Dökümanlarınızın tutulduğu, kodlarınızın tutulduğu bir repository üzerinde çalışacaksınız. Ve o repostory üzerinden version kontrol sistemi üzerinden değişikliklerinizi takip edeceksiniz, yeri gelecek yeni çalışmalar yapacaksınız bunu ana dosyaya, repostorye ekleme yapacaksınız gibi günlük IT dünyasında olmazsa olmazınız olacak.

Git nedir, neden kullanıyoruz ve nasıl bir avantaj sağlıyor bize bugün bunlardan bahsedeceğiz.

Bir terminal üzerinden bu dersi işleyeceğiz.

1. Git tanıtımı Git iş akışı nasıl yapılıyor--- bunları anladıktan sonra local repomuz üzerinde bir repostory oluşturacağız. (internete çıkmadan)
2. Branch ları anlayacağız, merge işlemine bakacağız, uyuşmazlıklar çıkabilir
3. Remote repo, git ve github farklı mıdır, farklı arayüzler var mı ona bakacağız
4. Public olanları fork edeceğiz,
5. İş yerlerinde bunları nasıl kullanıyoruz, en çok hangi işlemler yapılıyor bunları göreceğiz

\*\*\*Bugünkü dersimizde neler yapacağız

\*\* Git hakkında bilginiz var mı

Bir den fazla kişilerin bir program üzerinde aynı anda çalışmasını sağlar ve bunu internet olmadan yapıyoruz.

Git nedir ?

Open source, Linux torvıls çıkarmıştır.

Farklı kavramlar var, distrübütit bunlara da bakacağız.

Versiyon Kontrol sistemi ne demek, neden ihtiyaç duyuyoruz, bunun önemi nedir diye sorabilirler.

* Track changes , Git üzerinde, ya da bilgisayarınızda bir repo oluşturduğunuzda ve o oluşturduğunuz bir reponun altına bir dosya eklediğinizde artık GİT otomatik olarak o dosyaları traklıyor sizin yerinize.
* Unlimited Undo / Redo, dediğimiz limitsiz bir şekilde ileri geri gidebiliyorsunuz, istediğiniz zaman istediğiniz sürümüne, versiyona dönebiliyorsunuz.
* Time travel , , Dosya içinde yaptığınız tüm değişikliklerle beraber. Ve bir sınır yok 3 yıl önce 5 yıl önce diye bir sınır yok
* Colobrative, birden fazla kişi birbirlerini etkilemeden aynı doküman üzerinde çalışmasını sağlıyor. doküman ana merkezde duran dökümanı kimse değiştiremiyor onu kopyalayıp onun üzerinde çalışıyoruz.
* Bu değişikliklerde de ne değişiklik yapıldı, ne zaman yaptı, niçin yaptı ve kim hangi değişikliği yaptı. İzinsiz bilgilere ulaşılmıyor zaten, izin verilenler ulaşıyor.

Ana dosyaya herkes ekleme yapamayacak.

Şekil üzerinde açıklamak gerekirse;

Ana server da dosyanız olur……

Biz burada yapacağımız günlük şeyler

* Bir sıfırdan repo oluşturabiliriz
* Bir mevcut repoyu kendi lokalinize çekip kaydedebilirsiniz
* Bunun üzerinde bazı işlemler yapabilirsiniz
* Ve bu yaptığınız işlemleri kaydettikten sonra uzaktaki repoya push edebilirsiniz

Resimle anlatıyoruz—slayt 11

3 tane commit ile 3 tane dönülebilir nokta oluşturmuş oluyoruz.

\*\*

Zaman içinde dosyalarda yapılan değişiklikleri izler ve kaydeder

Her tür dosyayı izleyebilir, ancak en yaygın olarak kod için kullanılır

Tarih, yazar ve değişikliği açıklayan bir mesaj gibi ekstra bilgiler içerir

Dosyaların önceki sürümlerini herhangi bir zamanda alabilir

Yanlışlıkla silinen dosyaları kurtarın

Yerel olarak veya başkalarıyla ortaklaşa kullanılabilir

Git Nedir?

Git bir yazılımdı

İçerik traklamamıza yarıyordu

Dağıtık bir version kontrol sistemiydi

Linus Torvıls tarafından geliştirilmişti

Neden Git'e ihtiyacımız var?

* Backup, arşivleme, version ve history olarak (tarihçesine bakabiliyoruz, zaman bakımından)
* İstediğimiz tarihe gidebilme özelliği
* Karşılaştırabilme
* Takım halinde ortak çalışıp kodu oluşturabiliyorsunuz
* Ve kodu review edebiliyorsunuz, birisi yaptığı kod için pull request gönderdiğinde burada herkes görüp uygun olur ya da burada şunu ekleseydin daha iyi olurdu gibi yorumlar yapabiliyor
* Suçlama diyoruz burada da kimin nerede neyi yaptığını bildiğimiz için çalışan bir sistem artık çalışmıyorsa ya da en önemli kod yoksa bununla ilgili kimin ne yaptığı ispatlanabiliyor.

Git Basic

Backup

Burada genelde developerler günlük olarak pull yaptıkları için çökme olmaz ama olursa da bu bilgisayarlardan kopyalanabilir

Collobration

Bu taskları branchlara ayırabiliyoruz bunun üzerinde çalışma yapıp daha sonra birleştirilebiliyor

Repostory nedir

Repo nedir, Git kullanarak verilerinizi, dosyalarınızı versiyonlamak istiyorsanız işe repo oluşturarak başlamanız gerekiyor.

Repo iki türlü oluşturulabiliyor

* Github üzerinden oluşturabiliyorsunuz kendi localinize çekebiliyorsunuz
* Ya da git üzerinde kendi lokalinizde bir klasör açarak git init yapıyoruz, otomatik olarak .git oluşmuş oluyor ve bunun sayesinde dosyalarınız traklanmış oluyor

Uygulamaları yapıyoruz

git --version

git config --global user.name “Your Name” t

git config --global user.email “Your Email” t

git config --global core.editor “vim” t

git config –list

Project klasörü oluştur ve içine bir klasör daha oluştur, first proje

Git init yap

Bunun içindeki ve altındaki tüm dosya ve klasörleri traclar

Git help

Git commit - -help

Git clone adress yapalım

Workflow (İş Akışı)

Bir alışveriş arabası ya da roket fırlatması gibi.

Siz satın alana kadar eşyalar sizin değildir.

kayıt olmaya gidin ve ödeyin, o zaman sizindir, depo buzdolabınız veya kilerinizdir.

**Committed,** Son kaydetme anlık görüntüsündeki değiştirilmemiş değişiklikler

**Modified**, Son kayıt anlık görüntüsünden bu yana dosyalarda yapılan değişiklikler

**Staged**, Bir sonraki işleme anlık görüntüsüne eklenmek üzere işaretlenen değişiklikler

Touch index.html

İçini doldur, kaydet

Git status yap

Git add index.html git add . git -A tüm değişiklikleri

Git commit -m “ddddddd”

Git log ile commit listesi

Git commit -am “hhhh” dosyayı add yapmadan direk repoya atar. İlk defa add yaptıktan sonra kullanabilirim.

Git commit –amend comit mesajında değişiklik yapmak için nano içerisine girip yapıyoruz

Git log daki commitler arasında gezinebilirim, şu an master branch ındayım istediğim commit e nasıl geçebilirim

Git checkout ad\_numbergit

metin, çizgi, diyagram, el yazısı içeren bir resim

Açıklama otomatik olarak oluşturuldu

<https://git-school.github.io/visualizing-git/#free>