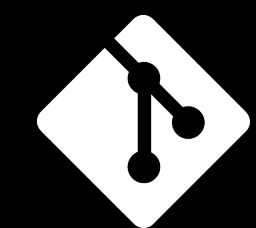


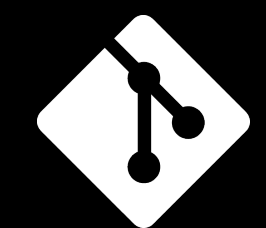
git

Kod Yönetimi

Git



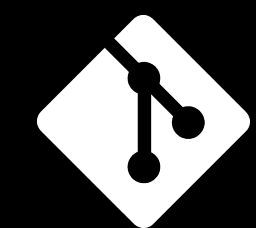
Versiyon Kontrol Sistemi Nedir?



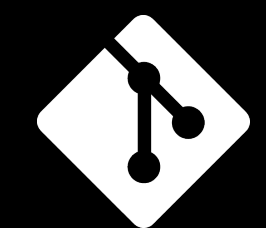
Versiyon Kontrol Sistemi Nedir?

“Yazılım ekiplerinin zaman içinde kaynak kodda yapılan değişiklikleri yönetmesine yardımcı olan yazılım araçlarıdır.”

“Versiyon kontrol, belirli versiyonların daha sonra çağrılabilmesi için zaman içerisinde bir dosya veya dosya grubundaki değişiklikleri kaydeden bir sistemdir.”



Versiyon Kontrol Sistemi Ne Sağlar?

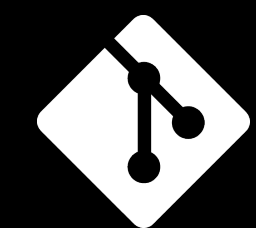


Versiyon Kontrol Sistemi Ne Sağlar?

Takip Edilebilirlik

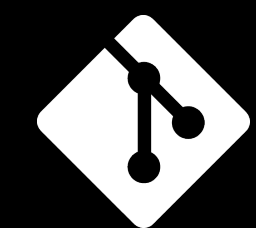
Güven

Kolaylık



Git Nedir?

"Git, yazılım geliştirme süreçlerinde kullanılan, hız odaklı, dağıtık çalışan bir sürüm kontrol ve kaynak kod yönetim sistemidir."

