





# Git ve Github

Alparslan Türkeş Science and Technology  
University

-  CheesyFrappe – Emirhan Balcı
-  BurakAhmet – Ahmet Burak Biçer



Git nedir?

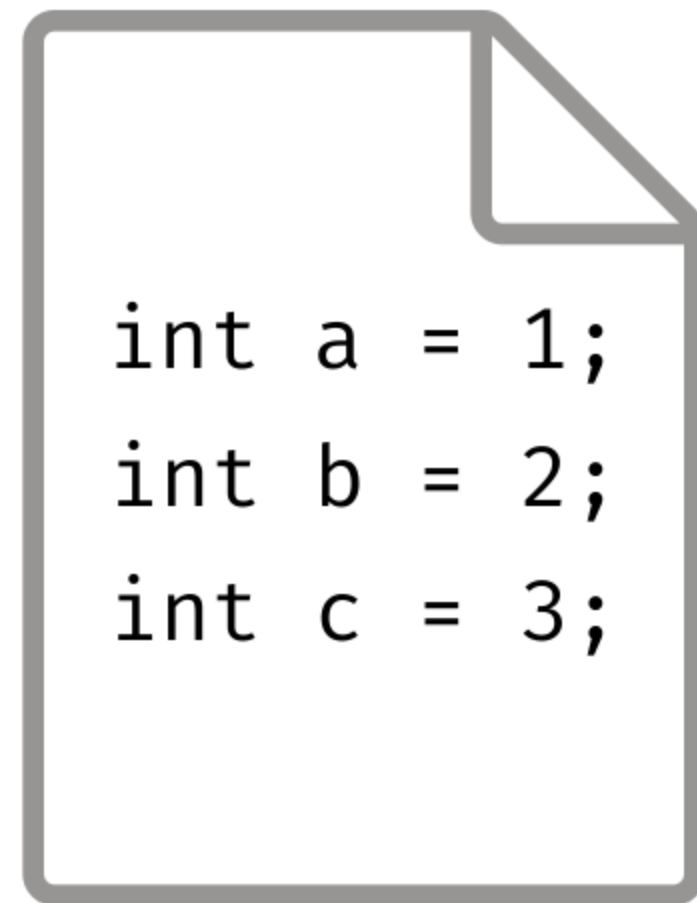


Git, yazılan kodu takip eden bir versiyon kontrol sistemidir.

- Projedeki değişiklikleri takip etmeyi kolaylaştırır.
- Kodu kişiler arasında senkonize eder.
- Kodun orijinalini kaybetmeden test yapabilme imkanı sunar.
- Projede yapılmış değişiklikleri geri almayı sağlar.



- Projedeki değişiklikleri takip etmeyi kolaylaştırır.



Create file



- Projedeki değişiklikleri takip etmeyi kolaylaştırır.

```
int a = 1;  
int b = 2;  
int c = 3;
```

Create file



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```

Add a line

- Projedeki değişiklikleri takip etmeyi kolaylaştırır.

```
int a = 1;  
int b = 2;  
int c = 3;
```

Create file



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```

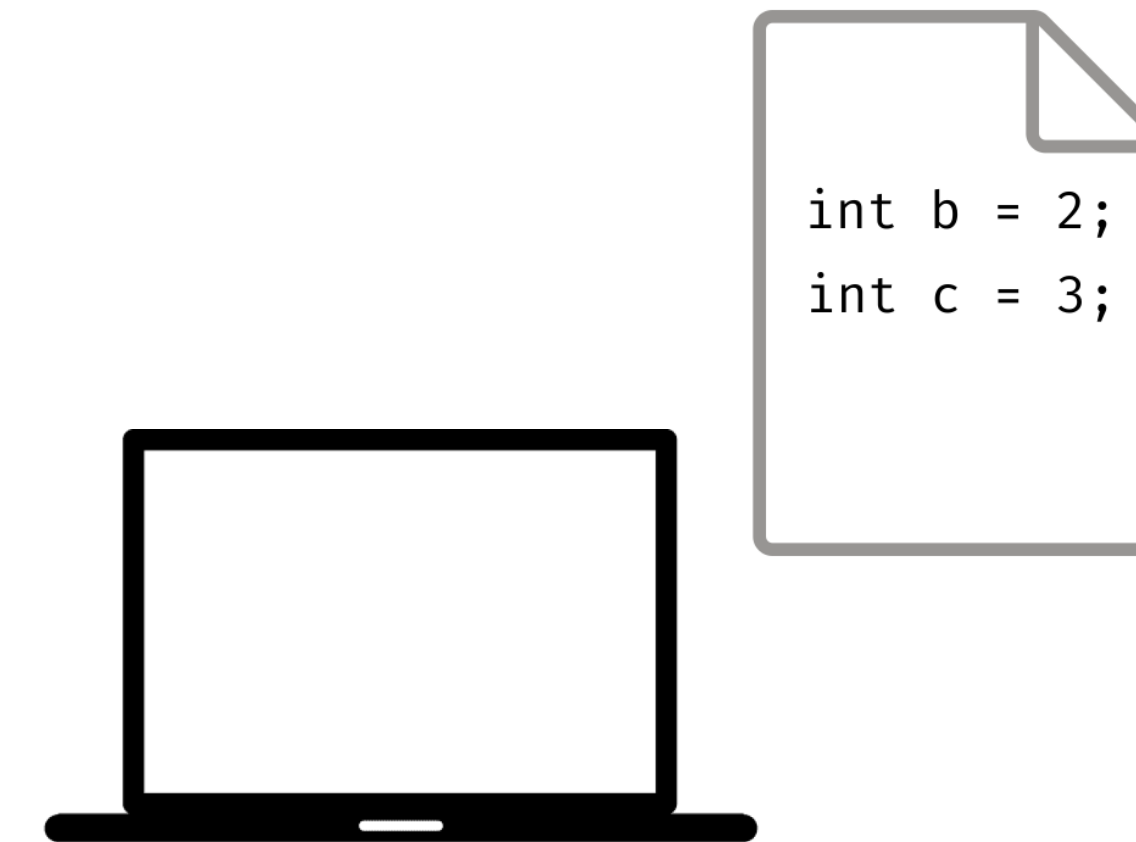
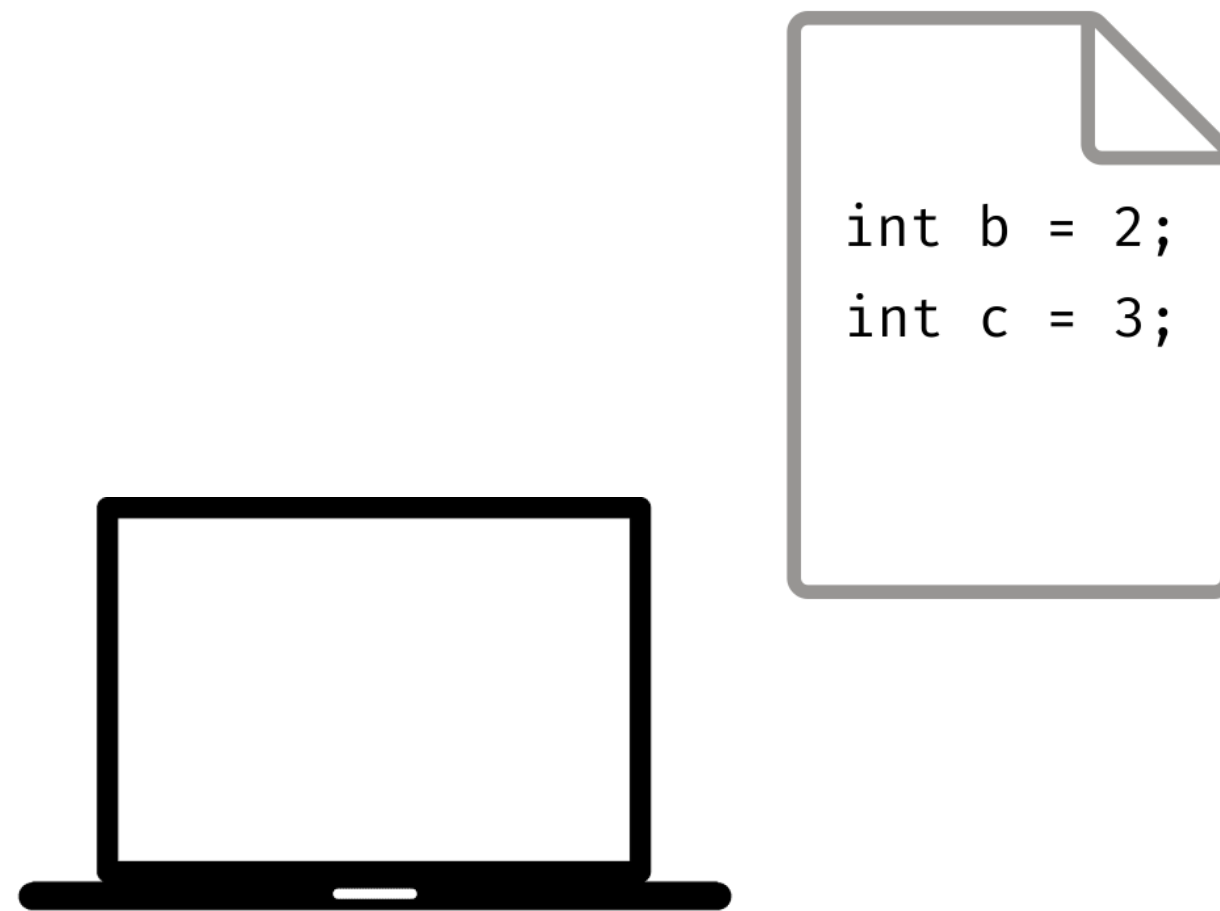
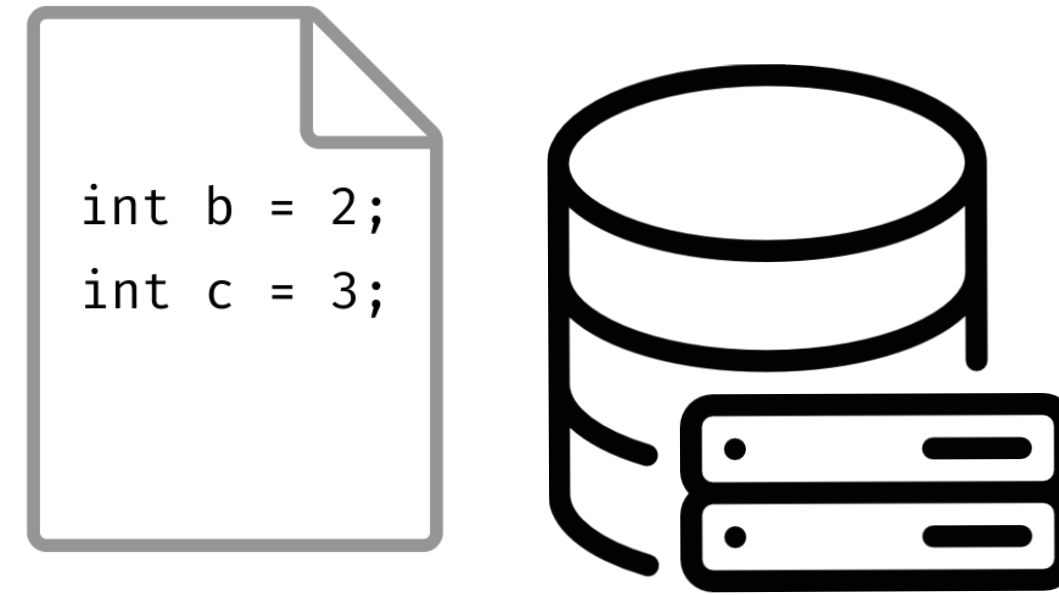
Add a line



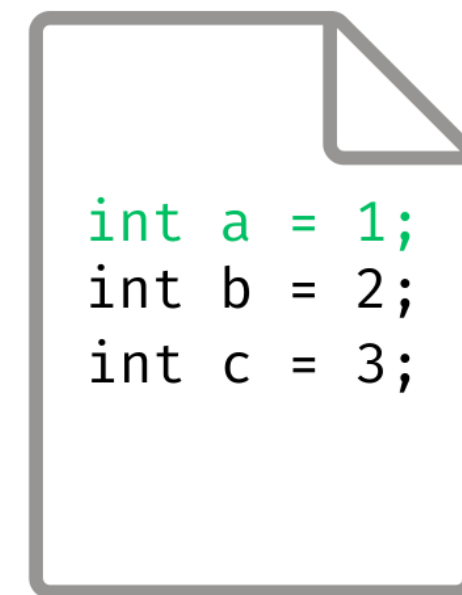
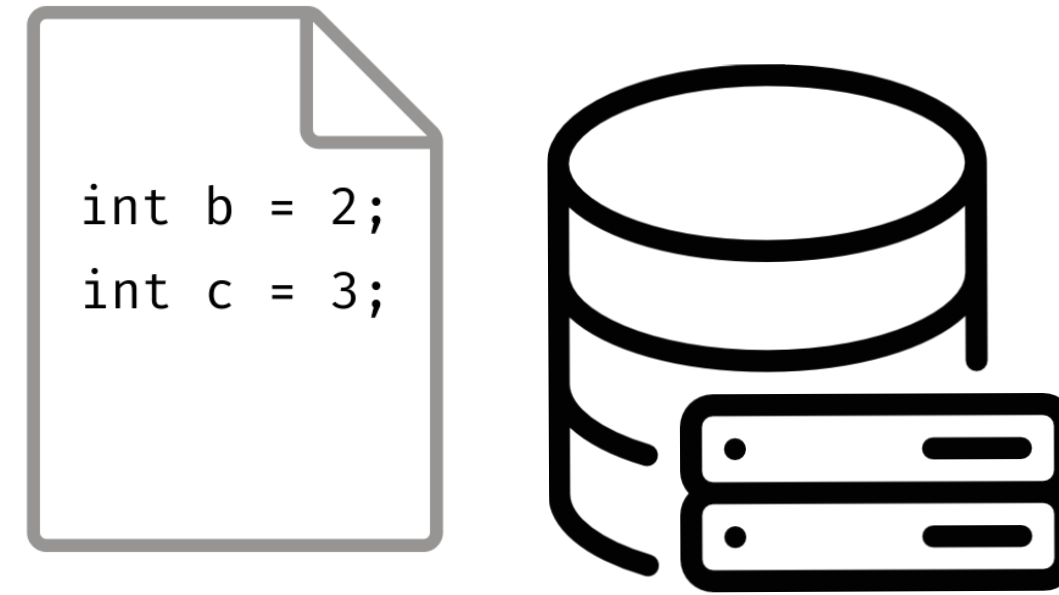
```
int a = 1;  
int b = 2;  
int d = 4;
```

Remove a line

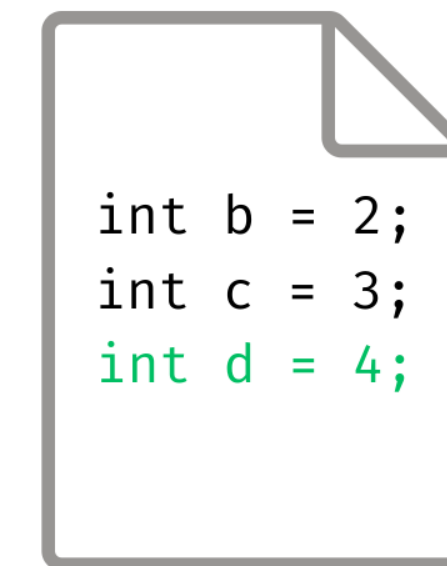
- Kodu kişiler arasında senkonize eder.



- Kodu kişiler arasında senkonize eder.



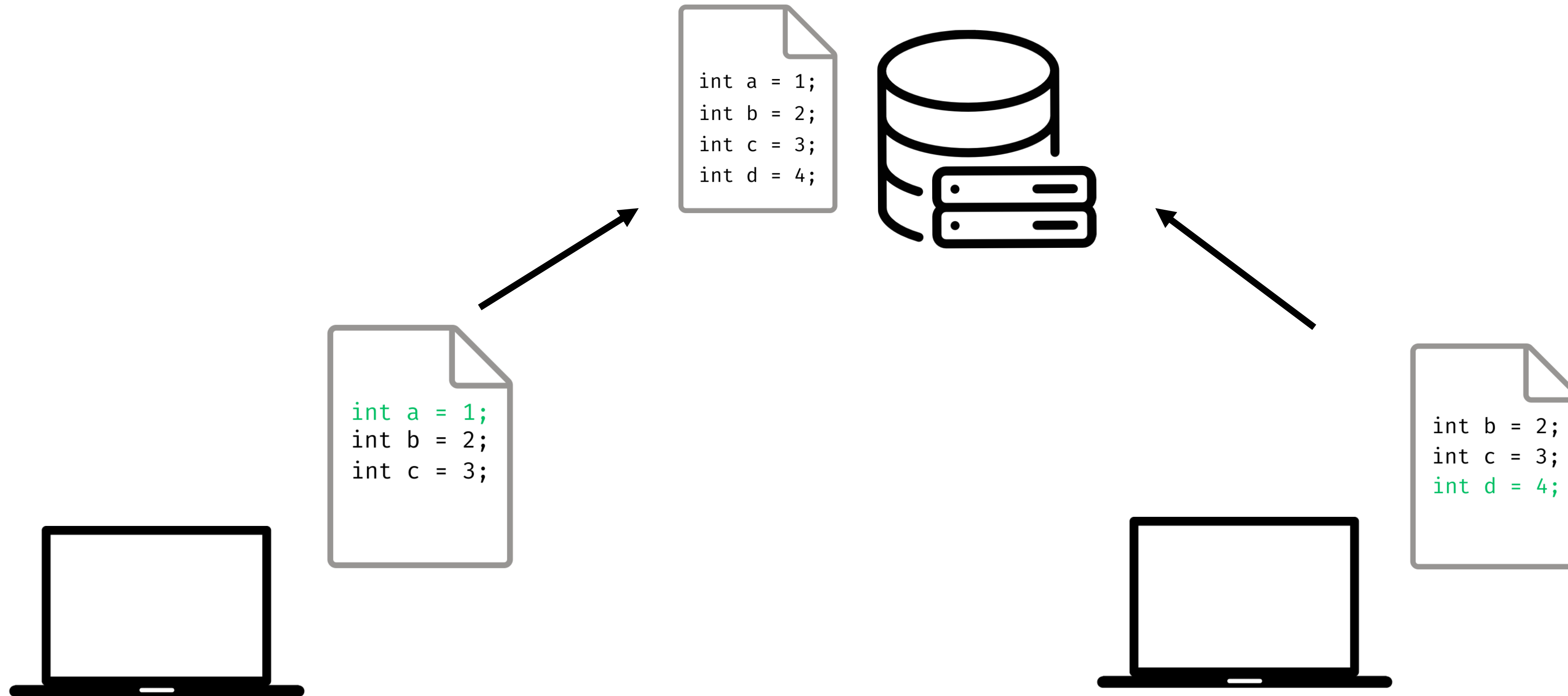
Add a line



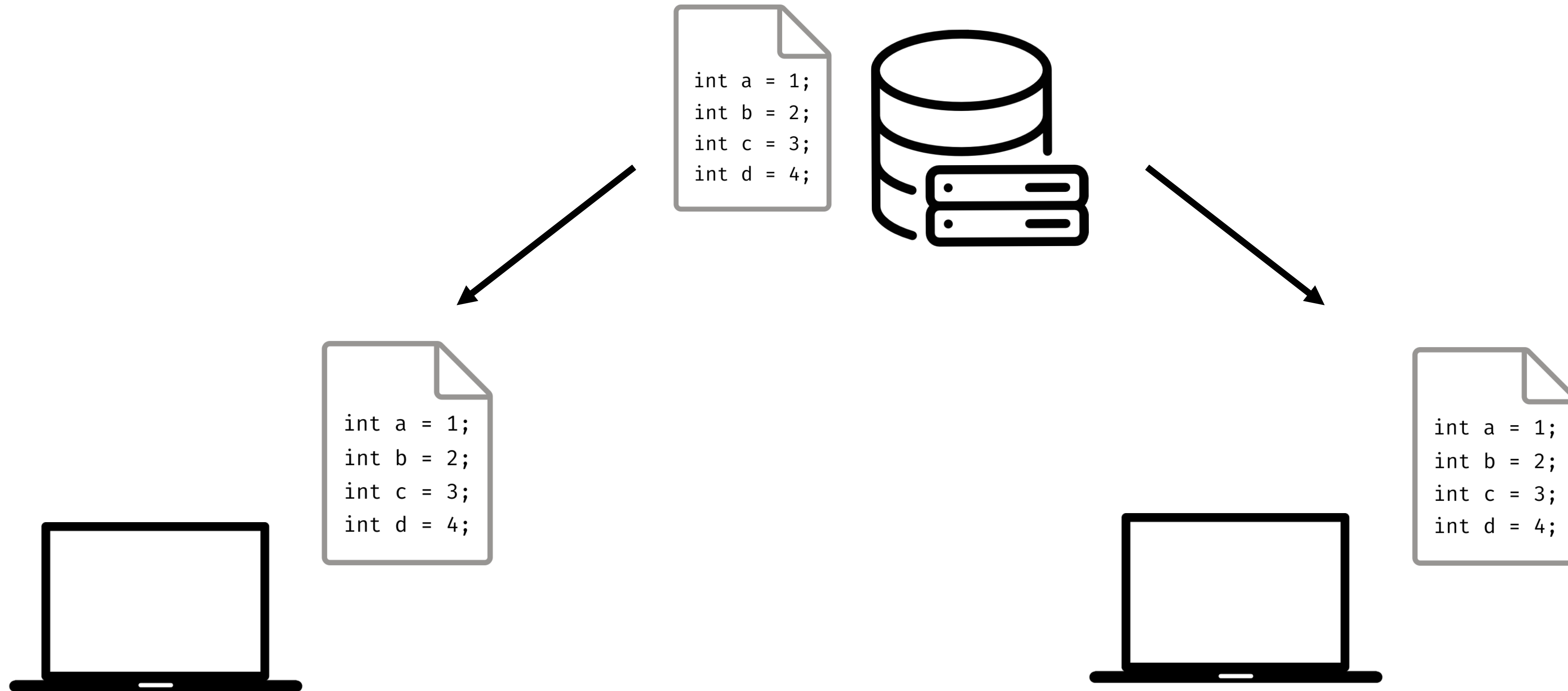
Add a line



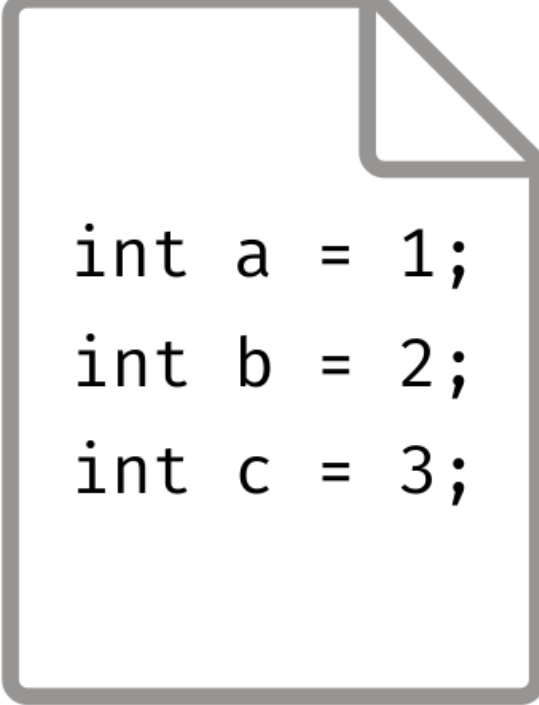
- Kodu kişiler arasında senkonize eder.



