

# **GIT Anleitung**

Author: Klerdy

19. März 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Konfiguration</b>	<b>2</b>
<b>3</b>	<b>Commands</b>	<b>3</b>
3.1	Projekte erzeugen . . . . .	3
3.2	Typischer Workflow . . . . .	3

# 1 Einleitung

Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die durch Linus Torvalds initiiert wurde. Bevor die technischen Konzepte detailliert beschrieben werden, muss eine Konfiguration auf dem jeweiligen Computer erfolgen damit Uploads und Änderungen gezielt einer Person zugeordnet werden können.

## 2 Konfiguration

Zu Beginn sollte sichergestellt werden, dass die installierte GIT Version möglichst aktuell ist.

```
git version (1)
```

```
git version 2.8.4.windows.1
```

Danach sollte man den Namen und die Email Adresse angeben.

```
git config --global user.name "Mister Nobody"  
git config --global user.email "Mister-Nobody@test.de" (2)
```

Im nächsten Kapitel werden alle wichtigen Befehle für die Versionierung aufgeführt!

## 3 Commands

### 3.1 Projekte erzeugen

GIT Projekt erstellen

```
git init [Projektname] (3)
```

Github Projekt kopieren

```
git clone github-url (4)
```

### 3.2 Typischer Workflow

Status abfragen

```
git status (5)
```

neue Datei erzeugen

```
echo "Test Git Quick Start demo"» start.txt (6)
```

Datei für die Versionierung vorbereiten

```
git add start.txt (7)
```

Neue Version speichern (lokal)

```
git commit -m "Adding start text file" (8)
```

Lokales Repository auf Externes überspielen

```
git push origin master (9)
```

Bei dem letzten Befehl ist es sehr wichtig zu klären, was es denn mit dem Befehl *origin* und *master* auf sich hat.

“origin“ ist der Name des remote repositories (entfernte Ablage) wo die Commits veröffentlicht werden. Konventionell wird hierbei ein remote repository standardmäßig als “origin“ bezeichnet. Man kann aber auch mit anderen remotes arbeiten, die über andere Namen verfügen.