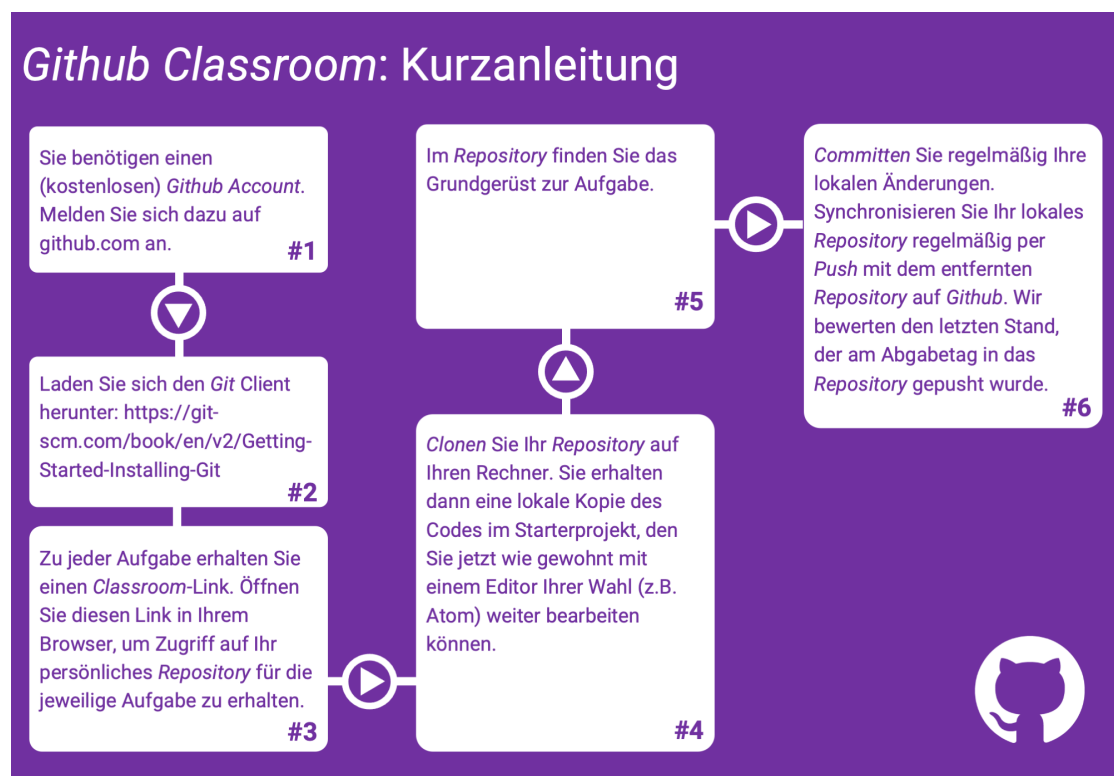


Workflow mit *GitHub Classroom*

Medieninformatik Webtechnologien - Prof. Dr. Markus Heckner

GitHub Classroom

GitHub Classroom ist ein Service von *GitHub*, der es ermöglicht den Quellcode für studentische Projekte zu verwalten. Gleichzeitig ist die Abgabe Ihrer Lösungen einfach, da Sie Ihren Code automatisch während der Bearbeitung mit *GitHub* synchronisieren. Dieses Handout beschreibt, wie Sie Ihr eigenes Repository einrichten können, und den Workflow mit *GitHub* während der Bearbeitung Ihrer Aufgabe.



Account bei GitHub anlegen

Für die Aufgabe steht Ihnen ein Starterprojekt zur Verfügung.

Um auf das Starterprojekt zugreifen zu können, und um Ihren Code in das Repository hochzuladen benötigen Sie einen *GitHub* Account: Legen Sie sich zuerst einen Account bei <http://github.com> an.

Hinweis: Bitte achten Sie darauf, dass Sie Ihr **echter Name** in Ihrem Benutzernamen klar zu erkennen ist (z.B. **lesternygaard, lester.nygaard, lnygaard, nygaardi, ...**). **Ansonsten können wir Ihre Aufgabe nicht zuordnen und bewerten.**

GitHub einrichten

Laden Sie sich hier

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git> einen *Git-Client* für Ihr System herunter.