- Kodu kişiler arasında senkonize eder.



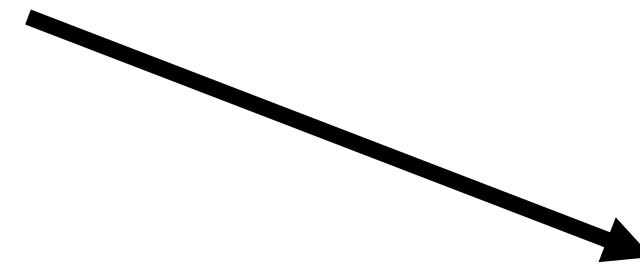
- Kodun orijinalini kaybetmeden test yapabilme imkanı sunar.

A document icon with a folded top-right corner, containing three lines of code.

```
int a = 1;  
int b = 2;  
int c = 3;
```

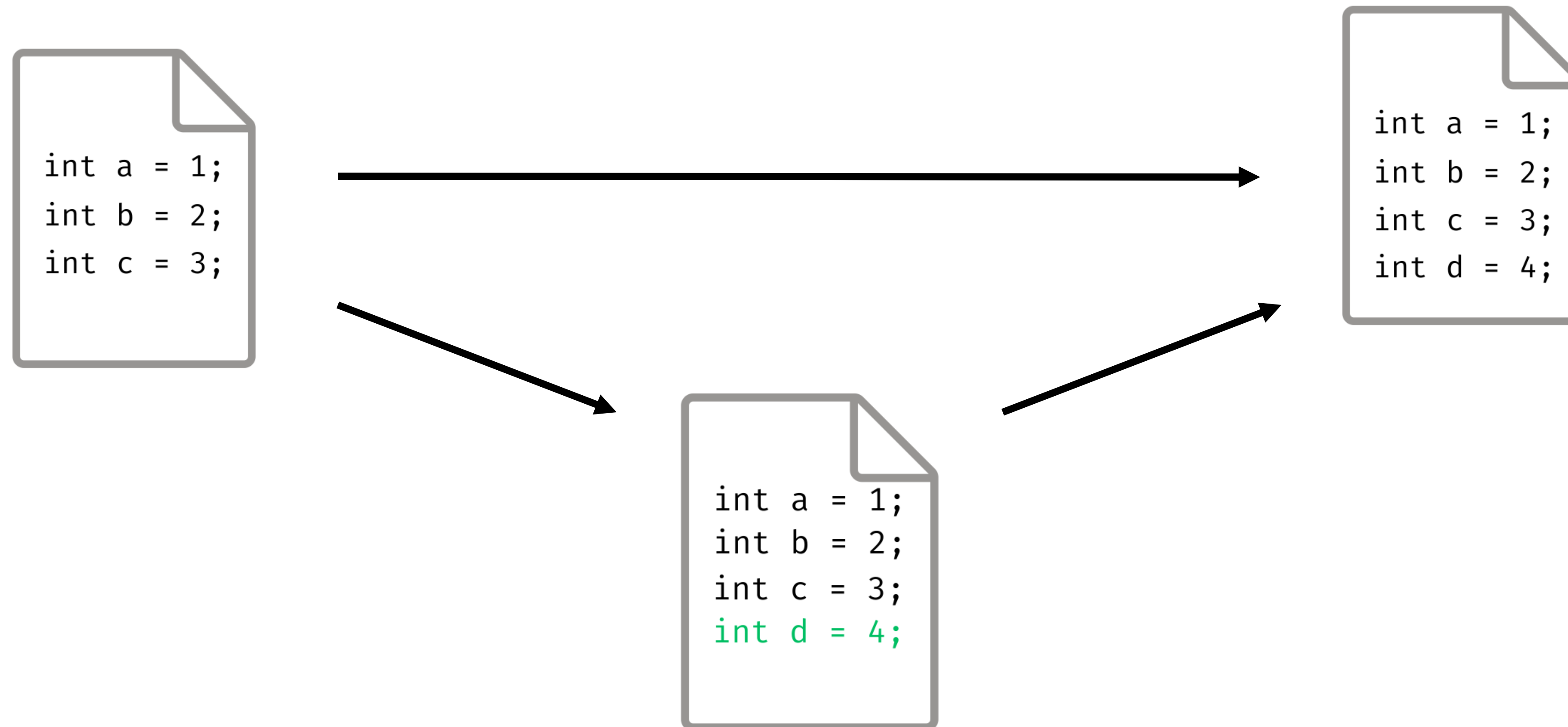
- Kodun orijinalini kaybetmeden test yapabilme imkanı sunar.

```
int a = 1;  
int b = 2;  
int c = 3;
```



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```

- Kodun orijinalini kaybetmeden test yapabilme imkanı sunar.





- Projede yapılmış değişiklikleri geri almayı sağlar.

```
int a = 1;  
int b = 2;  
int c = 3;
```

Create file



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```

Add a line



```
int a = 1;  
int b = 2;  
int d = 4;
```

Remove a line

- Projede yapılmış değişiklikleri geri almayı sağlar.

```
int a = 1;  
int b = 2;  
int c = 3;
```

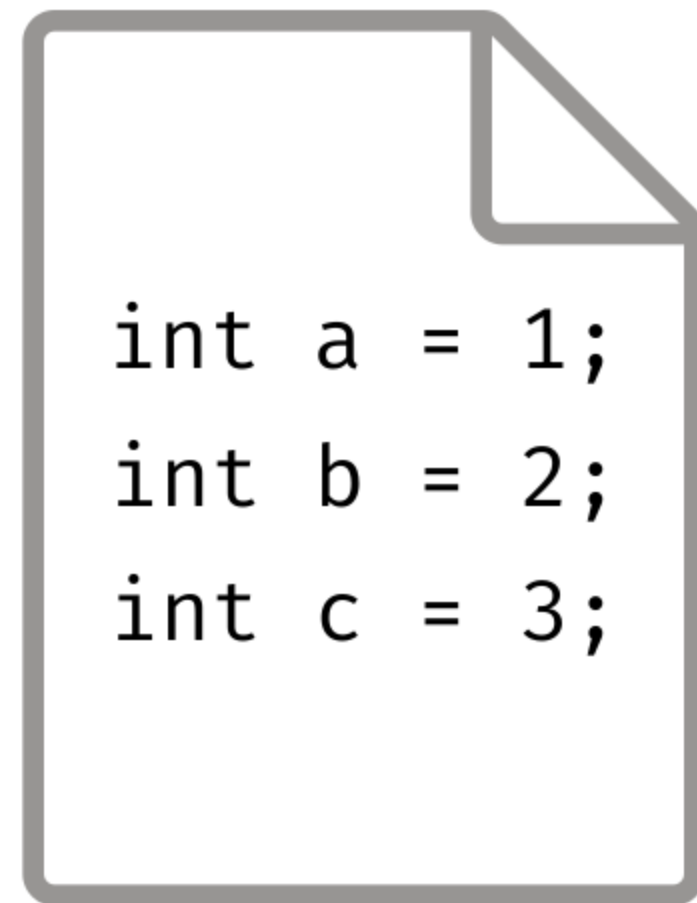
Create file



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```

Add a line

- Projede yapılmış değişiklikleri geri almayı sağlar.



Create file



<https://git-scm.com/downloads>



# GitHub nedir?



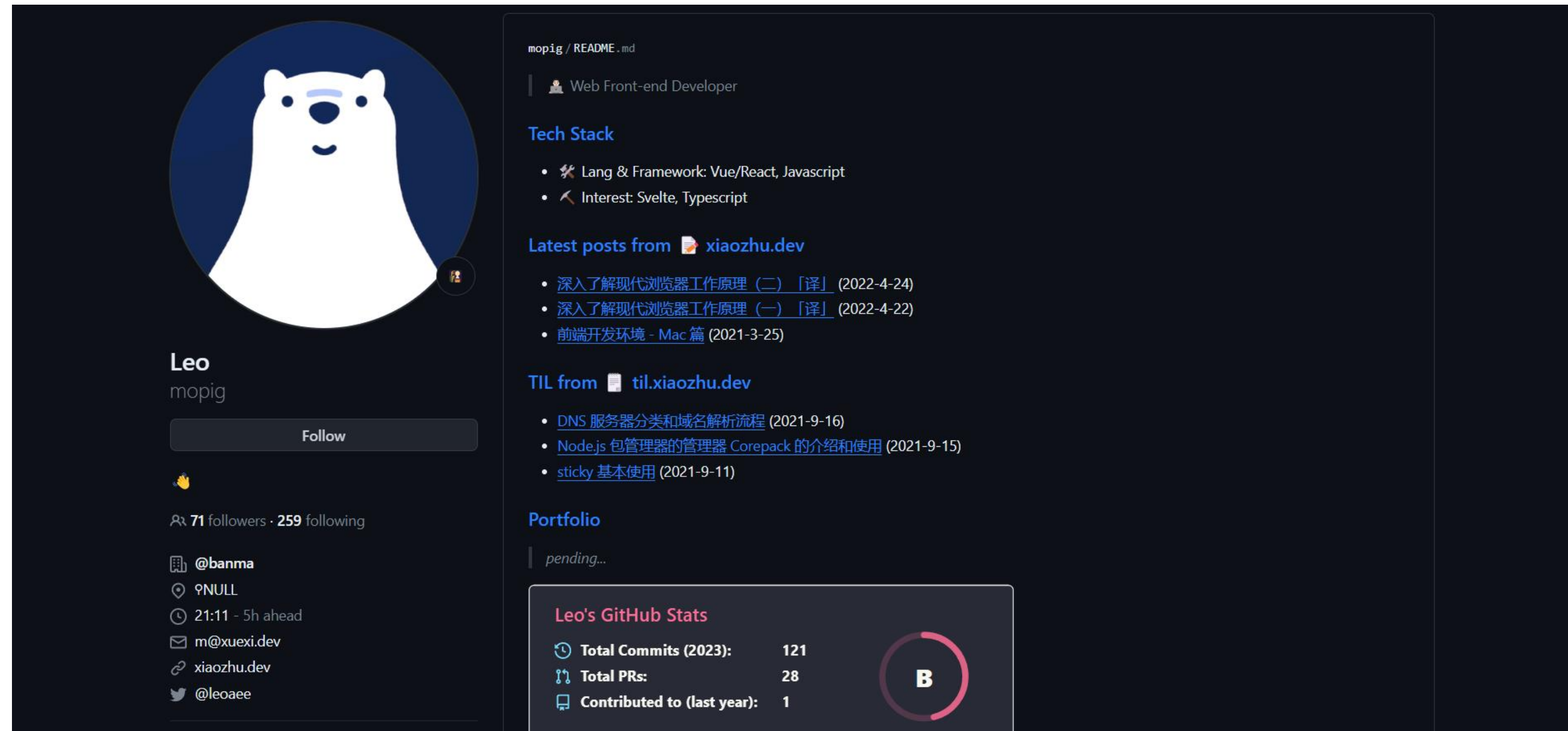


## GitHub git kullanan yazılım geliştirme projeleri için web tabanlı bir depolama servisi

- Projeyi dünyadaki diğer insanlarla paylaşmanıza olanak tanır.
- Projenin kolektif olarak geliştirilmesini sağlar. Açık kaynak kodlu projeler bu sayede geliştirilir.
- Geliştirilen projenin farklı versiyonlarını depolamayı sağlar.
- Projeye dair açıklayıcı bir dökümantasyon oluşturulmasını sağlar.
- Yazılım geliştiricileri için bir sosyal medya olarak düşünülebilir.



- Kullanıcılara bir arayüz sunar.



# Git ve GitHub'ın Farkı

## Git

- Yerel bir versiyon kontrol sistemidir.
- Depolama ve yönetim işlemleri kullanıcının yerel cihazında yapılır.
- Kod yönetimi ve takım çalışması için temel özellikler sunar.

## GitHub

- Bulut tabanlı bir web uygulamasıdır.
- Depolama ve yönetim için bulut üzerinden işlem yapılır.
- Kod yönetimi ve gelişmiş takım çalışması için özellikler sunar.

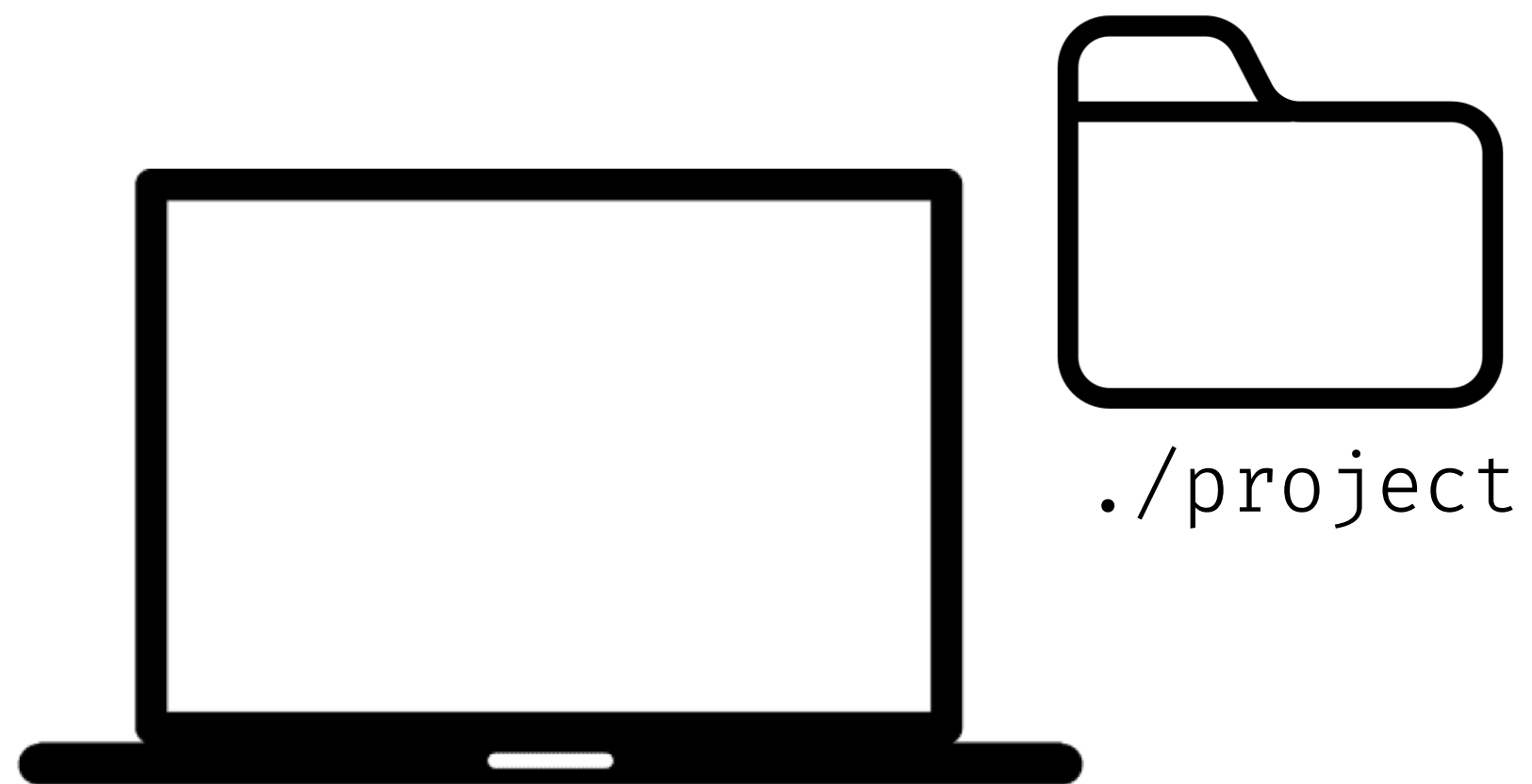


```
git init
```



# git init

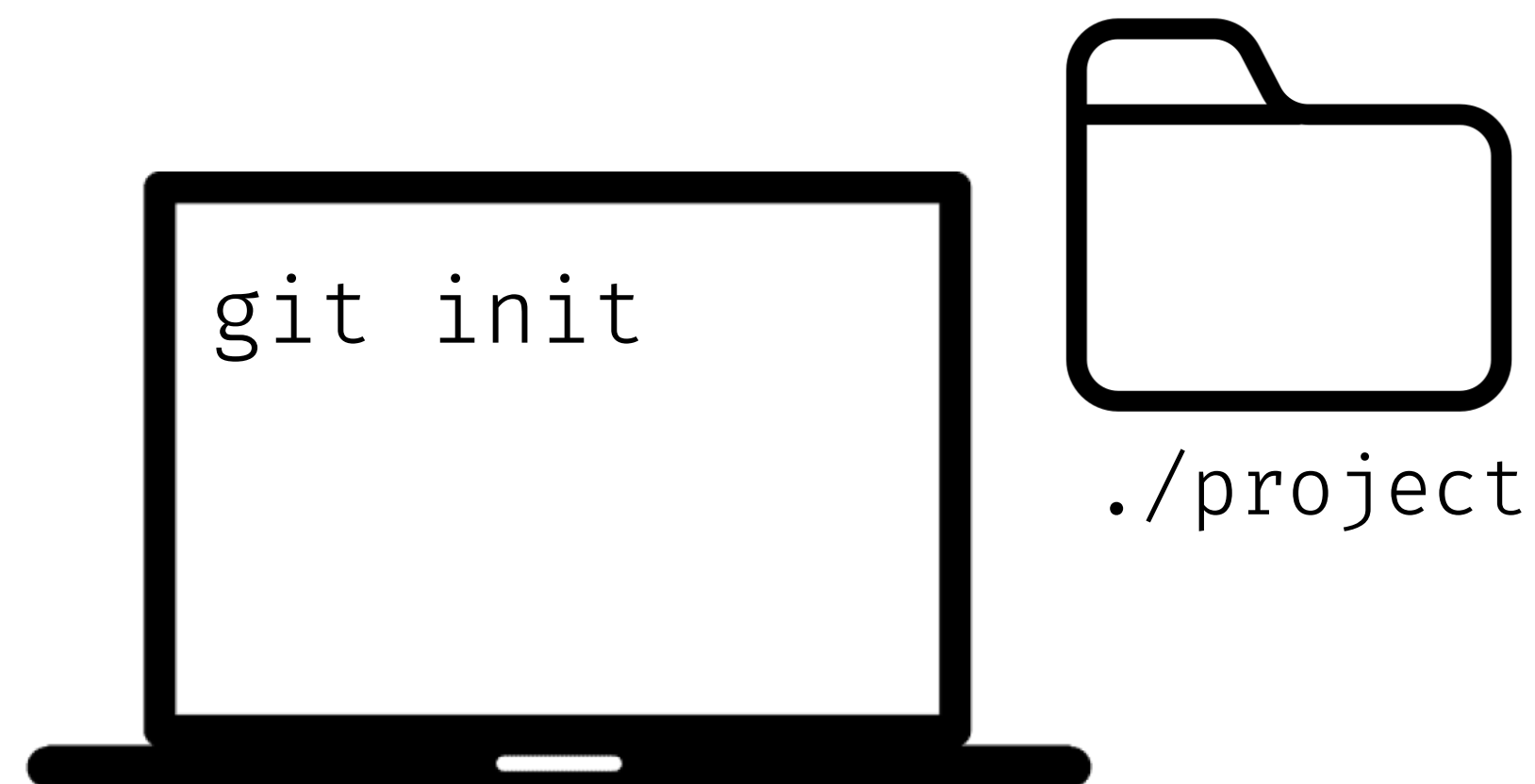
- Makinenizde yeni bir Git repository'si yaratır.
- ".git" adlı dosyayı ekleyerek, Github ile aradaki
- köprüyü sağlamış olur.





