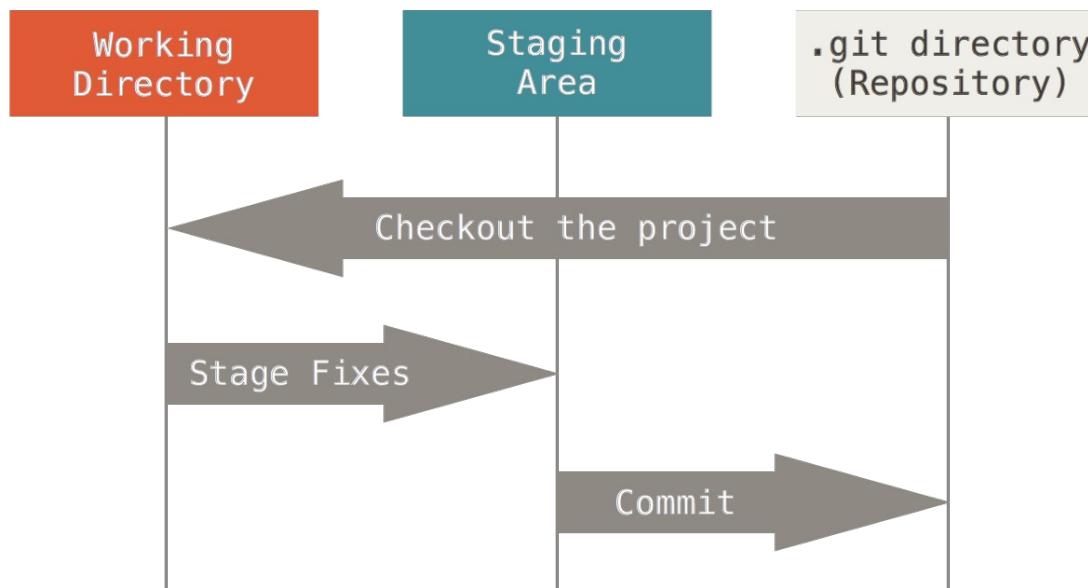




Git이란?



Git이란?

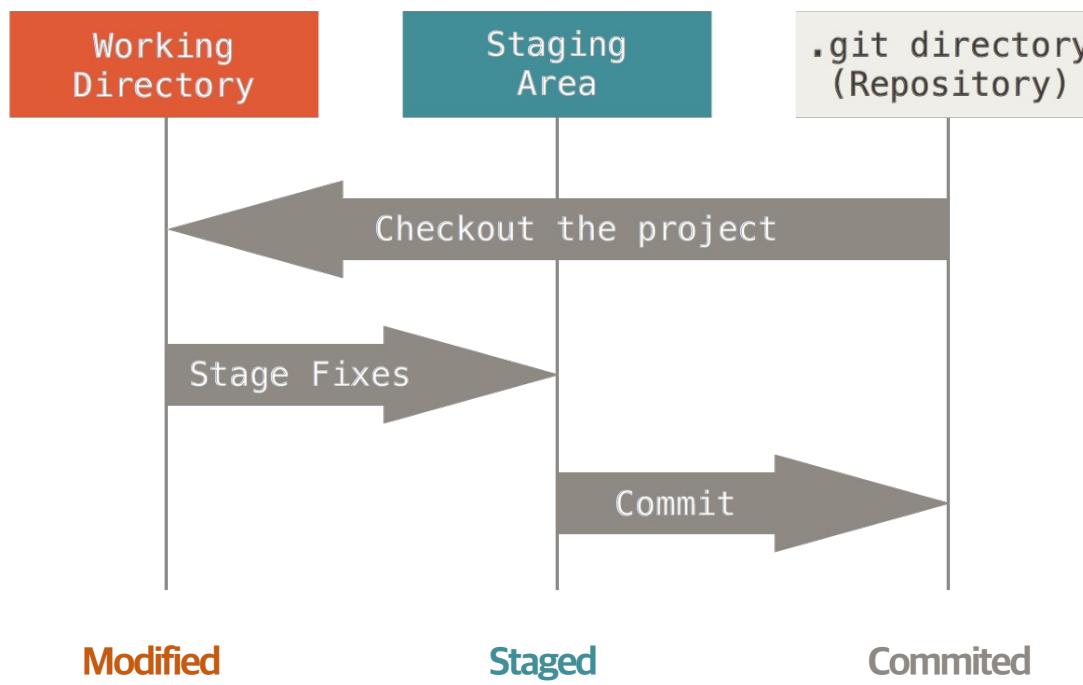
Git은 분산 버전 관리 시스템이다.

버전을 관리한다?

특정한 코드가 있다고 가정했을 때,
이를 수정하면서 해당 코드를 develop하게 된다.
이렇게 develop된 코드를 버전 2라고 할 수 있다.

이렇게, 각 버전을 관리하는 것이 분산 버전 관리 시스템인 Git이다.

Git이란?



Git에서의 3가지 상태

- Modified

Modified는 수정된 이라는 뜻을 가지고 있다.
즉, 코드를 수정하거나, 추가했는데 아직 git에 commit 하지 않은 상태이다.

