WHAT IS GIT?

It helps developers track changes in their code base.

Version control git tracs changes in your project source code,allowing you to review and revert.

Every developer working on a project has a complete copy of the code on their computer.

Developers can create branches to work on bug fixes independently without affecting the main code base.

Multiple people can work on the same project and git helps and merge their changes.

Git is a open source software this means that the source code is available and developers can freely contibute.

HOW TO CONFİG GİT?

```
$ git config --global user.name "funda"
```

```
$ git config --global user.email "fundaaugurlu@gmail.com"
```

```
$ git config --list
```

```
user.name=funda
```

```
user.email=fundaaugurlu@gmail.com
```

HOW TO PUSH A FİLE TO GİTHUB USEİNG GİT ?

Click to the file and open the terminator and follow these comands.

Create a empty repostory in github.

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git init
```

```
Reinitialized existing Git repository in
C:/Users/ASUS/OneDrive/Masaüstü/Python/.git/

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git add .

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git commit -m "yeter"

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git branch -M main

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git push -u origin mai
```

GIT BRANCH AND MERGİNG

// it shows which branch we are in

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git branch
```

// we are in the main branch

```
* main
```

//We created a new branch

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git branch user-backend

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git branch
```

//Now we have 2 branches.

```
* main

  user-backend

PS C:\Users\ASUS\OneDrive\Masaüstü\Python>
```

//We are in the main branch (Because of the symbol * and its green.).If we want to switch to the other branch we use this:

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git checkout user-backend
Switched to branch 'user-backend'
```

//We can combine these two branches.

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git merge main user-backend
```

//the user-backend branch, the main branch, and the origin/main branch all point to the same commit. So, the main branch and the user-backend branch are currently in the same state

```
commit 2b81a0c6c8fb46c5a654e6482b213435724afc80 (HEAD -> user-backend, origin/main, main)
```

//We can spot diffrances between these branches.

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git diff
```

//If you want to delete the branch you should checkout first.Then delete it.

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git branch -d user-backend
Deleted branch user-backend (was 2b81a0c).
```

// We use the branch that we created to send it to a remote repository

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git push origin main
Everything up-to-date
```

//We can use Git Status to check for changes in project files

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git add .
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git commit -m "Yeni klasör eklendi"

[main da1f162] Yeni klasör eklendi

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 "Yeni klas\303\266r/aa.txt"

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git status

On branch main

Your branch is ahead of 'origin/main' by 1 commit

//We can see the commits with git log

PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git log

commit da1f1621960feedc22ebaeeba1b7c2f8923c714c (HEAD -> main, origin/main)

Author: funda <fundaaugurlu@gmail.com>

Date:   Tue Jul 16 18:18:37 2024 +0300

 Yeni klasör eklendi

commit 2b81a0c6c8fb46c5a654e6482b213435724afc80

Author: funda <fundaaugurlu@gmail.com>

Date:   Tue Jul 16 16:48:08 2024 +0300
```

UNDO THE   COMMİT CHANGE WITH GIT REVERT

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python>  git revert 2b81a0c6c8fb46c5a654e6482b213435724afc80

On branch main

Your branch is ahead of 'origin/main' by 1 commit.

  (use "git push" to publish your local commits)
```

UNDO THE COMMIT CHANGE WITH GIT RESET

```
PS C:\Users\ASUS\OneDrive\Masaüstü\Python> git reset --hard f472d8e87e80ab6385bc56f32a2c0f181933f143
HEAD is now at f472d8e Revert "yeter"
```

WHATS THE DİFFRANCE BETWEEN GIT RESET AND GIT REVERT?

Git reset: Changes the history, deletes local changes.

Git Revert: It protects the past, performs retrieval.

OTHER GIT COMMANDS

1-GIT INIT

It creats a new blank repository or to make an existing project as a git project.

It also starts a head pointer for the master branch of the repository.

2-GIT LOG

We can use this to learn more details about our commits.

3-GIT BLAME FİLE NAME

This one tells us that who made what changes in the code.

4-GİT CLONE url

Creates a copy of an existing remote repository on your local computer.

5-GIT PUSH ORIGIN BRANCH

Sends your local changes to the specified branch on the remote repo.

6-GIT PULL

Gets changes from the specified remote branch and merges them into your current branch.