



git

163301015 - Bekir Can YUVA

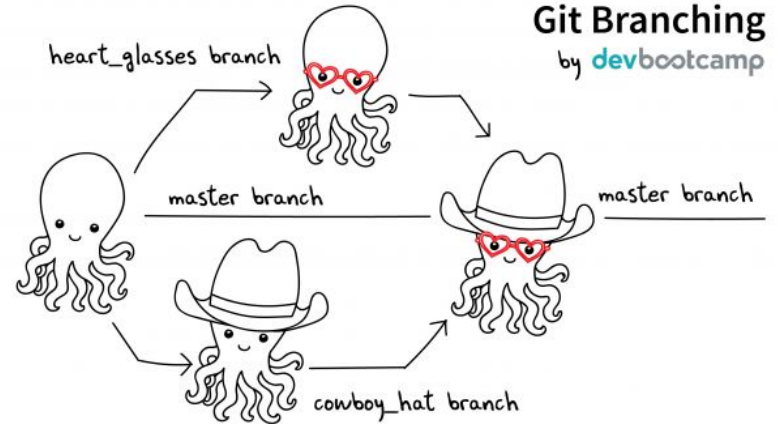
153301054 - Ramazan MUTLU

163301019 - Mücahit Sami RENKYORGANCI

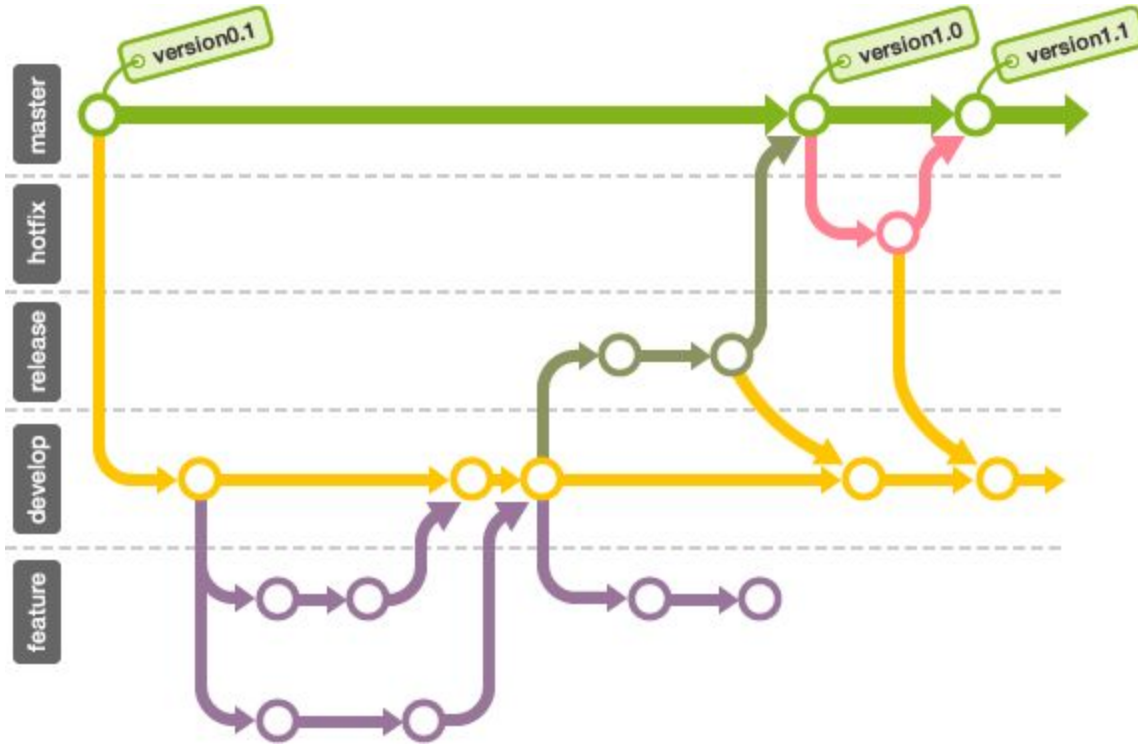
Branch (Dallanma)

Git'te bir dal, bu kayıtlardan birine işaret eden, yer değiştirebilen kıvrak bir imleçten ibarettir. Git'teki varsayılan dal adı master'dır. İlk kaydı yaptığınızda, son yaptığınız kaydı gösteren bir master dalına sahip olursunuz. Her kayıt yaptığınızda dal otomatik olarak son kaydı göstermek üzere hareket eder. Yeni kayıtlarla ilerlemenizi sağlayan yeni bir imleç oluşturulur.

```
git branch
git branch yeni
git checkout yeni
git push origin yeni
```



Branch (Dallanma)



Diff

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

İki commit arasındaki farkları görmek için :

```
git diff hash1 hash2
```

İki branch arasındaki farkları görmek için :

