



# git

*163301015 - Bekir Can YUVA*  
*153301054 - Ramazan MUTLU*

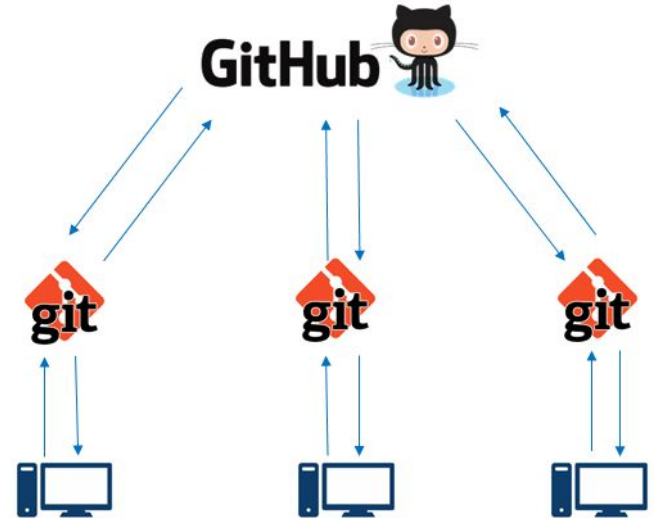
# Git Nedir?

---

Git, bir yazılımı geliştirirken büyük kolaylık ve hız sağlayan sürüm kontrol ve kod yönetim sistemidir.

- Arayüzü kolaydır.
- Hızlıdır.
- Tamamen dağıtılmıştır.
- Açık kaynak kodludur.

Yazıldığı Diller : C, Perl, Tcl, Python



# Sürüm Kontrolü Nedir?

---

Sürüm kontrolü, bir ya da daha fazla dosya üzerinde yapılan değişiklikleri kaydeden ve daha sonra belirli bir sürüme geri dönebilmenizi sağlayan bir sistemdir.

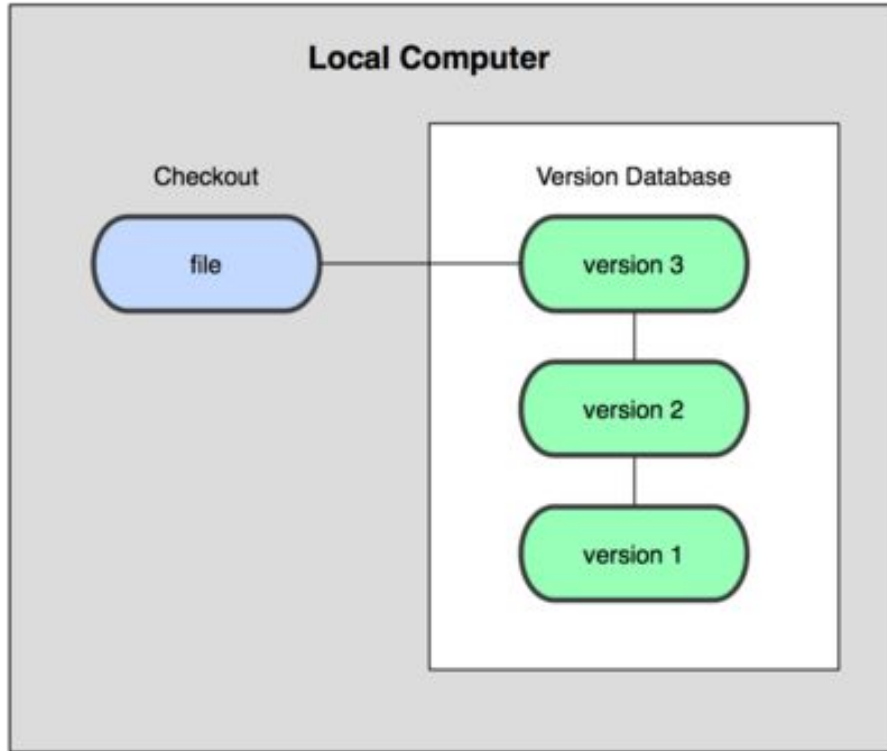
# Yerel Sürüm Kontrol Sistemi










---

Bu tür sistemler herhangi bir uzak sunucu (remote server) ile çalışmaz. Kaynak kodun tarihçesi de sadece yerel sistemlerde yapılır; yerelde versiyonlama yapar. Kişisel işler için kullanılabilir fakat ekip halinde çalışma imkanı sunmazlar.

