



.git

► What's Version Control?



- Track changes on text files / source files for you
- Unlimited Undo / Redo
- Time Travel
- Collaborative development environment
- Compare and Blame
 - ◆ What changed
 - ◆ When it changed
 - ◆ Why it changed
 - ◆ Who changed it

CLARUSWAY®
WAY TO REINVENT YOURSELF



Version kontrol sistemi versiyonlardaki değişiklikleri kontrol eder. Biz Git kullanacağız. Git bu değişikliği fark eder, kaydetmek istiyor musun diye uyarır.

Bir dosya veya klasörün sınırsız şekilde versiyonlarına ulaşabiliyoruz. İstediğin zamana dönebiliriz.

Bir dosyaya, bir klasöre farklı farklı kişiler müdahale edebiliyor. Birden fazla kişinin aynı dosya üzerinde çalışmasını sağlar.

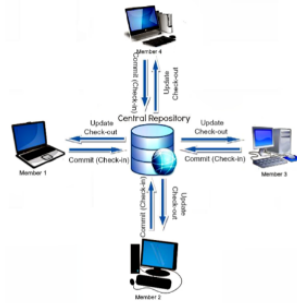
Versiyonu almak bizim verdiğimiz bir komutlar oluyor. Versiyonda ne değişikliği yapıldığını, ne zaman yapıldığını, neden yapıldığını ve kimin yaptığını görebiliyoruz.

► Version Control Systems



- **Centralized**

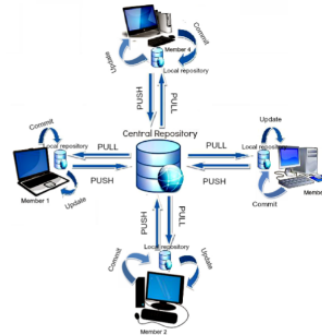
You need to be connected to the server



CLARUSWAY®
WAY TO REINVENT YOURSELF

- **Distributed**

You can work while offline



10

Centralized Server sisteminde sistemde herkes servera bağlanıyor ve bu server üzerinde çalışıyor. Proje üzerinde çalışmak isteyen herkes internete bağlı olmak zorunda.

Distributed Server sisteminde serverdan projeyi bilgisayara çekip offline olarak çalışıp sonra internet olduğunda sisteme yüklüyoruz. Öte yandan ana server çökse bile başka bilgisayarlarda indirilmiş en son versiyon elde olacaktır.

► What is Git?



- **Git** is a software
- Content Tracker
- Distributed Version Control System (VCS)
- Linus Torvalds



CLARUSWAY®
WAY TO REINVENT YOURSELF

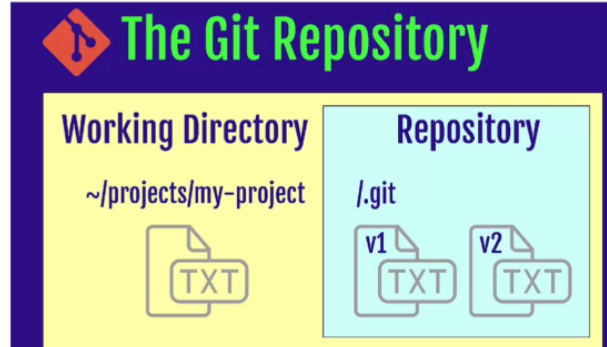
14

► Git Repository



What is a repository

- A directory or storage space where your projects can live.
- Local Repository
- Remote Repository