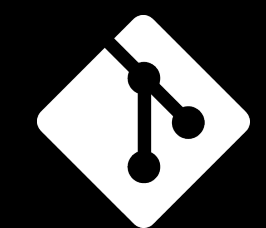


İlk Git

Linux Çekirdeği

2005

Linus Torvalds

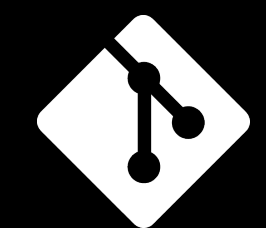


Linus Benedict Torvalds

Yazılım Mühendisi

Linux Çekirdeği Geliştirici ve Proje Yöneticisi

Açık Kaynak Geliştirme Laboratuvarı



Tarihçe

Linux Çekirdeği

BitKeeper

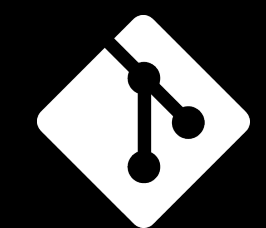
Andrew Tridgell

Tersine Mühendislik

Larry McVoy

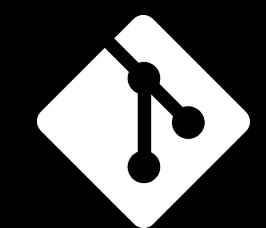
Telif Hakları

Git



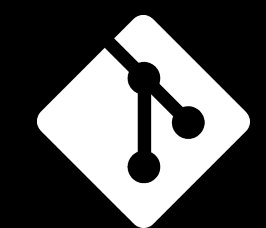
Tarihçe

Linus Torvalds



Tarihçe

Dağıtık Yapı

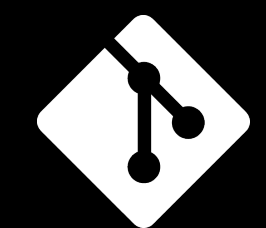


Tarihçe

30 sn

250 Tekrar

3 sn

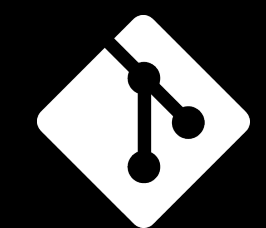


Tarihçe

Kararsızlık

Dağıtık Yapı

Güvenlik



Tarihçe

Tasarım Kriterleri

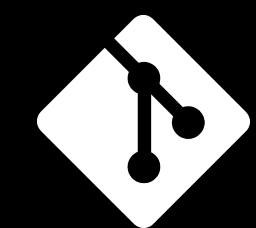
Dağıtık Yapı

Kararsızlık

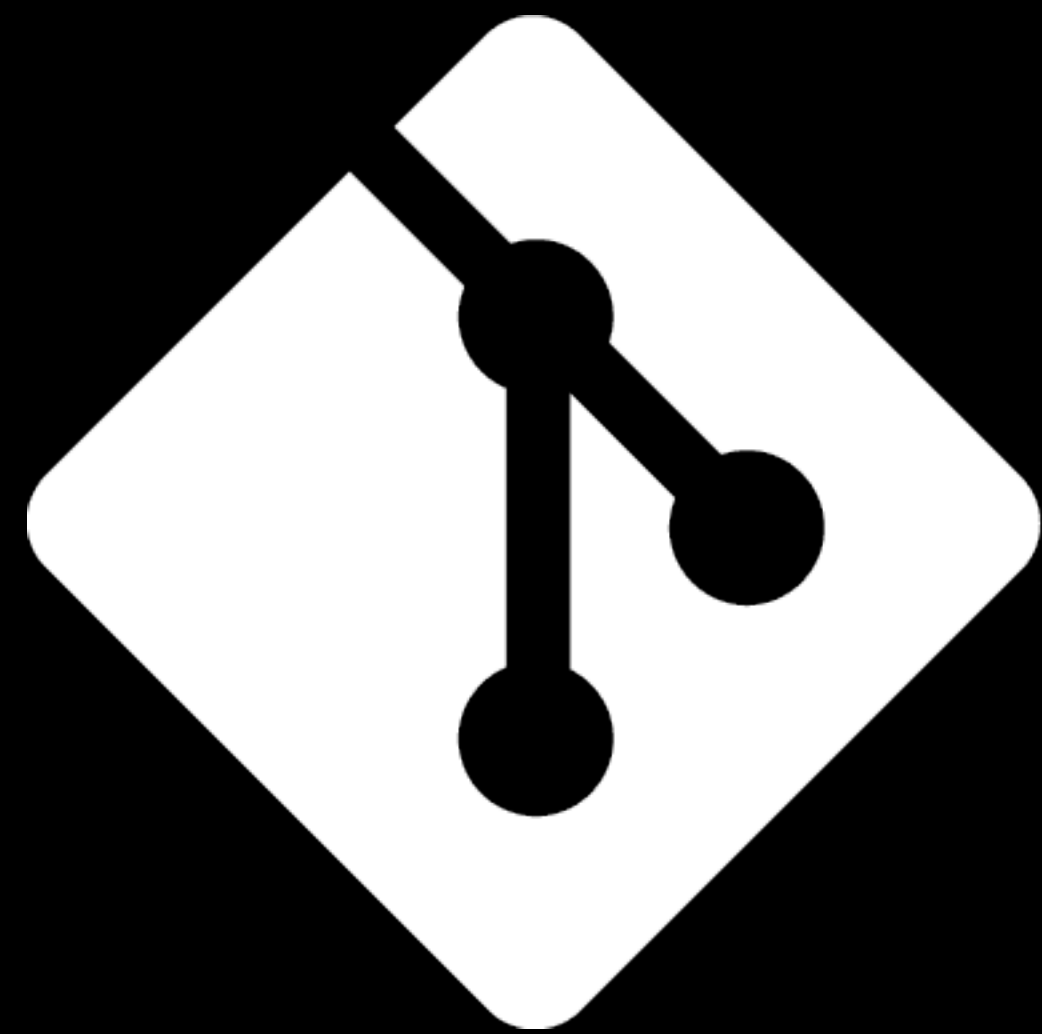
Güvenlik

“Monotone”? PERFORMANS!

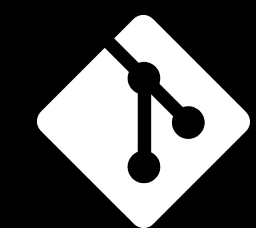
2.6.12-rc2



Neden

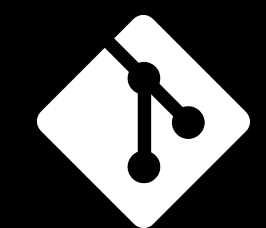


git



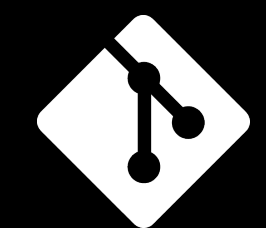
Neden Git?