- Staged

Modified된 코드를 commit할 것이라 예고하는 상태
(Git으로 버전 관리를 당하기 전에 들리는 임시 휴게소와 같은 개념)

- Committed

Local git database에 안전하게 버전이 저장되었다는 상태
(단, 만약 GitHub를 통한 Remote를 이용할 경우 그거와는 별개)
이렇게 Local git database에 commit된 버전을 Remote(GitHub, Gitlab...)에 push를 해줘야 한다.

Git0|란?

```
> nlp
⚡ __init__.py
⚡ config.py
⚡ data.py
⚡ function_class.py
⚡ function.py
⚡ gpu.py
⚡ model.py          M
⚡ optimizer.py
⚡ scheduler.py      U
📅 test.ipynb
≡ test.pkl
⚡ trainer.py         M
⚡ utils.py
```

실제 상황에서의 Git의 3단계

- Modified

작업 디렉토리에서 파일을 수정

Git이란?

✓	스테이징된 변경 사항	1
✚	model.py neuralflow	M
✓	변경 사항	5
✚	scheduler.py neuralflow	U
✚	trainer.py neuralflow	M
➕	cnn.ipynb test/CNN	M
➕	neural_network.ipynb test/neural_network	M
➕	lstm.ipynb test/RNN	M

실제 상황에서의 Git의 3단계

- Modified

작업 디렉토리에서 파일을 수정

- Staged

수정할 파일을 stage해서 commit할 스냅샷을 만듬
(Git으로 버전 관리를 당하기 전에 들리는 임시 휴게소와 같은 개념)

Git0|란?

```
commit 94e469c04b343afe04e9f63f20e0e31b59d12fc4 (origin/main, origin/HEAD)
Author: augustinLib <2bambitious@gmail.com>
Date:   Sun Jan 22 04:31:48 2023 +0900

    Docs: Fix README.md
```

```
commit 6e0523b1cf45e43b620838b21f1643f4792a9d01
Author: augustinLib <2bambitious@gmail.com>
Date:   Sun Jan 22 03:47:25 2023 +0900

    Docs: Update README.md
```

```
commit 53a07e6f926cafeb61b9562bbfb7f84a4b55b64
Author: augustinLib <2bambitious@gmail.com>
Date:   Sun Jan 22 03:29:08 2023 +0900
:■
```

실제 상황에서의 Git의 3단계

- Modified

작업 디렉토리에서 파일을 수정

- Staged

수정할 파일을 stage해서 commit할 스냅샷을 만듬
(Git으로 버전 관리를 당하기 전에 들리는 임시 휴게소와 같은 개념)

- Committed

Staging area에 있는 파일들을 커밋하여 git directory에 저장
(각 commit 값을 이용하여 이전 버전으로 돌아갈 수도 있다.)

Github란?



Github란?

Git 저장소이다.

정확한 비유는 아니지만, git을 사용하는 구글 드라이브의 느낌이다.
이를 이용하여 백업 비슷한 역할도 기대할 수 있고, 협업이 가능해진다.

GitHub를 통해 오픈 소스 소프트웨어들이 크게 발전할 수 있으며,
본인이 진행했던 프로젝트들을 포트폴리오의 형식으로 만들 수 있다.

(그리고 과제할 때 매우 유용하다)



GitHub의 요소

The screenshot shows a GitHub profile for a user named Sangam Lee (augustinLib). The profile includes a large circular profile picture of a person with glasses and a hoodie, a bio section with interests in NLP and IR, contact links (Gmail, Tistory, LinkedIn, Portfolio), a skill stack section with Python, R, C++, HTML5, CSS, JavaScript, TypeScript, Sass, TensorFlow, PyTorch, PyTorch Lightning, scikit learn, pandas, NumPy, weights and biases, Plotly, Selenium, Dart, Go, Rust, Node.js, React, and Flutter, and an environment section with macOS, Ubuntu, VSCode, PyCharm, Xcode, RStudio, Vim, Anaconda, Conda Forge, Git, Docker, GoLand, WebStorm, Slack, and Notion.

Projects 62 · Packages · Stars 62

WHO AM I?

I'm interested in Natural Language Processing(NLP) and Information Retrieval(IR) especially, interested in Retrieval-Augmented(RA) Language Model and Parameter-Efficient Fine-Tuning(PEFT)

if you want to contact me, see below and click button

Gmail Tistory LinkedIn Portfolio

STACKS

Python R C++ HTML5 CSS JavaScript TypeScript Sass TensorFlow PyTorch PyTorch Lightning scikit learn pandas NumPy weights and biases Plotly Selenium Dart Go Rust Node.js React Flutter

ENVIRONMENT

macOS Ubuntu VSCode PyCharm Xcode RStudio Vim Anaconda Conda Forge Git Docker GoLand WebStorm Slack Notion

Overview

본인(혹은 타인)의 프로필을 전반적으로 볼 수 있다.



GitHub의 요소

The screenshot shows a GitHub profile page for a user named Sangam Lee. At the top, there's a search bar labeled "Find a repository...", followed by filters for "Type", "Language", "Sort", and a "New" button. Below the search bar, the user's profile picture is displayed, along with their name "Sangam Lee" and email "augustinLib".