# Yerel Sürüm Kontrol Sistemi

---



Name ^	
	proje - Copy.c
	proje.c
	proje-son.c
	proje-sonnn.c
	proje-sonnnn.c
	proje-sonnnn-2.c
	projesonnn-22.c
	proje-sonnn-22final.c
	proje-sonnn-22final222.c

# Yerel Sürüm Kontrol Sistemi

---

## ✓ Yararları :

- Yerel bilgisayar üzerinden çalışıldığı için ulaşılması kolaydır.

## ✗ Zararları :

- Hatalara açıktır.
- Projeyi ekip halinde yürütmek zordur.
- Zaman çizelgesi tutmak zordur.

# Merkezi Sürüm Kontrol Sistemi

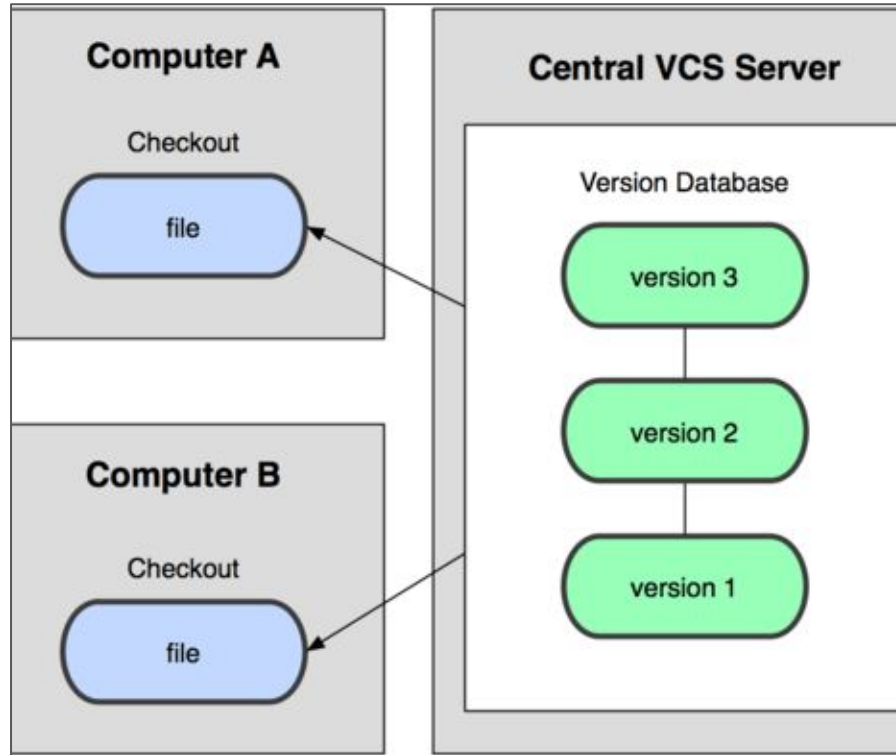
---

Geliştiriciler birbirleriyle etkileşimde olmak isterler.

Geliştirilen projenin ortak bir repository'de (depo) tutularak birden çok geliştiricinin aynı repository üzerinde çalışması yöntemine dayanan bu tasarım kısaca CVS (Concurrent Versions System) olarak adlandırılmaktadır.

# Merkezi Sürüm Kontrol Sistemi

---





# Merkezi Sürüm Kontrol Sistemi

---

## ✓ Yararları :

- Eş zamanlı birden çok geliştirici proje geliştirebilir.
- Her istemcide ayrı ayrı kurulu olan yerel veritabanlarını yönetmeye göre çok daha kolaydır.

## ✗ Zararları :

- Merkezi sunucu arızalanırsa proje sıkıntıya girebilir.
- Proje tek bir tarihçede tutulmak zorundadır.