Versiyon Yönetimi



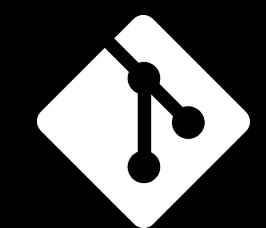
Neden Git?

Hız ve Boyut



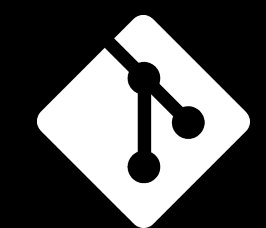
Neden Git?

Takım Çalışması



Neden Git?

Online/Offline Kullanım

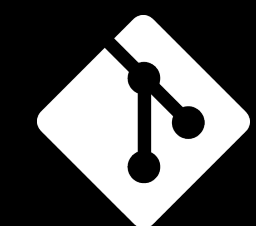


Git Nasıl Kurulur?

<https://git-scm.com>

```
git config --global user.name "John Doe"
```

```
git config --global user.email johndoe@example.com
```

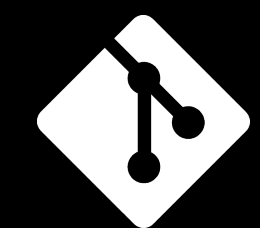


Git Kavramları

> init

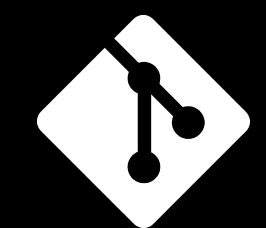
> clone

> repository



Git Kavramları

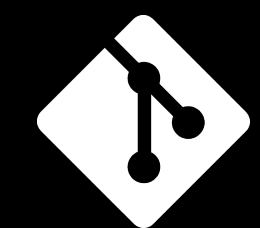
> .git



Git Kavramları

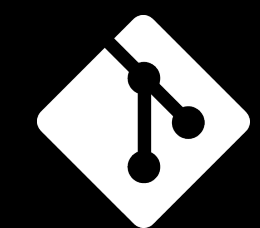
> master

> branch



Git Kavramları

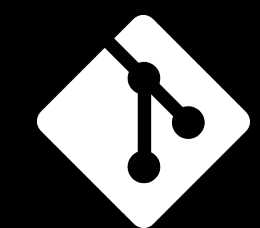
> checkout



Git Kavramları

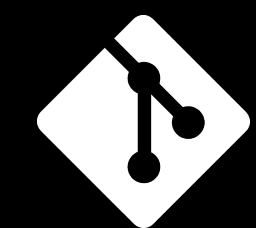
> add

> commit



Git Kavramları

> head

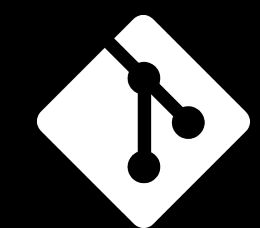


Git Kavramları

> modified

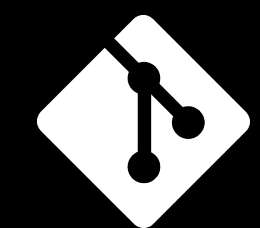
> tracked

> untracked



Git Kavramları

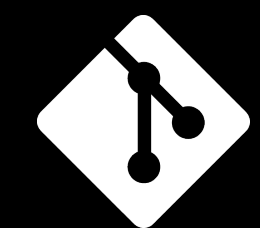
> .gitignore



Git Kavramları

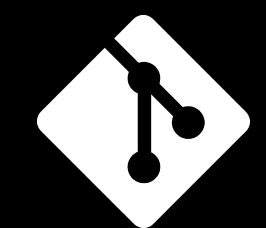
> staged

> stash



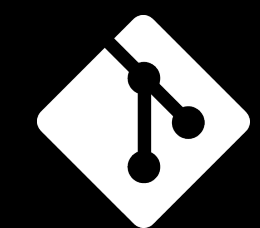
Git Kavramları

> grep



Git Kavramları

> hash



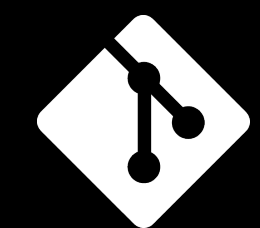
Git Kavramları

> status

> log

> diff

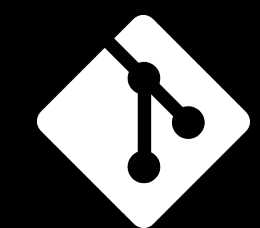
> undo



Git Kavramları

> reset

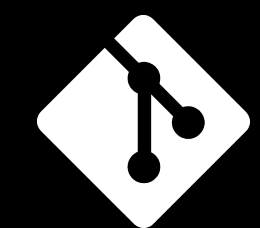
> revert



Git Kavramları

> merge

> rebase



