



Tag 3

Datei – I/O, Unit test und Git / GitHub

3. März 2023

Ablauf

- Rückblick Zweiter Tag
- Übungen besprechen
- Datei I/O
 - ggf. Umgang mit Text-Dateien rekapitulieren
 - JSON – was ist das?
 - JSON-Dateien lesen und schreiben
- Unit test
 - Was bringt`s
 - Wie wird`s gemacht?
 - Beispiel
- Git /GitHub
 - Weshalb Souce Code Management?
 - Wie funktioniert`s?

Rückblick

Zweiter Tag

Lernzeile

JSON File I/O

- Wissen, was JSON ist
- Können JSON Dateien lesen und schreiben

XML File I/O

- Haben Lesen/schreiben von XML Dateien gesehen

Datei I/O

- 1_json_file_io.ipynb/pdf
- 2_xml_file_io.ipynb/pdf

Übung «The Gang»

JSON einlesen und verarbeiten

- 1) Einlesen der JSON-Datei *the_gang.json*
- 2) Erstelle eine Funktion, welche das Durchschnittsalter berechnet und zurückgibt
- 3) Erstelle eine Funktion, welche die Haarfarben und deren Anzahl sammelt und als dict zurückgibt (Beispiel: {"Brown": 3, "Blonde": 1, ...})
- 4) Gib diese Werte und die Anzahl Personen aus
- 5) Speichere diese Werte wiederum als JSON, als dict mit den Schlüsseln «avg_age», «hair_colors» und «count» in *.results_the_gang.json*

Lernzeile

Unit Test

- Kennen die Vorteile von Unit test
- Haben mit einem Unittest-Framework gearbeitet
- Kennen den Aufbau eines Unit-Tests

Weshalb Unit Tests?

- Automatisiert
- Besserer Startpunkt als `main()`
- Dokumentieren die Verwendung
- Sicherheitsnetz bei Refaktorisierungen («Umbauten»)

Es gibt sogar Test First / Test Driven Design (TDD) als Methodologie

Asserts

Modul «unittest» ist Teil der Standardbibliotheken. Aufbau solcher Module ist immer ähnlich.

Methode

`assertEqual(a, b)`
`assertNotEqual(a, b)`
`assertTrue(x)`
`assertFalse(x)`
`assertIs(a, b)`
`assertIsNot(a, b)`
`assertIsNone(x)`
`assertIsNotNone(x)`
`assertIn(a, b)`
`assertNotIn(a, b)`
`assertIsInstance(a, b)`
`assertNotIsInstance(a, b)`

Test

`a == b`
`a != b`
`bool(x) is True`
`bool(x) is False`
`a is b`
`a is not b`
`x is None`
`x is not None`
`a in b`
`a not in b`
`isinstance(a, b)`
`not isinstance(a, b)`

Demo

→ `Code/unit_test.py`

Übung

Erstelle einen neuen Testcase für die Funktion, welche die Haarfarben sammelt

GIT

- Was ist Git?
- Was ist GitHub?
- Wie arbeitet man damit?
- Was ist *mergen* und wie geht das?

Lernziele GIT

- Wissen, was Git und GitHub ist
- Haben die wichtigsten Begriffe gehört
- Haben vom Standard-Workflow gehört
- Können *clonen* und *pullen*

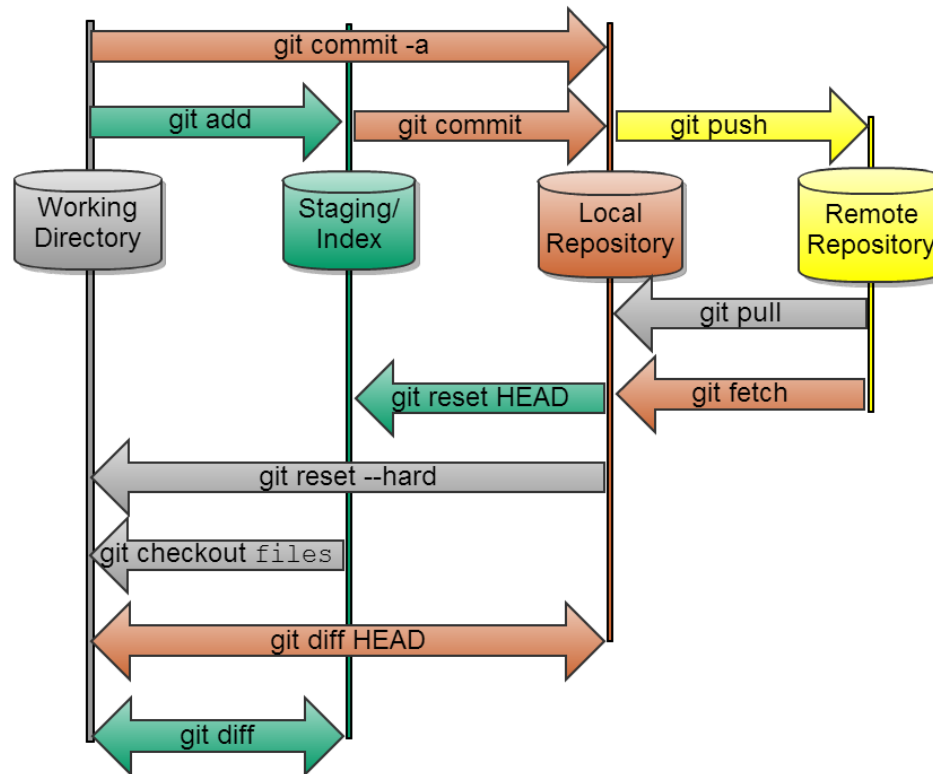
Was ist Git?

