

Arbeiten mit Git und co.

15. Lazarus-Konferenz

Jan Martin Reckel

23. September 2023

Metainformationen

Präsentation für den Vortrag „Arbeiten mit Git und Co.“ auf der
15. Lazaruskonferenz.

Kompilierdatum: 23. September 2023 13:55:49

Gliederung

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

- Was ist Git?
- Persönliche Gründe
- Aufgaben einer Versionsverwaltung
- Begriffsklärung
- Typischer Workflow

Arbeitsschritte

- Installation
- Ein Git-Repository erstellen
- Ein Git-Repository clonen
- Der Commit-Workflow
- Änderungen stagen
- Änderungen committen
- Pullen und Pushen
- Branches

Git und Lazarus

- Generelle Arbeitsweise
- Die .gitignore
- CI/CD

- 1 Theorie
- 2 Arbeitsschritte
- 3 Git und Lazarus

Theorie

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

1

Theorie

- Was ist Git?
- Persönliche Gründe
- Aufgaben einer Versionsverwaltung
- Begriffsklärung
- Typischer Workflow

Über Git

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

- Git ist eine dezentrale kommandozeilenbasierte Versionsverwaltung zur Verwaltung von Dateien und Ordner aller Art
- nicht-lineare Entwicklung (Zweige)
- kein zentraler Server
- kryptographische Sicherheit der Projektgeschichte durch Hashs gewährleistet
- Dateiversionierung und Speichersystem
- nicht nur für Programmierprojekte geeignet

Geschichte von Git

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

- Versionsverwaltung für den Linux-Kernel gesucht → keine geeignete Lösung gefunden
- 2005 Beginn der Entwicklung durch Linux Torvalds
- enorme Verbreitung in der Open-Source-Entwicklung (mehr als 70 Prozent aller Projekte)
- Online-Plattformen wie Github/Github setzen Entwicklung mit Git voraus
- Kommandozeilenwerkzeug, zahlreiche graphische Frontends (lazygit, Sublime Merge)

Persönliche Gründe

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

- seit ca. 10 Jahren Programmierung mit Lazarus
- Entwicklung einer Liedpräsentations-Software, da Nachfrage von Kirchgemeinden
- Langezeit Eigenbau ohne Versionsverwaltung
- Entwicklung schwer zu managen, geringe Backup-Möglichkeiten
- Einsatz einer Versionsverwaltung → Git, Github

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

The screenshot shows the Cantara application window titled "Song Selection (Cantara)". The window has a menu bar with "File", "Edit", and "Help". On the left, there is a list of songs, with "Amazing Grace" selected. The main area on the right displays a presentation slide titled "Presentation is running:". The slide features a background image of a stone cross on a grassy hill under a blue sky. The text on the slide reads: "1. Amazing grace how sweet the sound that saved a wretch like me. I once was lost but now am found, was blind, but now I see". Below this, there is a smaller text block: "2. It was grace that taught my heart to fear, and grace my fears relieved: how precious did that grace appear the hour I first believed." At the bottom of the window, there is a status bar with "Presentation..." on the left, a checked checkbox for "Multi-Screen-Mode (suitable for multiple screens)", and a "Settings..." button on the right.

Warum eine Versionsverwaltung?

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

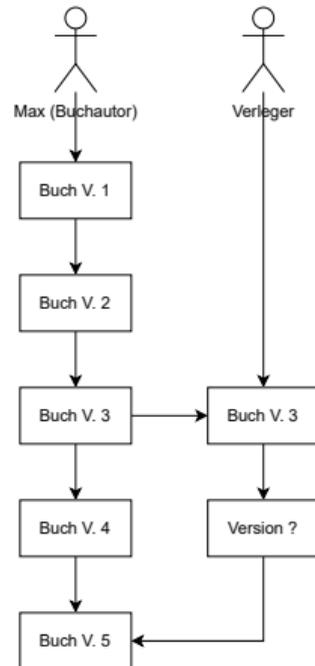
Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

Max möchte ein Buch schreiben.
Er aktualisiert das Dokument hin und wieder.
Schließlich sendet er dem Verleger eine Kopie
des aktuellen Stands.
Max arbeitet weiter. Der Verleger tätigt
parallel ein paar Korrekturen.
Schließlich erhält Max die
Änderungsvorschläge des Dokuments zurück.
Er muss nun zwei Dokumente
zusammenführen.



Aufgaben einer Versionsverwaltung

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

- Protokollierung von Änderungen auf vertikaler (und horizontaler) Ebene
- Zurücksetzen eines Dokuments auf den vorherigen Stand
- Zusammenfassen verschiedener Versionen
- Abgleichen verschiedener Orte

Git arbeitet *dezentral*, d.h. dass die Repositories beliebig verteilt sein können. Es wird immer in einem lokalen Repository gearbeitet und dieses anschließend synchronisiert.