Git Kavramları

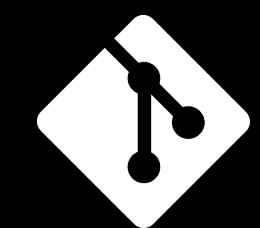
> remote

> origin

> fetch

> pull

> push



Git GUI

GitHub

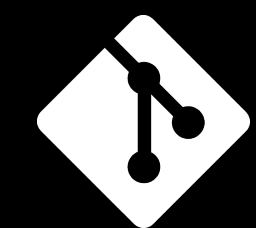
Sourcetree

GitLab

Bitbucket

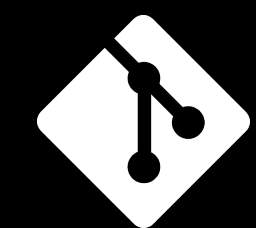
Visual Studio

IntelliJ IDEA



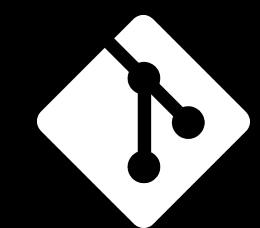
Git GUI

GitHub



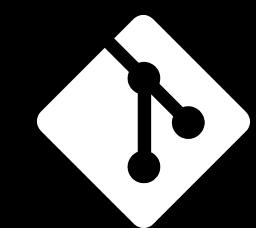
Git Alternatifleri

SVN



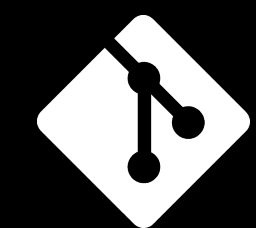
Git Alternatifleri

ClearCase



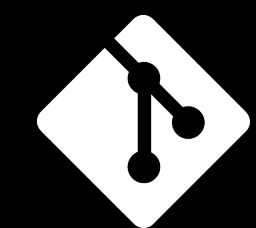
Git Alternatifleri

Mercurial



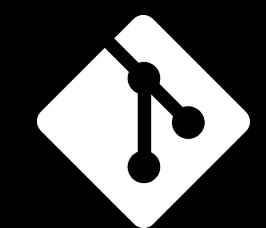
Git Alternatifleri

TFS (TFVC)



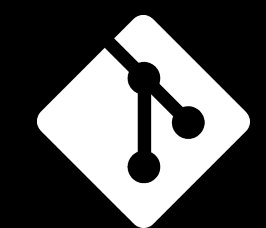
“Commit Message” Önerileri

Konuyu gövdeden boş bir satırla ayırın



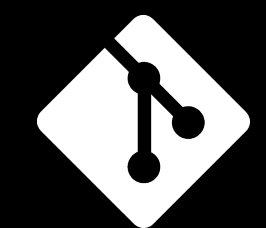
“Commit Message” Önerileri

Konu satırını 50 karakter ile sınırlayın



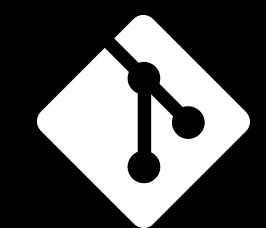
“Commit Message” Önerileri

Konu satırını nokta ile sonlandırmayın



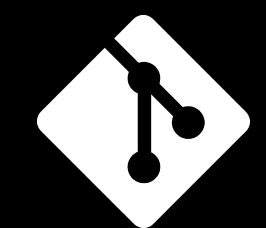
“Commit Message” Önerileri

Konu satırında emir kipi kullanın



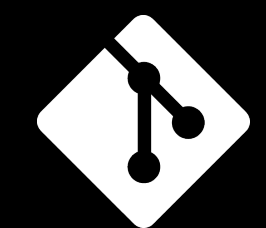
“Commit Message” Önerileri

Gövdeyi 72 karakter ile sarmalayın



“Commit Message” Önerileri

Neyi, neden ve nasıl olduğunu açıklamak için gövdeyi kullanın



“Commit Message” Örneği

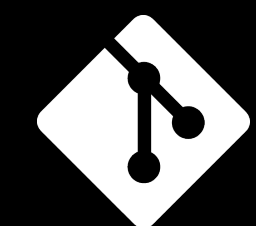
Simplify `serialize.h`'s exception handling

Remove the `'state'` and `'exceptmask'` from `serialize.h`'s stream implementations, as well as related methods.

