# **Workshop: Brug af GitHub i Programmeringsundervisningen**

## **Mål**

Brugen af GitHub som et værktøj til at dokumentere deres projekter og koder, samt at understøtte deres læring: Skriftlighed

## **Forberedelse**

1. **Opret en GitHub-konto**: Sørg for, at alle deltagere har en GitHub-konto.
2. **Visual Studio Code**: Anbefal at bruge VS Code som udviklingsmiljø for at lette integrationen med Github.
3. BrugeGitHub Desktop <https://github.com/apps/desktop>

## **Workshop Struktur**

### **1. Introduktion til GitHub**

* Forklar hvad GitHub er, og hvordan det fungerer som et versionsstyringssystem.
* Introducer begrebet "repository" og dets betydning for projektstyring.
* Gennemgå formålet med at dokumentere projekter i en README.md-fil.

### **2. Oprettelse og ‘clone’ af et Repository (30 minutter)**

* hvordan du arbejder med GitHub for at oprette en “fork” af et repository, hvorfor det er nødvendigt, og hvordan du kloner det forkede repository til din lokale maskine ved hjælp af GitHub Desktop.
* **Klon repository**:
  + XBI Vis, hvordan man fork repository'et til den lokale maskine ved hjælp af ‘fork’
  + XBI Vis, hvordan man clone en fork repo

Source:

<https://github.com/WordPress/twentytwentyfive>

* **Gruppediskussion:**

1. Hvorfor er det nødvendigt at forke et repository, når du arbejder med store projekter?

2 Hvilke fordele er der ved at have et fælles repository, når I arbejder på et teamprojekt?

3 Kan du identificere, hvilken type projekt dette er:[twentytwentyfive](https://github.com/WordPress/twentytwentyfive)?

4. Hvordan er filerne struktureret?

5 Hvilke mapper eller filer skiller sig ud, og hvad tror du, de bruges til?

6. README.md-filen?

Forklarer den, hvad projektet handler om?

**3. Tilføjelse af Dokumentation (30 minutter)**

* Opret et repo eller bruge et repo (jeres eget projekt)
* **Redigering af README.md-fil**:
  + Forklar strukturen af en README.md-fil og dens vigtighed for dokumentation.
  + Giv eksempler på, hvad der kan inkluderes:
    - Projektbeskrivelse
    - Installation og brug
    - Screenshots og figurer
    - Bidrag og author
* **Gruppediskussion: Markdown Syntax**:
  + Hvad er Github Markdown-syntax, herunder overskrifter, lister, links og billeder.

### **4. Code Review i Gruppe (15 minutter)**

* **Code Review**:
  + Gennemgå kode i gruppe
  + Fokusere på struktur, læsbarhed og funktionalitet.

Kodegennemgang kan finde sted i mange forskellige situationer under kodning, test eller frigivelses-/vedligeholdelsesfaserne af systemudviklingslivscyklussen (SDLC); det kan udføres på hver ny del af koden, før den flettes ind i hovedgrenen, eller lige før du udgiver version 1.0 af din applikation. <https://en.wikipedia.org/wiki/Code_review?utm_source=chatgpt.com>

### **5. Oprettelse af Issues fra Team Collaborator (20 minutter)**

* **Introduktion til Issues:**
  + Forklar hvad "issues" er, og hvordan de kan bruges til at rapportere fejl, foreslå nye funktioner eller stille spørgsmål.
* **Opret en Issue fra en Team Collaborator:**
  + Bed eleverne om at arbejde i grupper, hvor hver gruppe fungerer som et team.
  + En elev fra hver gruppe skal skabe en issue i et andet gruppes repository, som beskriver et problem, giver feedback, eller foreslår forbedringer.
  + XBI Giv en demonstration af, hvordan man opretter en issue på GitHub.

### **6. Afslutning (10 minutter)**

* Diskuter fordelene ved at bruge GitHub, som platform for vedligeholdelse af kode
* Opfordre jer til at fortsætte med at bruge GitHub til deres fremtidige projekter og opgaver.

## **Reference**

Seitz, M. (2023, June 26). How to write a perfect README for your GitHub project. DEV Community. <https://dev.to/mfts/how-to-write-a-perfect-readme-for-your-github-project-59f2>

Graber, J. (2022, January 11). How to create issue templates for GitHub? Improve & Repeat. <https://improveandrepeat.com/2022/01/how-to-create-issue-templates-for-github/>

Axolo. (2024). GitHub branching and merging: Step-by-step guide. Axolo Blog. <https://axolo.co/blog/p/github-branching-and-merging-step-by-step-guide>