Begriffsklärung

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

- **Projekt:** Eine Einheit von Ordnern und Dateien, die versioniert werden soll.
- **Repository:** Eine Kopie des Projekts, welches von Git verwaltet wird.
- **Commit:** Ein manuell definierter Arbeitsschritt, welche eine oder mehrere Änderungen enthält.
- **Branch:** Ein Arbeitszweig.
- **Pull:** Den aktuellen Stand von einem entfernten Repository *herunterziehen*.
- **Push:** Den aktuellen Stand auf ein entferntes Repository hochladen (*drücken*).

Github, Gitlab und co

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

- **Code-Hosting-Plattform:** Ein webbasierter Dienst, welcher Git und ein graphisches Frontend mit ergänzenden Funktionen anbietet (Issue-System, Bugtracker, Wiki, etc.)
- **Continuous Integration (CI):** Automatische Erstellung und Testung eines Softwareprojekts
- **Continuous Deployment (CD):** Automatische Auslieferung eines Softwareprojekts

Ein typischer Workflow

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

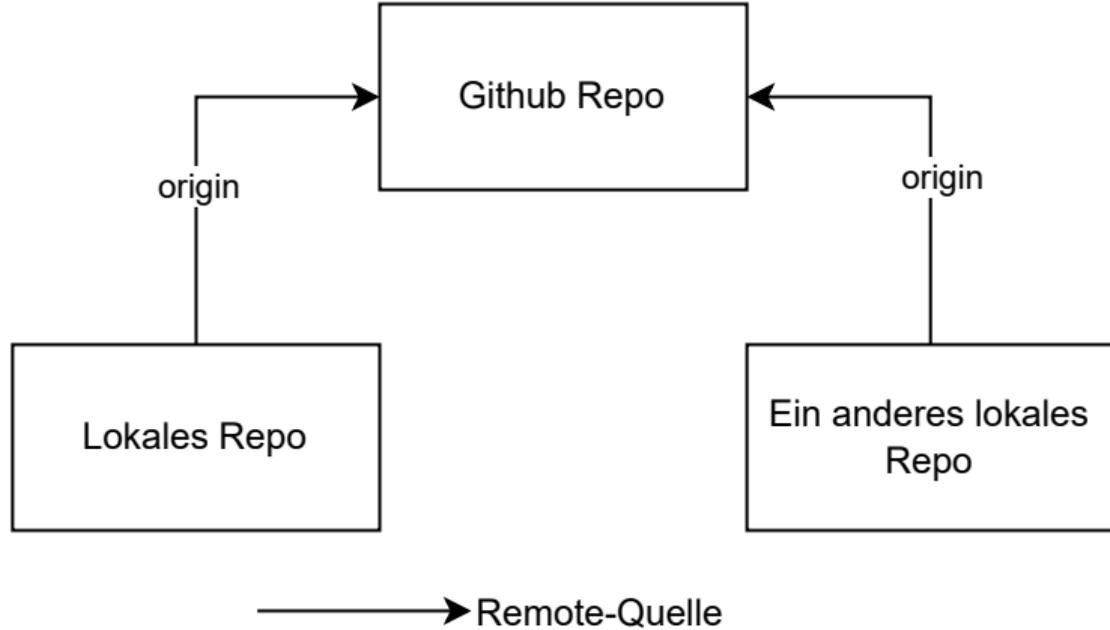
Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD



Arbeitsschritte

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

2 Arbeitsschritte

- Installation
- Ein Git-Repository erstellen
- Ein Git-Repository clonen
- Der Commit-Workflow
- Änderungen stagen
- Änderungen commiten
- Pullen und Pushen
- Branches

Installation

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

Git ist bei allen Betriebssystem über den Paketmanager verfügbar:

```
winget install -e --id Git.Git
sudo pacman -S git
sudo apt install git
```

Windows-Nutzer sollten die Power-Shell benutzen.

Zusätzlich existieren zahlreiche Gitplugins und graphische Frontends, die das Leben leichter machen (VS Code Plugin, IntelliJ Plugin, *lazygit*, *Sublime Merge*).

Ein Git-Repository erstellen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

git init

Beschreibung:

- erstellt ein Git-Repository im aktuellen Ordner für ein Projekt, welches **noch nicht mit Git versioniert wurde**.
- nachher existiert ein Unterordner `.git`, dieser darf nicht gelöscht werden

Ein Git-Repository clonen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git clone <adresse>
```

Parameter:

- **Adresse:** Die Adresse des entfernten Repositories, welches geclont wird.

Beschreibung:

- Clont ein entferntes Repository, sodass anschließend ein Unterordner mit einer entsprechenden Kopie entsteht.
- Das neu entstandene lokale Repository hat das entfernte Repository als origin-Quelle.

Der Commit-Workflow

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

- 1 (Repository pullen)
- 2 Änderungen vornehmen
- 3 Änderungen stagen
- 4 Änderungen commiten
- 5 (Repository pushen)

Änderungen stagen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git add <dateien und ordner>
```

Parameter:

- `dateien` und `ordner`: Die Dateien und Ordner, die für den nächsten Commit erfasst werden sollen. Wenn *alles* committed werden soll, einen Punkt `.` verwenden.

