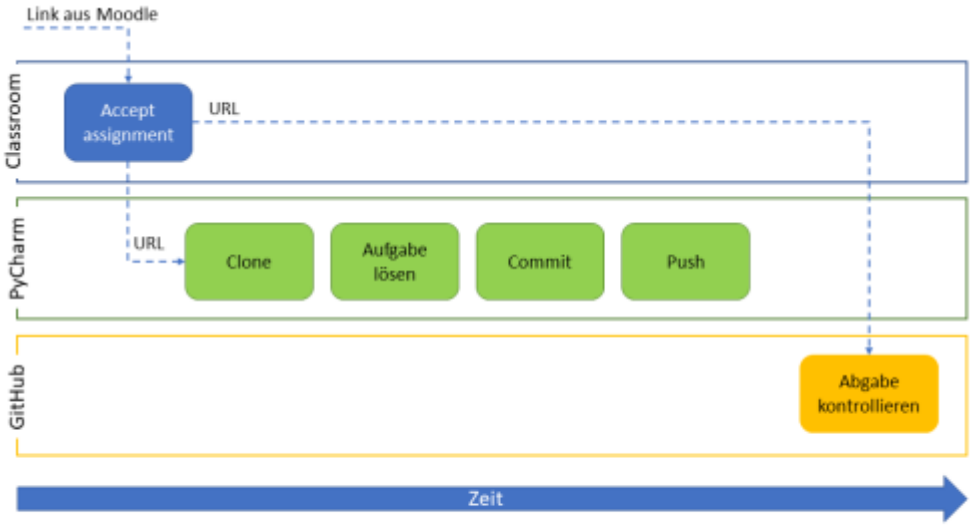
# LU04c - Arbeiten mit GitHub Classroom

Video zum Arbeiten mit GitHub, Classroom und PyCharm

## Überblick

Für die Programmieraufgaben im Modul 319 verwenden wir GitHub Classroom und PyCharm (oder eine ähnliche Entwicklungsumgebung). Dadurch erlernen Sie die wichtigsten Arbeitsschritt mit git und brauchen Ihre Lösungen nicht separat in Moodle hochzuladen.




## Voraussetzung

Bevor Sie mit GitHub Classroom arbeiten können, müssen Sie ...

- ... ein Konto bei GitHub erstellt haben.

