FW API 2021-W36 [08.09.-15.09.] Git / GitHub Einführung

**Einführungsaufgaben**

Aufgabe 1a Informieren

Lesen Sie die Theorie zu den verschiedenen git-Befehlen, den verschiedenen Workflows und wie man einen Merge durchführt.

**OneNote Seite**

[Git Terminal Befehle](onenote:../../../../BM%20IT%201%20-%20Know-how/BM%20IT%201%20-%20Know-how/d)%20Anleitungen/Git%20GitHub.one#klonen,%20commit%20und%20push%20Git%20Begriffe%20/%20Terminal&section-id={3FF95535-8452-40DE-809D-0A0D69931B0D}&page-id={DF42348A-5740-4700-82FA-93BEDC77398F}&base-path=https://d.docs.live.net/ecf086a16b405b46/OneNote Einzelordner)

**Online Ressourcen**

[git Buch](https://git-scm.com/book/de/v2/Erste-Schritte-Was-ist-Versionsverwaltung%3F)

[atlassian Dokumentation für Einsteiger](https://www.atlassian.com/de/git/tutorials/setting-up-a-repository)

[atlassian Dokumentation git-flow Workflow](https://www.atlassian.com/de/git/tutorials/comparing-workflows/gitflow-workflow)

Schreiben Sie eine Dokumentation über alles gelernte. Diese speichern Sie am Ende der zweiten Aufgabe im vorbereiteten Branch *documentation* des [Fachwoche Advanced-Repositorys](https://github.com/espas-bm-it/fw_git_advanced) ab. Erstellen Sie dort einen Ordner benannt nach ihrem Kürzel.

Sie können diese Dokumentation auch für ihr Lernjournal verwenden.

Aufgabe 1b Ein Repository erstellen

Erstellen Sie ein Repository auf dem GitHub-Konto für die eigene Webseite und verbinden Sie diese mit METANET. Dabei dienen folgende OneNote Seiten als Anleitung:

[GitHub Konto erstellen](onenote:../../../../BM%20IT%201%20-%20Know-how/BM%20IT%201%20-%20Know-how/d)%20Anleitungen/Git%20GitHub.one#GitHub%20Konto%20erstellen&section-id={3FF95535-8452-40DE-809D-0A0D69931B0D}&page-id={C3EBD248-F6C5-42EB-87E8-AE9233D9060E}&base-path=https://d.docs.live.net/ecf086a16b405b46/OneNote Einzelordner)

[GitHub Personal Access Token erstellen](onenote:https://d.docs.live.net/ecf086a16b405b46/OneNote%20Einzelordner/BM%20IT%201%20-%20Know-how/BM%20IT%201%20-%20Know-how/d)%20Anleitungen/GitHub.one#GitHub%20Personal%20Access%20Token%20erstellen&section-id={3FF95535-8452-40DE-809D-0A0D69931B0D}&page-id={CAF23084-E630-4000-94F4-0909A4C69BFC}&end)

[Subdomainhosting mit GitHub](onenote:https://d.docs.live.net/ecf086a16b405b46/OneNote%20Einzelordner/BM%20IT%201%20-%20Know-how/BM%20IT%201%20-%20Know-how/d)%20Anleitungen/xyz.bm-it.ch%20(Eigene%20Website)/xyz.bm-it.ch%20(METANET).one#Subdomainhosting%20mit%20GitHub&section-id={8BFAB47C-559E-4461-A1B8-C6993A4DF6A1}&page-id={2C8AD50F-FC8C-4DBA-BD85-6AA566409D33}&end)

Falls dies schon erledigt wurde, erstellen Sie ein weiteres Repository mit dem Namen "Fachwochen". Dieses Repository verwenden Sie für alle zukünftigen Fachwochen.

Workflow, was, wieviele gibt

Zentraler Workflow

Feature Branch Workflow

Git-flow Workflow

Forking Workflow

Merge, wie, gefahren

Fetch

Pull

Push

Commit

Add

Clone

**Intermediate Aufgaben**

Aufgabe 2a clone, commit und push

Klonen Sie das neu erstellte Repository in einer Programmierungsumgebung. Hierfür verwenden Sie bitte das Terminal.