# git init

- Makinenizde yeni bir Git repository'si yaratır.
- ".git" adlı dosyayı ekleyerek, Github ile aradaki
- köprüyü sağlamış olur.



```
emirbalci@abra-linux:~/Desktop$project/ git init
```

```
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:
```

```
hint:
```

```
hint:     git config --global init.defaultBranch <name>
```

```
hint:
```

```
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
```

```
hint: 'development'. The just-created branch can be renamed via this command:
```

```
hint:
```

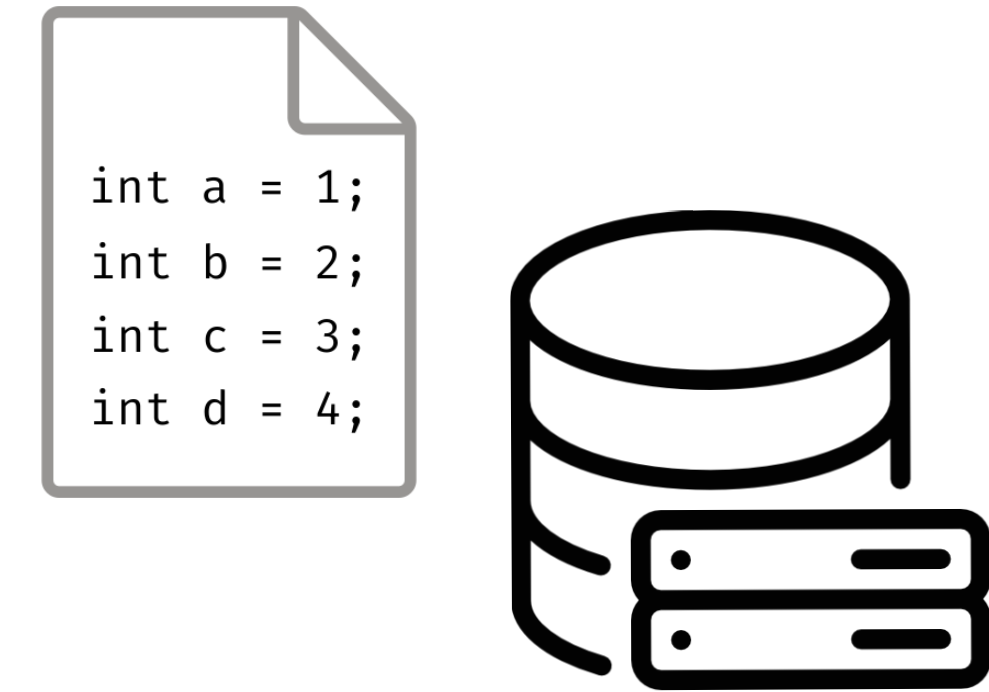
```
hint:     git branch -m <name>
```

```
Initialized empty Git repository in /home/emirbalci/Desktop/project/.git/
```

git clone

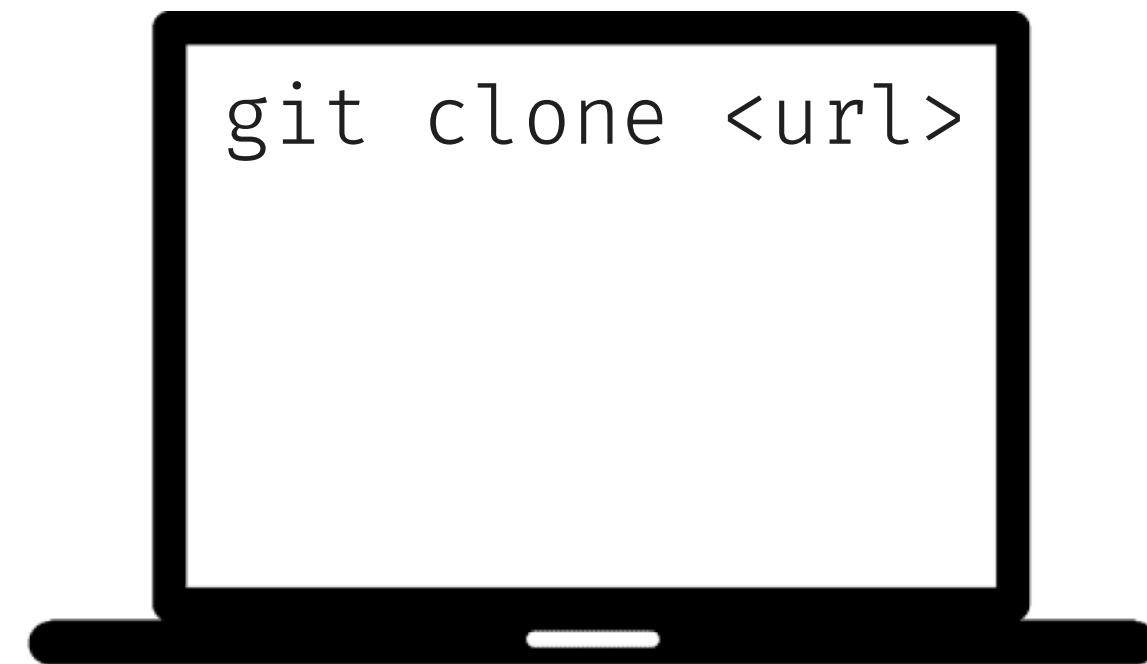
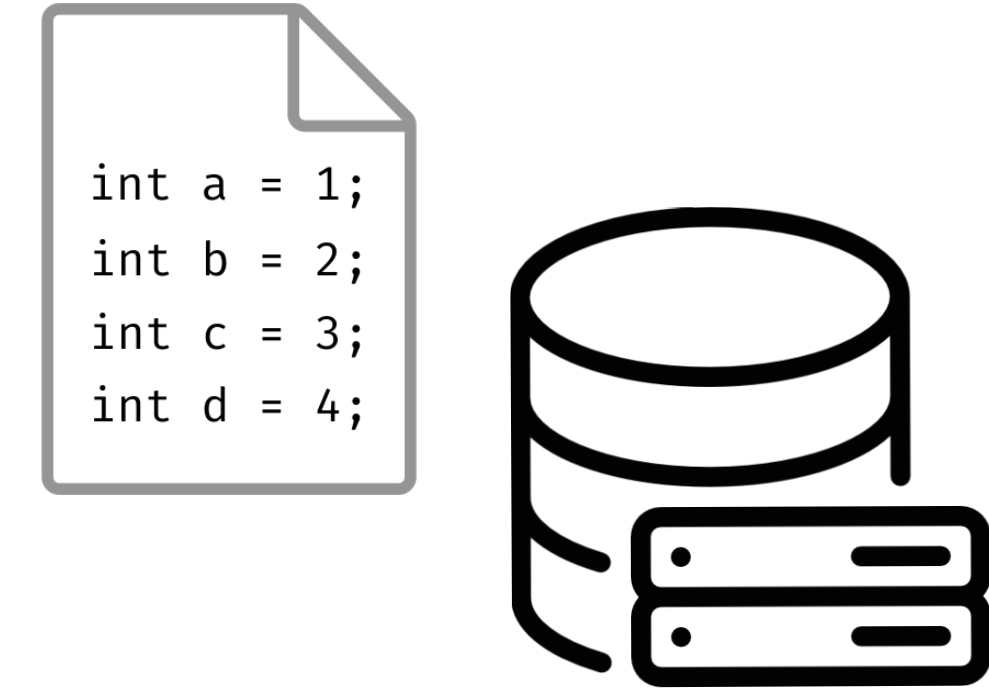
# git clone <url>

- Repository'nin kopyasını makineye indirir.
- "fork" ise başkasının repository'sini kopyalar.



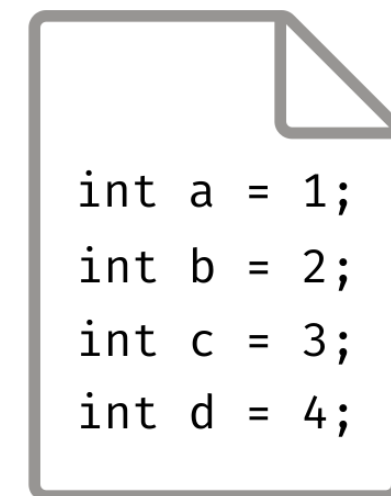
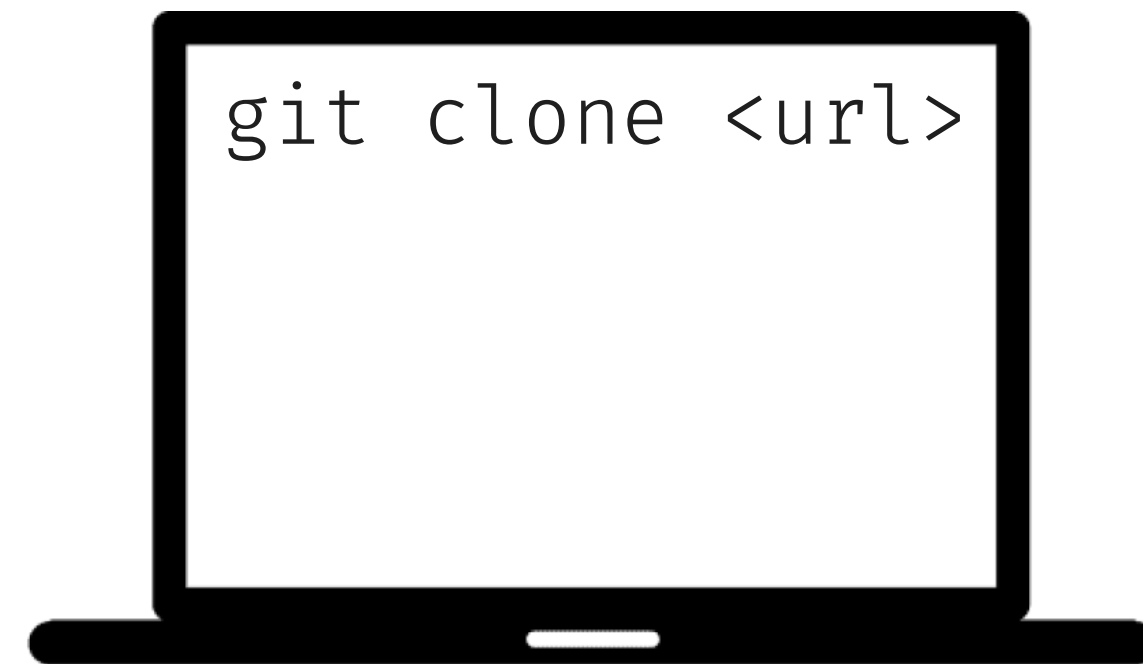
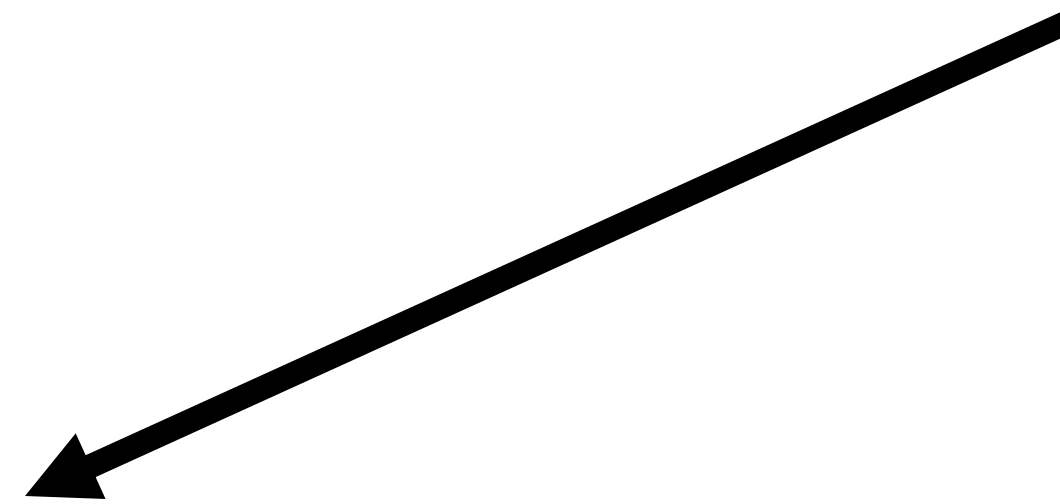
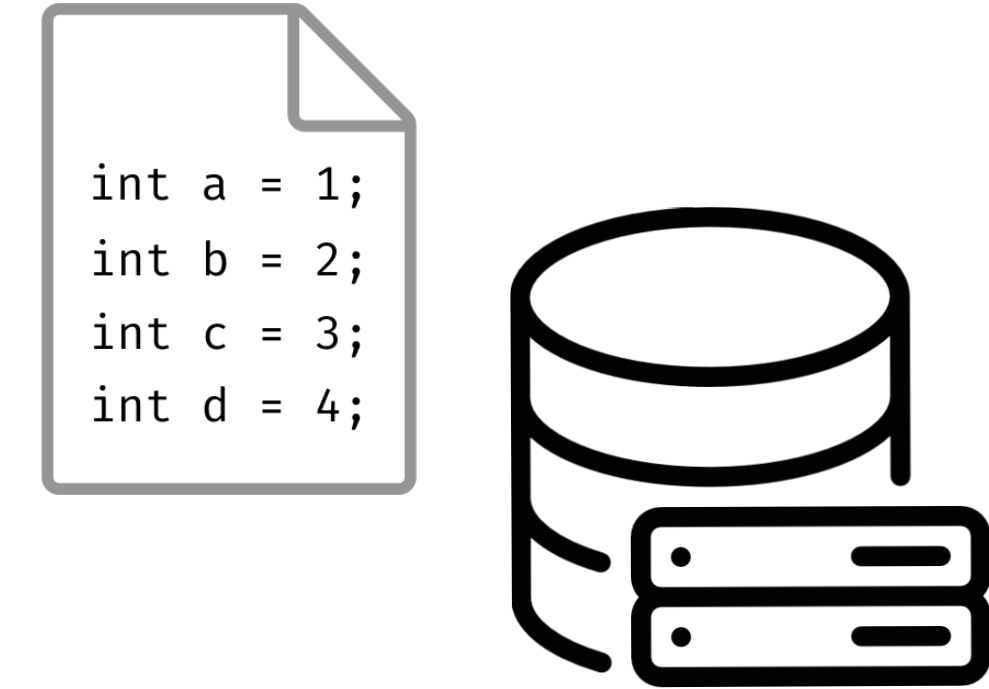
# git clone <url>

- Repository'nin kopyasını makineye indirir.
- "fork" ise başkasının repository'sini kopyalar.



# git clone <url>

- Repository'nin kopyasını makineye indirir.
- "fork" ise başkasının repository'sini kopyalar.





`git add`

# git add <filename>

- Yeni eklenen veya üzerinde değişiklik yapılan dosyayı “staged” statüsüne alır.
- Bir sonraki güncellemede dosyayı yüklemesi için talimat verir.

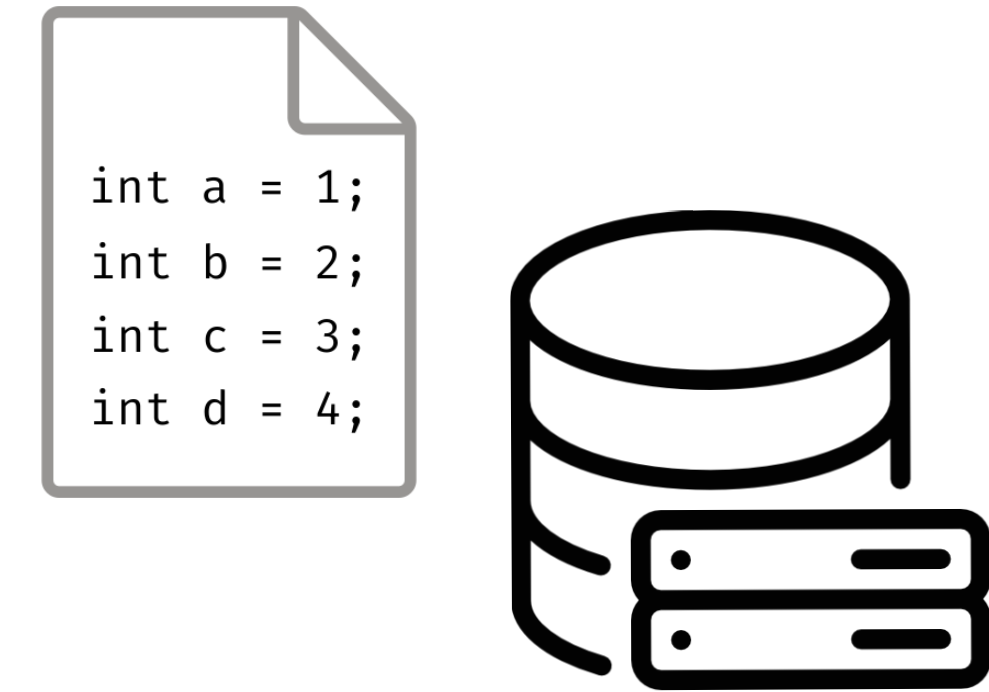
```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;  
int e = 5;
```

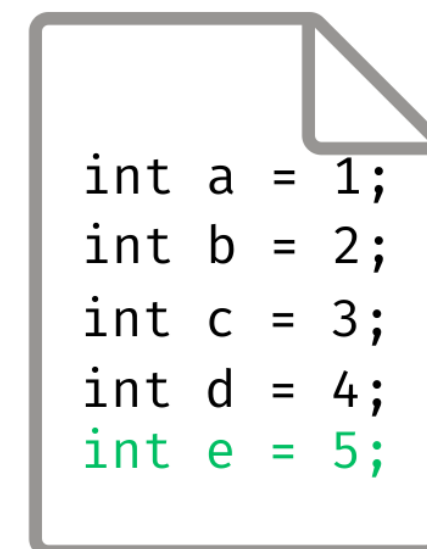
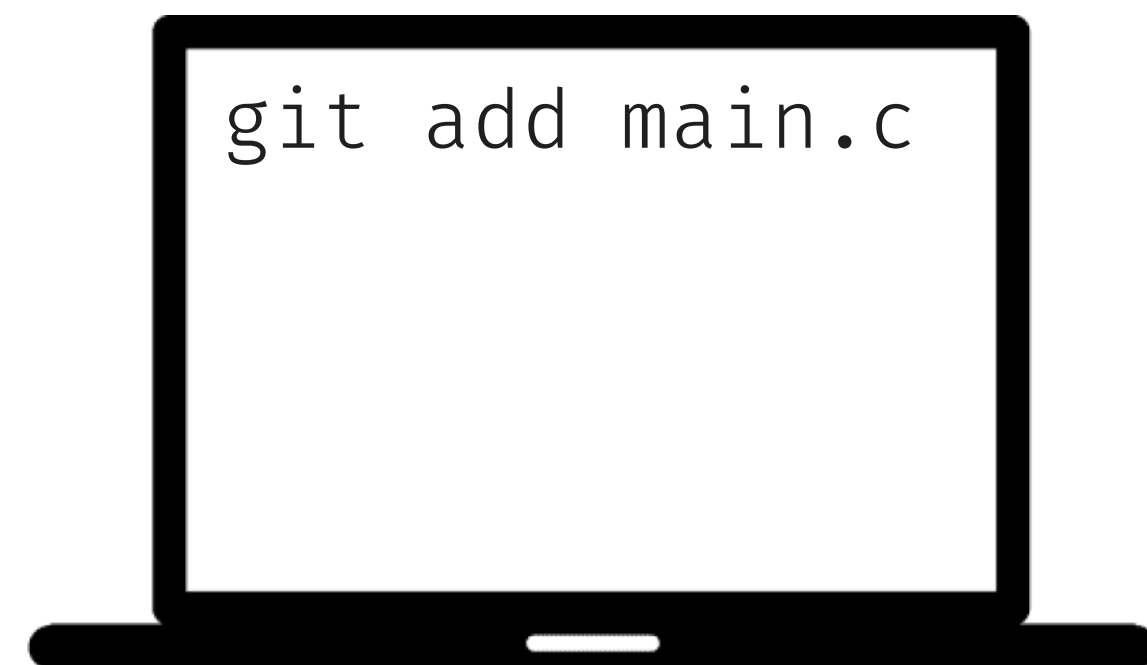
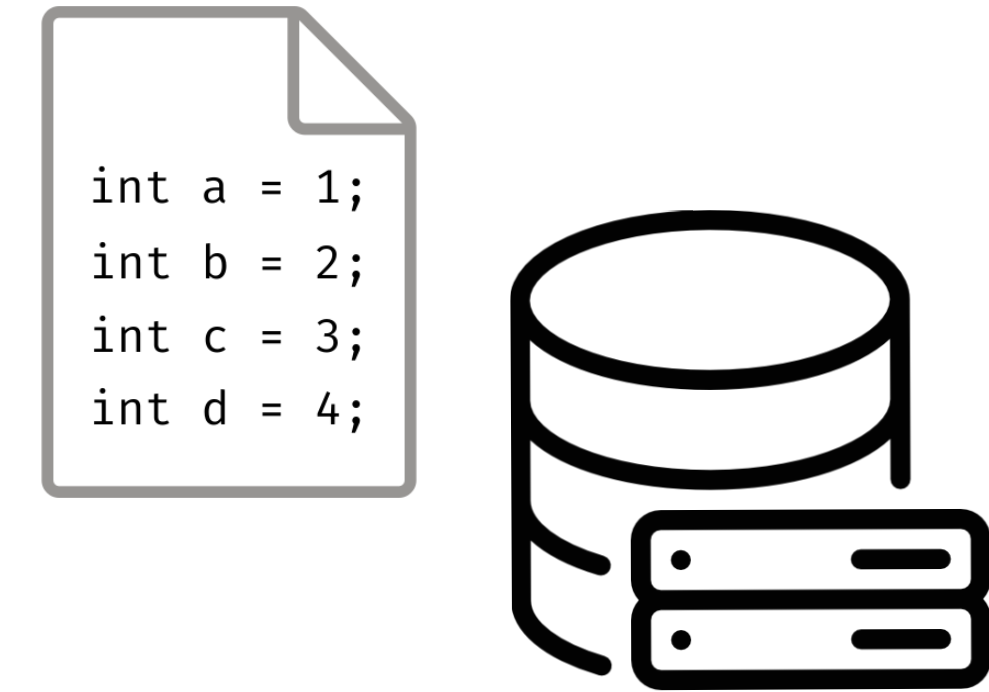
# git add <filename>

- Yeni eklenen veya üzerinde değişiklik yapılan dosyayı “staged” statüsüne alır.
- Bir sonraki güncellemede dosyayı yüklemesi için talimat verir.



