



git

183311088 Eren İNAL
173311021 Asım Tahir KARAKUŞ
173311008 Ali Vahap AYDIN

BRANCH(DALLANMA)

Dallar farklı özellikleri ayrı ayrı geliştirmek için kullanılır. Yeni bir depo oluşturduğunuzda master "varsayılan" daldır. Diğer dallar geliştirildikten sonra master'a birleştirilir.

```
git branch  
git branch yeni  
git checkout yeni  
git push origin yeni veya git push origin --all
```



DİFF

Versiyon kontrol sistemlerinde iki versiyon arasındaki değişikliklere İngilizce difference (fark) kelimesinin kısaltması olan diff denir.

İki commit arası farkı görmek için:

```
git diff hash1 hash2
```

İki branch arasındaki farkı görme için:

```
git diff master yenisurum
```

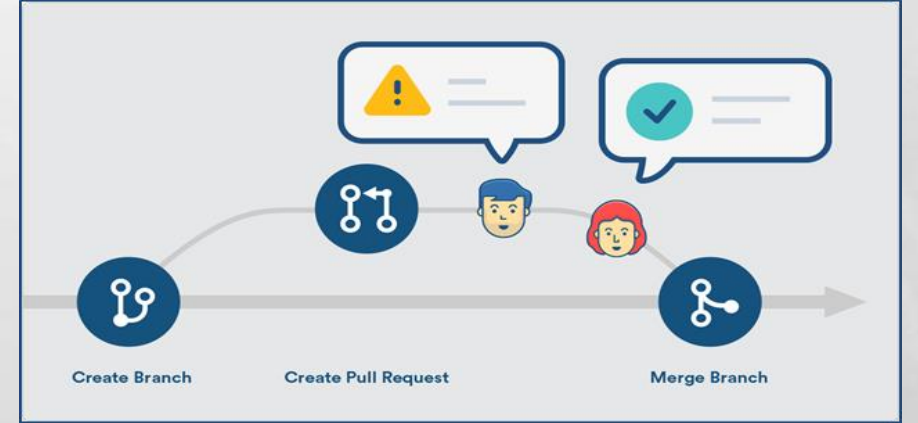
FETCH-PULL-MERGE

En son değişiklikleri yerel depomuza almak için fetch ve pull komutlarını kullanırız. Fakat fetch komutu yerel deponuzda bir değişiklik yapmaya kalkmaz. Pull komutu ise dosyaları hem yerel depomuza indirir hem de merge(birleştirme) yapmayı dener.

```
git pull  
git diff yenisurum master  
git merge master
```

PULL REQUEST

GitHub tarafında pull request yöntemi fork ve pull modeli ile gerçekleştirilmektedir. Yapılan katkılar bir projeye çekilmesi için gerekli proje ilk önce GitHub üzerinden fork (çatallama) işlemine tabi tutulmalıdır. Kopyalanan (fork'lanan) proje diğer geliştiricinin hesabında oluştuktan sonra yapılacak katkının önemliliğine göre projeye lokalde çalışılması için kendi makinesine bunu klonlaması gerekmektedir. Yapılan geliştirmeler ile oluşan işlem akışı Git aracılığıyla öncelikle kendi projesine (origin) gönderilir daha sonra GitHub servisiyle asıl projeye (upstream) katkılarının çekilmesi isteği ile sonlanır.



SORULAR



git



1) İki branch arasındaki farkı görebilmek için hangi komut kullanılır?

- A) git pull
- B) git checkout commithash1
- C) git branch master
- D) git diff commithash1 commithash2
- E) git diff master test

1) İki branch arasındaki farkı görebilmek için hangi komut kullanılır?

- A) git pull
- B) git checkout commithash1
- C) git branch master
- D) git diff commithash1 commithash2
- E) git diff master test

2) Test adlı branchı master branchıyla birleştirmek için hangi komut kullanılır?(master branchinde çalışmaktadır.)

- A) git checkout master
- B) git commit – m «commithash1»
- C) git merge test
- D) git push origin test
- E) git merge

2) Test adlı branchı master branchıyla birleştirmek için hangi komut kullanılır?(master branchinde çalışmaktadır.)

- A) git checkout master
- B) git commit – m «commithash1»
- C) git merge test
- D) git push origin test
- E) git merge

3) “Başka bir geliştiricinin projesi üzerinde çalışmak için önce ... yapılmalıdır. Geliştiricinin yapılan değişiklikleri onaylaması için ... yapmalıdır.”

Yukarıdaki boş alanlara sırasıyla ne gelmelidir?

- A) fork – pull request
- B) merge - pull
- C) pull – merge
- D) pull request – fork
- E) fork - merge

3) “Başka bir geliştiricinin projesi üzerinde çalışmak için önce ... yapılmalıdır. Geliştiricinin yapılan değişiklikleri onaylaması için ... yapmalıdır.”

Yukarıdaki boş alanlara sırasıyla ne gelmelidir?

- A) fork – pull request
- B) merge - pull
- C) pull – merge
- D) pull request – fork
- E) fork - merge

4)Aşağıdakilerden hangisi yaptığımız tüm commitleri görmemizi sağlar?

- A) git commit -m"mesaj"
- B) git log
- C) git pull
- D) git merge commit
- E) git config

4)Aşağıdakilerden hangisi yaptığımız tüm commitleri görmemizi sağlar?

A) git commit -m"mesaj"

B) git log

C) git pull

D) git merge commit

E) git config

5)Localimizde bulunan güncel veriyi birleştirme işlemi uygulamadan hangi komutla çekeriz?

- A) git fetch
- B) git pull
- C) git push origin --all
- D) git commit -m 'depo çekildi.'
- E) git merge master test

5)Localimizde bulunan güncel veriyi birleştirme işlemi uygulamadan hangi komutla çekeriz?

A) **git fetch**

B) git pull

C) git push origin --all

D) git commit -m 'depo çekildi.'

E) git merge master test

2)Aşağıdakilerden hangisi hem yeni branch oluşturup hemde o brancha geçiş yapar?

- A) git checkout yeni
- B) git log –oneline yeni
- C) git checkout –b yeni
- D) git branch yeni
- E) git merge yeni

2)Aşağıdakilerden hangisi hem yeni branch oluşturup hemde o brancha geçiş yapar?

- A) git checkout yeni
- B) git log --oneline yeni
- C) git checkout -b yeni
- D) git branch yeni
- E) git merge yeni