

Template-Repositories

Ein **Template-Repository** enthält den Startcode einer Aufgabe. Classroom 50 kopiert es beim Annehmen in ein privates Repository der lernenden Person. Die Python-Templates der BZZ liegen in der Organisation `templates-python`.

Aufbau

```
<SLUG>/
├── .github/
│   ├── autograding/           ← Bewertungs-Konfiguration (Fallback)
│   │   ├── unittests.json
│   │   ├── lint.json
│   │   └── pylintrc
├── tests/                     ← die Tests der Lehrperson
│   └── test_*.py
├── main.py                    ← Startcode / Stub
├── requirements.txt           ← für die lokale Arbeit der Lernenden
└── README.md                  ← Aufgabenstellung
```

Kein `.github/workflows/classroom.yml`. Der Bewertungs-Workflow kommt unter Classroom 50 zentral aus dem Config-Repo. Bleibt die alte Datei liegen, laufen bei jedem Push zwei Autograder nebeneinander — siehe [Migration](#).

Namensregeln

Der Repository-Name ist der **Slug** der Aufgabe. Er wird an drei Stellen identisch verwendet:

1. Name des Template-Repos
2. slug in `<CLASSROOM>/assignments.json`
3. Name der Aktivität *External Assignment* in Moodle

Kleinbuchstaben, Bindestriche als Trenner, Modul und Lerneinheit voran:

```
m323-lu01-a02-imperativer-ggt
m320-lu04-a4-objektkommunikation
```

Gegenüber GitHub Classroom entfällt die alte Einschränkung, dass der Name keine Bindestriche enthalten darf. Classroom 50 arbeitet durchgehend mit Slugs.

Neues Template anlegen

1. Repository in templates-python erstellen und unter **Settings → Template repository** als Template markieren.
2. Startcode, Tests und README.md mit der Aufgabenstellung einfügen.
3. .github/autograding/ mit unittests.json, lint.json und pylintrc anlegen — Aufbau siehe [Wie bewertet wird](#).
4. Aufgabe in <CLASSROOM>/assignments.json des Config-Repo eintragen, mit „autograder“: „default“ und ohne tests-Block.
5. In Moodle eine Aktivität *External Assignment* mit demselben Slug anlegen, siehe [Classroom 50: Noten nach Moodle übertragen](#).

requirements.txt

Die Datei ist für die **lokale** Arbeit der Lernenden da. Die Bewertung auf GitHub installiert sie bewusst nicht: pygrader50 bringt eigene, gepinnte Abhängigkeiten mit, damit die Bewertung reproduzierbar bleibt.

Die alten Pins `pytest==8.3.3` und `pylint==3.2.7` sind auf Python 3.14 nicht mehr lauffähig. Wer die Datei pflegt, sollte sie deshalb auf aktuelle Versionen heben — sonst weicht die lokale Umgebung der Lernenden von der Bewertung ab.

Änderungen an einem Template

Template-Änderungen erreichen bestehende Repositories nicht. Ein Studi-Repo trägt seine Kopie aus dem Moment der Annahme. Eine korrigierte `unittests.json` muss also entweder in die bereits angenommenen Repos nachgezogen werden — oder, besser, als Bundle ins Config-Repo wandern, wo sie sofort für alle gilt.

Das ist das stärkste Argument für die Bundles unter <CLASSROOM>/autograders/<SLUG>/: Eine Korrektur dort wirkt beim nächsten Push für die ganze Klasse.

Weiter

- [Das classroom50-Repository](#)
- [Wie bewertet wird](#)
- [PyCharm-Template](#)

From:

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

Permanent link:

<https://wiki.bzz.ch/howto/git/template>

Last update: **2026/08/14 11:11**