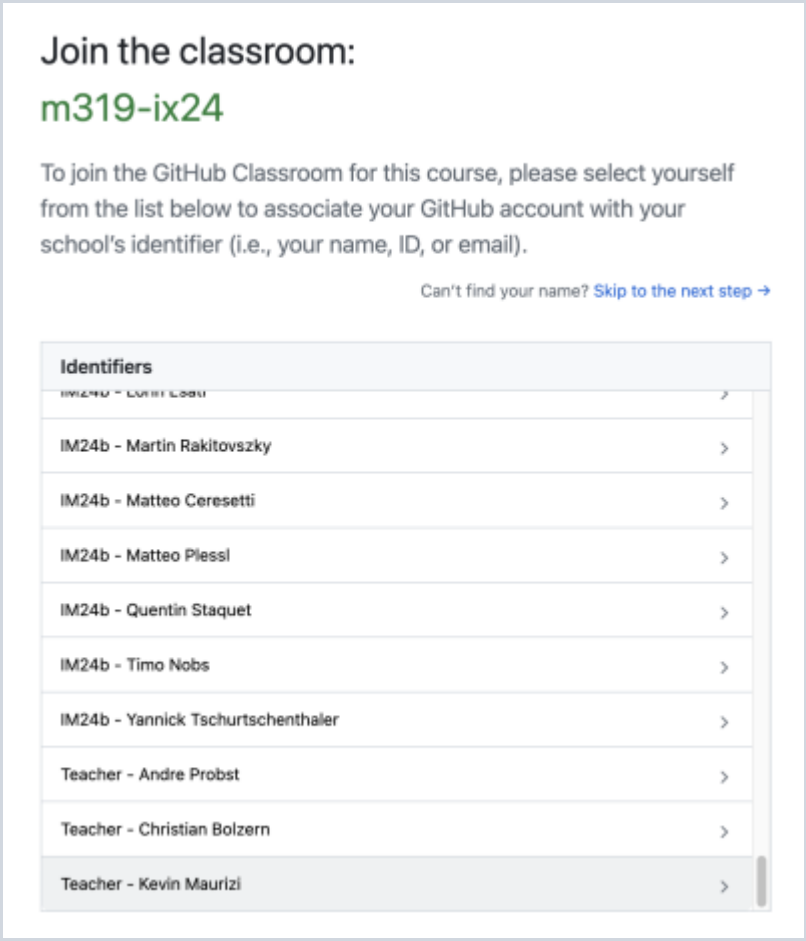
## Vorgehen

### 1. Aufgabe akzeptieren

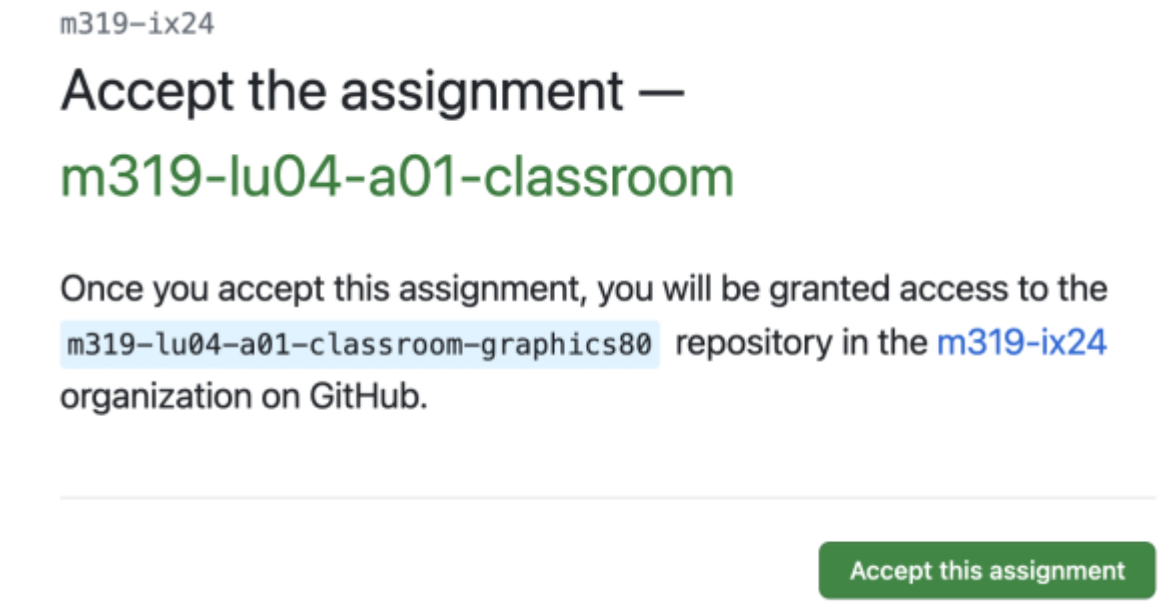
Im Moodle-Kurs finden Sie den Link zur Aufgabe im Classroom.

Bei der ersten Aufgabe in einem neuen Modul:

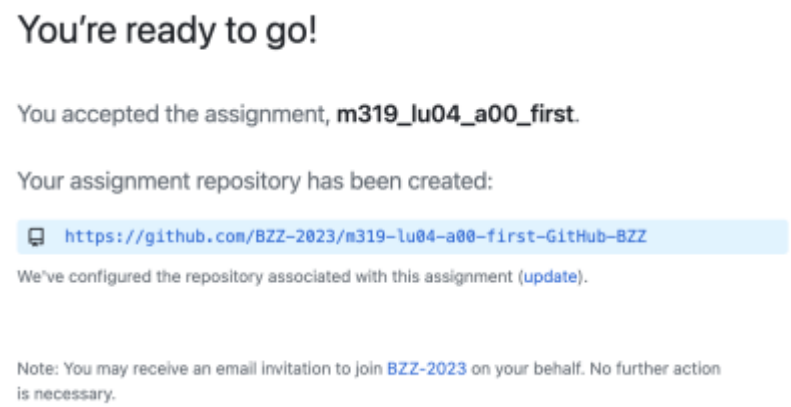
Wählen Sie ihren Namen aus der Liste aus:




Öffnen Sie diesen Link und akzeptieren Sie das *Assignment*.



