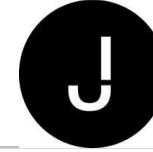




**JAVACREAM**

*Training  
Consulting  
Projectmanagement*

# GIT



- Name und Rolle im Unternehmen
- Themenbezogene Vorkenntnisse
- Aktuelle Problemstellung
- Individuelle Zielsetzung

# Einführung

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, ...
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, ...)
- Team-Zusammenarbeit

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, ...
  - Konsole
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, ...)
- Team-Zusammenarbeit

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, ...
  - Konsole, Integration ins Betriebssystem, Integration in Editoren und IDEs
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, ...)
- Team-Zusammenarbeit



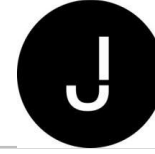
- Desktop
  - TortoiseGit
  - SourceTree
  - ...
- Plugins für Editoren
  - Eclipse
  - Visual Studio
  - Visual Studio Code
  - ...



- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, ...
  - Konsole, Integration ins Betriebssystem, Integration in Editoren und IDEs, Web Console
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, ...)
- Team-Zusammenarbeit



- GitHub
  - Server laufen in der Microsoft-Cloud
  - Öffentliche Ablage ist kostenlos, private Bereiche Lizenz-pflichtig
- BitBucket
  - Atlassian
  - Cloud-Service + Betrieb auf eigenen Servern
- GitLab
  - gitlab.com
  - Cloud-Service + Betrieb auf eigenen Servern
- Azure DevOps
  - Microsoft-Cloud



## First Contact

- Terminal-Fenster mit Git-Unterstützung
- Das Kommando “git” steht hierin zur Verfügung
  - Das ist **kein simples Command Line Interface**, das mit einem Git-Server kommuniziert
    - Es gibt keinen laufenden Git-Server-Prozess, kein Dämon, ...
  - `git --version`
  -

```
rainer@rainer-Aspire-VN7-572G:~$ git --version  
git version 2.32.0
```

- `git config <scope> <key-hierarchie> <value>`
  - `<scope>`
    - local
    - global (User-Profile)
    - system
  - `<key-hierarchie>`
    - “.” trennt die Hierarchie-Ebenen
  - `<value>`
    - irgendwas, Leerzeichen etc. aber in Anführungszeichen setzen
- `git config --global user.name “Rainer Sawitzki”`
- `git config --global user.email training@rainer-sawitzki.de`
- Auslesen `git config --get user.name`

# Einrichten eines Git-Projektverzeichnis

mkdir first  
cd first

git init

git status  
Fehlerfrei

Normales Verzeichnis -> Git-Projektverzeichnis

Git Workspace

Git Repository

Hinweis:  
In der Praxis entspricht diese Sequenz  
einem  
git clone server-repo first