# git add <filename>

- Yeni eklenen veya üzerinde değişiklik yapılan dosyayı “staged” statüsüne alır.
- Bir sonraki güncellemede dosyayı yüklemesi için talimat verir.



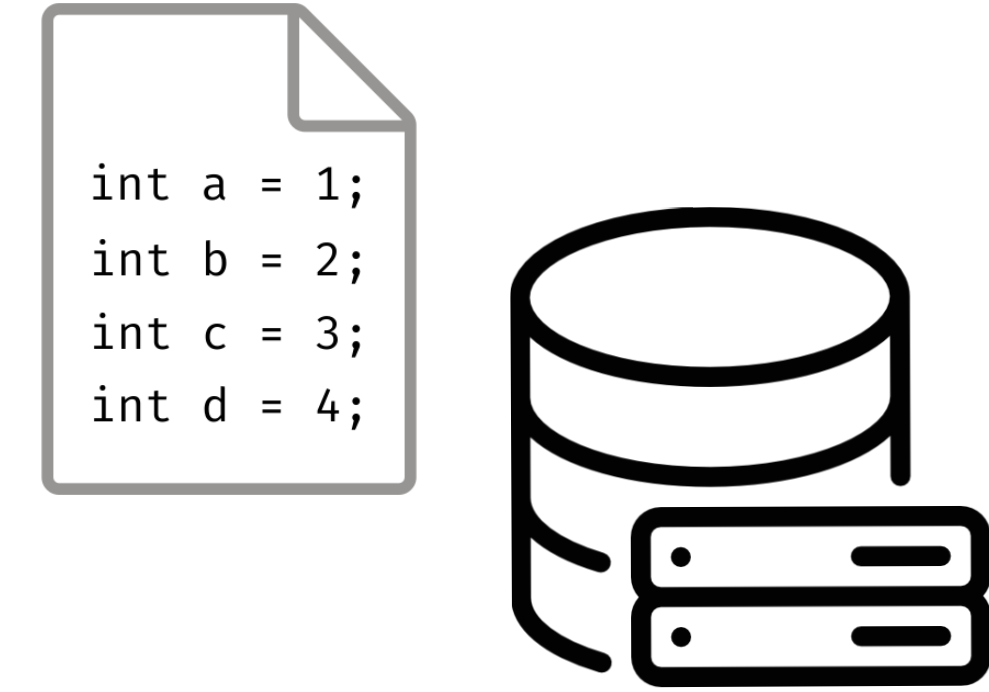
Changes to be committed:  
**modified: main.c**

git commit



# git commit -m "message"

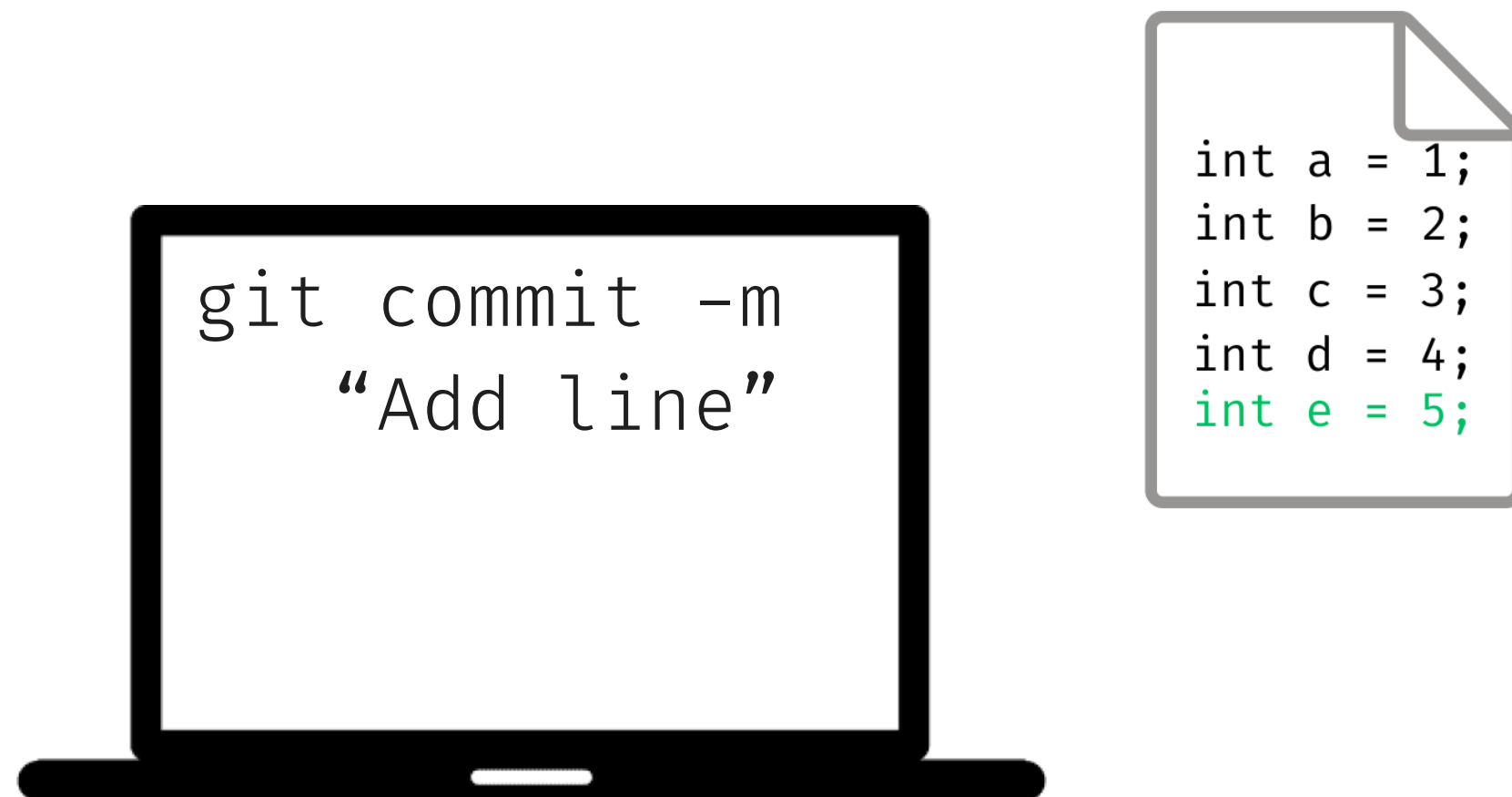
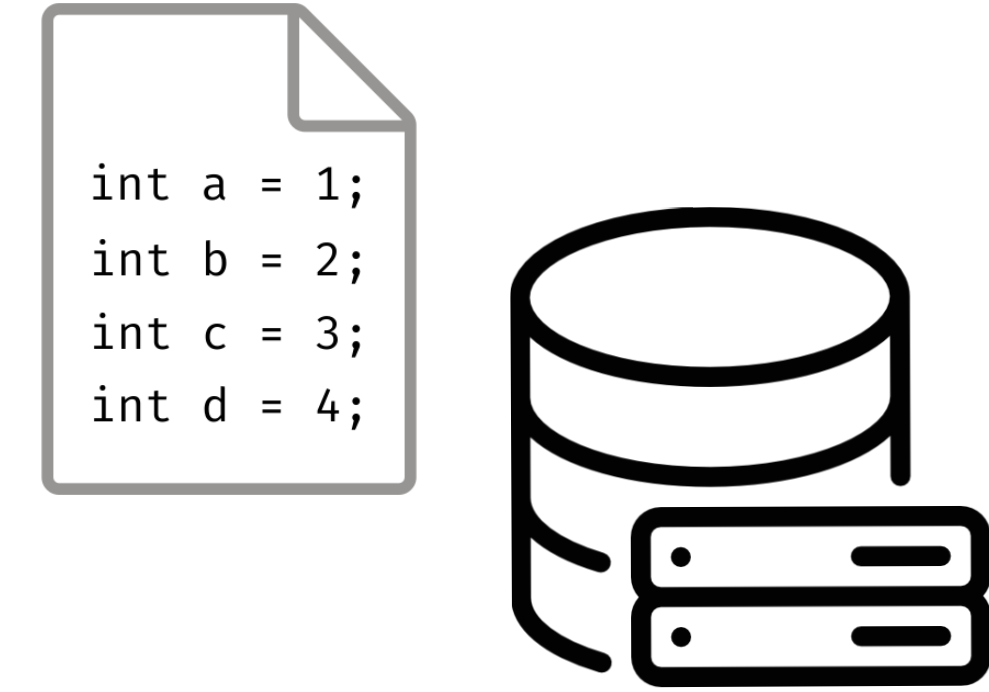
- Repository'de yapılan değişiklikleri yeni bir revizyon olarak kaydeder.
- Her revizyonun bir kayıt mesajı bulunur.
- git commit -am "message" komutu, ekleme ve kaydetme işlemlerini tek seferde yapar.





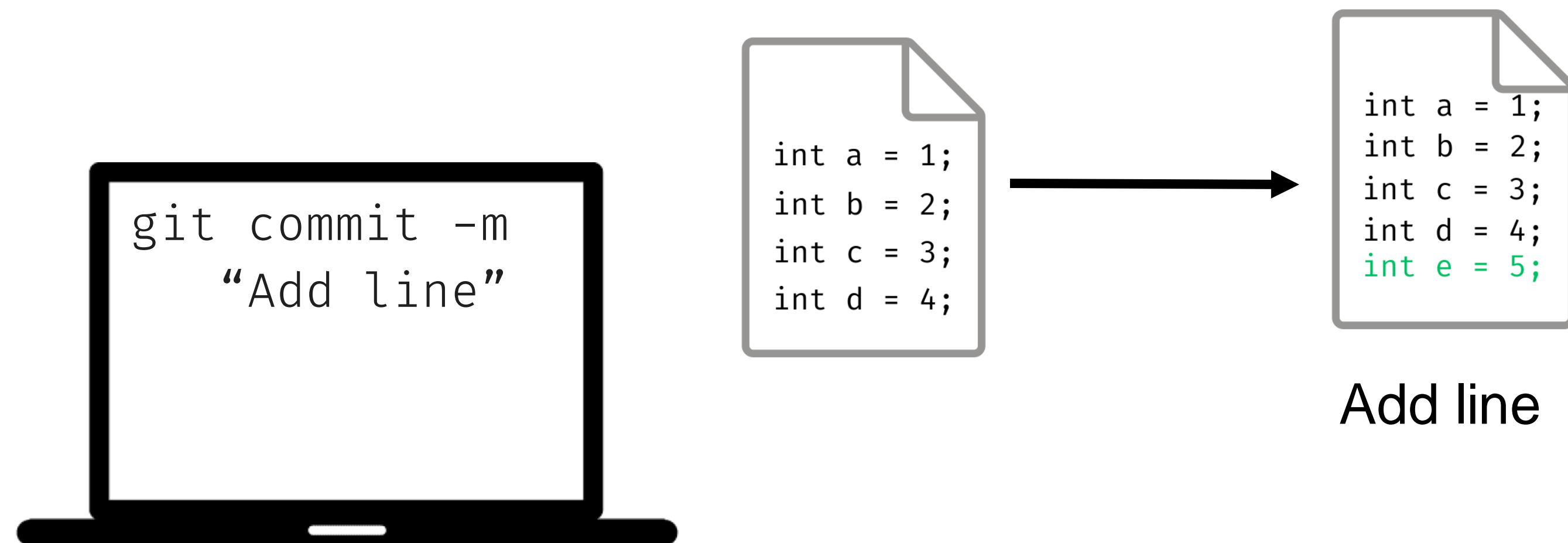
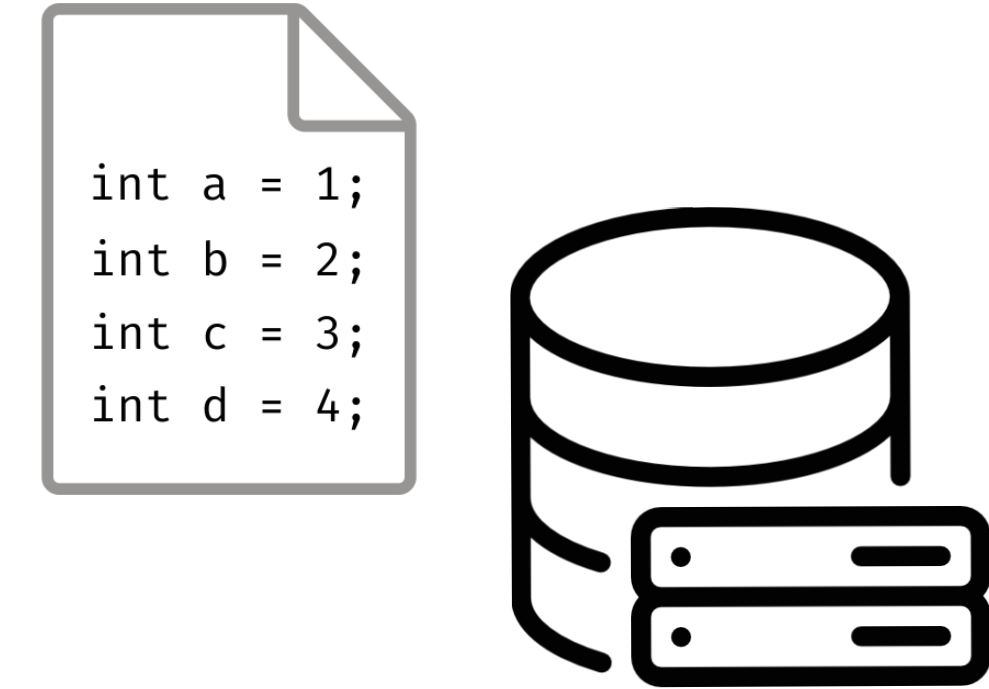
# git commit -m "message"

- Repository'de yapılan değişiklikleri yeni bir revizyon olarak kaydeder.
- Her revizyonun bir kayıt mesajı bulunur.
- git commit -am "message" komutu, ekleme ve kaydetme işlemlerini tek seferde yapar.



# git commit -m "message"

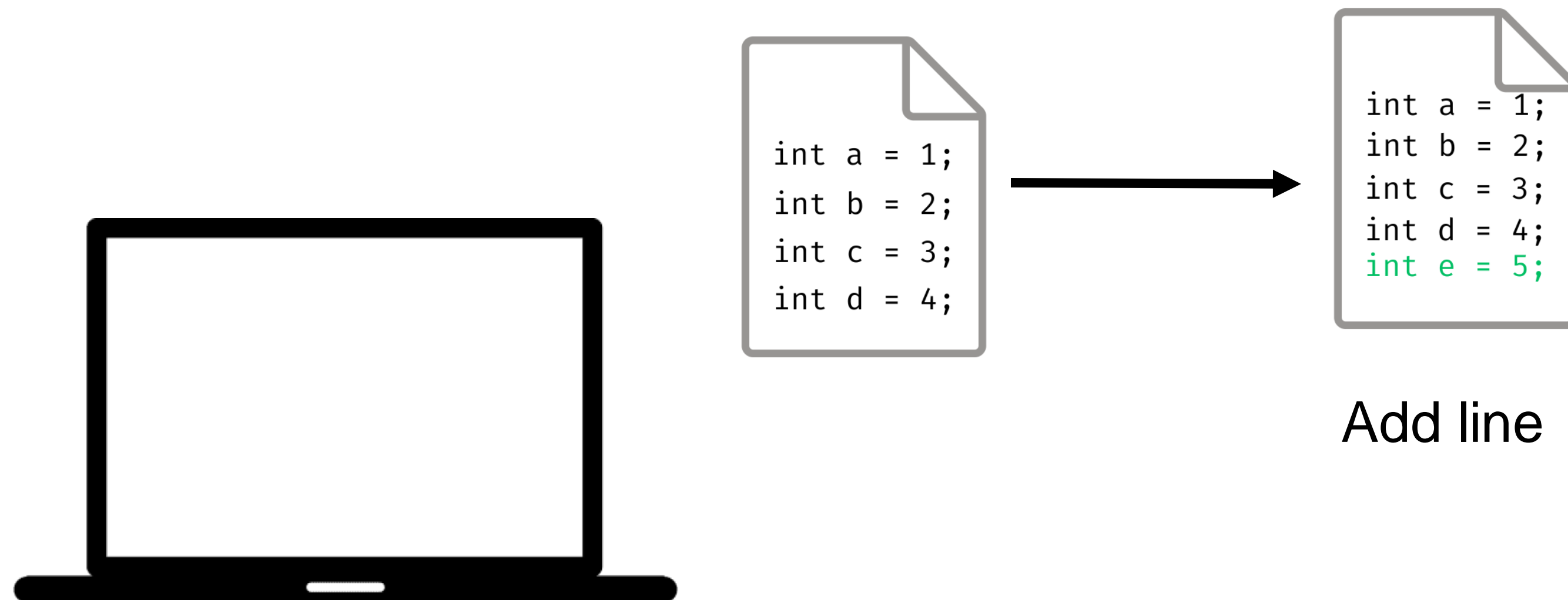
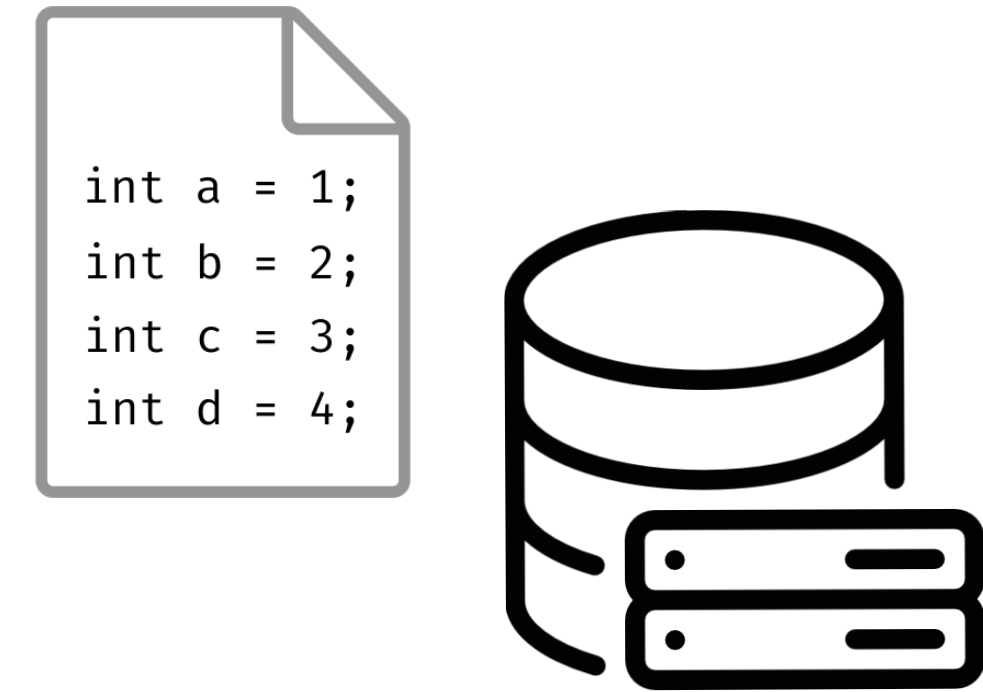
- Repository'de yapılan değişiklikleri yeni bir revizyon olarak kaydeder.
- Her revizyonun bir kayıt mesajı bulunur.
- `git commit -am "message"` komutu, ekleme ve kaydetme işlemlerini tek seferde yapar.