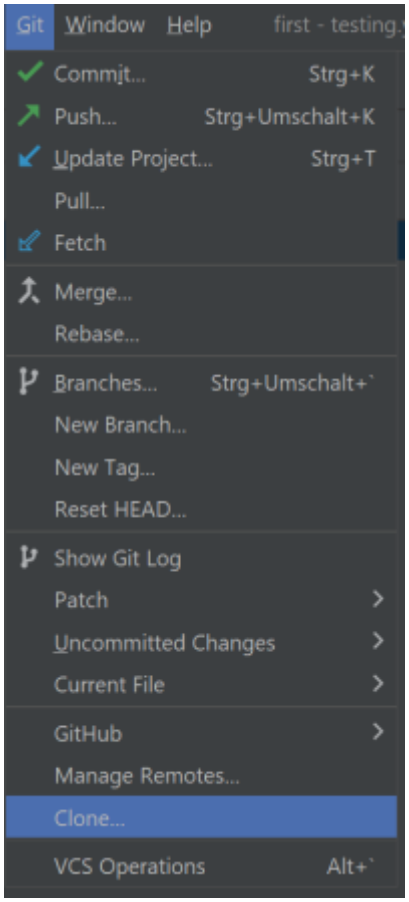

Warten Sie ein paar Sekunden und laden Sie die Webseite neu. Sie sehen die Adresse (URL) Ihres persönlichen Repositories. Kopieren Sie diesen Adresse.



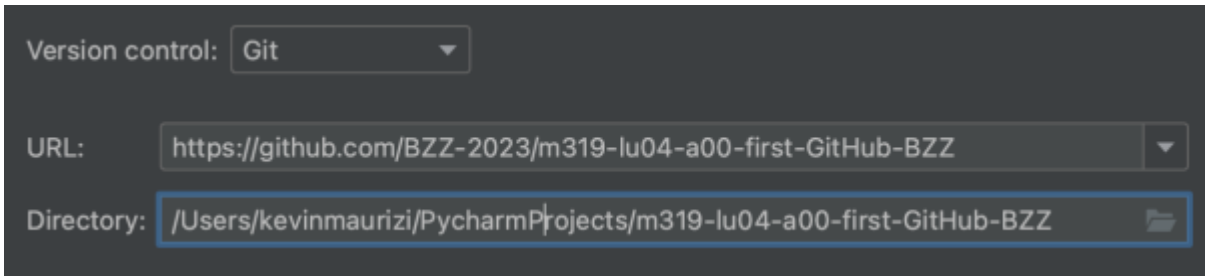

### 2. Repository klonen

Öffnen Sie nun PyCharm und klonen Sie mit dem kopierten Link das Repository als neues Projekt.

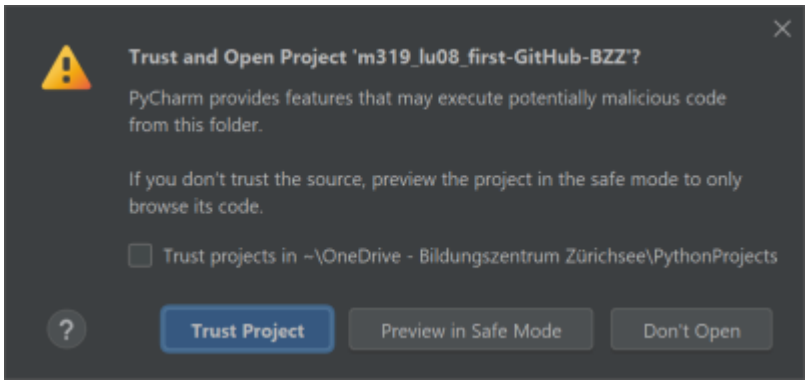
**oder**




Tragen Sie die kopierte URL ein. Passen Sie bei Bedarf den Ordnerpfad an, in dem das Projekt erstellt wird.

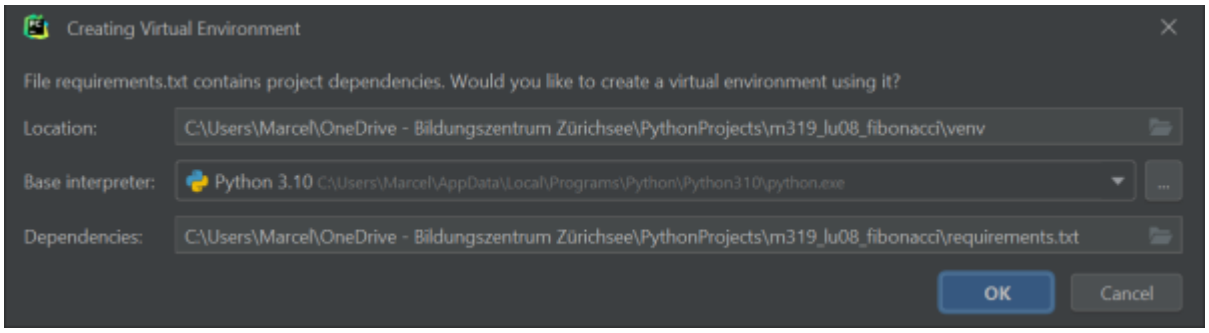



Anschliessend müssen Sie evtl. bestätigen, dass Sie diesem Projekt trauen.