- Source Code Management System (SCM)
- Mehrere Personen können gleichzeitig am selben Projekt arbeiten
- Sogar an der selben Datei
- Dezentral
- Workflow nicht vorgegeben
- Git heutzutage wohl am weitesten verbreitet. Früher war Subversion (und TFS, Team Foundation Server) auch gang und gäbe
- Extrem wertvoll, auch wenn man nur alleine an etwas arbeitet
 - Versions-Geschichte, Undo

Was ist GitHub?

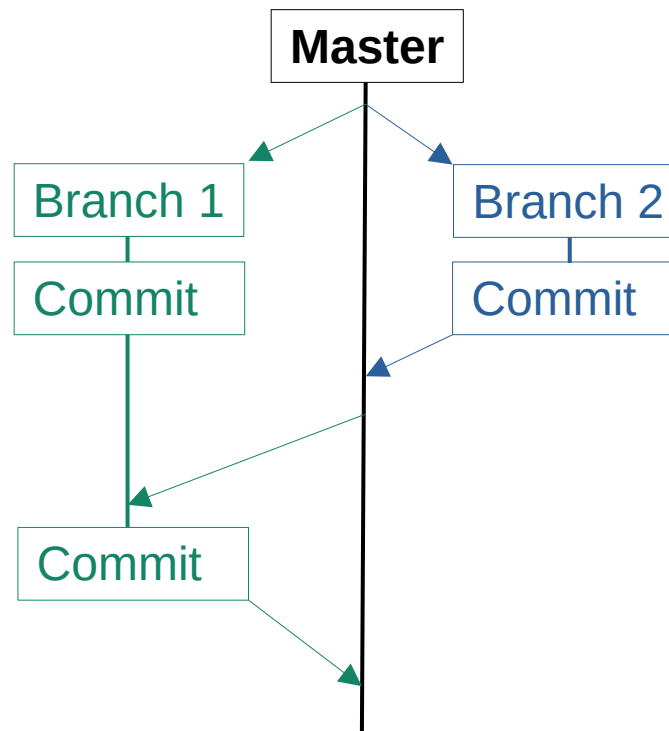
- Ein öffentlicher, frei verfügbarer Git-Server
- Plus Pull Request, Issues, Review support, CI/CD und und und....
- Kostenloses Angebot

Wie arbeitet man damit?

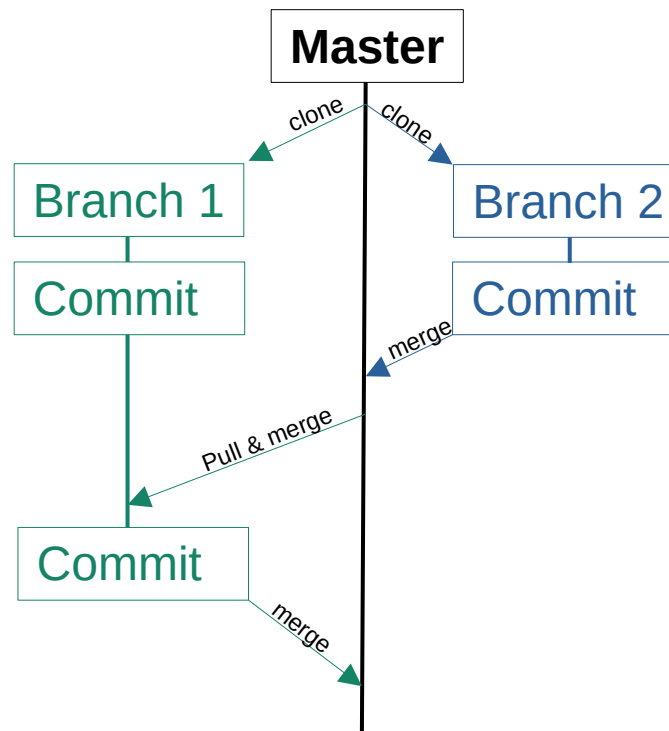


- DAS Buch: <https://git-scm.com/book/de/v2>
- Werkzeuge: Sourcetree und/oder VS Code (ggf.mit Plugin Git Graph)

Standard-Workflow



Standard-Workflow



Übung

- 1) Optional: Registriere dich bei GitHub
- 2) Optional: Installiere SourceTree
- 3) Öffne <https://github.com/MicCalo/Applikationsentwicklung> mit einem Webbrowser
- 4) Clone das Repository mit in VS Code