Personal access token anlegen

1. Öffnen Sie die URL: <https://github.com/settings/security> und loggen Sie sich mit Ihrem GitHub Usernamen und Passwort ein
2. Konfigurieren Sie dann die *two-factor-authentication* unter dem folgenden Link: [two-factor-authentication](#). Auch wenn GitHub das nicht vorschlägt ist die Authentifizierung per SMS (*Configuring two-factor authentication using text messages*) möglicherweise der schnellste Weg.
3. Gehen Sie auf <https://github.com/settings/tokens>
4. Klicken Sie auf *Generate new token*
5. Geben Sie im Feld *Note* einen Text ein z.B. *Medieninformatik*
6. Wählen Sie *No expiration* im Dropdown unter *Note*
7. Wählen Sie *repo* unter *Select scopes*
8. Klicken Sie auf *Generate token*
9. Markieren und kopieren Sie den *personal access token*. Dieser beginnt wahrscheinlich mit **ghp_**
10. Dieser Token funktioniert als Ihr Passwort bei der Nutzung des *Git-Clients*. Speichern Sie diesen unbedingt an einem sicheren Ort (z.B. Passwortmanager). Sie können Sie den Token später nicht noch einmal ansehen, d.h. Sie müssen dann einen neuen Token erstellen, falls Sie diesen Token verlieren sollten.

Repository anlegen und Starterprojekt klonen

1. Legen Sie einen Ordner auf Ihrem Rechner für den Code Ihres Projekts an.
2. Folgen Sie den Anweisungen unter dem Link im pdf der Aufgabenstellung.
Achtung: Sie müssen bei *GitHub* eingeloggt sein, ansonsten erhalten Sie nach Klick auf den Link eine Fehlermeldung. Wählen Sie dann Ihren Namen aus der Teilnehmerliste, um Ihren GitHub Account damit zu verknüpfen
3. Navigieren Sie dann zur Startseite Ihres Repositories und klicken Sie dort auf *Clone or download*. Kopieren Sie sich den dort erscheinenden Link (benötigen Sie im nächsten Schritt als **REPO-URL**).
4. Laden Sie das Starterprojekt, indem Sie eine Kommandozeile auf Ihren Rechner öffnen. Wechseln Sie dann zu dem neu angelegten Ordner und tippen Sie folgenden Befehl ein:
git clone REPO-URL
5. Geben Sie Ihren *personal access token* ein, falls der *Git-Client* Sie dazu auffordert.
6. Jetzt sollten Sie das Starterprojekt auf ihren Rechner geladen haben und Sie können den Code wie gewohnt bearbeiten.

Workflow während der Bearbeitung der Projektarbeit

Hinzufügen von Änderungen in das Repository

Sobald Sie ein neues Feature hinzufügen oder einen Bug fixen, sollten Sie eine neue Version in Ihrem Repository erzeugen.

Gehen Sie wie folgt vor, um Änderungen an Ihrem Code in das Repository zu übertragen. Achtung: Das Repository ist gleichzeitig die Abgabe für Ihr Projekt. Sie müssen alle Änderungen in das Repository übertragen, sonst kann Ihre Lösung nicht bewertet werden.

Hinzufügen der Änderungen (d.h. alle neuen Dateien werden für Übertragung in das lokale Repository markiert):

```
git add .
```

Hinzufügen der lokalen Änderungen in das lokale Repository:

```
git commit -m "Added CSS for first breakpoint at 950 pixels"
```

Hinweis: Erstellen Sie unbedingt eine aussagekräftige *Commit Message* auf **englisch**. Orientieren Sie sich an den Beispielen weiter oben in diesem Abschnitt.

Übertragen der Änderungen in das Remote Repository:

```
git push origin main
```

Führen Sie diese Befehle immer dann aus, wenn Sie relevante Änderungen in Ihrem Code vornehmen, d.h. neue Features implementieren oder Bugs fixen.

Wichtiger Hinweis: Alle Repositories werden nach dem Semester gelöscht. Sie sind selbst für ein Backup Ihrer Dateien verantwortlich!