

GIT & GITHUB Notları...

GİT: Bilgisayarınızın kendi içindeki (LOCAL deki) sanal alanı

GİT:Version kontrol sistemi

BRANCH:Bilgisayarınızdaki yaptığınız çalışmayı kaydettiğiniz dal (hat)

MERGE:Dallar arası geçiş

MASTER:Ana dal(Ana hat)

Version Kontrol sistemi => 1)Yerel Kontrol sistemi: kendi bilgisayarınızdaki (localinizdeki ) sistem

Localde DEPOLAMA Yapılmaz !!!

YEREL VKS ==> Kullanımı tercih edilmez

DAĞITIK VKS: (Bir sunucu ve birçok kullanıcının aynı anda kullandığı sistem)Her kullanıcının tüm verilere ulaşma imkanı sunar, yani her kı yedegidir. örneğin :git

PUBLIC: Görünürlüğü herkese açık repository

DESCRIPTION: Oluşturduğunuz repository nin açıklama kısmı (doldurulması zorunlu değildir)

COMMIT CHANGE:Herhangi bir çalışma versiyon olarak kaydedildiğinde commit değişikliği yapmak istediğinizde kullanılır

Commit Store: Commitlerin , değişikliklerin tutulduğu market:)

↑ 931 new messages ×

Working Space:Çalışma alanı

Staging Area:Çalışmalarımızın değişikliklerinin tamamlanması için beklediği alan(aktif dinlenme alanı)

[1:23 PM] CMD Komut Notları...

CLS:Temizlik komutumuz

mkdir:Klasör oluşturma

echo mesaj(dosyanın içerisinde yazılacak mesaj) > dosyaadı.dosyauzantısı

more: dosya içerisindeki mesajı gösterir

ren:dosya ismi değiştirme

del:silme

ls:listeleme(mac)

cat > dosyaadı.dosyauzantısı :Dosya oluşturma

ctrl+d : çıkış kısayolu

mv: Dosya ismi degistirme(mac

git commit -m "commit mesajı" -s

git init : klasörünüz içerisinde çalışmalrınız bulunmaktadır == Working Space

git config:Git kurulumundan sonra yazmamız gereken bir komuttur. Hangi servis üzerinde çalışıyorsak o ayarlar ilk burada yapılır.

git log , Git versiyon kontrol sisteminde oluşturulmuş commit geçmişini tarihsel olarak sondan başa doğru sırasıyla konsola yazan bir Git komutu oldukça sıkıcı bir görünüme sahip olan git log komutunu birkaç parametre ile daha tatlı bir hale getirebiliyoruz.

git log: Tek satırda özet

git add . : çalışmada değişiklikleri ekler

GIT & GITHUB Notları...GİT: Bilgisayarınızın kendi içindeki (LOCAL deki) sanal alanıGİT:Version kontrol sistemiBRANCH:Bilgisayarınızdaki yaptığınız çalışmayı kaydettiğiniz dal (hat)MERGE:Dallar arası geçişMASTER:Ana dal(Ana hat)Version Kontrol sistemi =>. 1)Yerel Kontrol sistemi: kendi bilgisayarınızdaki (localimizdeki ) sistemLocalde DEPOLAMA Yapılmaz !!!YEREL VKS ==> Kullanımı tercih edilmezDAĞITIK VKS: (Bir sunucu ve birçok kullanıcının aynı anda kullandığı sistem)Her kullanıcının tüm verilere ulaşma imkanı sunar, yani her kullanıcı aynı zamanda bir veri yedegidir. örneğin :gitPUBLİC: Görünürlüğü herkese açık repositoryDESCRIPTION: Olusturdugunuz repository nin açıklama kısmı (doldurulması zorunlu değildir)COMMIT CHANGE:yaptığımız çalışmayı version olarak kaydettikten sonra değişiklik yapmak istediğimizde kullanılırREPOSITORY: Localimizdeki Çalışmalarımızı kaydettiğimiz sanal alanREPOSITORY nin ÇALIŞMA ALANI : Çalışma yaptığımız klasörümüzün içinde .git klasörü oluşturur(klasörümüz gitHub a bağlantısını oluştugunda oluşur),Kendi localimizdeki bilgilerde buradadırSTAGING AREA:Çalışmamızı bitirdikten sonra 2. adım olarak bu çalışmayı commitleyip(paketleme olarak düşünebiliriz)Gecici olarak beklettiğimiz alandırGIT ADD ." komutu : yaptığımız çalışmamızı n üzerine değişiklikleri eklemeye kullanılırCommit Store: Commitlerin , değişikliklerin tutulduğu market:)Working Space:Çalışma alanıStaging Area:Çalışmalarımızın değişikliklerinin tamamlanması için beklediği alan(aktif dinlenme alanı)

[1:23 PM] CMD Komut Notları...CLS:Temizlik komutumuzmkdir:Klasör oluşturmaecho mesaj(dosyanın içerisinde yazılacak mesaj) > dosyaadı.dosyauzantısımore: dosya içerisindeki mesajı gösterirren:dosya ismi değiştirmedel:silmels:listeleme(mac)cat > dosyaadı.dosyauzantısı :Dosya oluşturmactrl+d : çıkış kısayolumv: Dosya ismi degistirme(macgit version: git surumunuzu gösterirgit config -global user.name "Ali Gel": Commitler buradaki isim ve mail ile eşleştiği için, team şeklindeki çalışmalarımızda , bizden giden bilgileri bu isimde görüntülergit init : klasörünüz içerisinde çalışmalrınız bulunmaktadır == Working Spacegit config:Git kurulumundan sonra yazmamız gereken bir komuttur. Hangi servis üzerinde çalışıyorsak o ayarlar ilk burada yapılır.git log , Git versiyon kontrol sisteminde oluşturulmuş commit geçmişini tarihsel olarak sondan başa doğru sırasıyla konsola yazan bir Git komutudur. Tek başına kullanıldığında oldukça sıkıcı bir görünüme sahip olan git log komutunu birkaç parametre ile daha tatlı bir hale getirebiliyoruz.git log: Tek satırda özetgit add . : çalışmada değişiklikleri eklerHATA SEBEBLERİ: 1) dosya oluşturmamış olabilirsiniz 2) dosya isminiz yavuz olmayabilir 3) kodu yazarken yazım hatası yamış olabilirsiniz 4)dosya nızı olusturduktan sonra git init yapmamış olabilirsiniz.