

Vorlage für Pycharm mit GitHub Classroom

Repository / Projekt erstellen



Um Konflikte mit der Benennung im Classroom zu vermeiden, sollten die Repositories und Assignments

- alles in Kleinbuchstaben geschrieben sein
- Worttrennungen mit Bindestrichen sein

Zum Beispiel: m399-lu19-a39-something

Variante A: Mit Template

1. Erstelle ein neues GitHub-Repository. Wähle als Template „BZZ-Commons/python-template“ aus.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk ().*

Single sign-on to see results in the bzz-fgict organization.

Repository template



Start your repository with a template repository's contents.

Include all branches

Copy all branches from BZZ-Commons/python-template and not just the default branch.

Owner *



Repository name *

/ m319_lu06_a02_names

m319_lu06_a02_names is available.

Great repository names are short and memorable. Need inspiration? How about [cautious-happiness](#) ?



Beim Auswählen des Templates wird der Owner zu BZZ-Commons geändert. Du musst hier den korrekten Owner z.B. templates-python auswählen.

2. Klon das neue Repository in PyCharm.

3. PyCharm fragt dich, ob du ein *Virtual Environment (venv)* erstellen willst, bestätige diese mit [OK].

Variante B: Leeres Repository

1. Erstelle ein neues, leeres Repository. Dieses Repo darf keine Dateien enthalten; auch kein README.md oder LICENSE!
2. Erstelle ein neues Projekt in PyCharm.
3. Erstelle die Dateien und Ordner in PyCharm.
4. Führe einen Push ins GitHub Repository durch.

Anpassungen

README.md

Lösche diese Datei oder passe die Beschreibung an.

classroom.yml

```
1. name: GitHub Classroom Workflow
2.
3. on: push
4.
5. permissions:
6.   checks: write
7.   issues: write
8.   actions: read
9.   contents: read
10.
11. jobs:
12.   copy-issues:
13.     if: ${{ contains(github.actor, 'classroom') }}
14.     uses: BZZ-Commons/workflows/.github/workflows/setup.yml@main
15.     with:
16.       SOURCE_REPO: templates-python/m319_lu04_a00_classroom # CHANGE
          owner/name of the repo with the issues
17.       TARGET_REPO: ${{ github.repository }}
18.       ADD_LABELS: "false"
19.       secrets:
20.         GHSECRET: ${{ secrets.GITHUB_TOKEN }}
21.
22.   grading:
23.     if: ${{ !contains(github.actor, 'classroom') }}
24.     uses: BZZ-Commons/workflows/.github/workflows/grading.yml@main
25.     with:
26.       MOODLE: false # CHANGE true=export grade to moodle
27.       secrets: inherit
```

Dieser Workflow umfasst 2 Jobs:

copy-issues

Kopiert die Issues aus einer Vorlage in das Repository des Lernenden. Wird nur durchgeführt, wenn ein Lernender das Assignment in GitHub Classroom akzeptiert.

Entweder du passt bei SOURCE_REPO auf Zeile 16 den Besitzer und Namen des Vorlage-Repositories an.

Oder du löscht diesen Job raus, indem du die Zeilen 12-20 entfernst.

grading

Führt automatisierte Tests durch und bewertet das Programm. Dieser Job wird bei jedem Push ausser beim Erstellen des Repositories durchgeführt.

Auf Zeile 26 kannst du angeben, ob die Bewertung und das Feedback in Moodle exportiert werden soll. Ändere den Wert auf true, damit der Export ausgeführt wird. Lies dazu die Anleitung [Bewertung vom Autograding in Moodle übernehmen](#)

copyissues.yml

Dies ist ein manueller Workflow, um die Issues aus einer Vorlage zu kopieren. Falls du bei diesem Projekt nicht mit Issues arbeitest, kannst du diesen Workflow löschen.

Template

https://github.com/bzz-fgict/M319_Template

m319_template-master.zip

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

Permanent link:
<https://wiki.bzz.ch/howto/pycharm/template>

Last update: **2024/03/28 14:07**