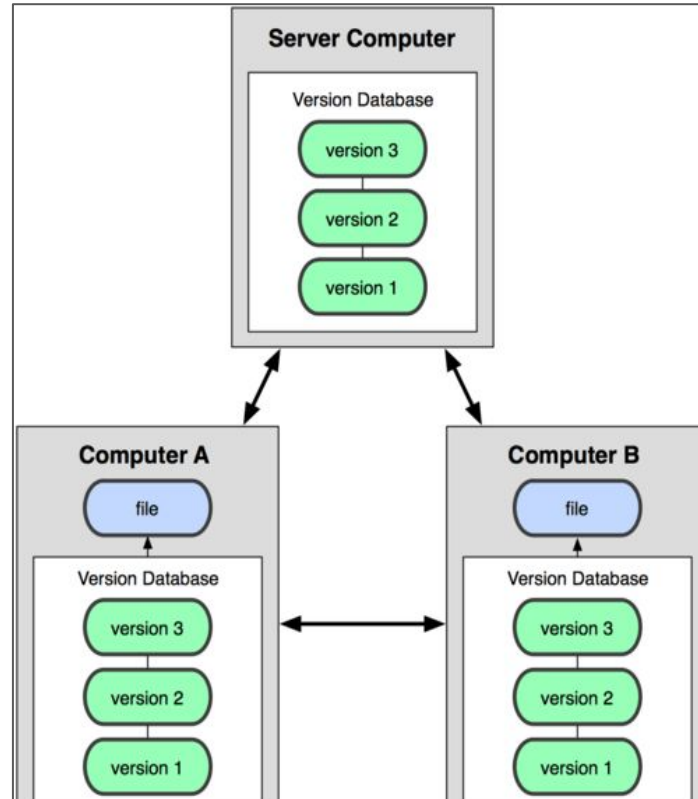
# Dağıtık Sürüm Kontrol Sistemi

---

Bir DSKS'de (Git, Mercurial, Bazaar vb.), istemciler (kullanıcılar) dosyaların yalnızca en son bellek kopyalarını almakla kalmazlar; yazılım havuzunu (repository) bütünüyle yansılarlar (kopyalarlar).

# Dağıtık Sürüm Kontrol Sistemi

---



# Dağıtık Sürüm Kontrol Sistemi

---

## ✓ Yararları :

- Eş zamanlı birden çok geliştirici dağıtık proje geliştirebilir.
- Uzak uç birimlerde (remote) geliştiriciler rahatlıkla çalışabilir.
- Dosyalar gelişmiş bir tarihçede tutulur.

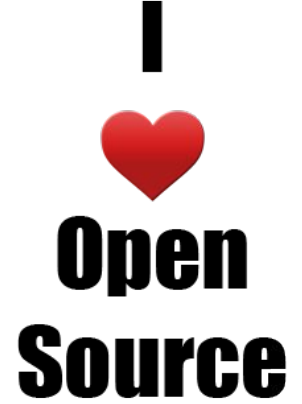
# Git'in Kısa Tarihçesi

---

Linux çekirdeği (kernel) oldukça büyük ölçekli bir açık kaynak kodlu yazılım projesidir. Linux çekirdek bakım ve geliştirme yaşam süresinin çoğunda (1991-2002), yazılım değişiklikleri yamalar ve arşiv dosyaları olarak tutulup taşındı. 2002 yılında, Linux çekirdek projesi, BitKeeper adında tescilli bir DSKS kullanmaya başladı.