The main content area lists several repositories:

- augustinLib** [Public] - Config files for my GitHub profile. Tags: config, github-config. Updated 2 hours ago.
- transformers** [Public] - Forked from huggingface/transformers. Description: Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX. Tags: Python, Apache License 2.0. Updated 7 hours ago.
- Rust-Explore** [Public] - Description: 근본을 찾아 떠나는 RUST 독학. Tags: Rust. Updated 2 days ago.
- AI-Archive** [Public] - Description: AI를 공부하고자 하는 사람들이 더 편하고 더 쉽게 공부할 수 있는 환경을 만들기 위한 프로젝트입니다. Tags: MIT License. Updated 2 days ago.
- LLM-Train-Template** [Public] - Description: Repository that contains LLM train template that makes LLM training easier. Tags: Python, MIT License. Updated 2 weeks ago.

At the bottom left, there are sections for "Achievements" (with four circular icons) and "Send feedback" (Beta). On the right side, there are two "Starred" repositories with green waveform activity indicators.

Overview

본인(혹은 타인)의 프로필을 전반적으로 볼 수 있다.

Repository

본인(혹은 타인)의 레포지토리 목록



GitHub의 요소

The screenshot shows a GitHub profile for a user named Sangam Lee. The top navigation bar includes 'Overview', 'Repositories 22', 'Projects' (which is the active tab), 'Packages', and 'Stars 39'. The 'Projects' section features a 'Welcome to the all-new projects' message and a 'Project' board with cards like 'Prototype', 'Game brief and go-no', and 'Initial concept art'. Below the board is a search bar with 'is:open' and a 'New project' button. A list of open repositories follows, with 'NLP/추천시스템 컨퍼런스' being the first, marked as private and updated 3 days ago. The sidebar on the left contains sections for 'Achievements' (with several icons) and 'Highlights' (Developer Program Member, PRO). At the bottom, there's an 'Organizations' section with a 'B' icon.

Overview

본인(혹은 타인)의 프로필을 전반적으로 볼 수 있다.

Repository

본인(혹은 타인)의 레포지토리 목록

Project

본인(혹은 타인)의 GitHub project

GitHub project란, Asana와 비슷한 느낌인데, 특정 프로젝트를 위해 Task 관리를 수행함에 있어서 유용하게 사용되는 기능이다.



GitHub의 요소

The screenshot shows a GitHub profile for a user named Sangam Lee. The top navigation bar includes 'Overview', 'Repositories 22', 'Projects' (which is the active tab), 'Packages', and 'Stars 39'. The 'Projects' section features a 'Welcome to the all-new projects' message and a 'Project' board with cards like 'Prototype', 'Game brief and go-no', and 'Initial concept art'. Below the board is a search bar with 'is:open' and a 'New project' button. A list of open repositories follows, with 'NLP/추천시스템 컨퍼런스' being the first, marked as private and updated 3 days ago. The sidebar on the left contains sections for 'Achievements' (with several icons) and 'Highlights' (Developer Program Member, PRO). At the bottom, there's an 'Organizations' section with a logo.

Overview

본인(혹은 타인)의 프로필을 전반적으로 볼 수 있다.

Repository

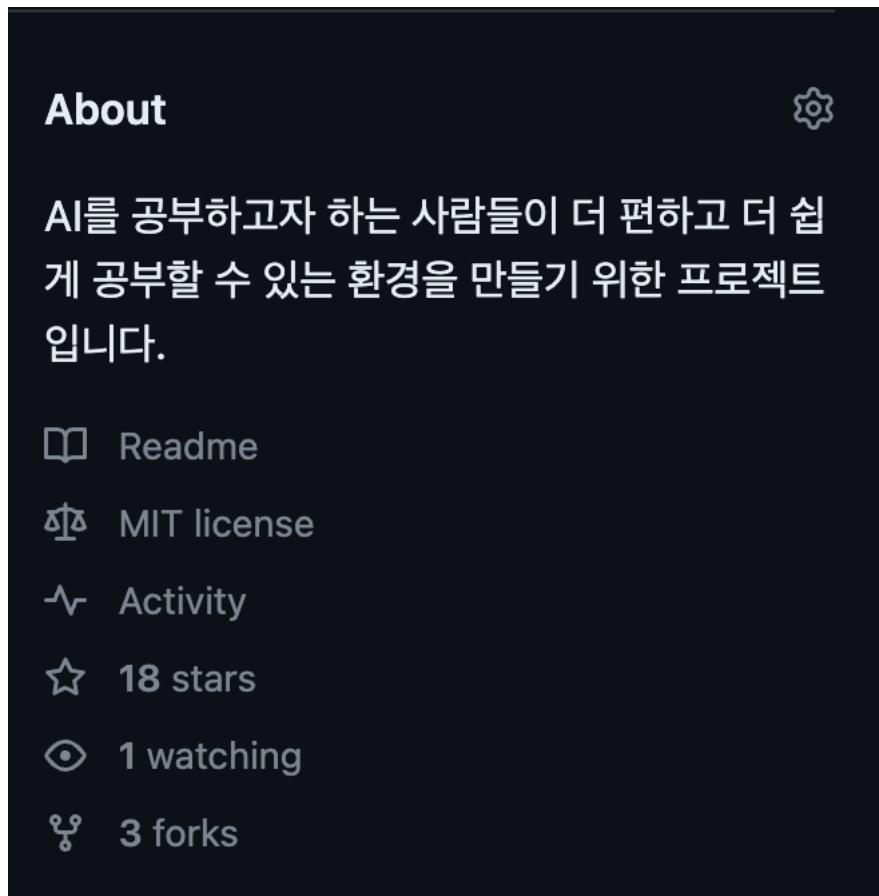
본인(혹은 타인)의 레포지토리 목록

Project

본인(혹은 타인)의 GitHub project

GitHub project란, Asana와 비슷한 느낌인데, 특정 프로젝트를 위해 Task 관리를 수행함에 있어서 유용하게 사용되는 기능이다.