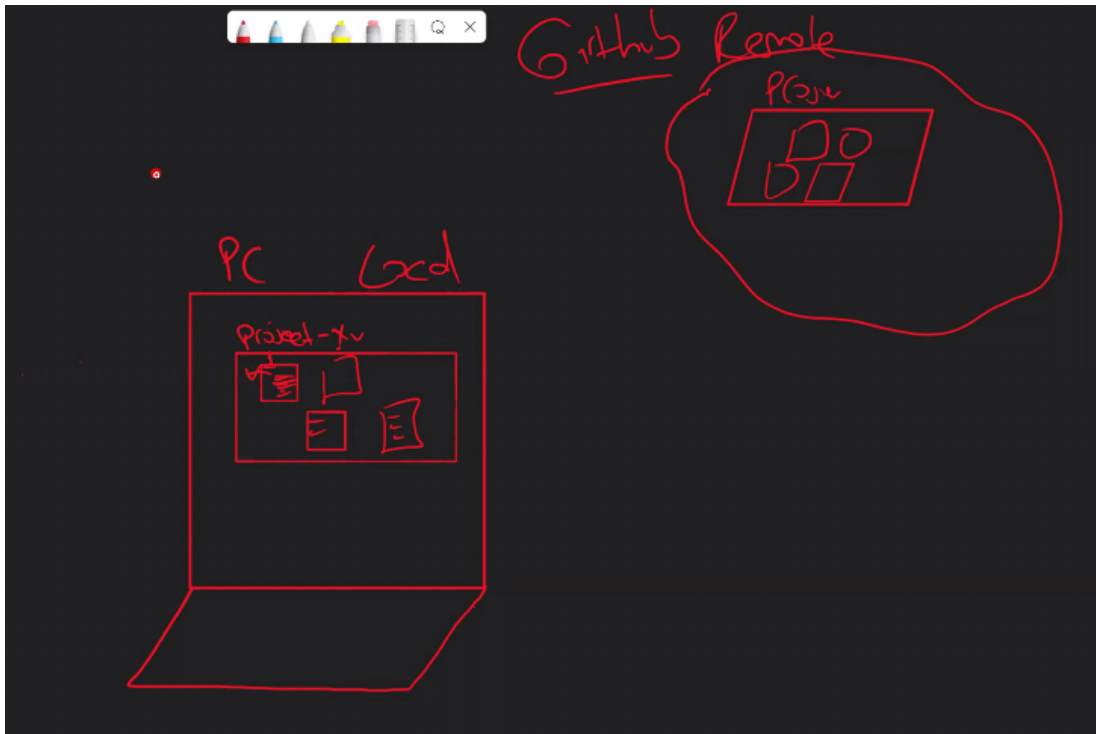


CLARUSWAY©
WAY TO REINVENT YOURSELF

15

Repository nedir? İki git reposu var.

Biri local diğeri remote:



Bash, linuxun terminali

Git Bash Git ile gelen bir terminal tool. Bu bir emilatör, windowsta sanki linux makineymiş gibi komutlarımızın çalıştırılmasını sağlıyor.

Git Bash te daha önce yazdığımız komutları görmek için üst tuşuna basacaz. Basınca komutlar sırasıyla geliyor.

Boşluk yerine alt çizgi kullanmayı alışkanlık haline getirmeliyiz.

Boş bir klasöre regular klasör diyoruz.

Dosyaları görmek için kullanacağımız komut ls dir.

Hangi dosyanın içinde olduğumuzu gösteren komut da pwd dir.

Klasörü değiştirmek için cd kullanıyoruz. (change directory, directory klasör)

```
a@DESKTOP-TPH8ITT MINGW64 ~  
$ pwd  
/c/Users/a  
  
a@DESKTOP-TPH8ITT MINGW64 ~  
$
```

Buradaki sonsuzluk işareti kullanıcı kısmında olduğumuzu simgeler.

Boş zamanlarınızda temel linux komutlarını çalışın.

“git help” komutuyla git le hangi komutları kullanabilirim onu gösterir:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson  
$ git help  
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]  
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]  
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]  
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]  
          [--super-prefix=<path>] [--config-env=<name>=<envvar>]  
          <command> [<args>]  
  
These are common Git commands used in various situations:  
  
start a working area (see also: git help tutorial)  
  clone Clone a repository into a new directory  
  init Create an empty Git repository or reinitialize an existing one  
  
work on the current change (see also: git help everyday)  
  add Add file contents to the index  
  mv Move or rename a file, a directory, or a symlink  
  restore Restore working tree files  
  rm Remove files from the working tree and from the index  
  
examine the history and state (see also: git help revisions)  
  bisect Use binary search to find the commit that introduced a bug  
  diff Show changes between commits, commit and working tree, etc  
  grep Print lines matching a pattern  
  log Show commit logs  
  show Show various types of objects  
  status Show the working tree status  
  
grow, mark and tweak your common history  
  branch List, create, or delete branches  
  commit Record changes to the repository  
  merge Join two or more development histories together  
  rebase Reapply commits on top of another base tip  
  reset Reset current HEAD to the specified state  
  switch Switch branches  
  tag Create, list, delete or verify a tag object signed with GPG  
  
collaborate (see also: git help workflows)  
  fetch Download objects and refs from another repository  
  pull Fetch from and integrate with another repository or a local branch  
  push Update remote refs along with associated objects  
  