2005 yılında, Linux çekirdeğini geliştiren toplulukla BitKeeper'ı geliştiren şirket arasındaki ilişki bozuldu ve aracın topluluk tarafından ücretsiz olarak kullanılabilmesi uygulamasına son verildi. Bu, Linux geliştirim topluluğunu (ve özellikle Linux'un geliştiricisi olan Linus Torvalds'ı) BitKeeper'ı kullanırken aldıkları derslerden yola çıkarak kendi araçlarını geliştirme konusunda harekete geçirdi. Yeni sistemin hedeflerinden bazıları şunlardı:

- Hız
- Basit tasarım
- Çizgisel olmayan geliştirim için güçlü destek (binlerce paralel dal (branch))
- Bütünüyle dağıtık olma
- Linux çekirdeği gibi büyük projelerle verimli biçimde başa çıkabilme (hız ve veri boyutu)



# Git Hosting Servisleri

---

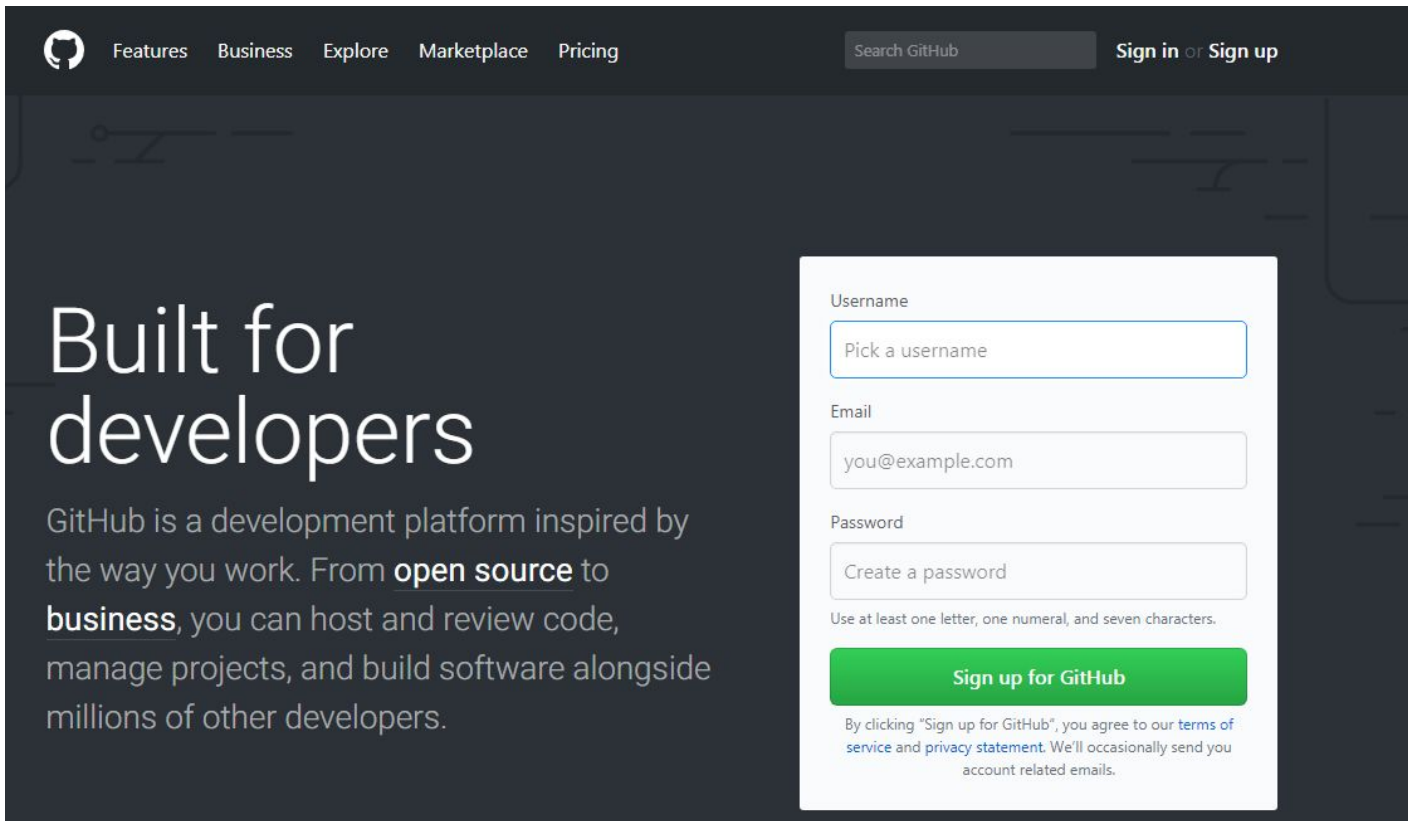
Aşağıdaki linkten karşılaştırmalı olarak git servislerine ulaşabilirsiniz :