GitHub의 요소



Star

유튜브의 좋아요 기능과 비슷한 개념이다
말 그대로 이 레포가 좋아요! 라고 하는 행위

watching

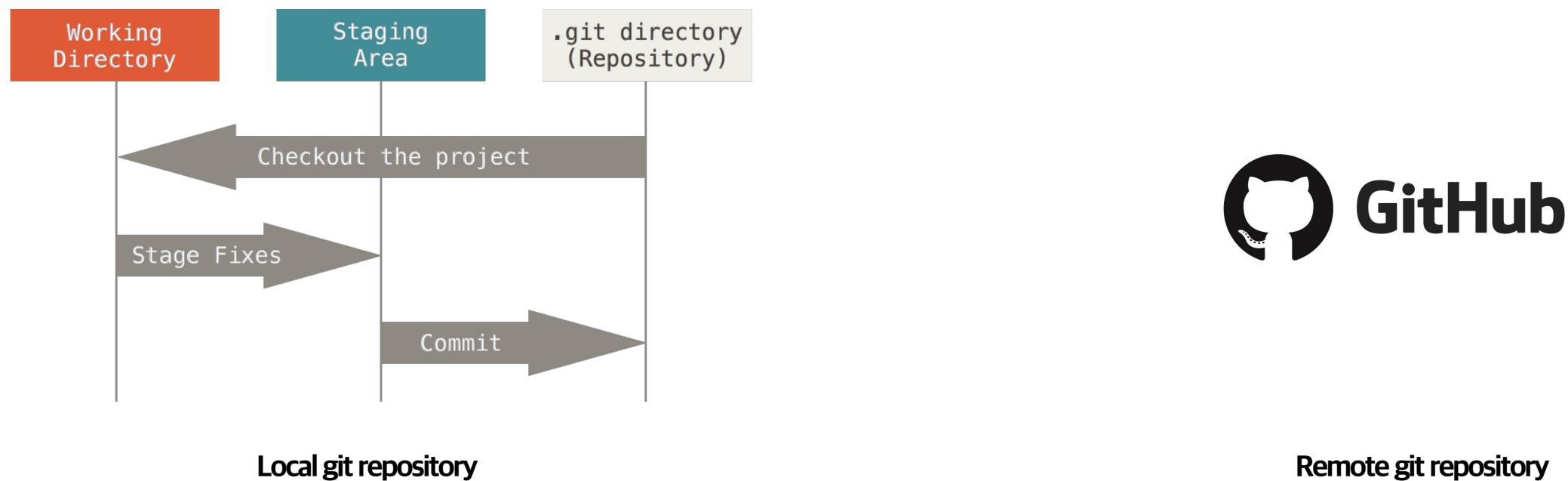
유튜브의 구독 기능이랑 비슷하다.
Watching을 설정해놓은 레포는 레포에서 진행되어지는
변경사항을 받아볼 수 있다.

Fork

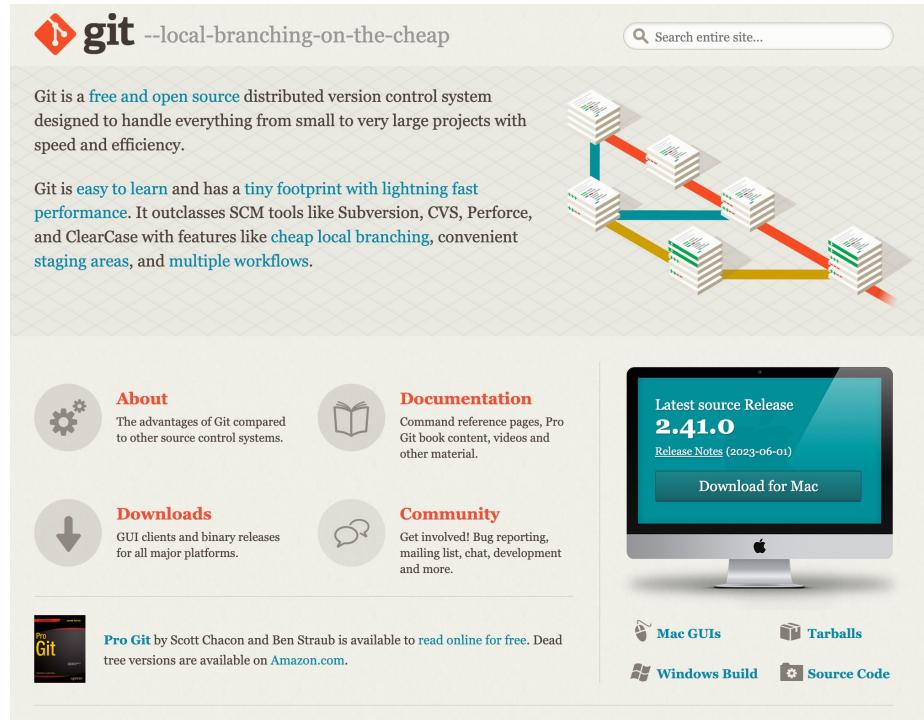
해당 레포를 내 레포에 그대로 복사해온다.
단순 git clone과 다른 점은, fork를 하면 본인의 수정본을
원본 레포에 pull request할 수 있다.

pull request : 나의 수정 사항을 원본 레포에도 수정 요청하는 것
(내가 이 코드 이렇게 고쳤으니깐 이거 반영해주세요!)

