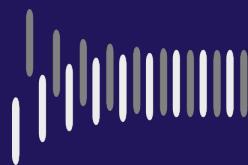




Git Grundlagen

Kursheft

Kapitel 5: Branching



DATAMICS
machine intelligence consulting services

Video #1 – Branching

falls du das Projekt noch nicht geklont/geforkt hast kannst du das von hier machen:

```
# git clone https://github.com/datamatics/football
```

```
cd football
```

```
git fetch
```

```
git pull
```

```
git status
```

```
git checkout master
```

Video #2 – Branch erstellen

```
pwd
```

```
git checkout -b feature-abc-12-Userlogin
```

```
# git branch feature-abc-12-Userlogin & git checkout feature-abc-12-Userlogin
```

```
echo "Mein neuer Code..." > user_login.py
```

```
git commit -m "abc-12: neuer user login" user_login.py
```

```
git status
```

```
git log
```

Video #3 – Branch wechseln (hotfix)

```
ls
```



```
git status
```

```
git checkout master
```

```
ls
```

```
git checkout -b hotfix
```

```
git branch
```

```
vim Dockerfile
```

```
# Füge ein Kommentar hinzu
```

```
git commit -m „fix docker file“ Dockerfile
```

Video #4 – Branch merge (hotfix)

```
git branch
```

```
ls
```

```
cat Dockerfile
```

```
git checkout master
```

```
cat Dockerfile
```

```
git merge hotfix
```

```
cat Dockerfile
```

```
git branch -d hotfix
```

```
git branch
```

```
git push origin master
```

Video #5 – Branch merge (feature)

```
git checkout feature-abc-12-Userlogin
```

```
echo „Neue Codezeile ...“ >> user_login.py
```

```
cat user_login.py
```

```
git status
```

```
git commit -a -m "abc-12: Erweiterung des user login"
```

```
git checkout master
```

```
git merge feature-abc-12-Userlogin
```

```
git branch -d feature-abc-12-Userlogin
```

```
git branch
```

```
git log
```

```
git push origin master
```

Video #6 – Challenge Merge

```
git clone https://github.com/datamatics/flair.git
```

```
cd flair
```

```
git branch
```

```
git branch -a
```



git checkout hotfix

git branch

git diff master hotfix

git checkout master

git merge hotfix

git branch -d hotfix

git log

git merge origin feature-101

git branch -d feature-101

git log

git branch

git push origin master

git push origin --delete feature-101

Video #7 – git merge conflict

git checkout -b feature-abc-13-Docker

vim Dockerfile # 2. Feature docker file



```
git commit -a -m „acb-13: feature dockerfile“
```

```
cat Dockerfile
```

```
git checkout master
```

```
cat Dockerfile
```

```
vim Dockerfile # 2. Fix Docker
```

```
git commit -m „fix docker file“ .
```

```
cat Dockerfile
```

```
git merge feature-abc-13-Docker
```

```
git status
```

```
vim Dockerfile # 2. Feature und Fix Dockerfile
```

```
git add .
```

```
git commit
```

```
git status
```

```
git branch -d feature-abc-13-Docker
```

```
git branch
```

Video #8 – Challenge Merge KONFLIKT

```
git clone https://github.com/datamics/flair.git
```

```
cd flair
```

```
git branch
```

```
git branch -a
```

```
git checkout feature-13
```

```
git branch
```

```
git diff master feature-13
```

```
git checkout master
```

```
git merge feature-13
```

```
git status
```

```
vim predict.py
```

```
git add .
```

```
git commit
```

```
git status
```

```
git branch -d feature-13
```

```
git push origin --delete feature-13
```

```
git log
```



```
git push origin master
```

Video #9/10 – git rebase

Teil 1:

```
pwd
```

```
git fetch
```

```
git pull
```

```
git status
```

```
git checkout master
```

```
git checkout -b feature-abc-20-Experiment
```

```
echo "Mein neuer Code..." > experiment.py
```

```
git commit -m "abc-20: neues Experiment," experiment.py
```

```
git status
```

```
git log
```

```
ls
```

```
git status
```

```
git checkout master
```

```
git branch
```



```
vim gfootball/replay.py
```

```
# Füge ein Kommentar hinzu
```

```
git commit -m „fix example usage“ gfootball/replay.py
```

Teil 2:

```
git checkout feature-abc-20-Experiment
```

```
git rebase master
```

```
git checkout master
```

```
git merge feature-abc-20-Experiment
```

```
git branch
```

```
git branch -d feature-abc-20-Experiment
```

```
git push origin master
```

Video #11 – git stash

```
git status
```

```
echo "Änderung für das stash Beispiel" >> datei_1.txt
```

```
git stash # git stash save
```

```
vim README.md
```



```
git commit -m „fix readme“ .
```

```
git stash apply
```

```
git commit -m „Fertig mit den stash Änderungen“ .
```

```
git stash list
```

```
git stash drop
```

```
git stash list
```