Beschreibung:

- Übernimmt geänderte Dateien und Ordner in die Staging-Area.
- Werden diese Dateien in der Zwischenzeit wieder verändert, müssen sie neu hinzugefügt werden.
- Rückgängig machen mit `git reset`

Änderungen commiten

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git commit -m <commit-nachricht>
```

Parameter:

- `commit-nachricht`: Die Beschreibung des Commits: Was wurde warum verändert?

Beschreibung:

- Führt einen Commit auf dem aktuellen Branch aus
- Wird der Parameter `-m` *weggelassen*, dann öffnet sich ein Editor zur Eingabe einer Commit-Nachricht.

Änderungen commiten

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git pull
```

```
git push
```

Beschreibung:

- **pull**: Pullt (zieht sich) das entfernte Origin-Repository und mergt dieses mit dem lokalen
- **push**: Pusht das lokale Repository auf den entfernten Ort und veranlasst dort einen Merge.

Einen Branch erstellen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git branch <branchname>
```

Parameter:

- **branchname**: Der Name des neuen Branches

Beschreibung:

- Erstellt einen neuen Branch unter dem Namen **branchname**, der mit dem aktuellen identisch ist.
- Git verbleibt beim alten Branch (**checkout** zum wechseln)

In einen Branch wechseln

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

```
git checkout <branchname>
```

Parameter:

- **branchname:** Der Name des Branches, zudem gewechselt werden soll

Beschreibung:

- Wechselt zu dem neuen Branch und stellt diesen im Arbeitsverzeichnis her
- **Vorsicht bei aktuell geöffneten Dateien!**

Zwei Branches zusammenführen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

```
git merge <branchname>
```

Parameter:

- **branchname:** Der Name des Branches, der in den aktuellen Branch zusammengeführt werden soll.

Beschreibung:

- Mergt den benannten Branch in den aktuellen Branch
- Beide Branches existieren weiter.

Git und Lazarus

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

3

Git und Lazarus

- Generelle Arbeitsweise
- Die .gitignore
- CI/CD

Generelle Arbeitsweise

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

- Lazarus kennt Git nicht, das heißt, die Versionsverwaltung erfolgt **außerhalb**
- keine Binärdateien committen (`.gitignore` hilft hier)
- Vorsicht beim Branch wechseln, wenn die Datei noch geöffnet ist (Gefahr des Überschreibens)
- CI/CD nützlich, am besten mit automatischen Tests (selbst noch am Lernen)

.gitignore

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

C/CD

```
# Created by https://www.gitignore.io/api/lazarus
# Edit at https://www.gitignore.io/?templates=lazarus

### Lazarus ###
# Lazarus compiler-generated binaries (safe to delete)
*.exe
*.dll
*.so
*.dylib
*.lrs
*.res
*.compiled
*.dbg
*.ppu
*.o
*.or
*.a
*.dbg

# Lazarus autogenerated files (duplicated info)
*.rst
```

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

Praxisbeispiel Cantara

- Github Actions
- Distribution via Snap

Weiterführende Informationen

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository clonen

Der Commit-Workflow

Änderungen stagen

Änderungen commiten

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

Graphische Anwendungen/Git-Tools

- Lazygit: Git-Frontend für die Kommandozeile
- Sublime Merge: Kommerzielle graphische Git-Verwaltung
- Beyond Compare: Kommerzielles Vergleichstool für Diffs, gut für Pascal geeignet
- Meld: Vergleichssoftware für Linux

Tutorials/zum Vertiefen:

- Visualizing Git: Die Befehle und die Arbeitsweise von Git verstehen. <https://git-school.github.io/visualizing-git/>
- Git-Tutorial von Atlassian: sehr gute und ausführliche Dokumentation in deutscher Sprache. <https://www.atlassian.com/de/git>
- CI/CD-Beispiel im Cantara-Projekt auf Github und im Lazarus-Wiki: <https://github.com/reckel-jm/cantara> und https://wiki.freepascal.org/Create_snap_package

Danke für die Aufmerksamkeit!

Arbeiten mit Git und co.

Jan Martin Reckel

Gliederung

Theorie

Was ist Git?

Persönliche Gründe

Aufgaben einer Versionsverwaltung

Begriffsklärung

Typischer Workflow

Arbeitsschritte

Installation

Ein Git-Repository erstellen

Ein Git-Repository klonen

Der Commit-Workflow

Änderungen stagen

Änderungen committen

Pullen und Pushen

Branches

Git und Lazarus

Generelle Arbeitsweise

Die .gitignore

CI/CD

Fragen, Kritik, Anregungen?