'git help -a' and 'git help -g' list available subcommands and some  
concept guides. See 'git help <command>' or 'git help <concept>'  
to read about a specific subcommand or concept.  
See 'git help git' for an overview of the system.  
  
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson  
$ |
```

Linux komutlarında “mkdir” yeni klasör oluşturur.

Linux sisteminde noktayla başlayan dosyalar gizli dosyalardır. Ls komutu gizli dosyaları göstermez. Bunun için ls -a komutunu kullanmamız gerekir.

"git init" komutunu girdiğimizde .git şeklinde bir gizli klasör oluşturuldu. Bu bir local repo dur. Bu şekilde bir local repo oluşturduk. Bu birinci yöntemidir.

"git status" komutu

```
MINGW64/c:/Users/a/Desktop/git_lesson/birinci_ders
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson
$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson
$ ls
birinci_ders/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson
$ cd birinci_ders/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders
$ git init
Initialized empty Git repository in C:/Users/a/Desktop/git_lesson/birinci_ders/.git/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ ls -a
./ ../ .git/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

► Git Repository



→ to create a new remote repo and connect it with your local repo (after you create a remote repo on Github/Bitbucket etc.)

git clone address

Git de repo oluşturmanın ikinci yöntemi ise klonlamaktır. Klonlamakta daha yaygın olan bir yöntemdir. Github da oluşturulan remote repo bilgisayara kopyalanır.

Workflow

Working Directory

Where you work.
Create new files,
edit files delete
files etc.



Staging Area (Index)

Before taking a
snapshot, you're
taking the files to
a stage. Ready
files to be
committed.



Repository

Committed
snapshots of your
project will be
stored here with
a full version
history.



Bilgisayara baktığım gördüğüm alan yani içinde olduğumuz klasör working directory dir.

Amacımız working directory de yaptığımız çalışmaları depolamaktır.