### 3. Virtual Environment einrichten

Das Virtual Environment von Python enthält alle externen Pakete, die unser Projekt benötigt. Pycharm kann dieses automatisch für uns einrichten




Bestätigen Sie die Einrichtung des Virtual Environments mit [OK]

### 4. Aufgabe lösen

Sie haben nun eine lokale Kopie des Projekts auf Ihrem Computer. Das Projekt enthält einige Dateien und Ordner, von denen aktuell nur zwei für Sie von Bedeutung sind:

- main.py: Das Python-Skript mit der Aufgabenstellung.
- main_test.py: Ein Python-Skript mit dem Sie Ihre Lösung prüfen können.

Suchen Sie in der Datei `main.py` die Zeile mit dem Kommentar `# TODO ...`. Ändern Sie die Zeile gemäss den Anweisungen im Kommentar. Löschen Sie danach den Kommentar.

### 5. Lösung testen

Das automatisierte Testen Ihrer Lösung hat zunächst nichts mit git oder GitHub Classroom zu tun. Es ermöglicht Ihnen, ein direktes Feedback zu Ihrer Lösung zu erhalten.

Öffnen Sie die Datei `main_test.py` und führen Sie die Funktion `test()` aus. Falls Ihre Lösung korrekt ist, sehen Sie folgendes im Fenster „run“:

```
============================= test session starts
==============================
collecting ... collected 1 item

main_test.py::test PASSED
[100%]

============================== 1 passed in 0.01s
==============================
```

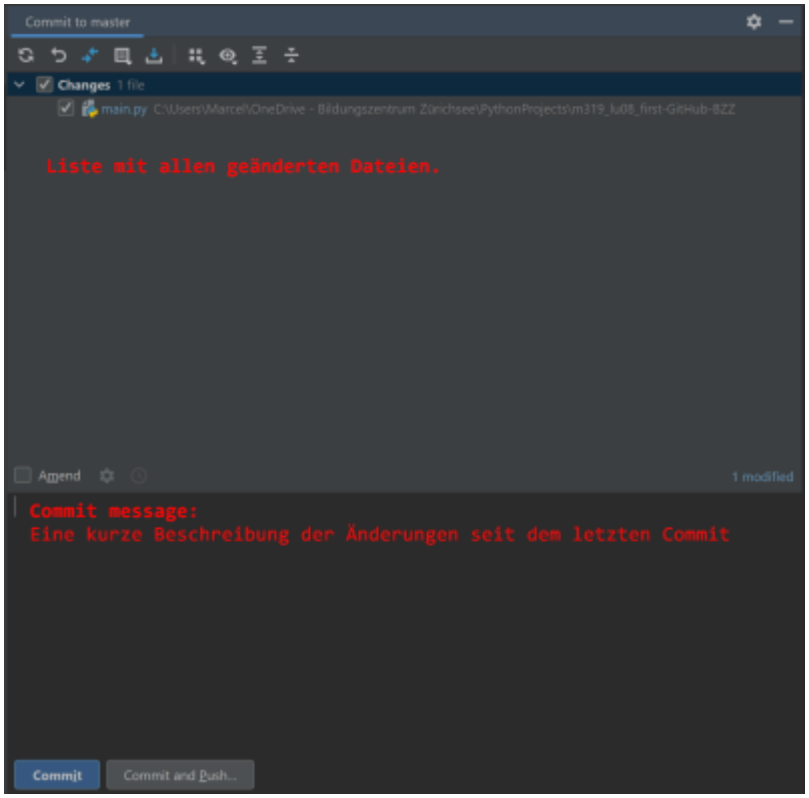
Falls noch ein Fehler in Ihrer Lösung ist, sieht die Ausgabe in etwa so aus:

```
============================= test session starts
==============================
collecting ... collected 1 item

main_test.py::test FAILED
[100%]
main_test.py:2 (test)
'Total: 47.25\n' != 'Total: 78.75\n'

Expected :'Total: 78.75\n'
Actual   :'Total: 47.25\n'
```

### 6. Commit

Nachdem Sie die Aufgabe gelöst haben, führen Sie einen Commit durch. Damit wird der aktuelle Stand Ihrer Dateien festgeschrieben.