<https://www.git-tower.com/blog/git-hosting-services-compared/>



# GitHub'da Hesap Oluşturmak

---

A screenshot of the GitHub website's sign-up page. The page has a dark background with white text. At the top, there is a navigation bar with the GitHub logo, links for Features, Business, Explore, Marketplace, and Pricing, a search bar, and links for Sign in or Sign up. The main content area on the left says "Built for developers" and describes GitHub as a development platform. On the right, there is a white sign-up form with fields for Username, Email, and Password, a green "Sign up for GitHub" button, and a disclaimer about terms of service and privacy statement.

Features Business Explore Marketplace Pricing Search GitHub Sign in or Sign up

## Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username  
Pick a username

Email  
you@example.com

Password  
Create a password

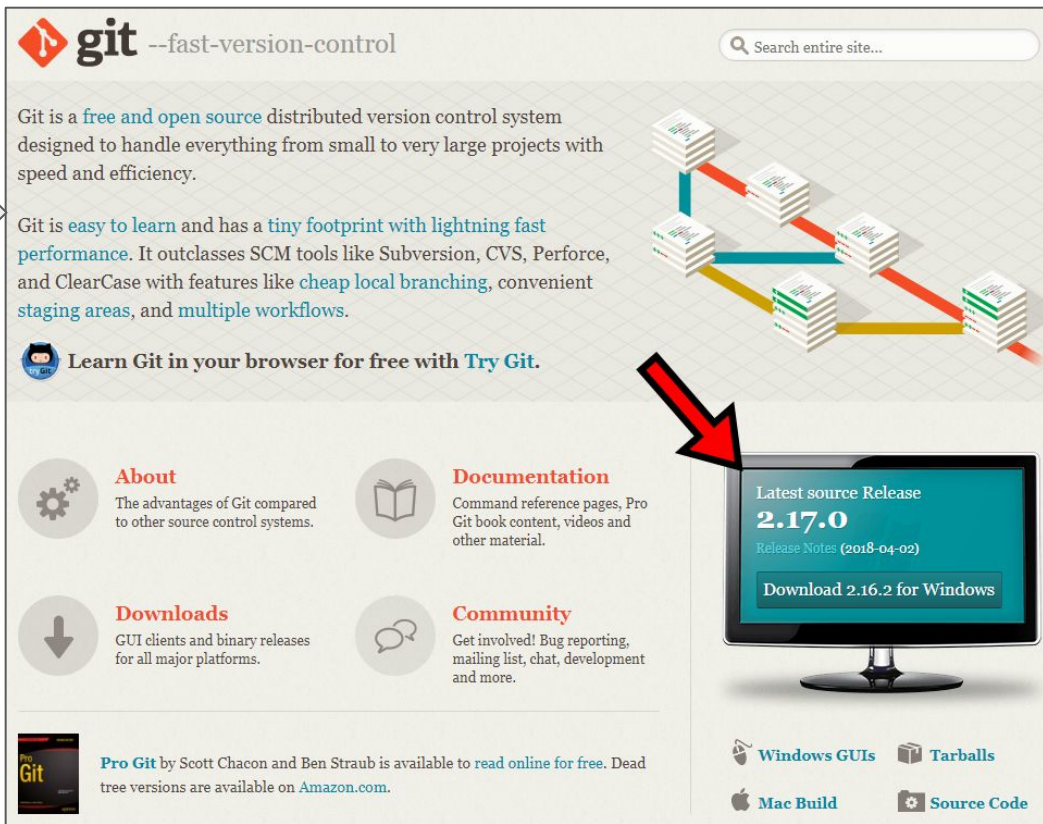
Use at least one letter, one numeral, and seven characters.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

# Git Bash/GUI Kurulumu

<https://git-scm.com>




The screenshot shows the Git website homepage. A white box on the left contains the URL <https://git-scm.com> with a grey arrow pointing to the top of the page. A red arrow points from the 'Downloads' section to a computer monitor displaying the latest source release information.