Local Git과 GitHub



Git 설치



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

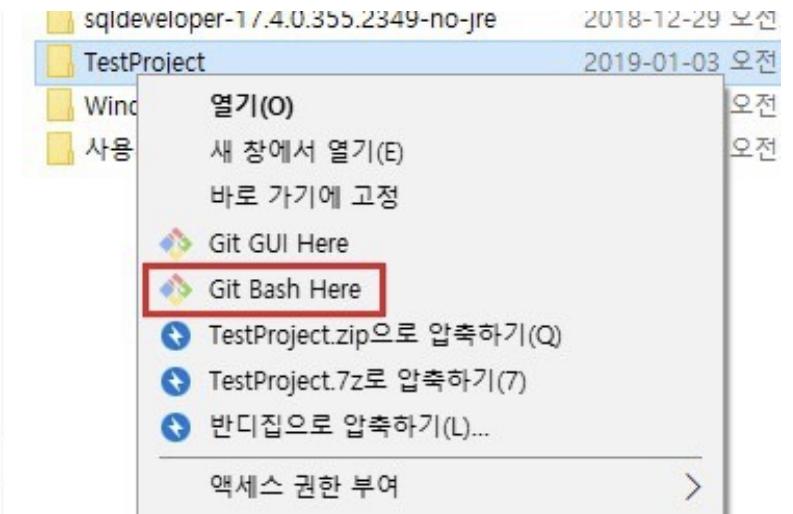
터미널 키기

Mac/Linux

```
augustin@isangamcBookPro:~/project
Last login: Mon Jan 23 03:09:53 on ttys002
augustin ➤ cd project
augustin ➤ ~/project
▶
```

Window

PC
3D 개체
다운로드
동영상
문서
바탕 화면
사진
음악
로컬 디스크 (C:)
시스템 예약 (D:)



터미널 키기 이후 작업할 디렉토리로 경로 변경

윈도우의 경우, 작업할 디렉토리 우클릭 -> Git Bash Here 하거나 WSL 사용

개행문자 설정



```
# 맥, 리눅스 사용자
$ git config --global core.autocrlf input

# 윈도우 사용자
$ git config --global core.autocrlf true
```

개행문자 설정

개행문자 : `\n`, 즉 줄바꿈이다.

각 OS마다 사용하는 개행문자가 다르다

윈도우 : CRLF

맥, 리눅스 : LF

윈도우 사용자가 올린 코드에 들어간 개행문자 CRLF가 맥 사용자가 받은 뒤 단 한줄만 코치면 이를 인식하지 못해서 모든 부분이 고쳐졌다고 할 것이다.

(다만 그렇다고는 하는데 경험해본 적은 없음)

개행문자 설정

core.autocrlf	FALSE	input	TRUE
git commit	LF => LF	LF => LF	LF => LF
	CR => CR	CR => CR	CR => CR
	CRLF => CRLF	CRLF => LF	CRLF => LF
git checkout	LF => LF	LF => LF	LF => CRLF
	CR => CR	CR => CR	CR => CR
	CRLF => CRLF	CRLF => CRLF	CRLF => CRLF

개행문자 설정

개행문자 : \n, 즉 줄바꿈이다.

각 OS마다 사용하는 개행문자가 다르다

윈도우 : CRLF

맥, 리눅스 : LF

윈도우 사용자가 올린 코드에 들어간 개행문자 CRLF가 맥 사용자가 받은 뒤 단 한줄만 코치면 이를 인식하지 못해서 모든 부분이 고쳐졌다고 할 것이다.

(다만 그렇다고는 하는데 경험해본 적은 없음)

