Spickzettel: GitHub Codespaces – Cloud-basierte Entwicklungsumgebungen

Ziel

Sofort einsatzbereite, cloudbasierte Entwicklungsumgebungen direkt auf GitHub nutzen – für schnelles Arbeiten ohne lokale Einrichtung.

Was ist GitHub Codespaces?

- Cloud-IDE (basierend auf VS Code)
- Läuft vollständig in der GitHub-Umgebung
- Unterstützt mehrere Sprachen & Frameworks
- Komplette Entwicklungsumgebung inkl. Terminal, Git, Editor, Docker

---

Voraussetzungen

- Aktivierung über GitHub.com → „Code“ → Tab „Codespaces“
- Codespaces ist kostenpflichtig für Privatnutzung (bei Organisationen ggf. inkludiert)
- GitHub-Account & unterstütztes Repository

Codespace erstellen

1. Gehe zu einem Repository
2. Klicke auf "Code" → "Codespaces" → "Create codespace"
3. Wähle Branch & Konfiguration

Optional per CLI:

```
gh codespace create --repo owner/repo --branch main
```

Entwicklungscontainer konfigurieren

- Datei: `.devcontainer/devcontainer.json`
- Definiert Umgebung, Erweiterungen, Befehle, Abhängigkeiten

Beispiel:

```
{
  "name": "Python Dev",
```

```
  "image": "mcr.microsoft.com/devcontainers/python:3.10",
  "features": {},
  "postCreateCommand": "pip install -r requirements.txt",
  "customizations": {
    "vscode": {
      "extensions": ["ms-python.python"]
    }
  }
}
```

Was wird synchronisiert?

- Automatisch synchronisiert: Git-Projekt, `.vscode/`, `.devcontainer/`
- Persistente Daten (z. B. Datenbanken) nur über Docker-Volumes

---

Best Practices

- Nutze `.devcontainer` zur Definition einer einheitlichen Umgebung im Team
- Automatisiere Projekt-Setup mit `postCreateCommand`
- Halte Codespaces klein (nicht zu viele Hintergrundprozesse)
- Verwende Stopp-Zeitlimits, um Kosten zu sparen
- Nutze Prebuilds für große Projekte mit langer Buildzeit

GitHub Codespaces ermöglichen sofortiges Loslegen – besonders nützlich für Onboarding, schnelle Fixes, prototypisches Arbeiten und Kontributoren.