**git** --fast-version-control

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

 Learn Git in your browser for free with **Try Git**.

**About**  
The advantages of Git compared to other source control systems.





**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

**Latest source Release**  
**2.17.0**  
Release Notes (2018-04-02)  
[Download 2.16.2 for Windows](#)

**Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

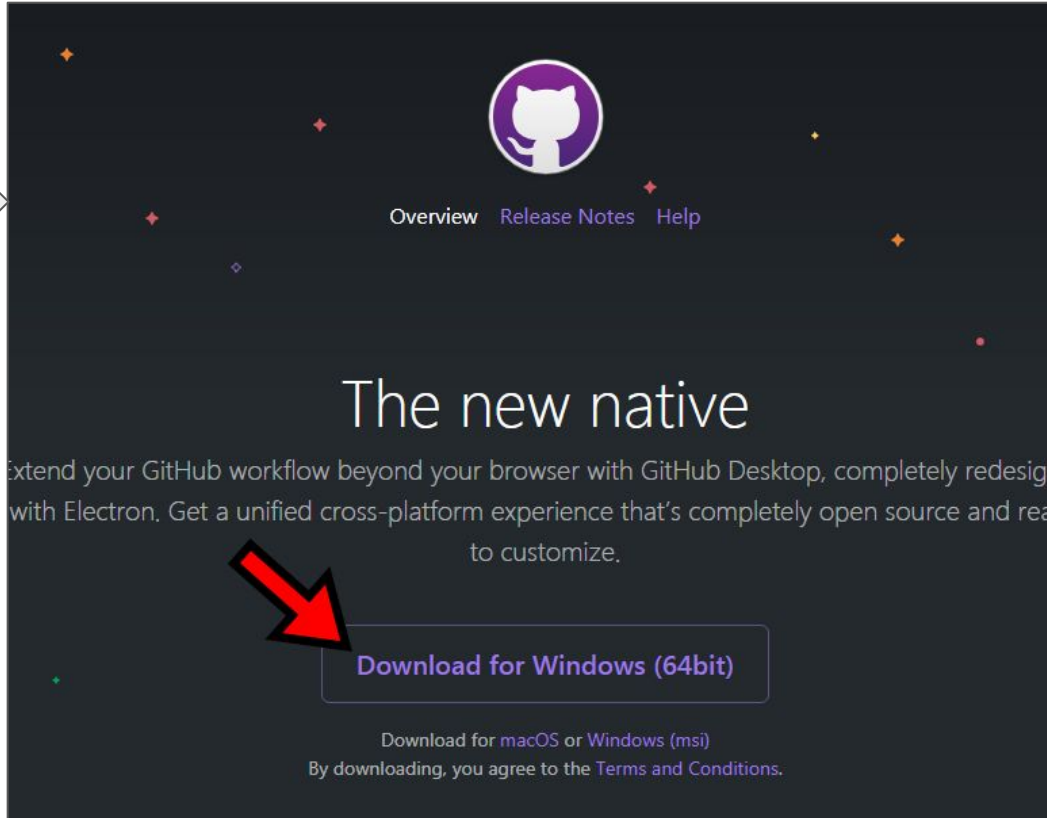
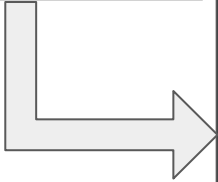
 **Windows GUIs**  **Tarballs**  
 **Mac Build**  **Source Code**



# Git Desktop Kurulumu

---

<https://desktop.github.com>



# Başlıca Git Komutları

In case of fire



1. git commit



2. git push



3. leave building

# config

---

Kullanıcı adı, mail, dosya formatları gibi genel yapılandırma ayarlarını yapacağınız komuttur. Aşağıda yazdığımız config komutunda GitHub hesabımızın kullanıcı adını ve mailini yazıyoruz.

```
git config --global user.email "gitsunu@mail.com.tr"  
git config --global user.name "gitsunu"
```