git status

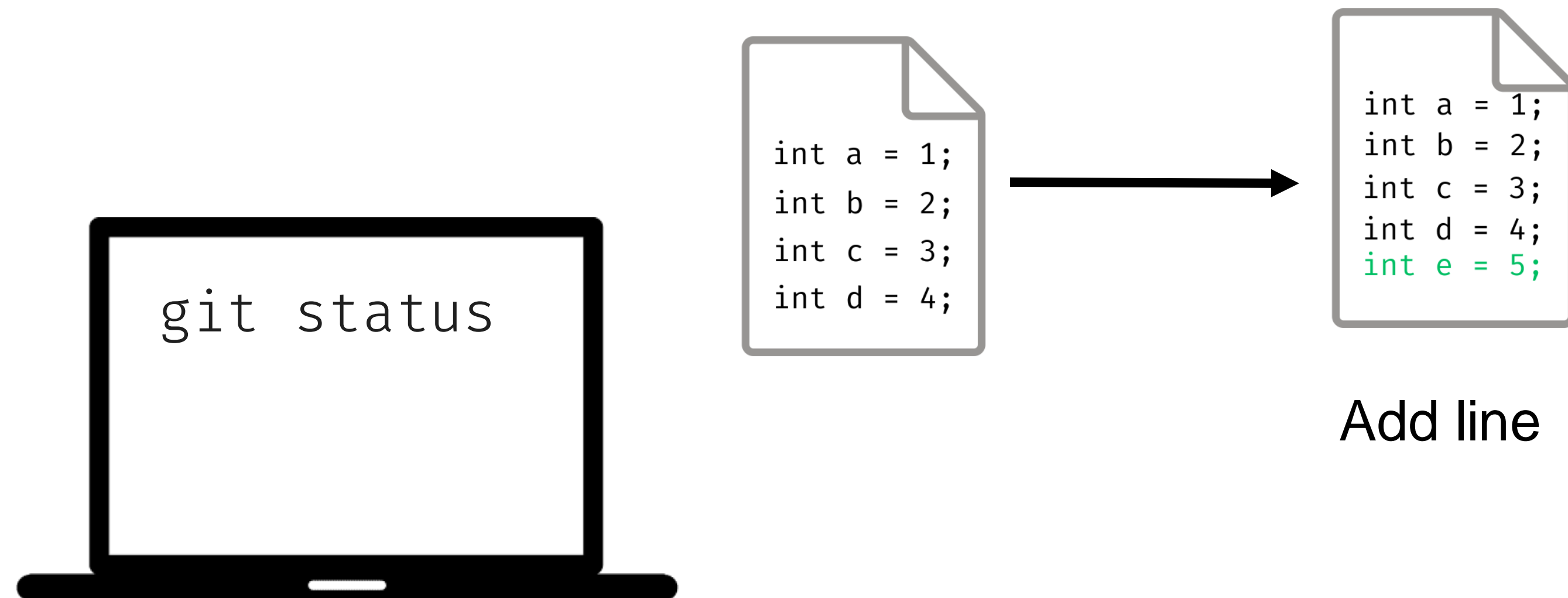
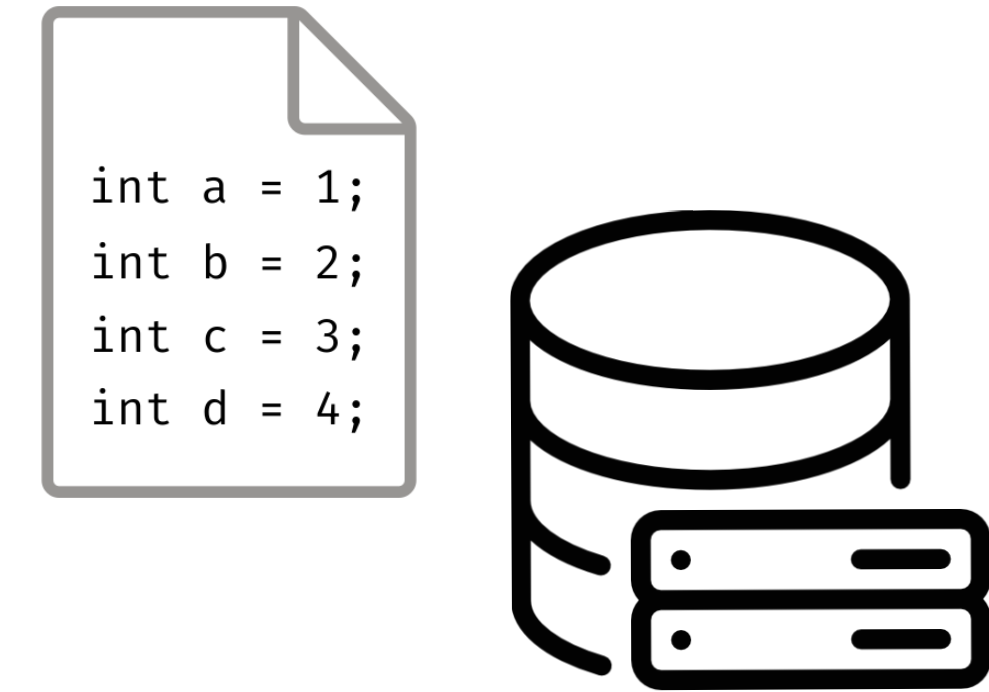
# git status

- Repository'nin güncel statüsünü verir.



# git status

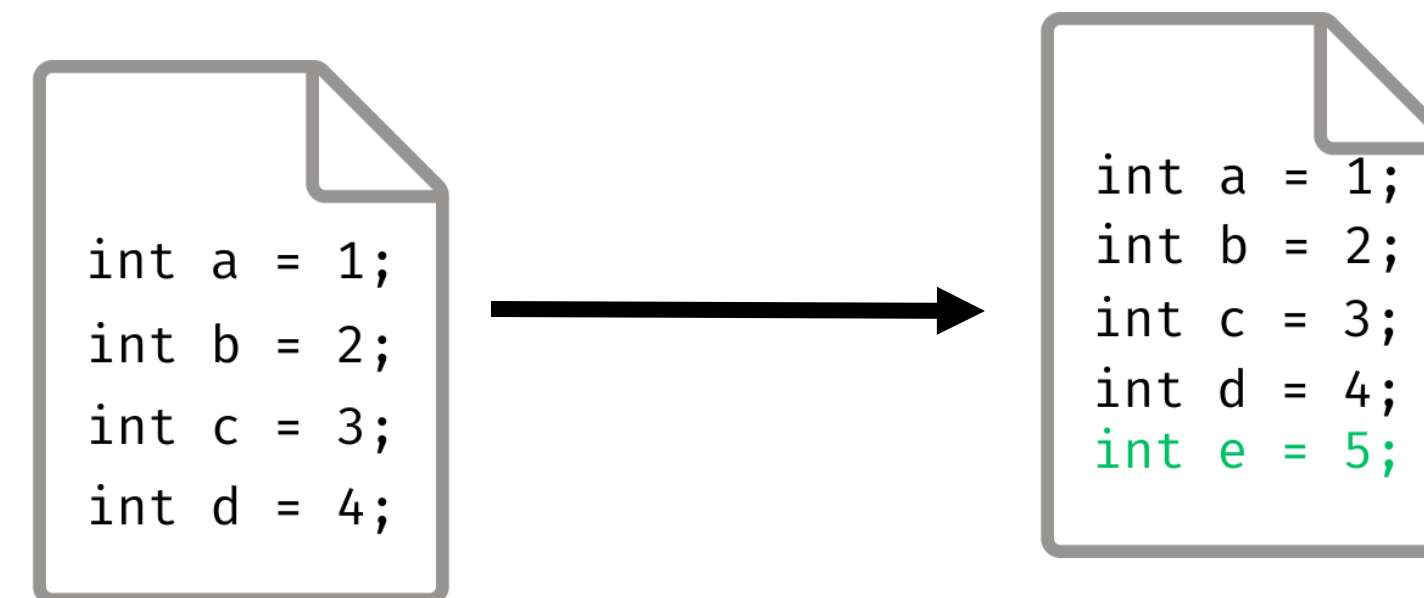
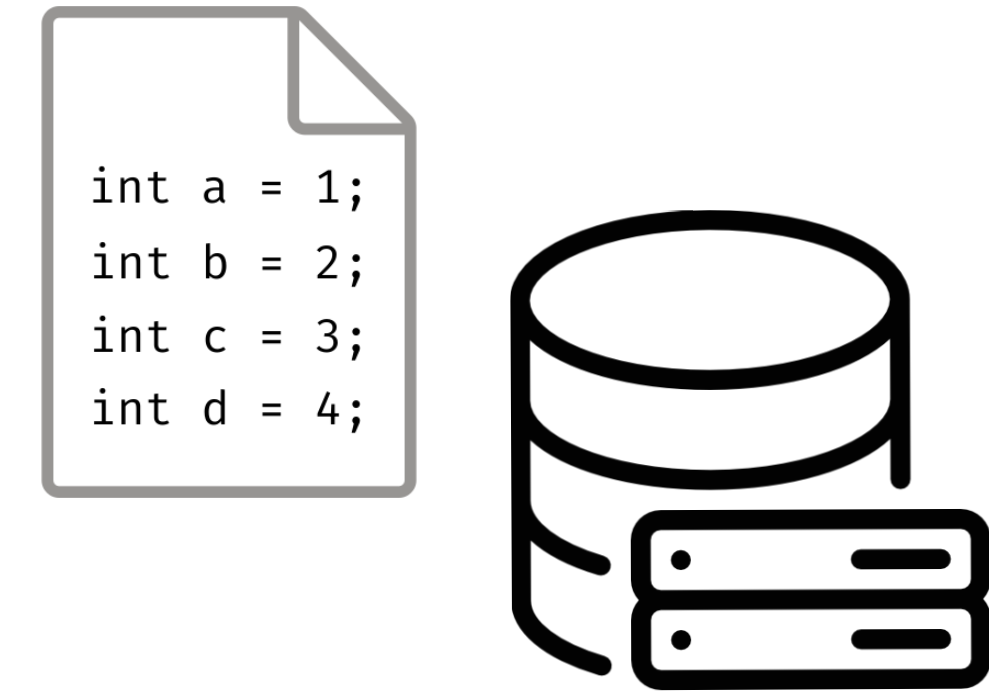
- Repository'nin güncel statüsünü verir.





# git status

- Repository'nin güncel statüsünü verir.



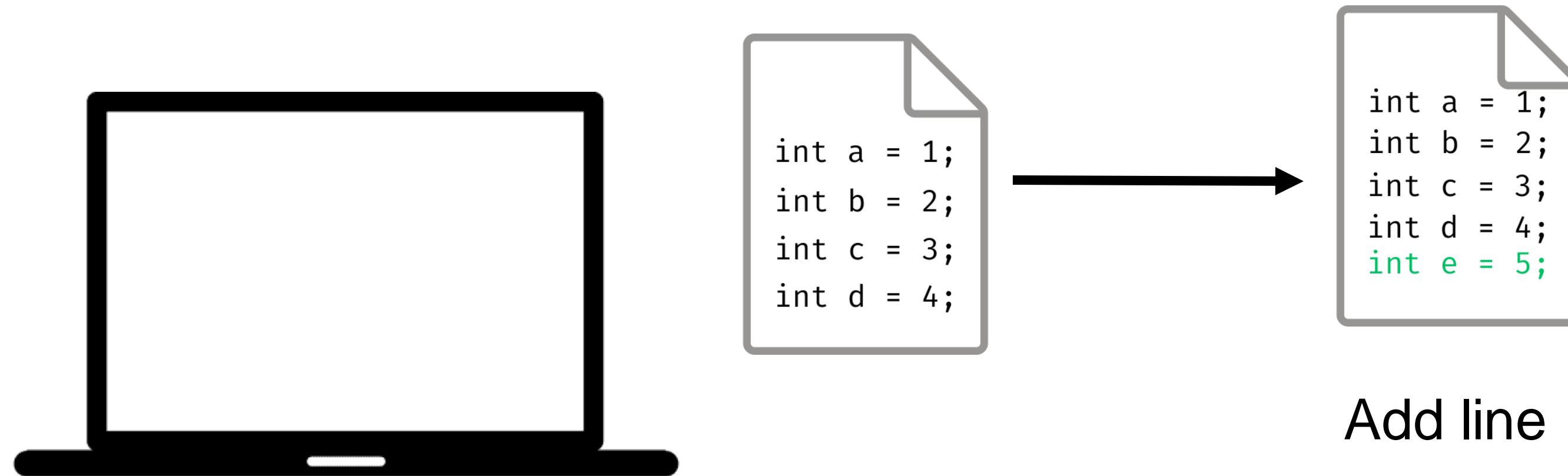
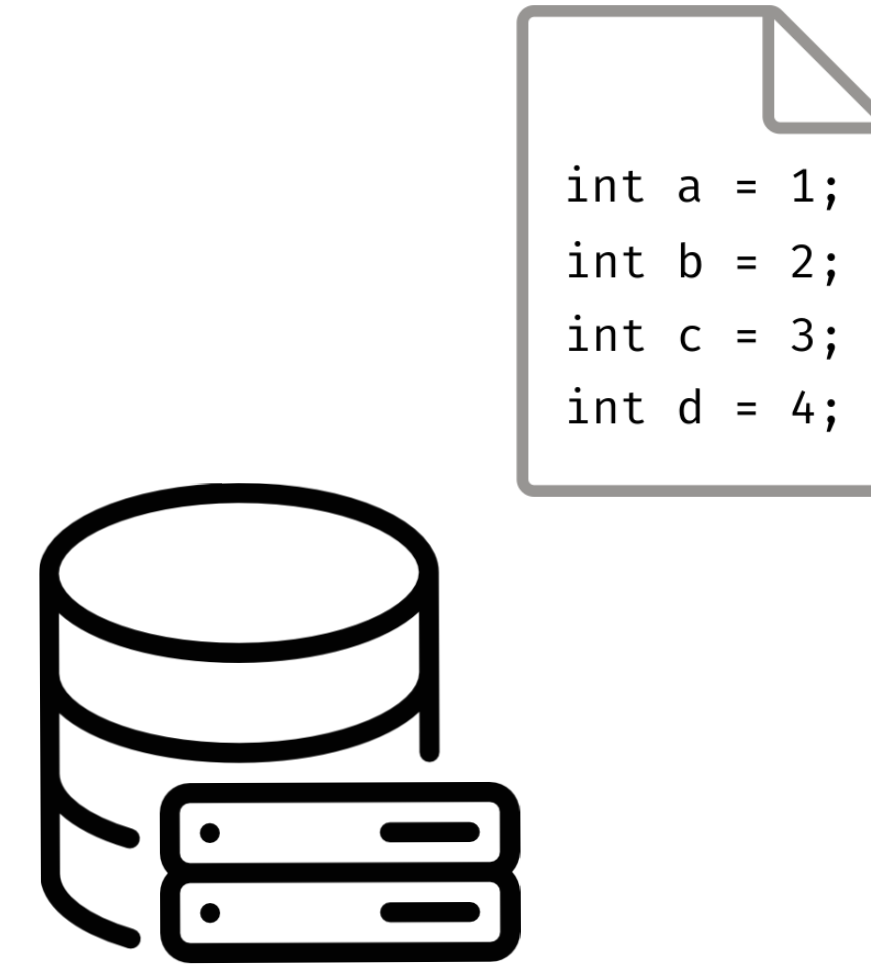
On branch master. **Add line**  
Your branch is ahead of 'origin/master' by 1 commit.  
( use "git push" to publish your local commits.)

git push



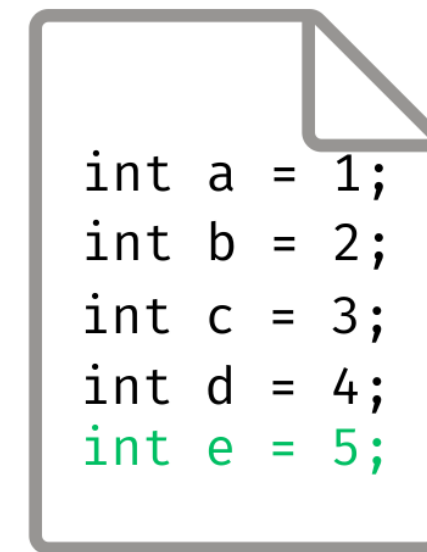
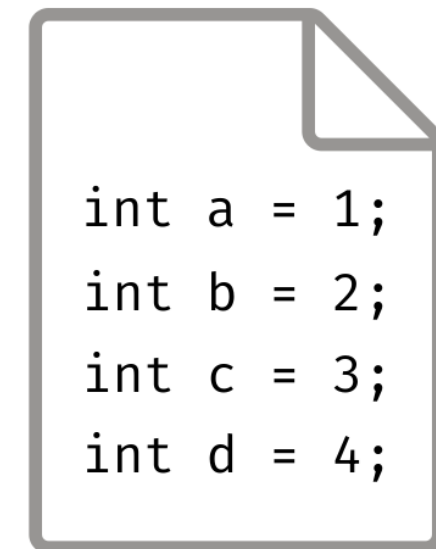
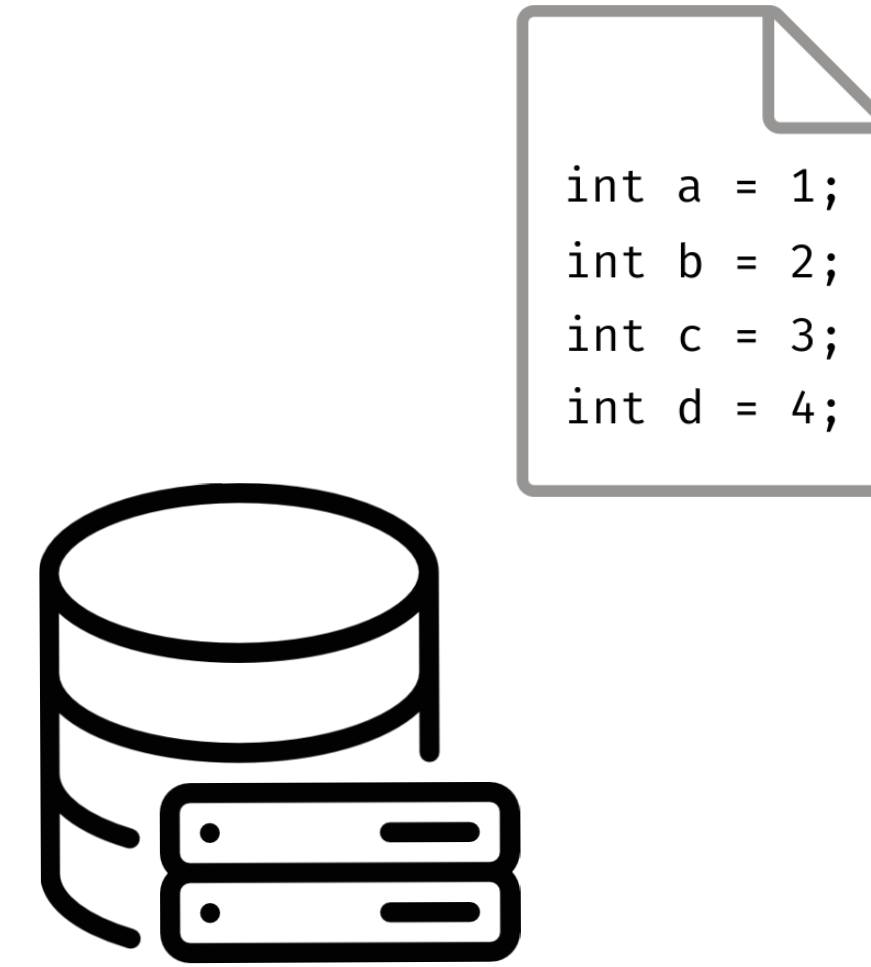
# git push

- Kaydedilen değişiklikleri kaynak repository'e gönderir.



# git push

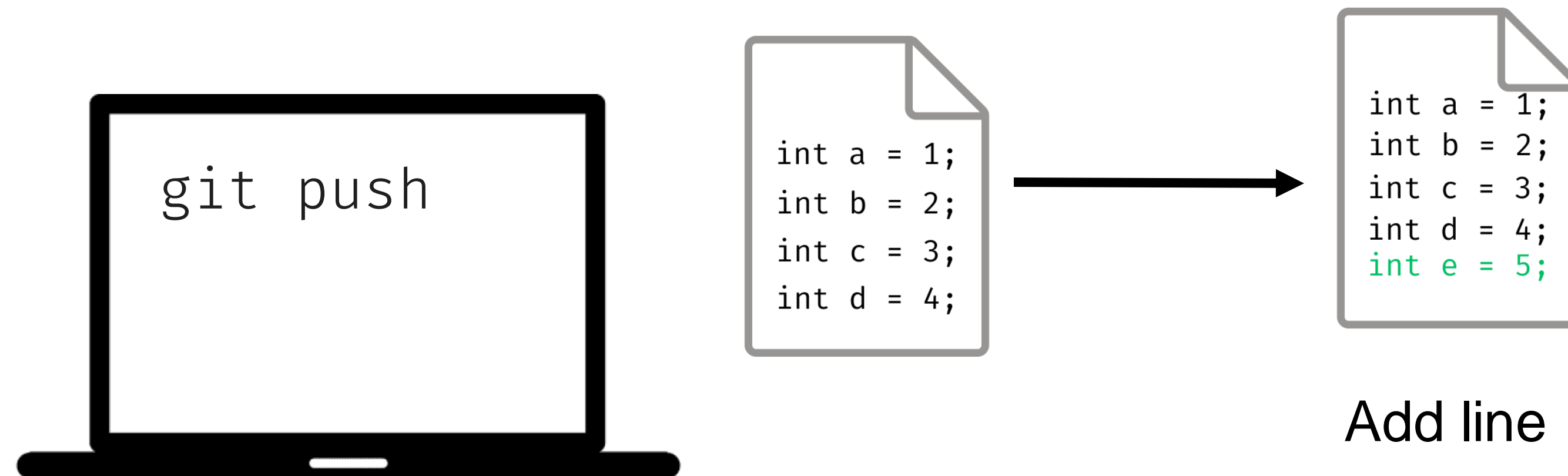
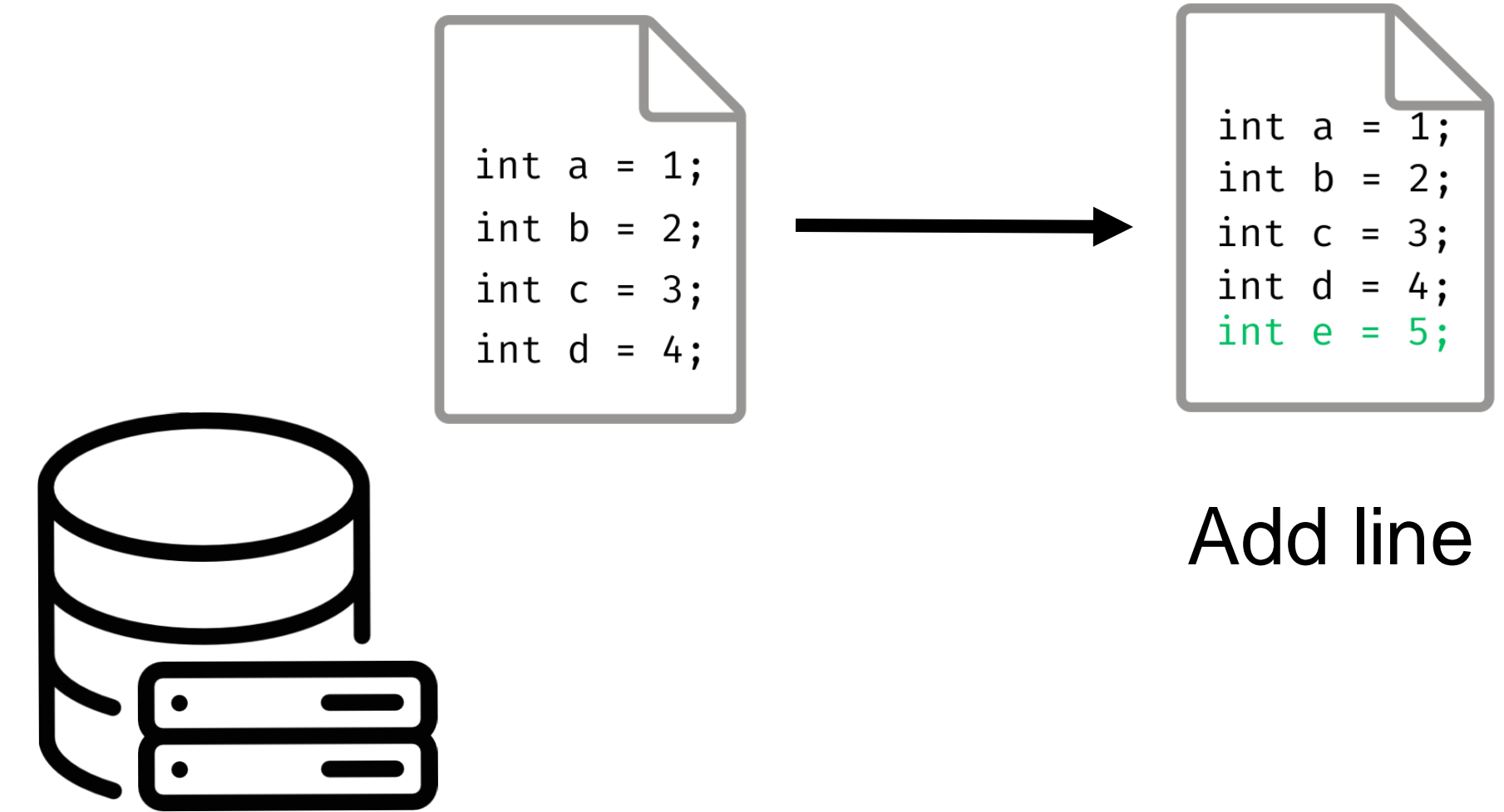
- Kaydedilen değişiklikleri kaynak repository'e gönderir.



