

SWT Uebung 01

Namen :

Ruilin Gao

Jing Chen

Chang Liu

Marlon Schacher

1、Vorbereitung des Git-Repositorys

2、Git-Begriffe

a)

1. Remote Branches sind einer der Referenzen (Zeiger) in Remote-Repositorys. Man kann eine vollständige, ausführliche Liste von Remote-Referenzen bekommen, wenn man die Anweisungen `git ls-remote <remote>` oder `git remote show <remote>` ausführt, für Remote-Branches sowie für weitere Informationen. Mit dem Behehl `git fetch <remote>` können wir unsere Arbeit mit einem bestimmten Remote synchronisieren.

2. Remote-tracking Branches sind Referenzen auf den Zustand von remote-Branches. Sie sind lokale Referenzen, die wir nicht manuell ändern können, sie werden automatisch für uns geändert, sobald wir irgendeine Netzwerkkommunikation durchführen. Mit dem Befehl `git fetch` werden die Remote-Tracking-Branches aktualisiert.

3. Tracking-Branches sind lokale Branches, die eine direkte Beziehung zu einem Remote-Branch haben. Sie werden nach dem Checkout eines lokalen Branches von einem Remote-Branch automatisch erzeugt. Wenn wir uns auf einem Tracking-Branch befinden und `git fetch` eingeben, weiß Git automatisch, von welchem Server Daten abzuholen sind und in welchen Branch diese einfließen sollen.

b)

Working tree ist ein Directory von dem Dateisystem, das relativ mit lokalen Repository ist. Man kann einfach eine neue Datei in working tree erstellen. Dann speichert man

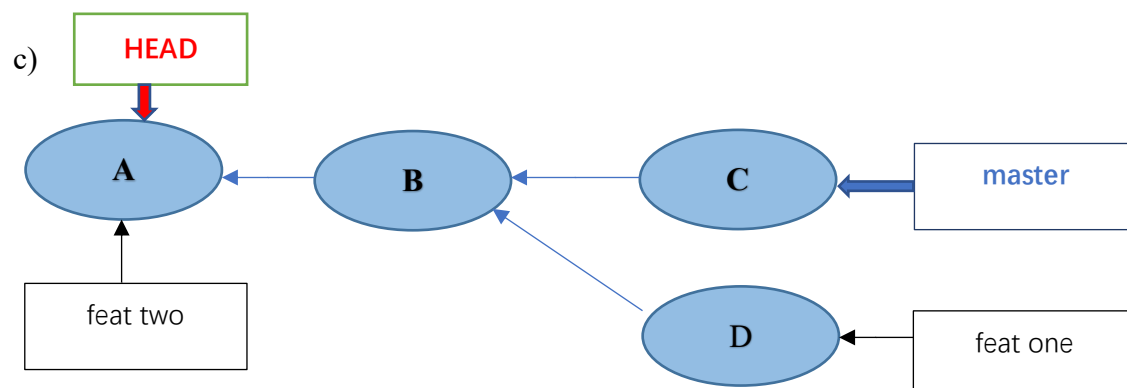
mit dem Befehl (git add) die neue erstellte Datei in Staging area. Staging area ist ein platz für Dateien, was schon bei git verfolgt werden. Mit dem Befehl (git commit) werden Dateien in Staging area auf lokales Repository übertragen. Local Repository ist ein Directory in unsere eigenen Laptops, wo kann Dateien von remote Repository heruntergeladen halten. Es haltet auch Dateien, was man nach remote Repository pushen möchte.

3、Ändern lokaler Repositorys

- a) git commit
git checkout B
git commit
git checkout master

- b) D darf nicht gepusht werden, weil Commit D nicht in einem Branch geschafft wird. Detached Head weist darauf hin, dass HEAD auf einer Commit zeigt. Um das Problem in 4(b) zu lösen, kann einfach ein neu Branch in Commit D erstellt.

git commit D
git checkout b- newbranch
git commit



4、Bedeutung von .gitignore

1.gitignore :

Die Gitignore-Datei wird verwendet, um die Änderungen an der angegebenen Datei oder dem angegebenen Ordner zu ignorieren. Die in der .gitignore-Datei aufgezeichneten Dateien oder Ordner können von git nicht verfolgt werden.

Mit anderen Worten, ignorierte Dateien werden nicht in Repository abgelegt.

Jedes Git-Projekt, das wir durchführen, benötigt eine ".gitignore" -Datei.

Mit dieser Datei können wir Git mitgeteilt, welche Dateien nicht zur Versionsverwaltung hinzugefügt werden müssen.

Aber es gilt nicht alle Fälle. Zum Beispiel, wenn die Dateien noch nicht “track” werden, dann kann diese Dateien ignoriert werden. Im Gegensatz wenn einige Dateien in die Versionsverwaltung aufgenommen worden sind, ist die Änderung von .gitignore ungültig. Dies liegt daran, dass wir zuvor direkt die Datei in git add.

Daher können die verfolgten Dateien nicht ignoriert werden. Dann wir können “target” und anderen Verzeichnissen auf der Webseite sehen.

Lösung :

```
git rm -r --cached .
```

```
git add .
```

```
git commit -m 'update .gitignore'
```

Deshlb ist es sehr wichtig und sinnvoll, das Anlegen einer .gitignore auch außerhalb eines Repositorys zu sein.