Yaptığımız dosyaları direk repoya koyamıyoruz. Şöyle ki süpermarkete gittiğimizde alışveriş sepetine koyduğumuz sütler henüz bizim değildir. Ücrtini verene kadar bizim değildir. Ancak başkası da gelip almaz. Staging area işte o market sepetidir.

```
MINGW64/c:/Users/a/Desktop/git_lesson/birinci_ders
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders
$ git init
Initialized empty Git repository in C:/Users/a/Desktop/git_lesson/birinci_ders/.git/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ ls -la
./ ../ .git/
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ touch limonata.txt
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

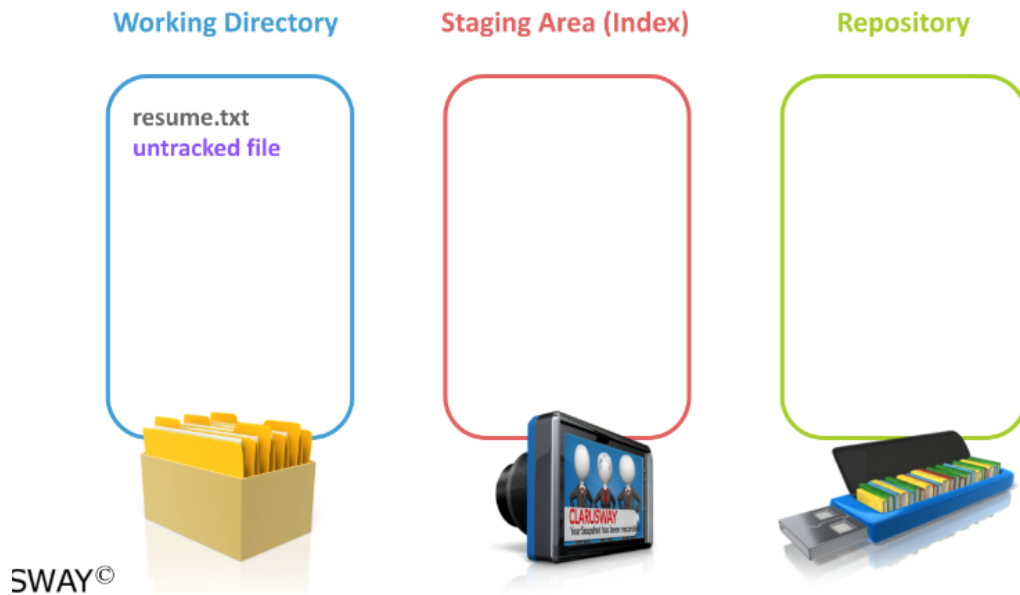
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    limonata.txt

nothing added to commit but untracked files present (use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

touch limonata.txt komutuyla yeni bir dosya oluşturduk.

Önceki git status komutunda bir şey görünmemişken dosya oluşturduktan sonra track edilmeyen bir dosya olduğunu belirtti.

Takip edilmeyen demek repoda versiyonlanmayan bir değişiklik yapılmış. Track etmek için git add komutunu kullan diyor.



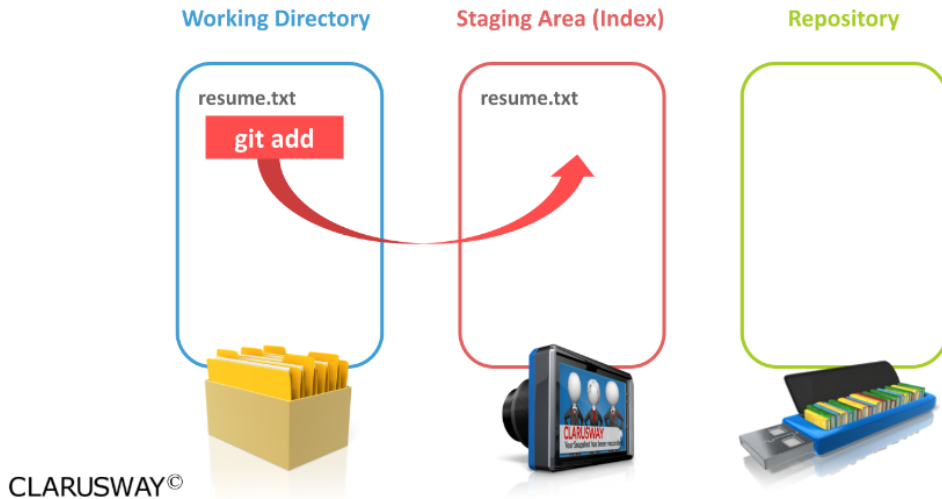
Biz bu dosyayı direk repoya atamayız. Önce staging area ya atmamız gerekiyor.

► Stage files options

- stage one file
git add filename
- stage all files (new, modified)
git add .
- stage modified and deleted files only
git add -u

"git add limonata.txt komutuyla bu dosyayı staging area ya attr:

► Track/stage a file



```
MINGW64:/c/Users/a/Desktop/git_lesson/birinci_ders
nothing to commit (create/copy files and use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ touch limonata.txt
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        limonata.txt

nothing added to commit but untracked files present (use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git add limonata.txt
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