Add line

# git push

- Kaydedilen değişiklikleri kaynak repository'e gönderir.



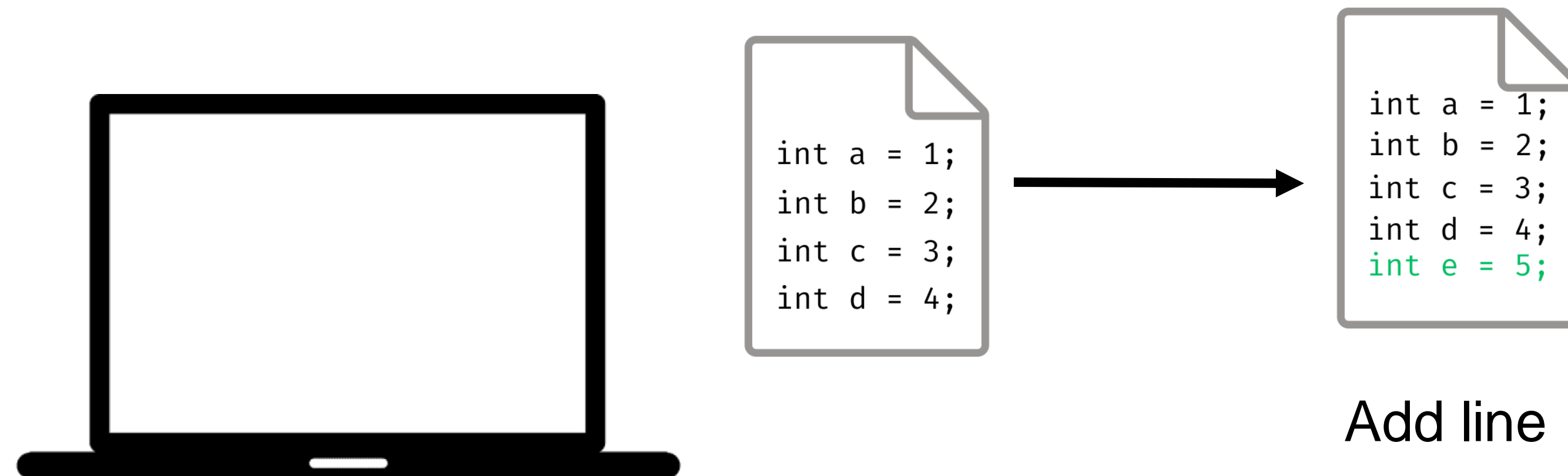
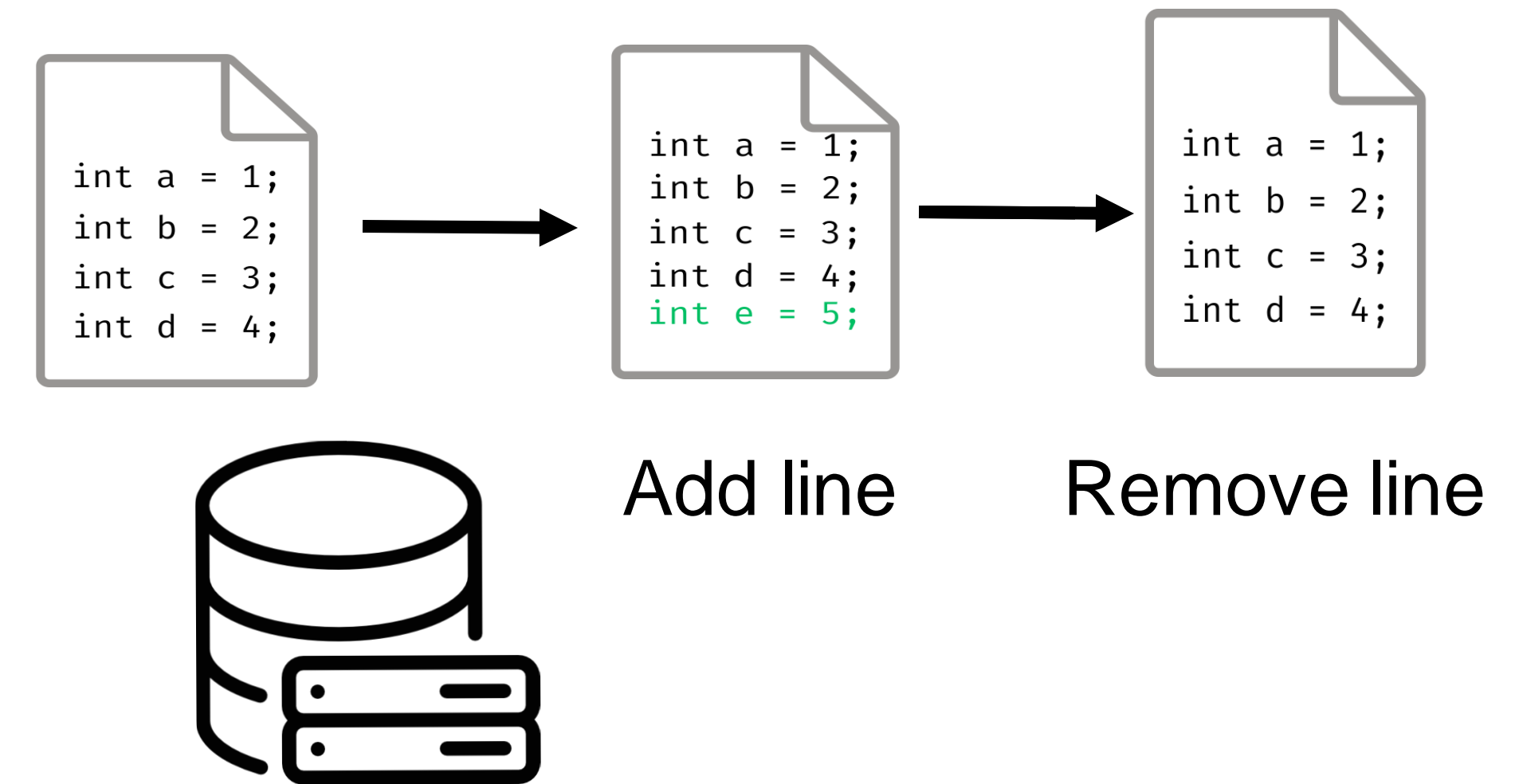


```
git pull
```



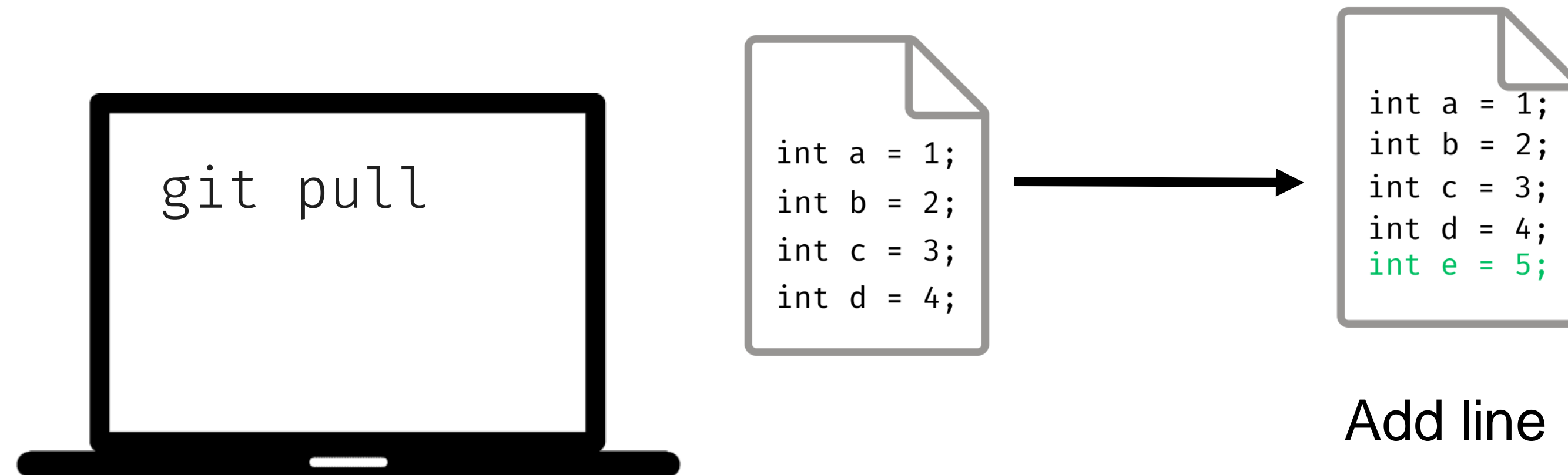
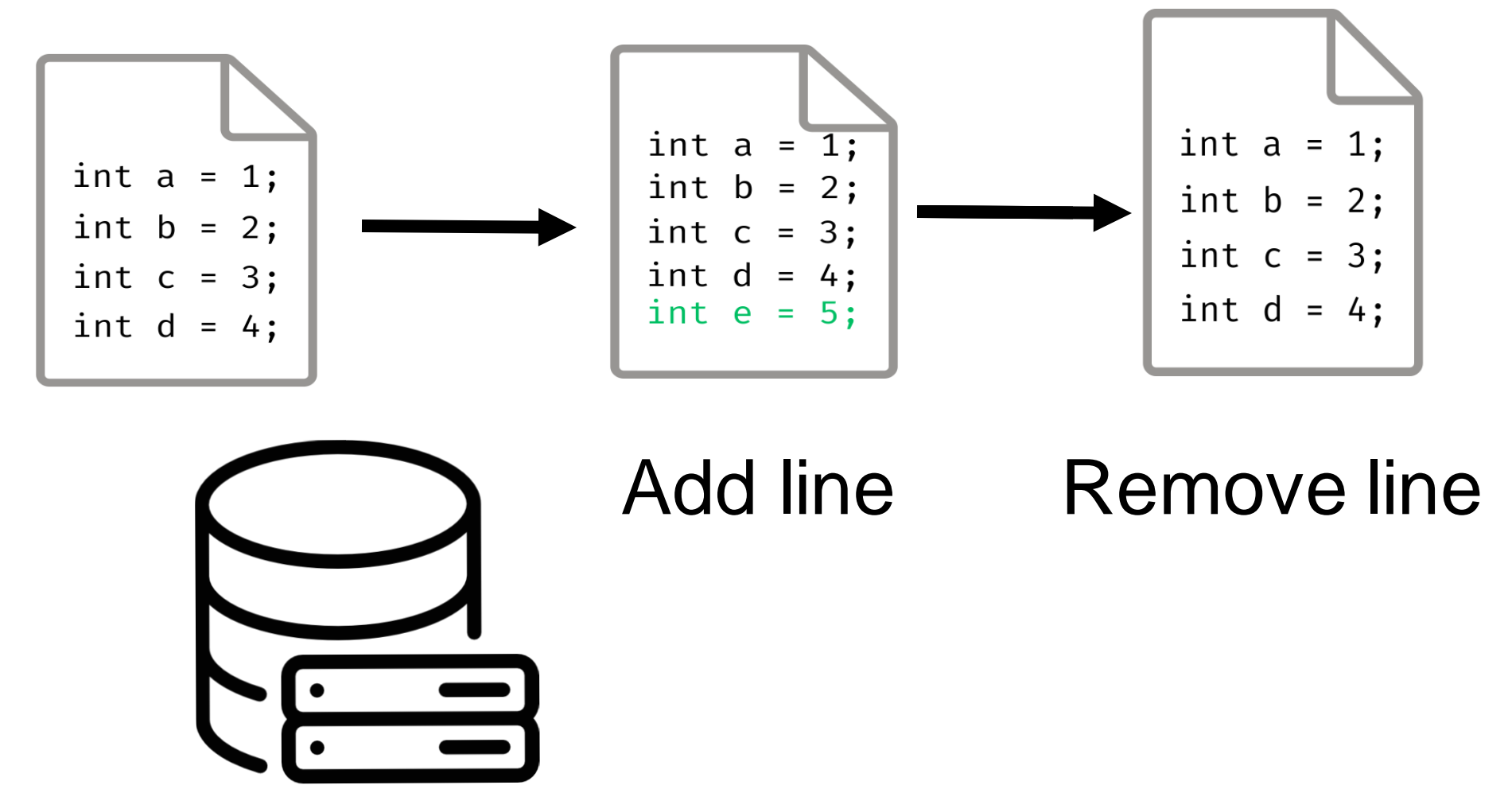
# git pull

- Güncel değişiklikleri kaynak repository'den alır.



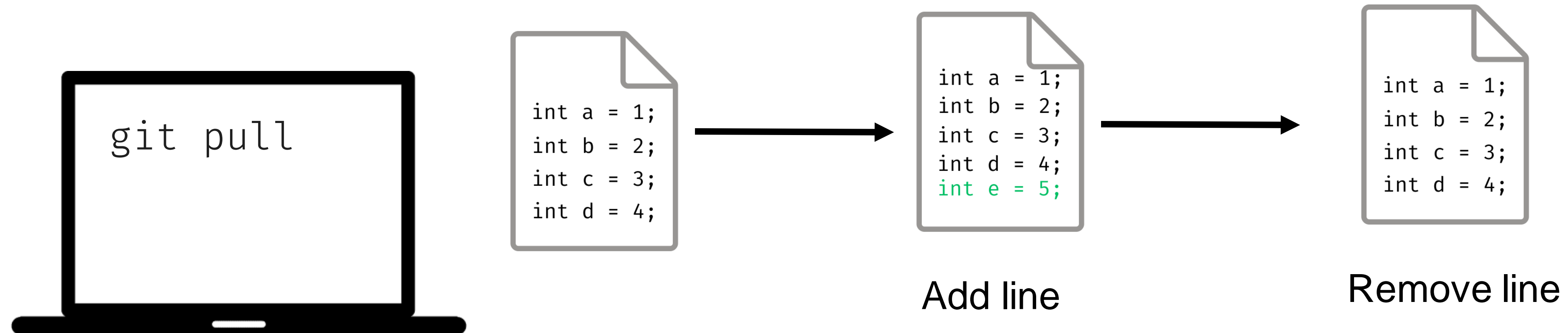
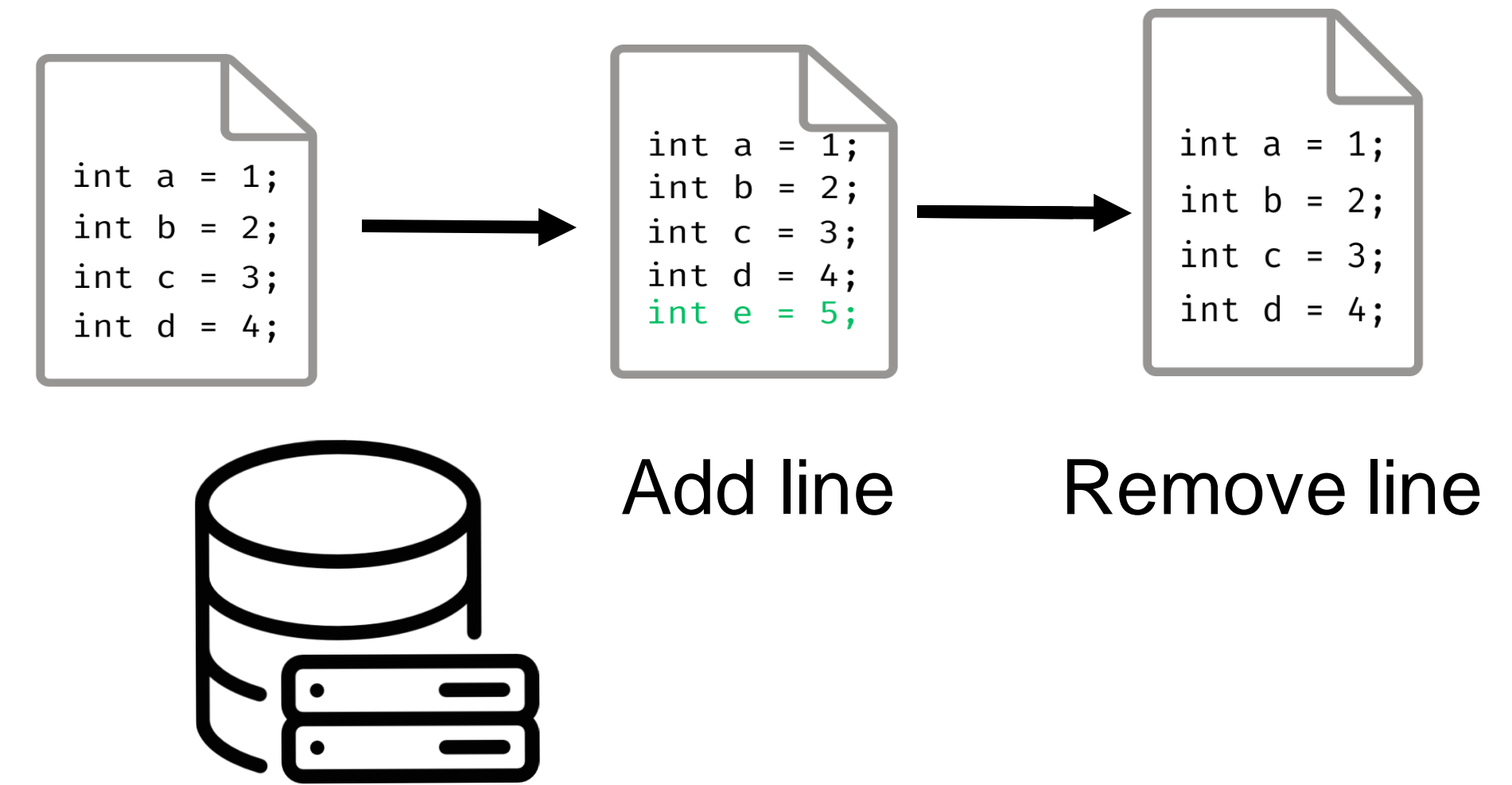
# git pull

- Güncel değişiklikleri kaynak repository'den alır.



# git pull

- Güncel değişiklikleri kaynak repository'den alır.



git log



# git log

- Commit geçmişini ve özetini gösterir.