# init

---

Git Bash'de bulunduğumuz dizini bir repository (depo) olarak gösterir.

```
git init
```

# clone

---

Yerelde veya uzak bir sunucuda yer alan bir deponun çalışan bir kopyasını bulunduğumuz dizine kopyalar.

```
git clone /yol/repo  
git clone https://github.com/gitsunu/ilkrepo.git
```

# add

---

Belirtilen dosyayı ya da tüm projeyi çalışma dizinine (index) gönderir.  
Başka bir deyişle commit'lemeye hazır hale getirir.

```
git add --all  
git add README.md
```

# rm

---

Belirtilen dosyayı ya da tüm projeyi çalışma dizininden (index) siler.

```
git rm README.md
```

# commit

---

Çalışma dizinine eklenen dosyaları HEAD denilen .git klasörü içindeki kısıma ekler. Commit işlemi yerelde gerçekleşen bir işlemdir. Uzak sunucudaki depoda bir değişiklik yapmaz.

-m parametresiyle açıklama kısmı olmadan hızlı bir biçimde commit atmamız mümkün.

```
git commit -m "ilk commit"
```



# push

---

Commit kısmında projemizin yerel depomuzdaki HEAD dosyasında saklandığını belirtmiştik. Şimdi ise push komutu ile saklanan projemizi uzak sunucuya gönderebiliriz.

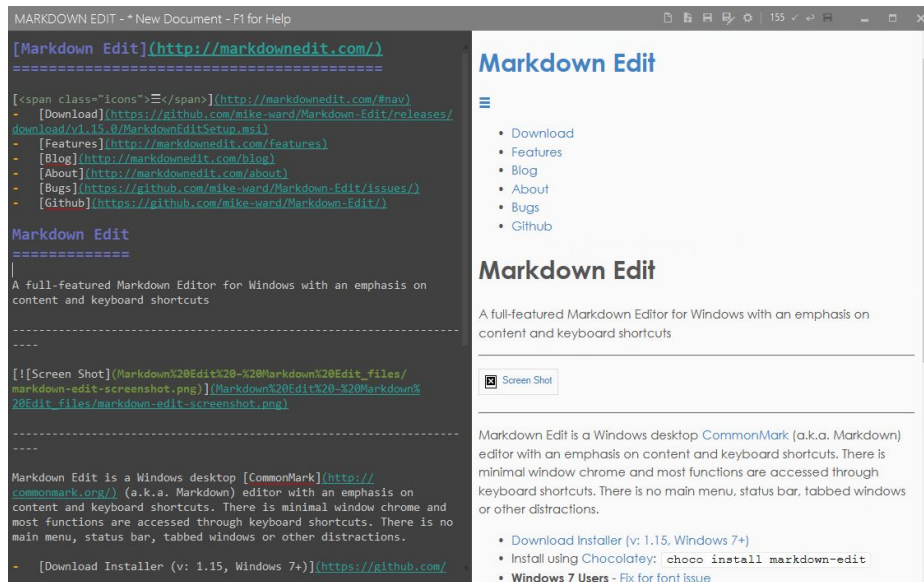
origin bizim uzak sunucumuzu (remote), master ise uzak sunucudaki ana branch (branş) 'imizi temsil ediyor.

```
git push origin master
```

# Markdown

Markdown ayrıntılı dökümantasyonuna aşağıdaki linkten ulaşabilirsiniz.

<https://guides.github.com/features/mastering-markdown/>



# Sorularınız?



# Sorularımız



## 1) Git nedir?

- A-) Versiyon kontrol sistemidir.
- B-) Kontrol sistem versiyonudur.
- C-) Sistemlerin versiyonlanmış kontrolüdür.
- D-) Geri dönüşümlü internet teknolojisi.
- E-) Dosya paylaşımı platformudur.

## 1) Git nedir?

- ☒ A- Versiyon kontrol sistemidir.
- ☐ B-) Kontrol sistem versiyonudur.
- ☐ C-) Sistemlerin versiyonlanmış kontrolüdür.
- ☐ D-) Geri dönüşümlü internet teknolojisi.
- ☐ E-) Dosya paylaşımı platformudur.