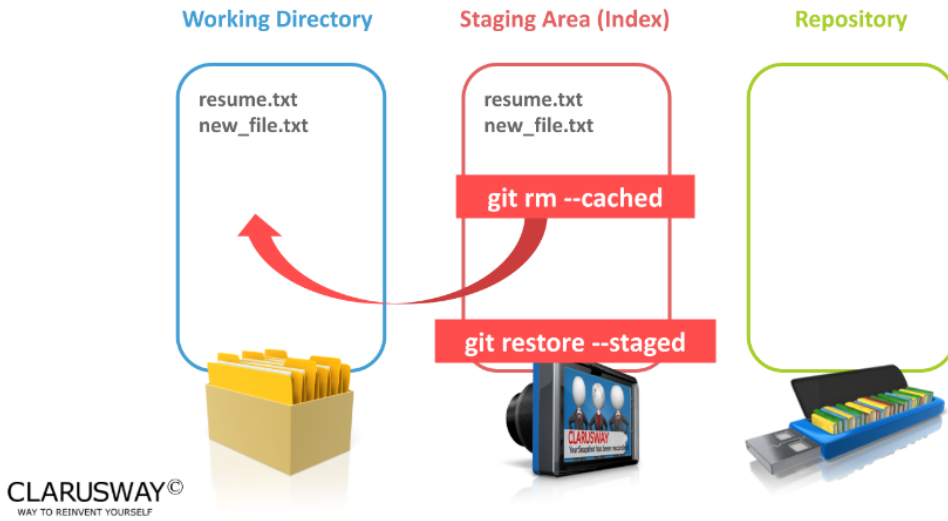
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   limonata.txt
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

bu kez "git status" komutunu girdiğimizde Staging area ya giren dosya yeşil renkle gösterilir.

Bize diyor ki commit etmek istemiyorsan bu dosyayı "git rm - -cached" komutuyla unstage edebilirsin.

Burada rm remove demek. Bu komutu girdiğimizde staging areadan çıkarır:

► Remove from stage



```
MINGW64:/c/Users/a/Desktop/git_lesson/birinci_ders

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git add limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git rm --cached limonata.txt
rm 'limonata.txt'

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    limonata.txt

nothing added to commit but untracked files present (use "git add" to track)

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

Burada ilk önce staging areadan çıkardık ve "git status" komutuyla kırmızı renkli olduğu için staging areadan çıkardığını gördük.

Bu kez git add . Komutuyla (bu komut bütün dosyaları staging area ya ekler) tekrar staging area ya ekledik.

Şimdi limonata dosyasını git commit -m "" komutuyla repo ya ekleyeceğiz. Tırnak içerisine neden commit ettiğimizi açıklayacak bir ifade yazıyoruz:

► Commit

- Commit the files on the stage

```
git commit -m "message"
```

- Add and commit all tracked files

```
git commit -am "message"
```

- amend commit message

```
git commit --amend
```

```
MINGW64~/c/Users/a/Desktop/git_lesson/birinci_ders
limonata.txt
nothing added to commit but untracked files present (use "git add" to track)
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git add .
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   limonata.txt

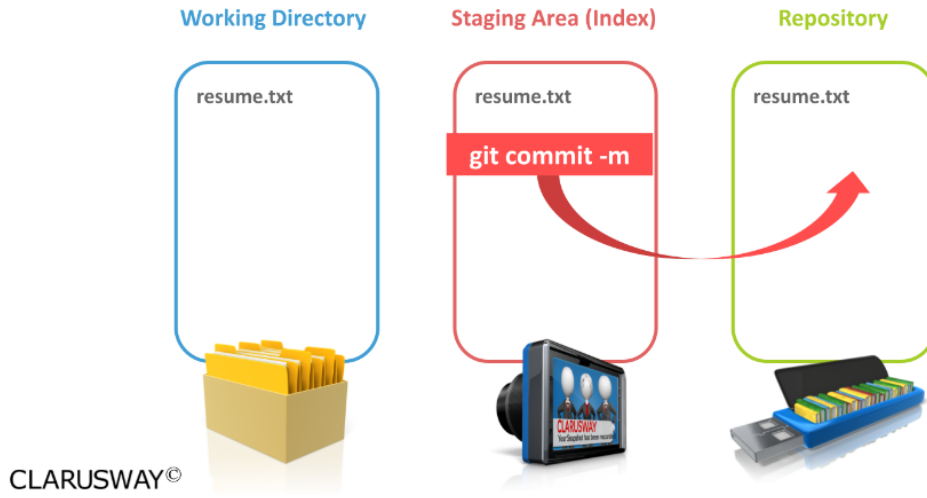
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git comit -m "limonata tarifi dosyasını oluşturdum"
git: 'comit' is not a git command. See 'git --help'.