# git log

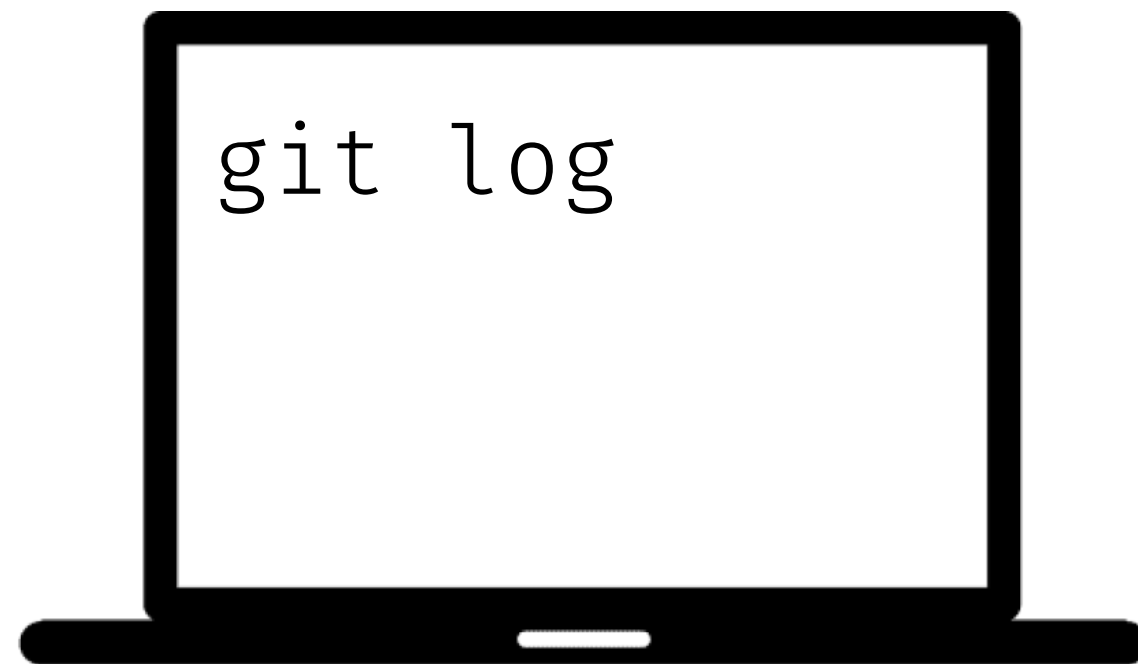
- Commit geçmişini ve özetini gösterir.

```
commit 4857647529537265925739267e
Author: Emir Balci <emirbalci360@gmail.com>
Date: Tue Oct 11 22:09:37 2023 -0400
```

Remove a line

```
commit 7493015470823768316743895e
Author: Emir Balci <emirbalci360@gmail.com>
Date: Tue Oct 11 22:02:18 2023 -0400
```

Add a line

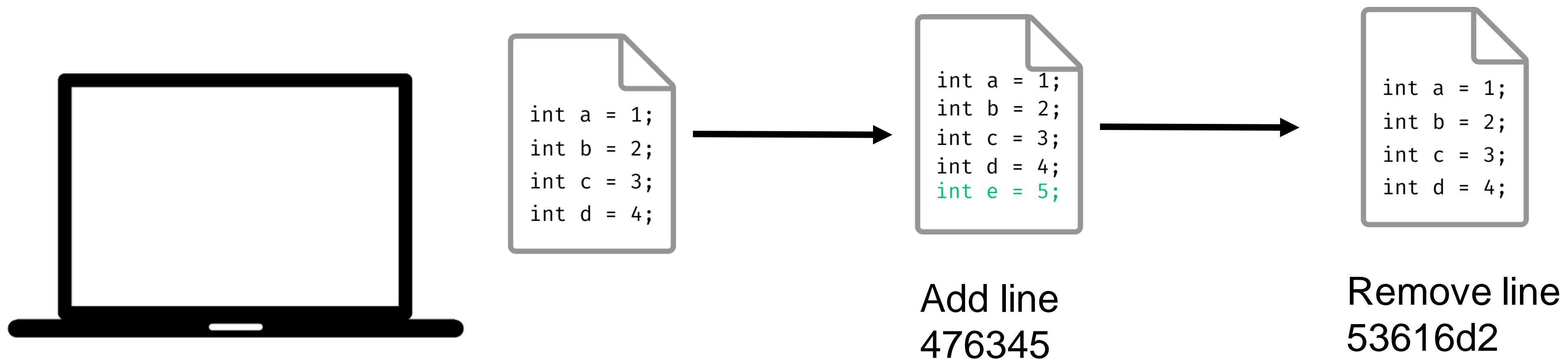


`git reset`



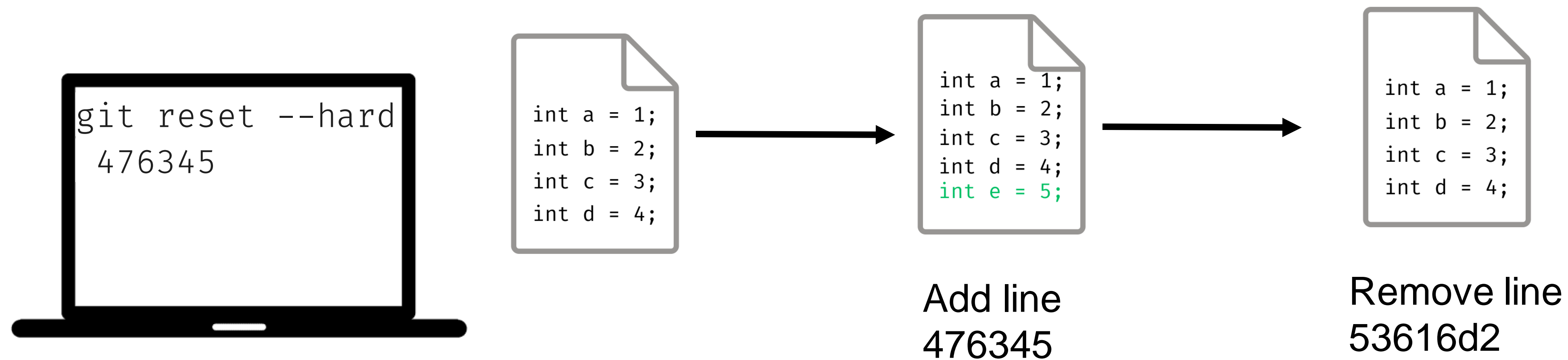
# git reset

- `git reset --hard <commit>`
- Kodu bir önceki commit'e alır.
- `git reset --hard origin/master`
- Kodu kaynak repository versiyonuna alır.



# git reset

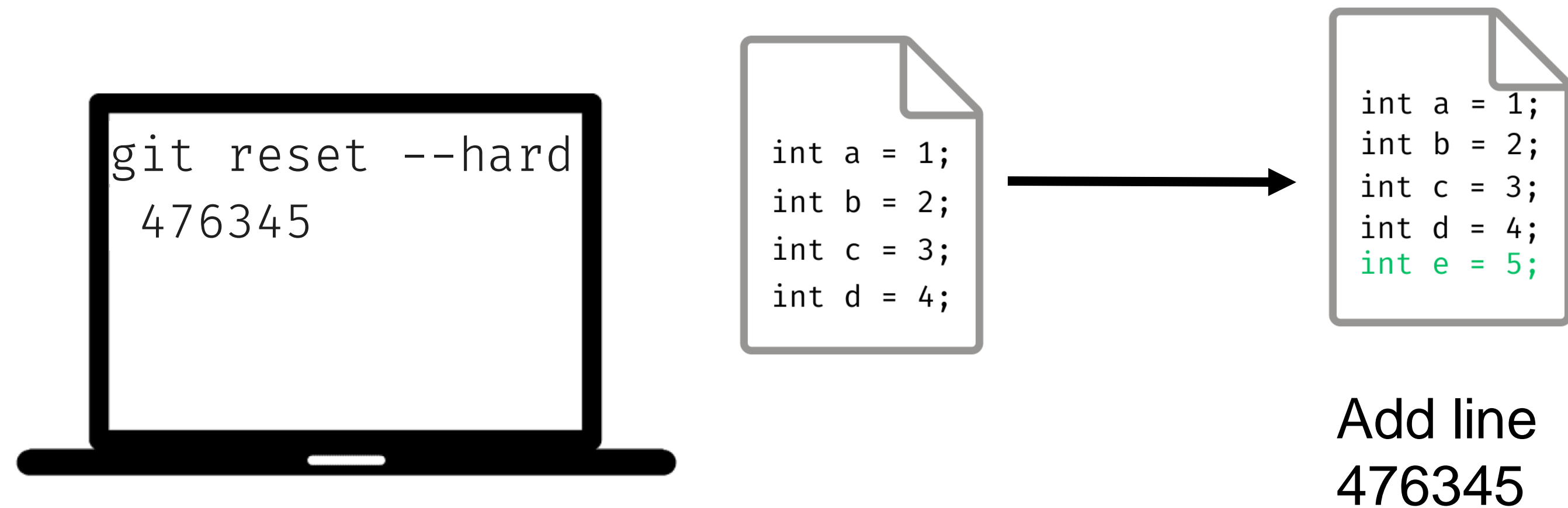
- `git reset --hard <commit>`
- Kodu bir önceki commit'e alır.
- `git reset --hard origin/master`
- Kodu kaynak repository versiyonuna alır.





# git reset

- `git reset --hard <commit>`
- Kodu bir önceki commit'e alır.
- `git reset --hard origin/master`
- Kodu kaynak repository versiyonuna alır.



# Branch



# Branch nedir?

- Branch, repository'nin farklı bir versiyonudur.
- Her branch, kendine ait commit geçmişi ve güncel bir
- versiyonu vardır.



git branch



# git branch

- Kodtaki tüm branch'leri gösterir.
- Yeni bir branch yaratmak için:
- `git branch <branch_name>`
- Farklı bir branch'e geçiş yapmak için:
- `git checkout <branch_name>`

master

```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```





# git branch

- Kodtaki tüm branch'leri gösterir.
- Yeni bir branch yaratmak için:
- `git branch <branch_name>`
- Farklı bir branch'e geçiş yapmak için:
- `git checkout <branch_name>`

master

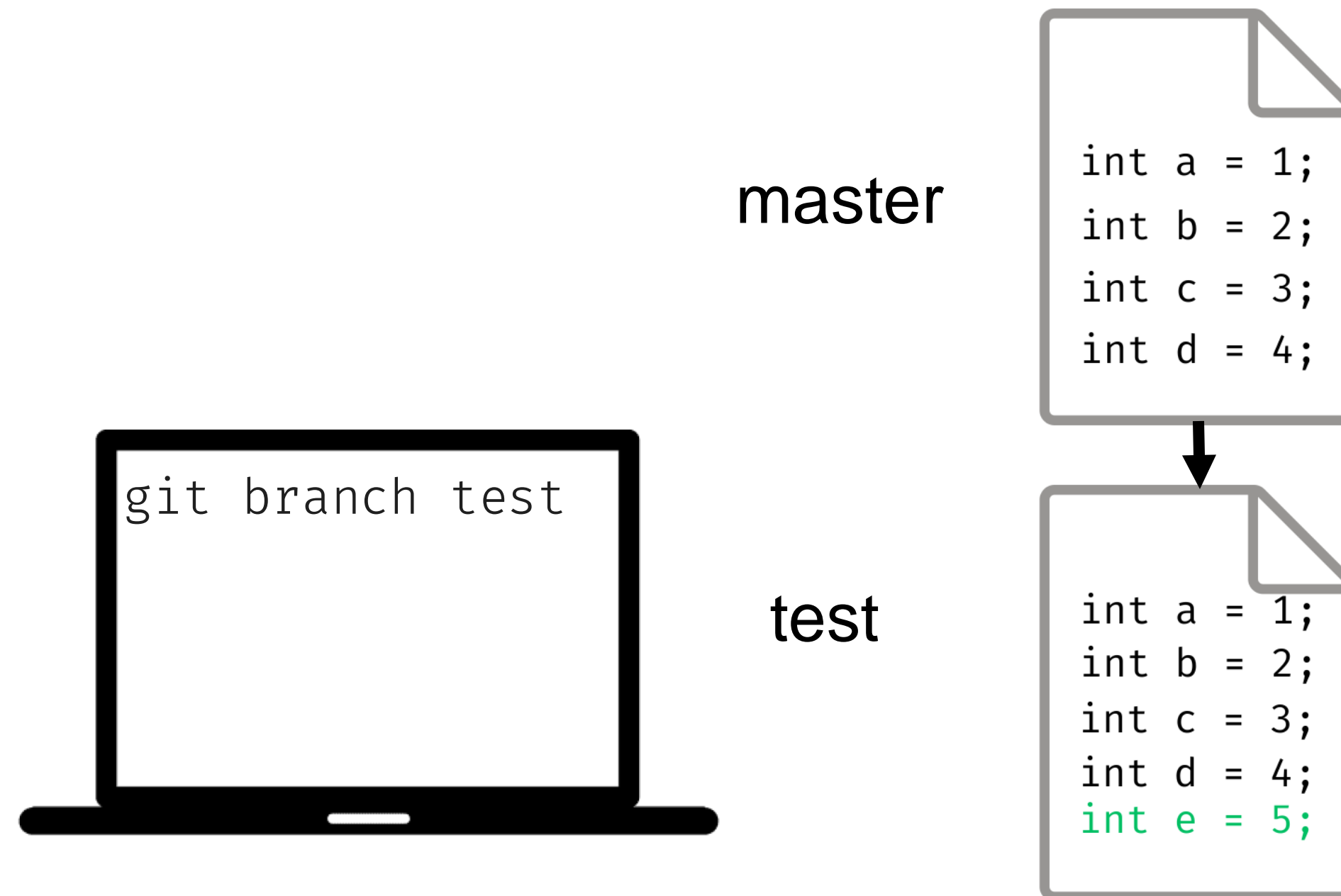
```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;
```



```
git branch test
```

# git branch

- Kodtaki tüm branch'leri gösterir.
- Yeni bir branch yaratmak için:
- `git branch <branch_name>`
- Farklı bir branch'e geçiş yapmak için:
- `git checkout <branch_name>`

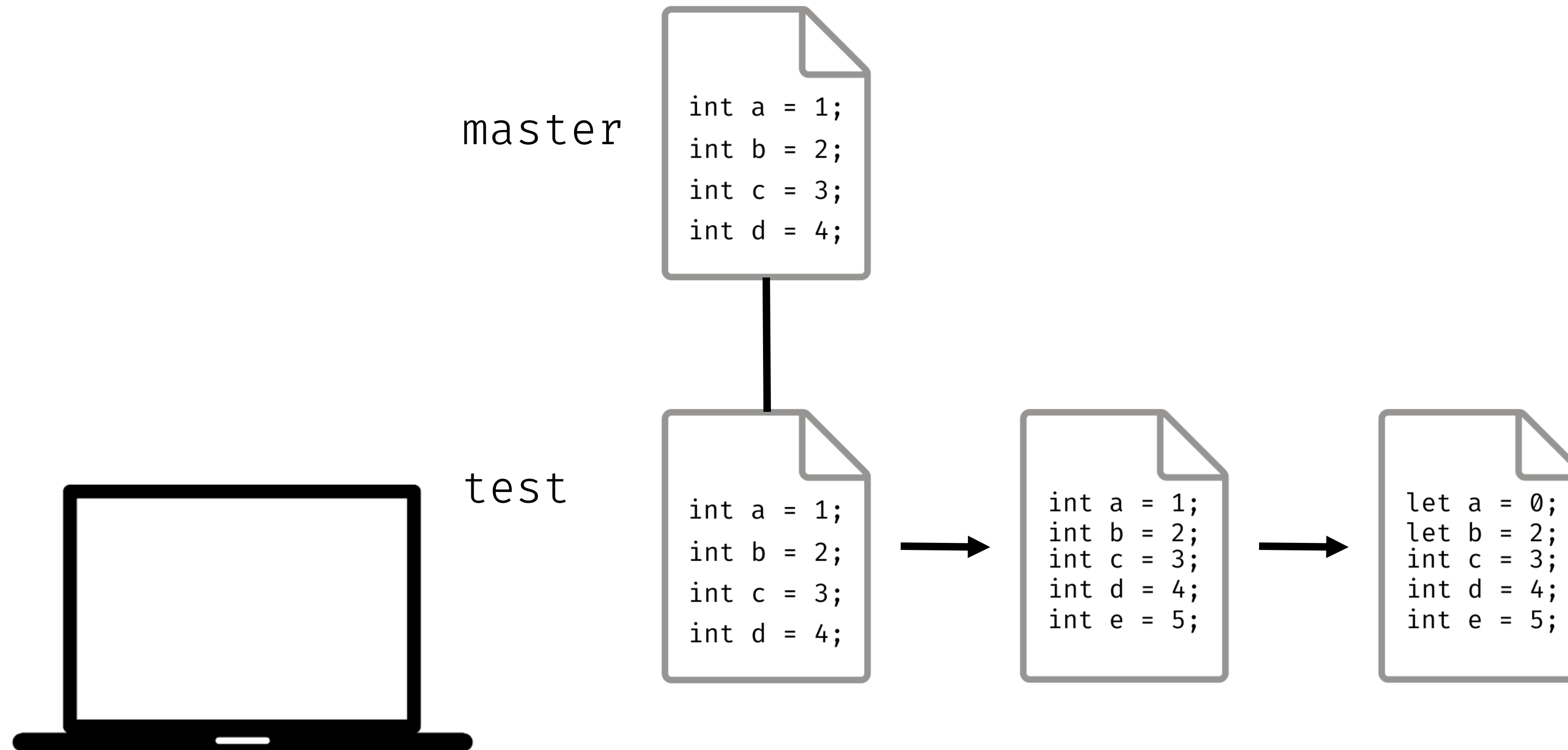


git merge



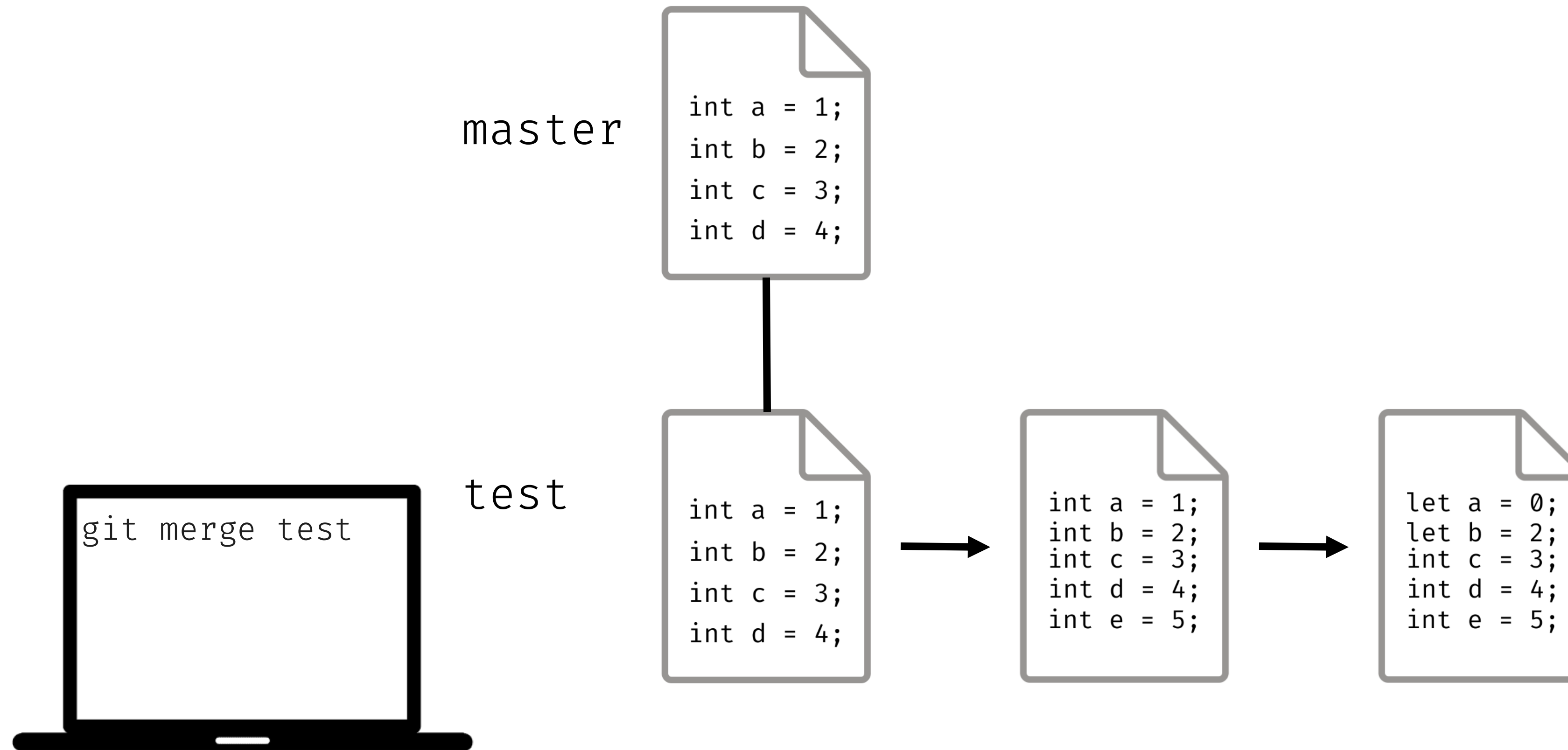
# git merge

- `git merge <branch_name>` komutu geçerli branch ile "branch\_name" isimli branch'i birleştirir.



# git merge

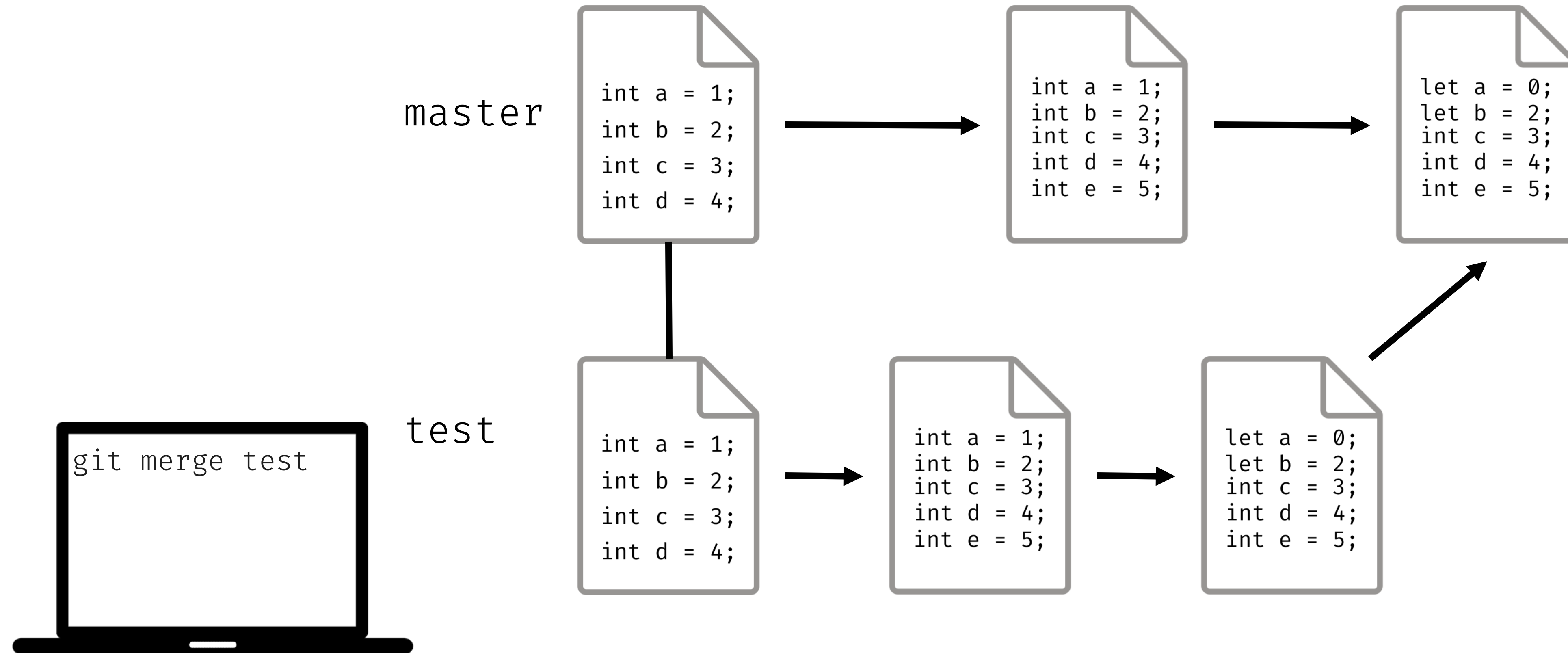
- `git merge <branch_name>` komutu geçerli branch ile "`branch_name`" isimli branch'i birleştirir.





# git merge

- `git merge <branch_name>` komutu geçerli branch ile "`branch_name`" isimli branch'i birleştirir.

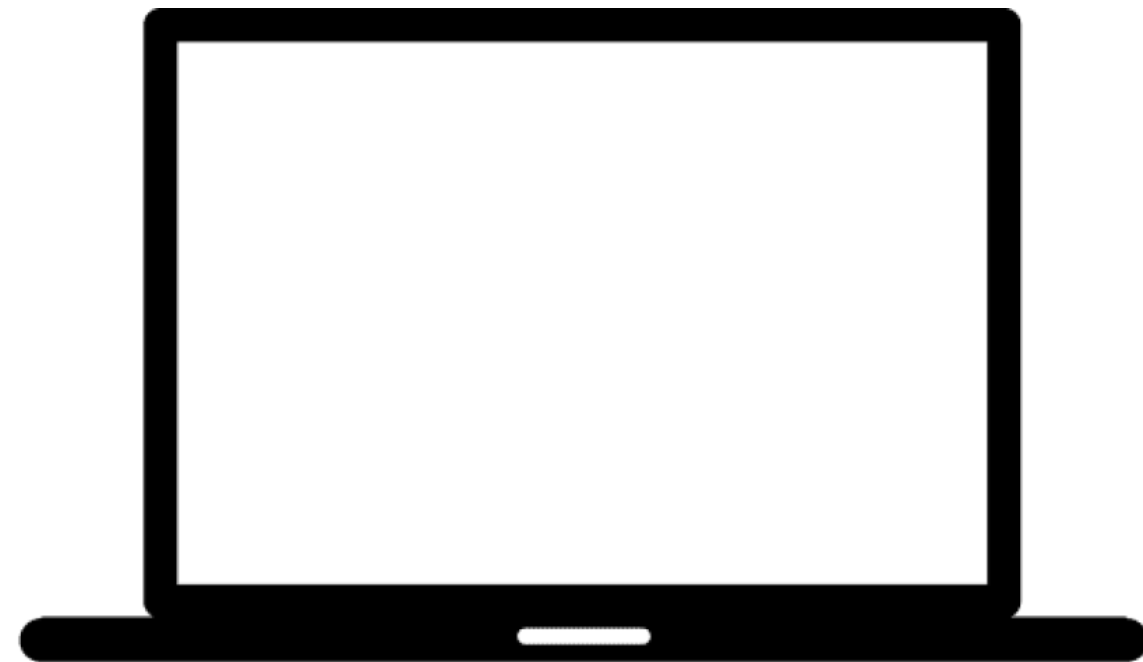


# Merge Conflict



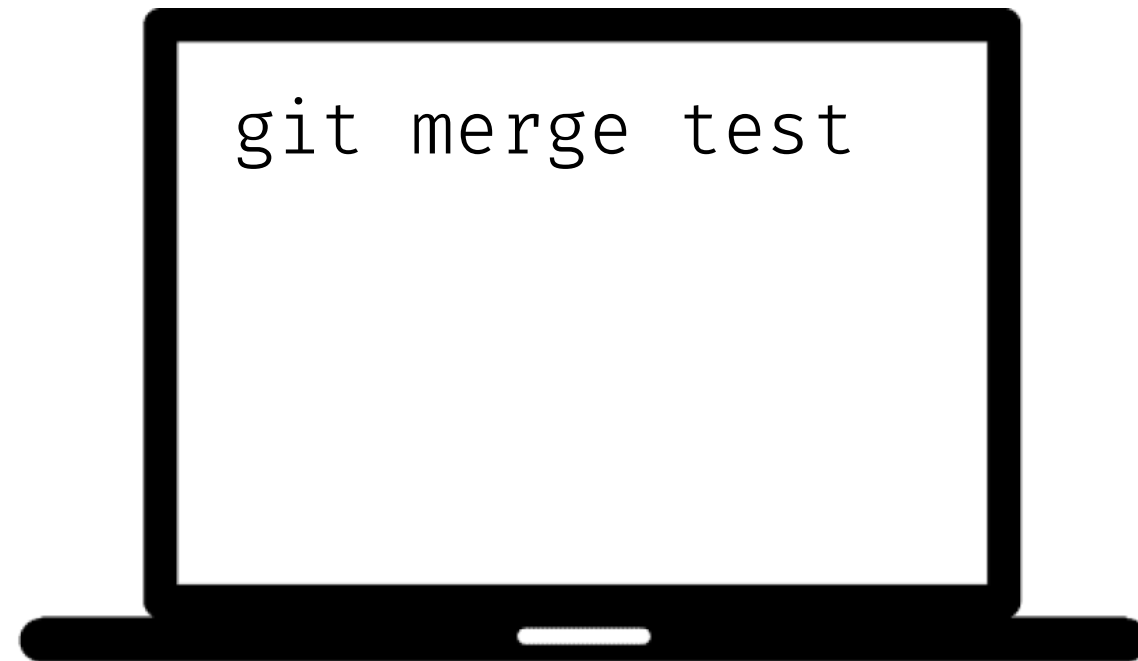
# Merge Conflict

- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.



# Merge Conflict

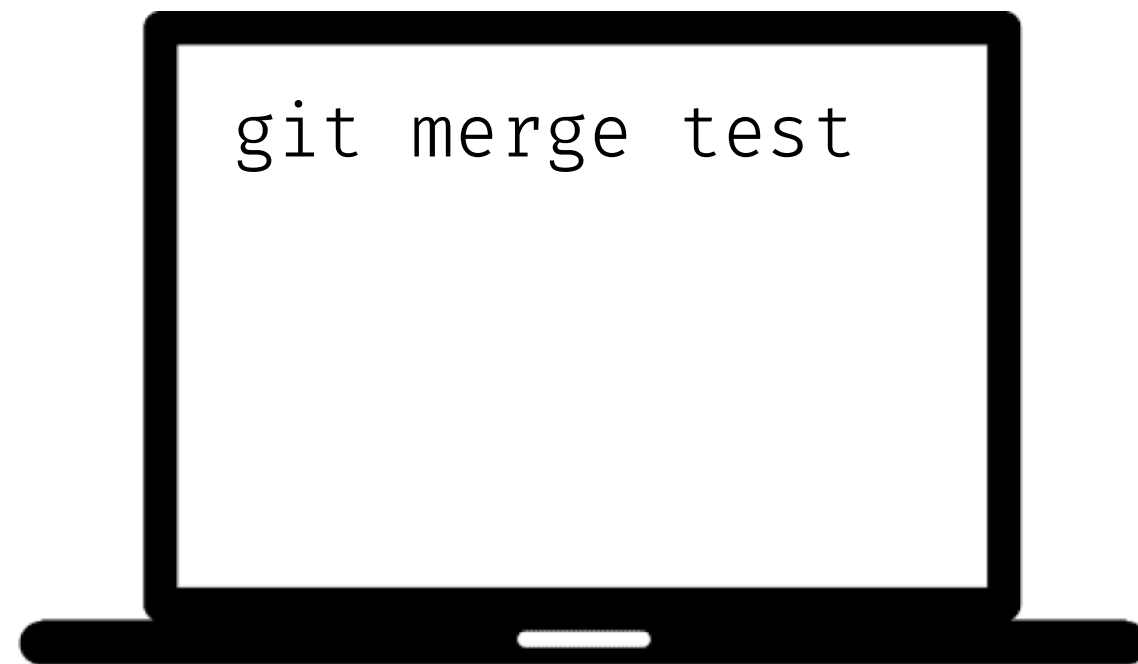
- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.



CONFLICT (content): Merge conflict in main.c  
Automatic merge failed; fix conflicts and then  
commit the results.

# Merge Conflict

- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.