Führen Sie auf der eigenen Webseiten Repository einen ersten Commit und push durch.

Beantworten Sie folgende Fragen:

* Was ist der Unterschied zwischen einem Commit und Push?

Pull-Requests sind mehr als bloße Benachrichtigungen – sie bieten ein dediziertes Forum zur Besprechung eines vorgeschlagenen Features. Sollten Probleme bei den Änderungen auftreten, können Teamkollegen im Pull-Request Feedback geben und das Feature sogar nachjustieren, indem sie nachfolgende Commits pushen. All diese Aktivitäten werden direkt im Pull-Request verfolgt.

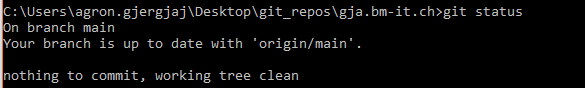
* Was ist ein "Working-Tree"?

Der Working Tree erlaubt es mehrere Branches zur selben Zeit auszuchecken. Linked Working Trees <git worktree add> können vom Main Working Tree entfernt werden <git worktree remove>.

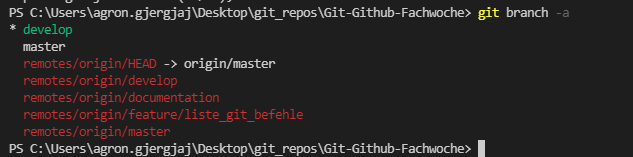
Aufgabe 2b Git Befehle

Führen Sie die Folgende Befehle im Terminal aus und halten Sie in Ihrem Lernjournal und Ihrer Dokumentation fest, was diese bewirken.

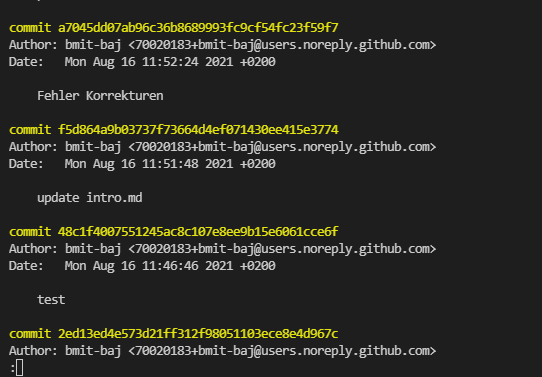
git status



git branch –a



git log (mit q kann man die Anfrage beenden)



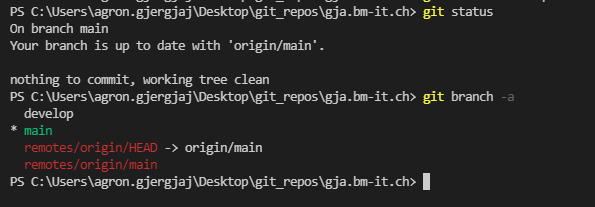
Aufgabe 2c Branches und Checkout

Erstellen Sie einen Develop-Branch im Repository Ihrer eigenen Webseite. Im Terminal kann das mit folgendem Befehl gemacht werden.

git branch develop



Führen Sie die git status und git branch -a Befehle aus. Was stellen Sie fest?

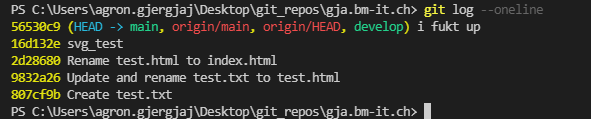


Der git checkout Befehl ist ein mächtiges Tool in einem Repository. Er ist für die Navigation zuständig und erlaubt zwischen den Branches zu wechseln. Dieser Befehl ist einer Suchfunktion ähnlich und verändert in einem Repository nichts. Er ist für die Versionskontrolle wichtig, weil man nicht nur auf Branches wechseln kann sondern auch auf einem Commit.

Suchen Sie einen Commit Hash-Code aus und führen Sie anschliessend darauf einen Checkout aus.