Git에 자신의 깃허브 정보 등록하기

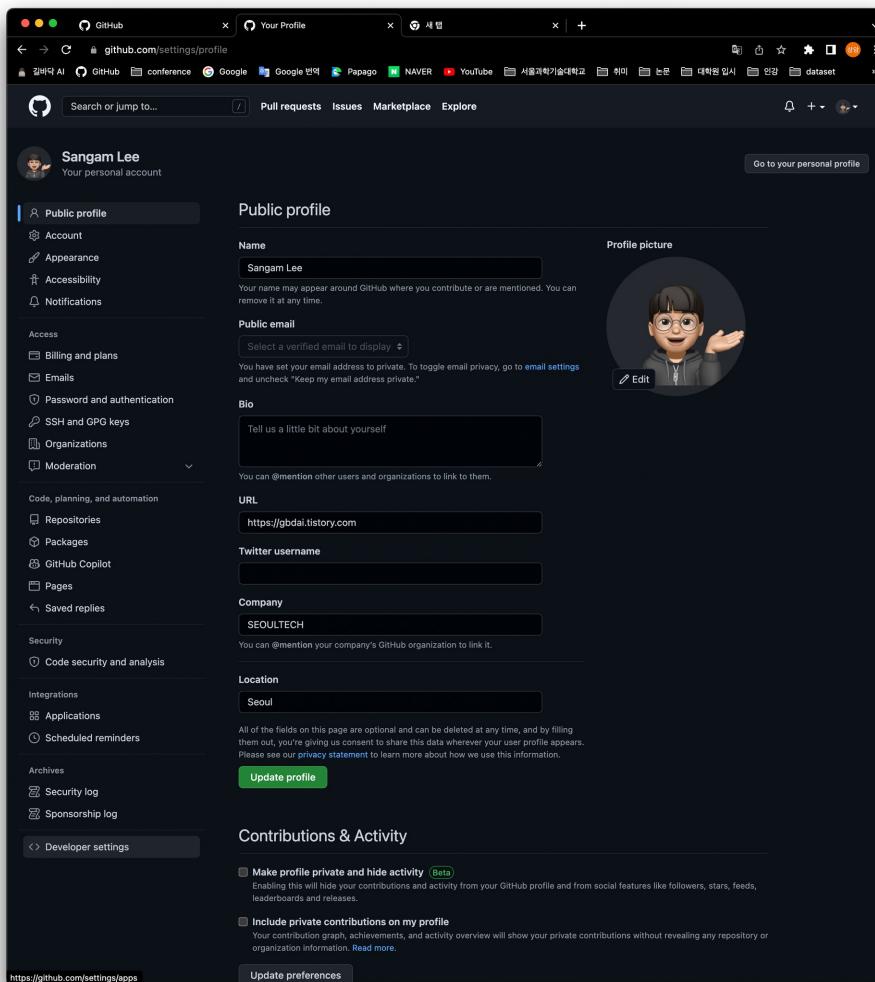
```
$ git config --global user.name 본인의 깃허브 닉네임  
$ git config --global user.email 본인의 깃허브 가입 시 사용한 이메일  
  
# 확인해보기  
$ git config --global user.name  
$ git config --global user.email
```

비밀번호(GitHub token)

The screenshot shows a GitHub profile page for a user named 'augustinLib'. The profile picture is a cartoon character of a person with glasses and a hoodie. The bio section includes a 'WHO AM I?' section with a brief description of the user's education and interests. Below the bio are sections for 'STACKS', 'ENVIRONMENT', and 'HOBBY'. On the right side of the page, a vertical sidebar lists various GitHub features like Overview, Repositories, Projects, Packages, Stars, and Gists. A dropdown menu is open, showing options such as 'Your profile', 'Your repositories', 'Your codespaces', 'Your organizations', 'Your projects', 'Your stars', 'Your gists', 'Upgrade', 'Feature preview', 'Help', and 'Settings'. The 'Settings' option is highlighted with a blue background. At the bottom of the sidebar, there are links for 'Edit profile', 'Sign out', and 'Send feedback'.

profile-setting

비밀번호(GitHub token)



The screenshot shows the GitHub Profile settings page. On the left, there's a sidebar with various tabs like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, SSH and GPG keys, Organizations, Moderation, and more. The 'Developer settings' tab is highlighted at the bottom of the sidebar. The main area is titled 'Public profile' and contains fields for Name, Profile picture, Public email, Bio, URL, Twitter username, Company, Location, and a note about optional fields. At the bottom, there's a green 'Update profile' button. Below the main profile section, there's a 'Contributions & Activity' section with two checkboxes: 'Make profile private and hide activity' (Beta) and 'Include private contributions on my profile'. At the very bottom of the page, there's a 'Update preferences' button.

맨 마지막의 developer settings 클릭



비밀번호(GitHub token)

The screenshot shows the GitHub developer settings interface. The URL in the address bar is <https://github.com/settings/tokens>. The left sidebar has a 'Personal access tokens' tab selected. The main content area contains a heading 'GitHub Apps' with a sub-section for 'Personal access tokens'. A note below the heading encourages users to build GitHub Apps and provides a link to developer documentation. At the bottom of the page, there is a footer with links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About pages.