- I - Yerel sürüm kontrol sistemi
- II - Sanal sürüm kontrol sistemi
- III - Merkezi sürüm kontrol sistemi
- IV - Dağıtık sürüm kontrol sistemi

2) Yukarıdakilerden hangileri versiyon kontrol sistemlerinin çeşitlerindendir?

- A-) I ve II
- B-) I-II ve III
- C-) I ve IV
- D-) III ve IV
- E-) I-III ve IV

- I - Yerel sürüm kontrol sistemi
- II - Sanal sürüm kontrol sistemi
- III - Merkezi sürüm kontrol sistemi
- IV - Dağıtık sürüm kontrol sistemi

2) Yukarıdakilerden hangileri versiyon kontrol sistemlerinin çeşitlerindendir?

- A-) I ve II
- B-) I-II ve III
- C-) I ve IV
- D-) III ve IV
- ☒ E-) I-III ve IV



- Hatalara açıktır.
- Projeleri ekip halinde yürütmek zordur.
- Yerel bilgisayar üzerinden çalışıldığı için ulaşılması kolaydır.
- Zaman çizelgesi tutmak zordur.

3) Yukarıda verilen özelliklere sahip sürüm kontrol sistemi aşağıdakilerden hangisidir?

- A-) Sanal sürüm kontrol sistemi
- B-) Dağıtık sürüm kontrol sistemi
- C-) Yerel sürüm kontrol sistemi
- D-) Merkezi sürüm kontrol sistemi
- E-) Evrensel sürüm kontrol sistemi

- Hatalara açıktır.
- Projeleri ekip halinde yürütmek zordur.
- Yerel bilgisayar üzerinden çalışıldığı için ulaşılması kolaydır.
- Zaman çizelgesi tutmak zordur.

3) Yukarıda verilen özelliklere sahip sürüm kontrol sistemi aşağıdakilerden hangisidir?

A-) Sanal sürüm kontrol sistemi

B-) Dağıtık sürüm kontrol sistemi

**C-) Yerel sürüm kontrol sistemi**

D-) Merkezi sürüm kontrol sistemi

E-) Evrensel sürüm kontrol sistemi

4) Aşağıdakilerden hangisi Git'in başlıca özelliklerinden biri değildir?

A-) Hızlıdır.

B-) Arayüzü kolaydır.

C-) Tamamen dağıtılmıştır.

D-) Çizgisel geliştirim kullanılır.

E-) Açık kaynak kodludur.

4) Aşağıdakilerden hangisi Git'in başlıca özelliklerinden biri değildir?

A-) Hızlıdır.

B-) Arayüzü kolaydır.

C-) Tamamen dağıtılmıştır.

☒ D-) Çizgisel geliştirim kullanılır.

E-) Açık kaynak kodludur.

5) Yerel depodaki bir projeyi uzak sunucuya göndermek için hangi git komutu kullanılır?

A-) git commit

B-) git config

C-) git clone

D-) git remote

E-) git push

5) Yerel depodaki bir projeyi uzak sunucuya göndermek için hangi git komutu kullanılır?

A-) git commit

B-) git config

C-) git clone

D-) git remote

☒ E-) git push

- `git add --all`
- .....
- `git push origin master`

6) Yukarıdaki boş bırakılan yere hangi komut yazılmalıdır?

A-) `git commit -m "ilk commit"`

B-) `git config --global user.name "isim"`

C-) `git rm -r .`

D-) `git clone https://github.com/isim/ilkrepo.git`

E-) `git reset --hard`

- `git add --all`
- .....
- `git push origin master`

6) Yukarıdaki boş bırakılan yere hangi komut yazılmalıdır?

- A-) `git commit -m "ilk commit"`
- B-) `git config --global user.name "isim"`
- C-) `git rm -r .`
- D-) `git clone https://github.com/isim/ilkrepo.git`
- E-) `git reset --hard`



# Haftaya g r   mek  zere