1. Wählen Sie im Menu **git** den Punkt **commit** aus. Es öffnet sich eine neue Ansicht:




2. Geben Sie in der **commit message** an, was Sie geändert haben.

3. Klicken Sie auf [Commit]. Sie sollten nun eine Bestätigung erhalten, dass der Commit erfolgreich war.

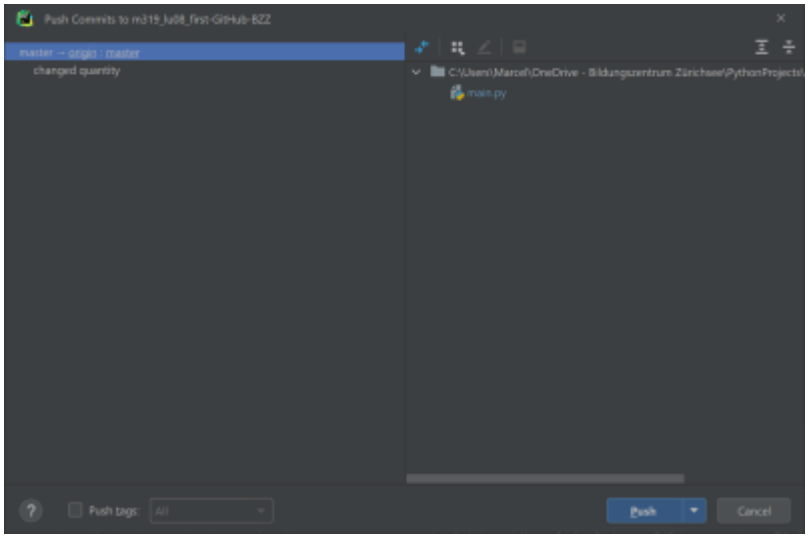
Falls Sie den Kommentar mit `# TODO …` nicht entfernt haben, erhalten Sie eine Warnung. Sie können

den Kommentar entfernen und den Commit erneut durchführen. Alternativ können Sie auch auf [Commit anyways] klicken.

### 7. Push

Um die gelöste Aufgabe abzugeben, führen Sie einen **Push** Ihres Repositories durch. Dabei werden die Änderungen aus Ihrem lokalen Repository in GitHub Classroom hochgeladen.

1. Wählen Sie im Menu **git** den Punkt **push ...** aus. Es öffnet sich ein neues Fenster:




Dieses Fenster zeigt Ihnen, welcher Branch (z.B. master) an welches entfernte Repository (z.B. origin) gesendet werden. Das Arbeiten mit verschiedenen Branches werden Sie später erlernen. Für den Moment können Sie diese Einstellungen einfach übernehmen.

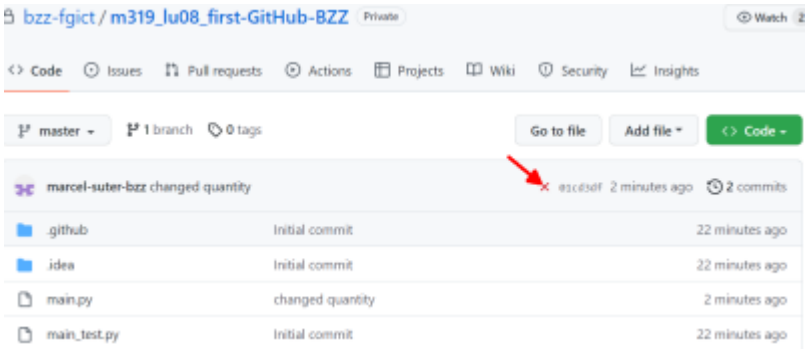
2. Klicken Sie auf [Push] und Sie erhalten eine Bestätigung, dass der Push erfolgreich war.

Manchmal erhalten Sie statt der Bestätigung eine Meldung, dass Sie sich zunächst bei GitHub autorisieren müssen.

Wählen Sie in diesem Fall [Login via GitHub ...]. Dadurch öffnet sich eine neue Seite in ihrem Webbrowser.

### 8. Abgabe überprüfen

Sie können Ihre Abgabe prüfen, indem Sie ihr Repository in einem Webbrowser anschauen. Öffnen Sie dazu den Link, mit dem Sie bei **Projekt klonen** ihr lokales Projekt erstellt haben.




Ein kleines Symbol zeigt Ihnen an, ob Ihre Lösung korrekt ist:

- Gelber Kreis: Die Überprüfung läuft noch. Warten Sie 1-2 Minuten und laden Sie die Seite neu.
- Rotes Kreuz: Einer oder mehrere Tests sind nicht erfolgreich verlaufen. Klicken Sie auf das rote Kreuz für genauere Angaben.
- Grüner Haken: Alle Tests waren erfolgreich.

---

m319-E1F

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
**https://wiki.bzz.ch/modul/m319/learningunits/lu04/gitanwendung?rev=1725977402**

Last update: **2024/09/10 16:10**