The most similar command is
    commit

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git commit -m "limonata tarifi dosyasını oluşturdum"
[master (root-commit) 4ad879e] limonata tarifi dosyasını oluşturdum
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

Commit



"git log" komutuyla bu yaptığımız commit i görebiliyoruz:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git commit -m "limonata tarifi dosyasını oluşturdum"
[master (root-commit) 4ad879e] limonata tarifi dosyasını oluşturdum
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit 4ad879ee8294e2c87accb9b70d7cd6d055457957 (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

Şimdi "git status" komutunu girdiğimizde commit edilecek bir şey olmadığını söyleyecek. Çünkü biz dosyamızı repoya gönderdik:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git commit -m "limonata tarifi dosyasını oluşturdum"
[master (root-commit) 4ad879e] limonata tarifi dosyasını oluşturdum
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 limonata.txt


a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit 4ad879ee8294e2c87accb9b70d7cd6d055457957 (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
nothing to commit, working tree clean

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

Şimdi limonata dosyamızın içine girerek limonata için gerekli malzemeleri yazdık:

 limonata - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

limon
şeker
su

<

Dosyada bir değişiklik yaptıktan sonra git status yaptığımızda:

```

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit 4ad879ee8294e2c87accb9b70d7cd6d055457957 (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
nothing to commit, working tree clean

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   limonata.txt

no changes added to commit (use "git add" and/or "git commit -a")
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ |

```

modified limonata.txt uyarısıyla dosyada bir değişiklik olduğunu ve commit olmadığını görüyoruz.

"git add ." komutuyla bütün değişiklikleri staging areaya ekliyoruz ve "git status" komutuyla kontrol ediyoruz:

```

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   limonata.txt

no changes added to commit (use "git add" and/or "git commit -a")

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git add .

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ |

```

"git commit -m" komutuyla mesajımızı ekleyerek dosyayı commit eder ve repoya gönderdik. "git log" komutuyla da iki versiyonu görüyoruz.

```

$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   limonata.txt

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git commit -m "tarifi yazdım"
[master b7db8e7] tarifi yazdım
1 file changed, 3 insertions(+)

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
nothing to commit, working tree clean

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit b7db8e71fbb5248642743eab556c0772c7b70646 (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date:   Mon Oct 31 22:01:47 2022 +0100

    tarifi yazdım

commit 4ad879ee8294e2c87accb9b70d7cd6d055457957
Author: yemre <yemreaydeniz1@gmail.com>
Date:   Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$

```

Sonra tarifimi geliştirmek istiyorum ve nane ile buz u ekliyorum:

```

limonata - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
limon
şeker
su
nane
buz

```

<

Şimdi yaptığımız değişikliği önce staging areaya sonra repoya göndereceğiz:

Sırasıyla;

"git status" ile yeni bir düzenleme olduğunu gördük.

"git add ." Ve "git commit - m" komutlarını yazıyoruz:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   limonata.txt

no changes added to commit (use "git add" and/or "git commit -a")

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git add .

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git commit -m "tarifimi geliřtirdim"
[master 9adec28] tarifimi geliřtirdim
1 file changed, 2 insertions(+)

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit 9adec2875cb700d7561d0bffb63458e16b73c8cc (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date:   Mon Oct 31 22:06:26 2022 +0100

    tarifimi geliřtirdim

commit b7db8e71fbb5248642743eab556c0772c7b70646
Author: yemre <yemreaydeniz1@gmail.com>
Date:   Mon Oct 31 22:01:47 2022 +0100