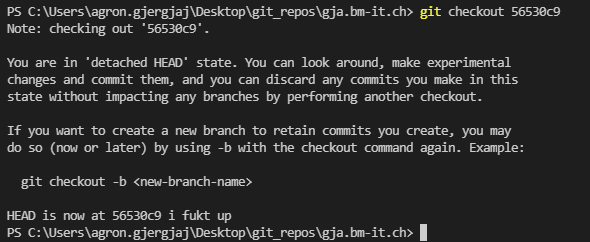
//anzeigen aller bisherigen Commits

git log --oneline

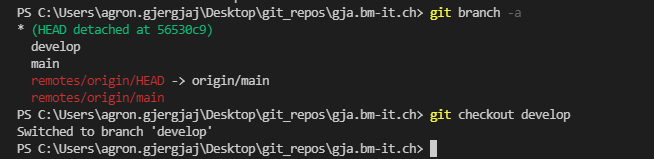


//wechsel zu bestimmten Branch/Commit

git checkout 64a51a0



Wie kommen Sie auf den Develop-Branch, den Sie erstellt haben?



**Advanced Aufgaben**

Einleitung

Ihre Aufgabe ist es im Team eine Webseite über Git/GitHub zu erstellen. Hierzu verwenden Sie ein vorbereitetes Repository und das darin enthaltene Template und die in den vorigen Aufgaben erarbeiteten Informationen.

Aufgabe 3a Feature-Branch erstellen

Erstellen Sie in ClickUp ein Feature und weisen Sie sich diesen Task zu. Klonen Sie das [Fachwoche-Advanced Repository](https://github.com/espas-bm-it/fw_git_advanced) und wechseln Sie in den Develop-Branch. Erstellen Sie für Ihren neuen Feature einen Branch aus develop und folgen dabei der Namenskonvention.

Hierzu den Kurzbefehl für das Terminal:

//wechseln zum Develop-Branch

git checkout develop

//erstellen und wechseln zum Feature-Branch

git checkout -b feature/*name\_des\_features*

Dieser Befehl erstellt einen neuen Branch und führt sofort einen Checkout darauf aus. Sie können dies mit git status überprüfen. Damit Ihre Mitentwickler Ihren neuen Branch sehen können, sollen Sie diesen noch pushen. Das Terminal wird folgenden Befehl erwarten:

git push --set-upstream origin feature/*name\_des\_features*

Aufgabe 3b Feature entwickeln

Nun geht es an die Arbeit. Kodieren Sie alleine oder zu zweit an diesem Feature.

Arbeiten Sie mit ClickUp, dieses ist Ihr stärkstes Kommunikationstool!

Aufgabe 3c Erstellen eines Pull-Requests

Wenn der Feature bereit ist, soll es in den Develop-Branch wieder hinzugefügt werden. Erstellen Sie auf GitHub einen Pull-Request und setzen Sie beide Git-Specialists im Review.

[atlassian Tutorial pull requests](https://www.atlassian.com/de/git/tutorials/making-a-pull-request)

[GitHub pull request](onenote:../../../../BM%20IT%201%20-%20Know-how/BM%20IT%201%20-%20Know-how/d)%20Anleitungen/Git%20GitHub.one#GitHub%20pull%20request&section-id={3FF95535-8452-40DE-809D-0A0D69931B0D}&page-id={E8B0D635-83E9-4D1B-AC60-E2810ECB831E}&base-path=https://d.docs.live.net/ecf086a16b405b46/OneNote Einzelordner)

Das erste Tutorial ist für BitBucket angepasst. In GitHub sieht es etwas anders aus. Bei Fragen fragen.

Der Pull-Request wird von den Git-Specialist überprüft. Wenn der Code sauber ist, wird er angenommen, ansonsten werden Sie gebeten nochmals darüber zu schauen. Sobald Ihr Pull-Request angenommen wird, können Sie den Merge durchführen.

Jetzt können Sie am nächsten Feature arbeiten.

Feature-List:

[Link zur Feature-List](https://app.clickup.com/1371872/v/b/19vq0-14161?pr=2445872) (ClickUp)