As `exceptmask` always included `'failbit'`, and `setstate` was always called with `bits = failbit`, all it did was immediately raise an exception. Get rid of those variables, and replace the `setstate` with direct exception throwing (which also removes some dead code).

As a result, `good()` is never reached after a failure (there are only 2 calls, one of which is in tests), and can just be replaced by `!eof()`.

`fail()`, `clear(n)` and `exceptions()` are just never called. Delete them.

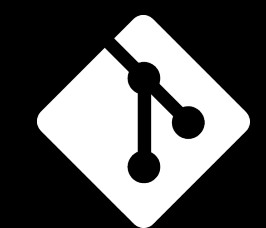


Proje Örneği

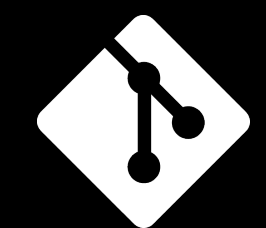
Staj2021

Local Proje Oluşturma

GitHub Fork

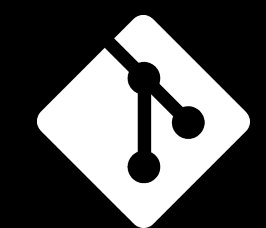


Sorular?



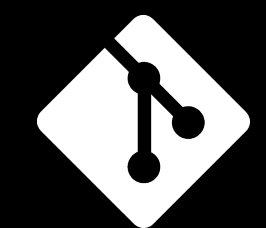
Sorular?

Git tasarım kriterleri nelerdir?



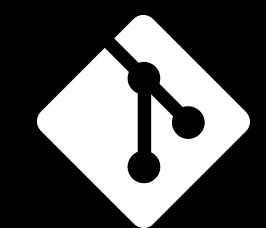
Sorular?

Neden versiyon kontrol sistemi kullanmalıyız?



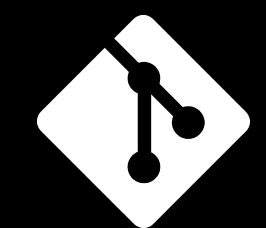
Sorular?

Git kavramlarından 3 tanesi nedir, ne işe yarar?



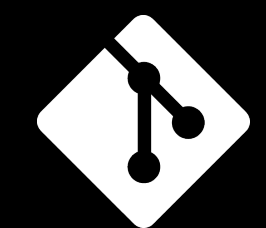
Sorular?

Git ve GitHub arasındaki fark nedir?



Sorular?

Sizce versiyon kontrol sistemlerinin en önemli özelliđi nedir?



Kaynaklar

[Git Wiki] -> <https://tr.wikipedia.org/wiki/Git> (yaz%C4%B1l%C4%B1m)

[Docs] -> <https://git-scm.com/doc>

[Edu] -> <https://education.github.com>

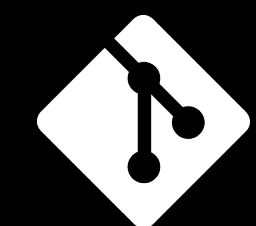
[Learn] -> <https://learngitbranching.js.org/>, <https://www.codecademy.com/courses/learn-git/lessons/git-workflow/exercises/hello-git>

[Book] -> <https://github.com/ITboy/book/blob/master/Version%20Control%20with%20Git%2C%202nd%20Edition.pdf>

[Links] -> <https://git-scm.com/doc/ext>

[Guis] -> <https://git-scm.com/downloads/guis/>

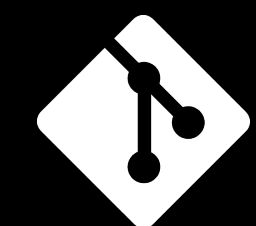
[Cheat] -> <https://education.github.com/git-cheat-sheet-education.pdf>



ÖDEV

Projeyi local bilgisayarınıza indirerek, proje ana dalından yeni bir dal oluşturup, “Readme.md” dosyasını aşağıdaki formata göre değiştirip, kodu tekrar ana dal üzerinde birleştiriniz.

“@Ad @Soyad, @Üniversite - @Bölüm, @Sınıf”



Teşekkürler

Yılmaz KORKMAZ

yilmaz.korkmaz@atez.com.tr