```
int a = 1;  
<<<<<<< HEAD  
int b = 2;  
=====  
int b = 0;  
>>>>>>> 5468973206973204352e  
int c = 3;  
int d = 4;  
int e = 5;
```

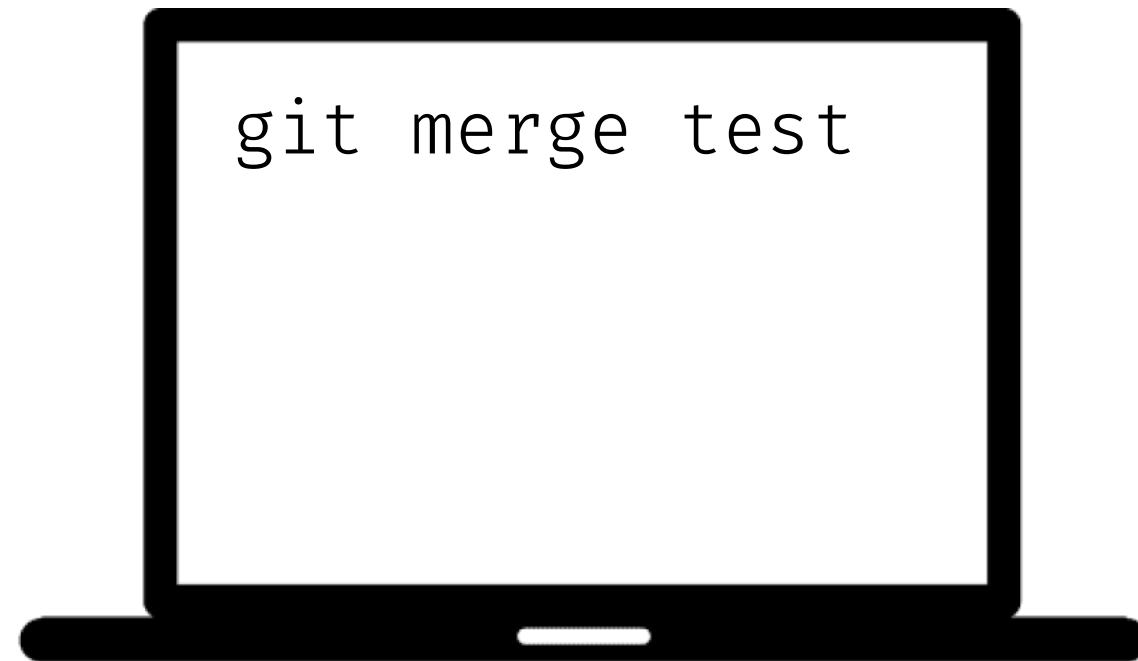


# Merge Conflict

- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.

Yaptığım değişiklik

Kaynaktaki değişiklik



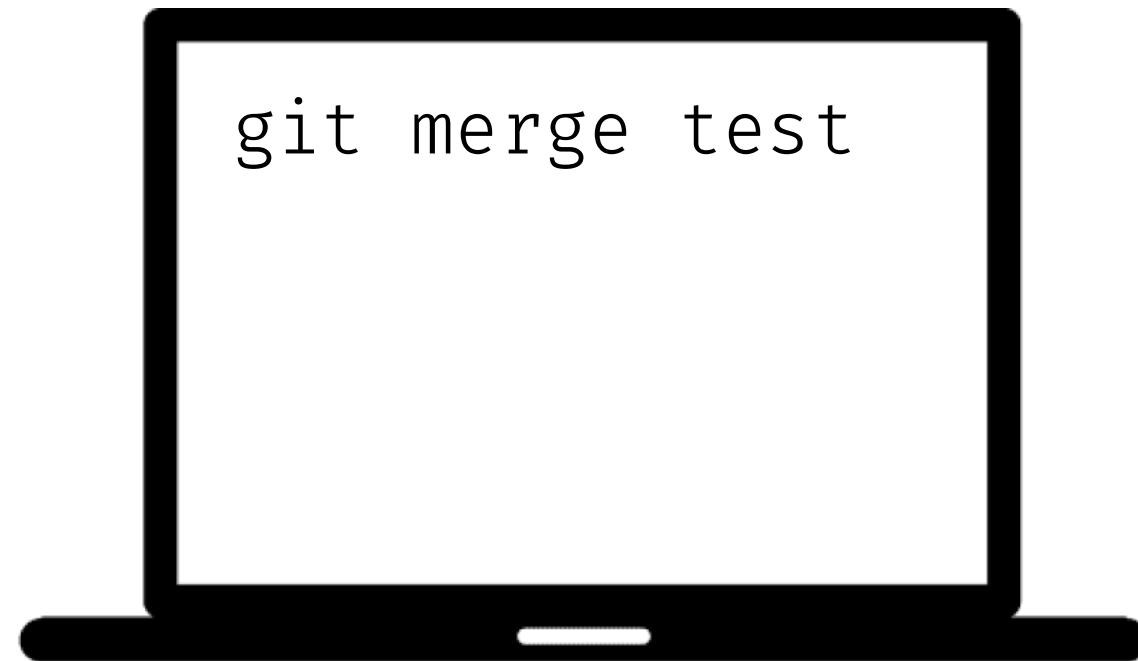
```
int a = 1;
<<<<<<< HEAD
int b = 2;
=====
int b = 0;
>>>>>>> 5468973206973204352e
int c = 3;
int d = 4;
int e = 5;
```

conflicting commit

↓

# Merge Conflict

- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.



```
int a = 1;
```

```
int b = 2;
```

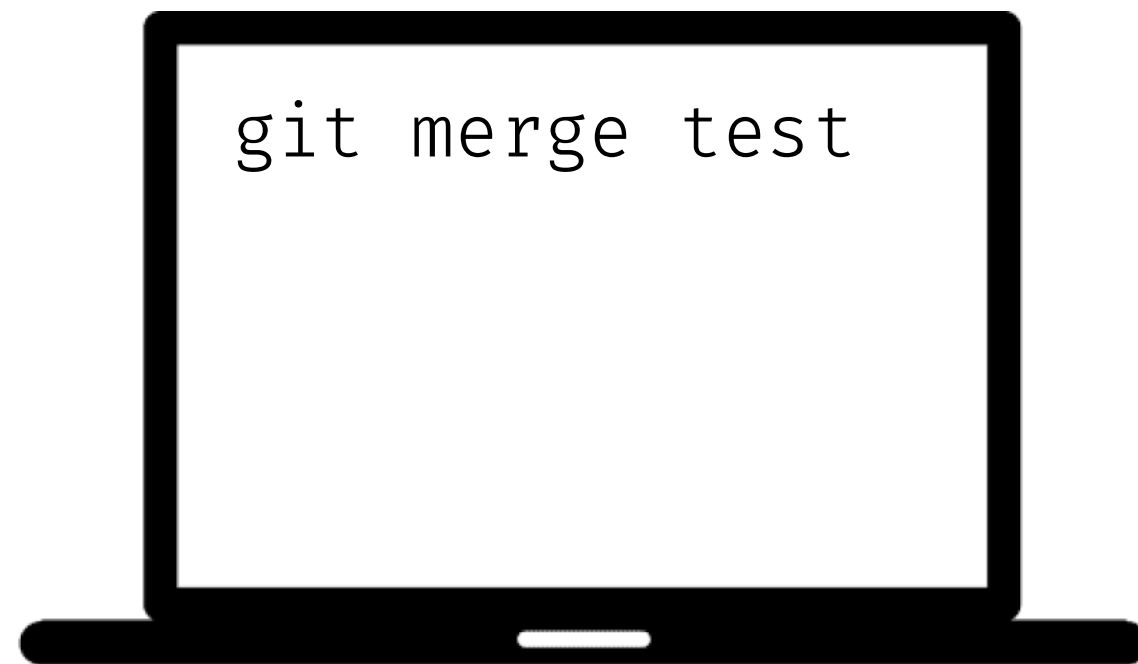
```
int c = 3;
```

```
int d = 4;
```

```
int e = 5;
```

# Merge Conflict

- İki farklı commit merge edilemez.
- Düzeltilmesi gerekir.



```
int a = 1;  
int b = 2;  
int c = 3;  
int d = 4;  
int e = 5;
```

# Pull Request

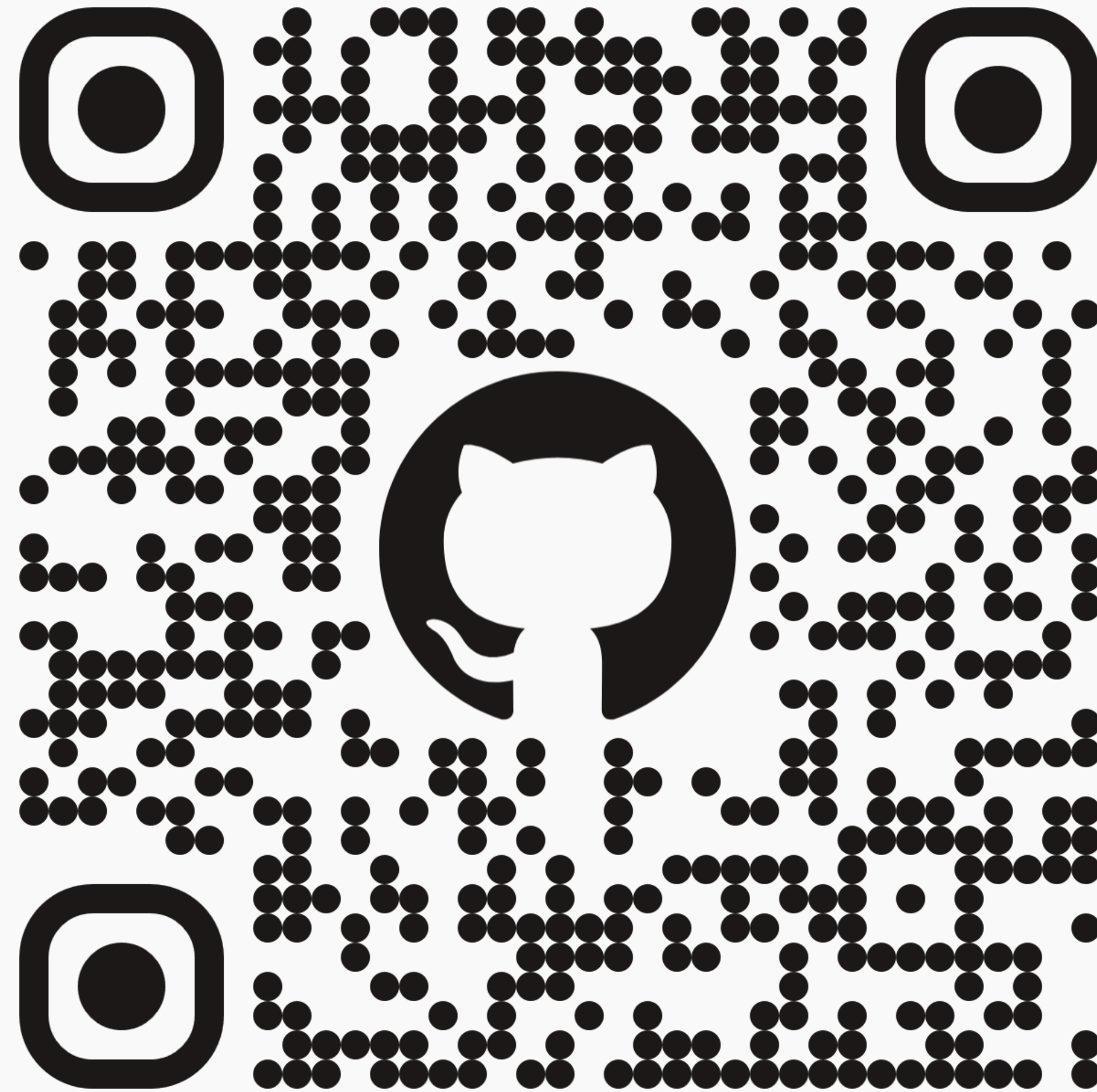


Git, yazılan kodu takip eden bir versiyon kontrol sistemidir.

- Projedeki değişiklikleri takip etmeyi kolaylaştırır.
- Kodu kişiler arasında senkonize eder.
- Kodun orijinalini kaybetmeden test yapabilme imkanı sunar.
- Projede yapılmış değişiklikleri geri almayı sağlar.



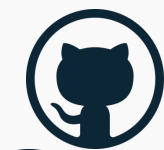






<https://github.com/Google-Developer-Student-Clubs-ATU>

# Git ve Github



CheesyFrappe – Emirhan Balcı



BurakAhmet – Ahmet Burak Biçer