterhin mittels git-svn nutzbar
- Aktuell noch fehlende IDE-Unterstützungen
- Teilweise kryptische Befehle