personal access tokens 클릭

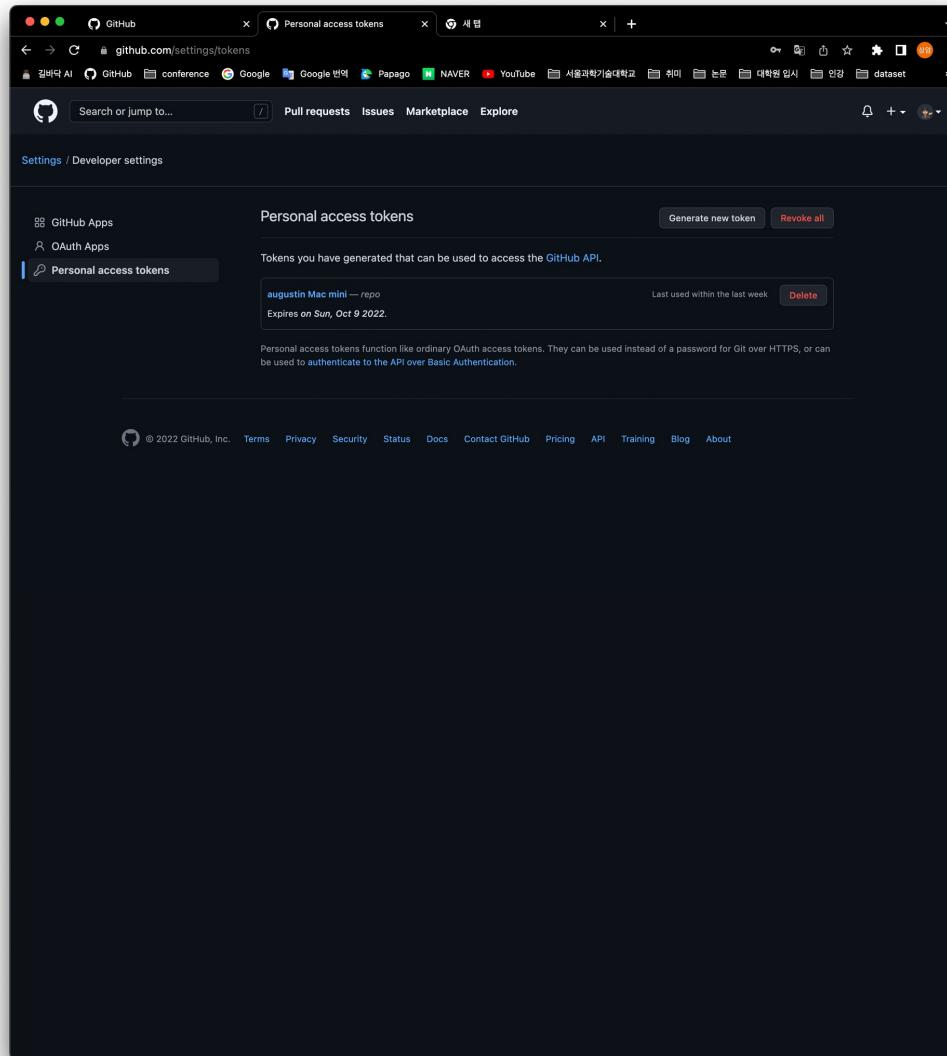


비밀번호(GitHub token)

The screenshot shows a web browser window with the GitHub URL `github.com/settings/tokens` in the address bar. The page is titled "Settings / Developer settings". On the left, there is a sidebar with three tabs: "GitHub Apps" (selected), "OAuth Apps", and "Personal access tokens". The main content area is titled "GitHub Apps" and contains a sub-section for "Personal access tokens". A sub-headline reads: "Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API. You can also read more about building GitHub Apps in our developer documentation." Below this, there is a "New GitHub App" button. At the bottom of the page, there is a footer with links: "© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About".

personal access tokens 클릭

비밀번호(GitHub token)



여기서 generate new token을 하면 token을 발급받을 수 있는데,
0| token을 password로 입력!

(다시 못보기 때문에 복붙해놓고 잘 보관해야함)



SSH Key 등록

The screenshot shows the GitHub 'Personal access tokens' page. The URL is `github.com/settings/tokens`. The page has a dark theme. On the left, there's a sidebar with links for GitHub Apps, OAuth Apps, and Personal access tokens. The 'Personal access tokens' link is currently selected. The main content area displays a table with one row of data:

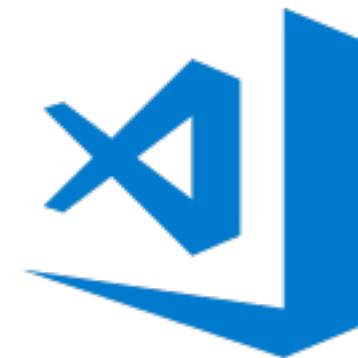
User	Repository	Last used	Action
augustin	Mac mini — repo	Within the last week	Delete

Below the table, there's a note: "Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication." At the bottom of the page, there's a footer with links: Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, About.

여기서 generate new token을 하면 token을 발급받을 수 있는데,
0| token을 password로 입력!

(다시 못보기 때문에 복붙해놓고 잘 보관해야함)

Git with VScode



Visual Studio Code

Fork와 Branch??

BOAZ에서 개발한, 사용자에게 인사를 건네는 프로그램인
say-hello가 운영되고 있다 가정해보자

[BOAZ/say-hello](#)

main(master)