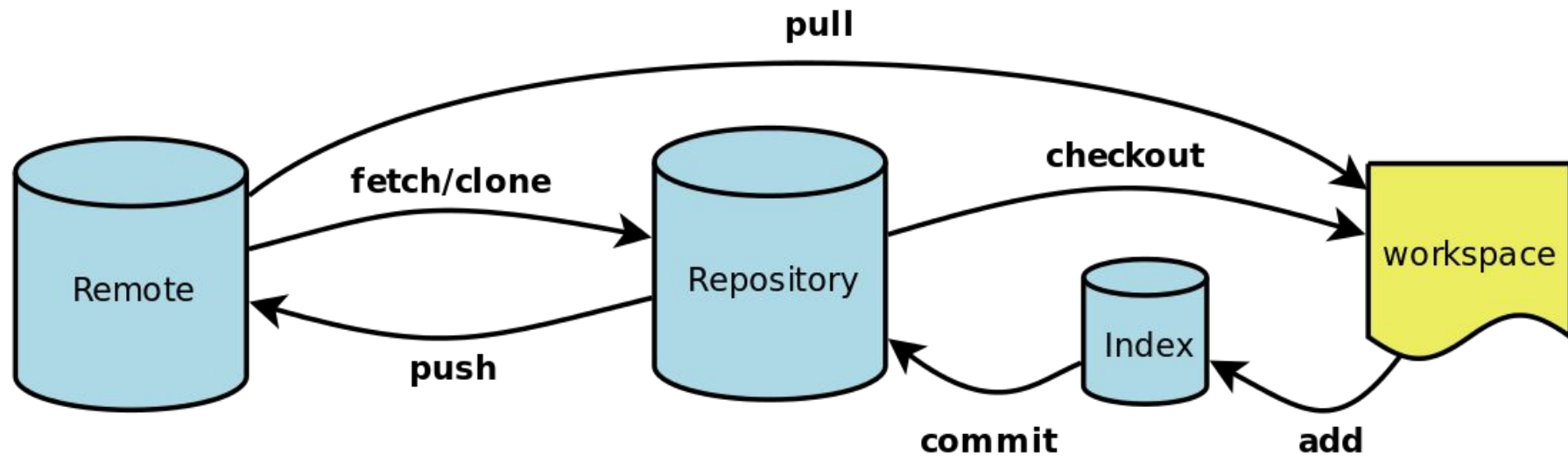
```
git diff master yenisurum
```

Pull - Merge

En son değişiklikleri (commit) yerel deponuza almak için “**git pull**” komutunu çalıştırın. Bu değişiklikleri al (fetch) ve birleştir (merge) yapacaktır. Aktif dala (örn. master) başka bir dalı birleştirmek için “**git merge <dal>**” her iki durumda da git değişiklikleri otomatik birleştirmeyi (auto-merge) dener. Maalesef, bu her zaman mümkün olmaz ve çakışmalarla (conflict) sonuçlanır. Git tarafından gösterilen dosyaları elle düzenleyerek bu çakışmaları birleştirmek size düşer. Değişikliklerden sonra, dosyaları eklemek için “**git add <dosyaadı>**” değişiklikleri birleştirmeden önce, önizleme yapmak için “**git diff <kaynak_dal> <hedef_dal>**”

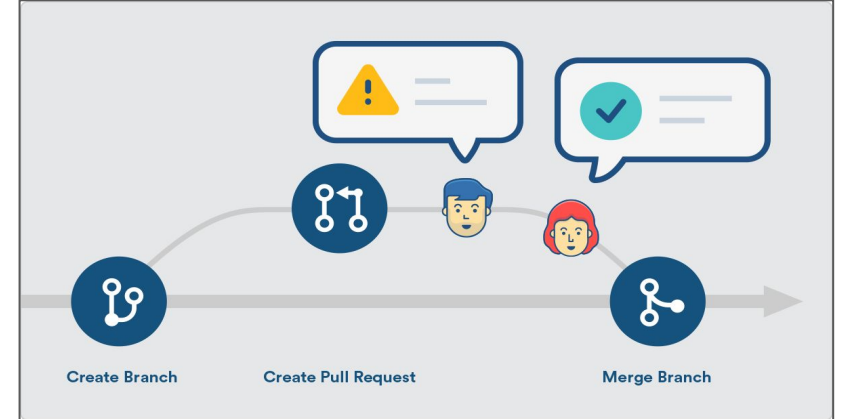
```
git pull
git diff yenisurum master
git merge master
```

Pull - Merge



Pull Request

GitHub tarafında pull request yöntemi fork ve pull modeli ile gerçekleşmektedir. Yapılan katkıların 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.



Sorularınız?



Sorularımız



1) İki commit arasındaki farkları görmek için hangi git komutu kullanılır?

A-) git checkout branch2

B-) git diff master branch2

C-) git diff commit1hash commit2hash

D-) git merge commit2

E-) git pull

1) İki commit arasındaki farkları görmek için hangi git komutu kullanılır?

A-) git checkout branch2

B-) git diff master branch2

☒ C-) git diff commit1hash commit2hash

D-) git merge commit2

E-) git pull

2) deneme adlı branch'ı master branch'ıyla birleştirmek için hangi git komutu kullanılmalıdır? (deneme branch'ında çalışılmaktadır)

A-) git merge deneme master

B-) git push master deneme

C-) git checkout master

D-) git merge master

E-) git branch deneme

2) deneme adlı branch'ı master branch'ıyla birleştirmek için hangi git komutu kullanılmalıdır? (deneme branch'ında çalışılmaktadır)

A-) git merge deneme master

B-) git push master deneme

C-) git checkout master

☒ D-) git merge master

E-) git branch deneme

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

3) Yukarıdaki boş alanlara sırasıyla ne yazılmalıdır ?

A-) fork - pull request

B-) merge - pull request

C-) fork - merge

D-) pull request - fork

E-) fork - fork

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

3) Yukarıdaki boş alanlara sırasıyla ne yazılmalıdır ?

- ☒ A-) fork - pull request
- ☐ B-) merge - pull request
- ☐ C-) fork - merge
- ☐ D-) pull request - fork
- ☐ E-) fork - fork

4) Depoda bulunan bütün dalları ve aktif dalı gösteren git komutu nedir?

A-) git merge

B-) git checkout

C-) git fetch

D-) git rebase

E-) git branch

4) Depoda bulunan bütün dalları ve aktif dalı gösteren git komutu nedir?

A-) git merge

B-) git checkout

C-) git fetch

D-) git rebase

☒ E-) git branch

Haftaya g r   mek  zere

