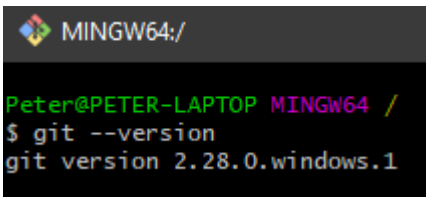
LU01a - Git

Installation

Installieren Sie Git inklusive der Git Bash

- https://git-scm.com/downloads

Öffnen Sie die Git Bash und führen Sie den folgenden Befehl aus, um Ihre Installation zu überprüfen:




```
git --version
```

Aliase

Mit dem Befehl `alias` können Sie einen alternativen Befehl für häufige Befehle definieren. Diese bleiben so lange gültig, bis die Konsole wieder geschlossen wird.

```
alias ..='cd ..'
alias ls='ls -1Ap --color=auto --show-control-chars'
```

Beim Start einer neuen Konsole wird in der Regel die Datei `.bash_profile` beziehungsweise `.bashrc` aufgerufen. Diese Datei befindet sich standardmässig unter `C:\Users\<USER>`. Wenn Sie Ihre Aliase also in dieser Datei hinzufügen, sind sie bei jeder neuen Bash-Konsole automatisch verfügbar.

Beispiel `.bashrc`:

```
...

# Meine benutzerdefinierten Aliase
alias git_sync='git stash; git pull; git push; git stash pop; git status'
```

Config

Stellen Sie sicher, dass Ihr Name `user.name` und Ihre E-Mail-Adresse `user.email` korrekt hinterlegt ist, damit Sie später auf GitHub zugreifen können.

```
git config --list
```

Sie können bei Bedarf die folgenden Befehle verwenden, um die entsprechenden Angaben zu setzen bzw. zu überschreiben.

```
git config --global user.name "<FIRST_NAME> <LAST_NAME>"
```

```
git config --global user.email "<MY_NAME@example.com>"
```

SSH

SSH-Schlüssel erlauben den Zugriff auf GitHub, ohne dass der Benutzername und das Passwort bei den Zugriffen erneut eingegeben werden müssen. Überprüfen Sie, ob auf Ihrem Gerät ein Schlüsselpaar existiert und ob der öffentliche Schlüssel auf GitHub hinterlegt ist.

```
ls ~/.ssh
```

Ob die Authentifikation möglich ist, kann mit folgendem Befehl überprüft werden:

```
ssh -T git@github.com
```

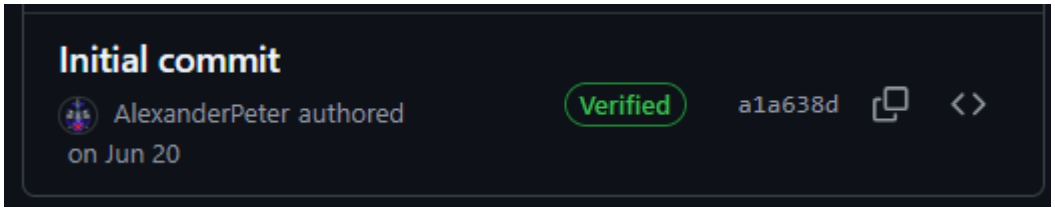
Erstellen Sie bei Bedarf einen neuen SSH-Key gemäss folgender Anleitung:

- https://docs.github.com/de/authentication/connecting-to-github-with-ssh
- https://docs.github.com/en/authentication/connecting-to-github-with-ssh

Der private Schlüssel bleibt immer auf dem Gerät und wird nicht herausgegeben, selbst wenn in einem Moodle-Quiz danach gefragt wird. Geben Sie anstelle des Schlüssels „ThePrivateKeyIsSecret“ als Antwort. Bei mehreren Geräten wird jeweils ein neues Schlüsselpaar generiert.

Signaturen

Schlüssel können ebenfalls verwendet werden, um commits zu signieren.



Dadurch kann jeder Benutzer verifizieren, dass der Commit von der Person erstellt wurde. Mehr Informationen finden Sie unter folgendem Link:

- https://docs.github.com/de/authentication/managing-commit-signature-verification
- https://docs.github.com/en/authentication/managing-commit-signature-verification

Git-Knigge

Es sollte in der Regel nicht mehr als ein Feature pro Commit gemacht werden. Jeder Commitete Codestand sollte aber auch compilierbar sein. Dadurch werden müssen voneinander abhängige Code-Teile miteinander commited werden. Unabhängige Code-Teile von anderen Features sollten hingegen separat commited werden. Bei einem Problem kann dadurch nur der tatsächlich problematische Teil rückgängig gemacht werden und andererseits können Änderungen anderweitit übernommen werden, ohne dass irrelevante Änderungen mitkommen (z. B. Cherry-Picking).

Commit-Nachrichten erfüllen einen wichtigen Zweck. Sie geben Auskunft über den Commit, so dass im Nachhinein verständlich ist, welche Änderungen vollzogen wurden, ohne die Details anzusehen. Schlechte Git Commit-Nachrichten wie im folgenden Beispiel erschweren das Verstehen und Nachvollziehen der Änderungen. Sie sind somit tabu.

Daher sind folgende Best Practices zu beachten!

- Den ersten Buchstaben des Commits grossschreiben
- Die Commit-Nachricht nicht mit einem Punkt beenden
- Die Imperativform verwenden
- Die Nachricht sollte folgenden Satz vervollständigen:

`If applied, this commit will` …

Beispiele:

- `If applied, this commit will` **Refactor subsystem X for readability**
- `If applied, this commit will` **Update getting started documentation**
- `If applied, this commit will` **Remove deprecated methods**
- `If applied, this commit will` **Release version 1.0.0**
- `If applied, this commit will` **Merge pull request #123 from user/branch**
- `If applied, this commit will` ~~Fixed bug with Y~~
- `If applied, this commit will` ~~Changing behavior of X~~
- `If applied, this commit will` ~~More fixes for broken stuff~~

- `If applied, this commit will` ~~Sweet new API methods~~

Weitere Best Practices können auf https://cbea.ms/git-commit/ eingesehen werden.

Notfall

Fehler können passieren. Für den Fall, dass etwas schiefgeht, ist folgende Seite ganz nützlich. Zum Beispiel, wenn man die Commit-Nachricht im Nachhinein abändern möchte. Es lohnt sich also mal ein Blick darauf zu werfen und den Link für Notfälle abzuspeichern.

- https://ohshitgit.com/de
- https://ohshitgit.com/