    tarifi yazdım

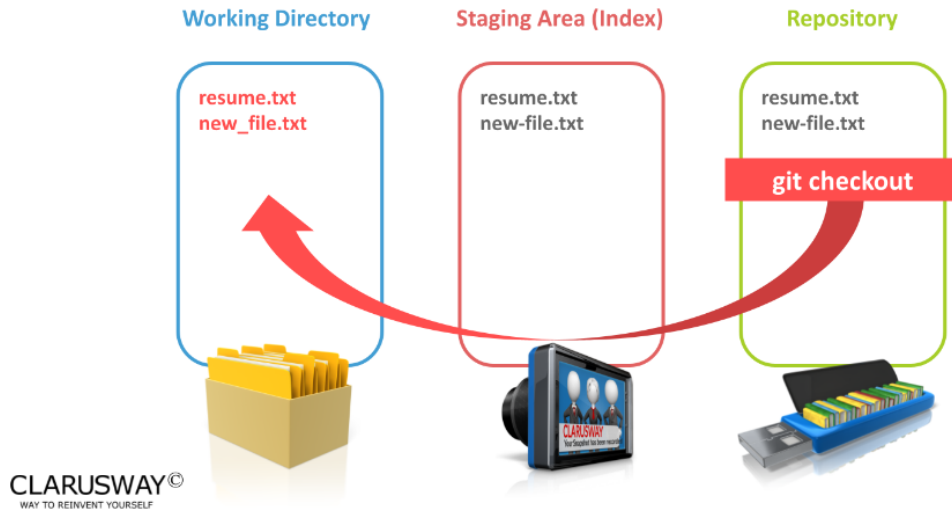
commit 4ad879ee8294e2c87accb9b70d7cd6d055457957
Author: yemre <yemreaydeniz1@gmail.com>
Date:   Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluřturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$
```

Versiyonlar arasında checkout komutuyla dolařıyoruz:

► Checkout from Repo



Şimdi checkout komutuyla daha önceki bir versiyona gideceğiz:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git checkout b7db8e71fbb5248642743eab556c0772c7b70646
Note: switching to 'b7db8e71fbb5248642743eab556c0772c7b70646'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at b7db8e7 tarifi yazdım

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders ((b7db8e7...))
$ git log
commit b7db8e71fbb5248642743eab556c0772c7b70646 (HEAD)
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 22:01:47 2022 +0100

    tarifi yazdım

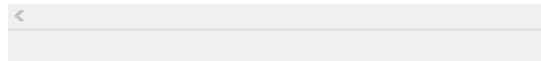
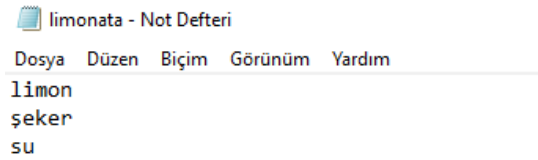
commit 4ad879ee8294e2c87accb9b70d7cd6d055457957
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders ((b7db8e7...))
$
```

"checkout" komutuyla birlikte gitmek istediğimiz versiyonunda commit in yanında yazılı id yi yazdık ve o versiyona gittik. Sonra da "git log" komutuyla hangi versiyonda olduğumuzu tekrar gördük.

Şimdi limonata dosyamıza tekrar bakalım:



İkinci versiyona gittiğimiz için nane ve buz un silindiğini gördük.

"git checkout master" komutuyla da üçüncü yani en son versiyona geri dönebiliriz:

Ardından "git log" komutuyla da geri döndüğümüzü görelim:

```
a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders ((b7db8e7...))
$ git checkout master
Previous HEAD position was b7db8e7 tarifi yazdım
Switched to branch 'master'

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ git log
commit 9adec2875cb700d7561d0bffb63458e16b73c8cc (HEAD -> master)
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 22:06:26 2022 +0100

    tarifimi geliştirdim

commit b7db8e71fbb5248642743eab556c0772c7b70646
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 22:01:47 2022 +0100

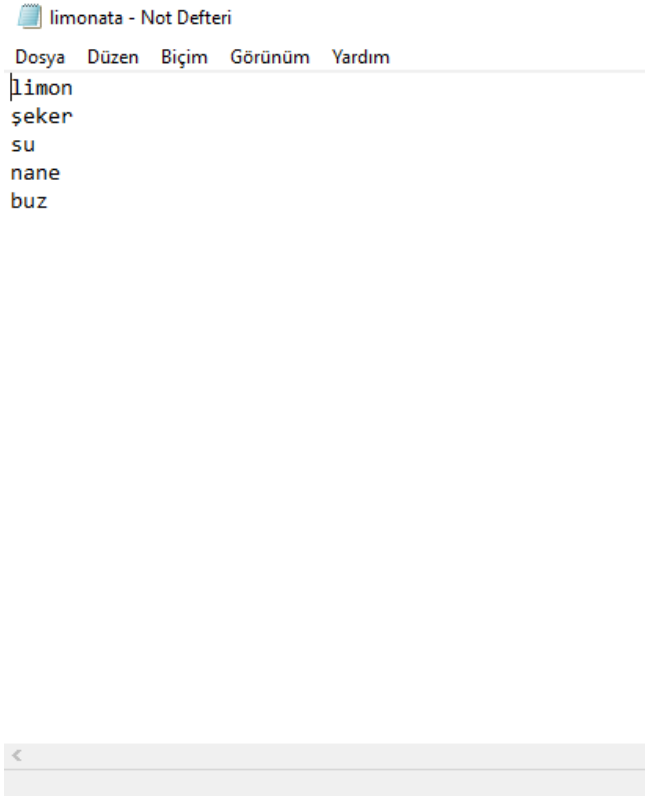
    tarifi yazdım

commit 4ad879ee8294e2c87accb9b70d7cd6d055457957
Author: yemre <yemreaydeniz1@gmail.com>
Date: Mon Oct 31 21:49:08 2022 +0100

    limonata tarifi dosyasını oluşturdum

a@DESKTOP-TPH8ITT MINGW64 ~/Desktop/git_lesson/birinci_ders (master)
$ |
```

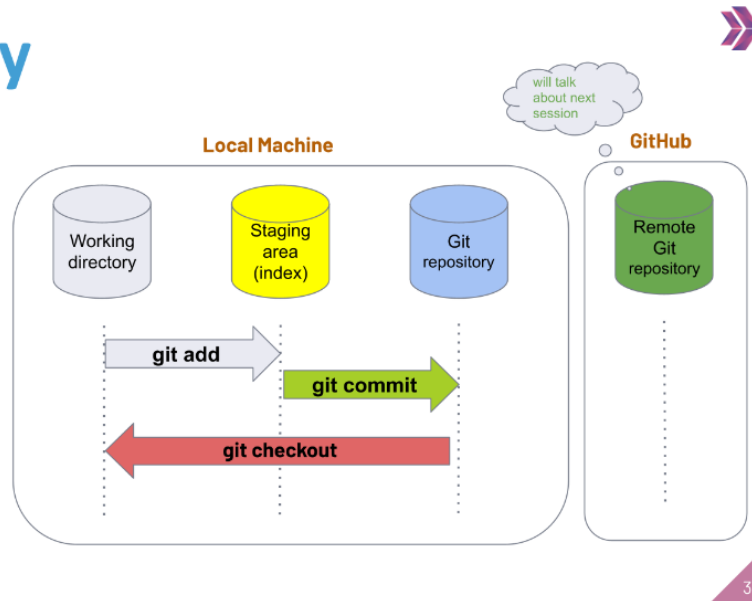
Limonata dosyamızı açtığımızda en son versiyona geri döndüğümüzü yani nane ve buzun tekrar eklendiğini orada da görüyoruz:



Özetle:

► Summary

git init
git status
git add .
git commit -m "abc"
git log
git checkout



CLARUSWAY®
WAY TO REINVENT YOURSELF

39

.git klasörünü silersek bu klasör Git tarafından kontrol edilmez.

Ya da "ls -a" ve rm -rf .git" komutuyla terminalde silebiliriz.

Bir git klasörünün altında başka bir git klasörü olmaması lazım. Hata verir.

Git Bash kapandığında komutlar silinir. "history" komutuyla daha önce girdiğimiz komutları görebiliriz.



Github repodan tüm repoyu değilde reponun içinden bir klasörü/dosyayı indirmek için aşağıdaki komutlar gireriz.

```
sudo yum install git -y
mkdir repo && cd repo
git init
git remote add origin <origin-url>
git config core.sparseCheckout true
echo "subdirectory/under/repo/" >> .git/info/sparse-checkout # do not put the repository folder name in the beginning
git pull origin <branch-name>
```

GitHub repoyu dosyaya bağlama

```
git init
git add README.md
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/<your github name>/mygithubrepo.git
git push -u origin main
```