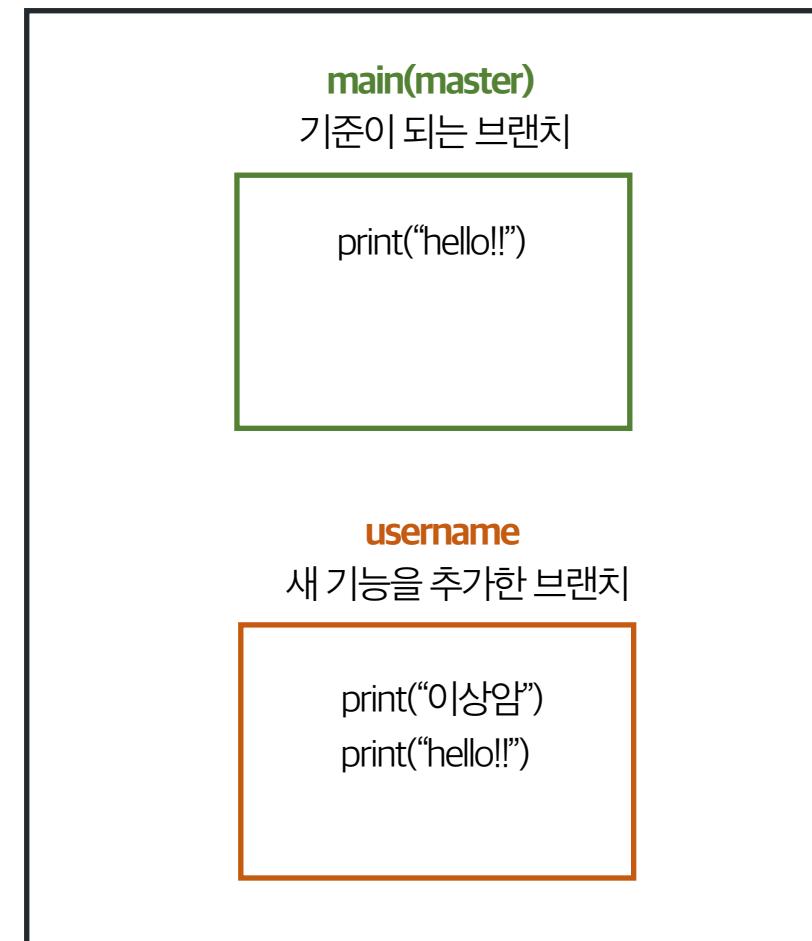
기준이 되는 브랜치

```
print("hello!!")
```

Fork와 Branch??

BOAZ 팀 내부에서 사용자의 이름도 같이 출력해주는 기능을 넣고 싶어한다.
이때, 팀 내부에서 분화하는 것이기 때문에 branch를 생성한다

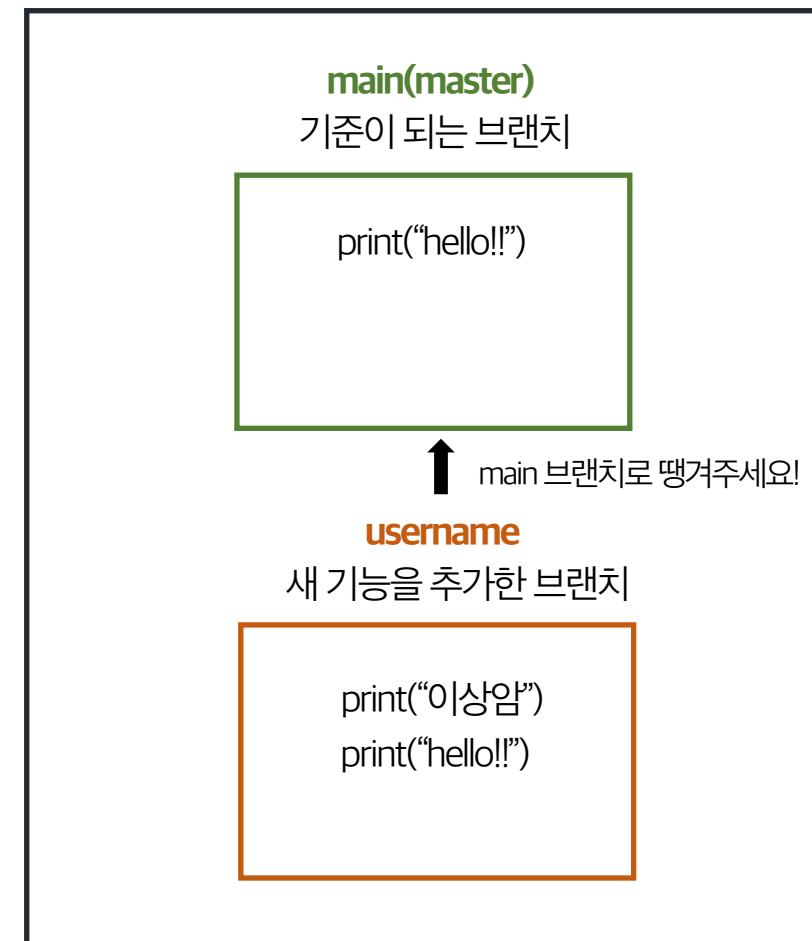
BOAZ/say-hello



Fork와 Branch??

사용자 이름 출력 기능을 개발한 이후, 본 프로그램에 추가하려고 한다.
이를 위해 main branch에 PR(pull request)를 한다.

BOAZ/say-hello



Fork와 Branch??

충분한 검토 끝에, 문제가 없음을 확인한 후 main branch로 merge한다

BOAZ/say-hello

main(master)

기준이 되는 브랜치

```
print("이상암")
print("hello!!")
```

username

새 기능을 추가한 브랜치

```
print("이상암")
print("hello!!")
```

Fork와 Branch??

충분한 검토 끝에, 문제가 없음을 확인한 후 main branch로 merge한다

[BOAZ/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("이상암")
print("hello!!")
```

Fork와 Branch??

이후, BOAZ 팀이 아닌 제 3의 인물이 코드가 맘에 들지 않거나, 다양한 이유로 해당 프로그램을 발전시키려 한다.

BOAZ/say-hello

main(master)

기준이 되는 브랜치

```
print("이상암")
print("hello!!")
```

Fork와 Branch??

그래서 그 사람은 fork를 하여 say-hello 레포를 본인 레포로 복사해왔다

[BOAZ/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("이상암")
print("hello!!")
```

[augustinLib/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("이상암")
print("hello!!")
```

Fork와 Branch??

이후, 새 기능을 추가하였다.

[BOAZ/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("0상암")  
print("hello!!")
```

[augustinLib/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("0|상암")  
print("hello!!")  
print("have a nice day")
```

Fork와 Branch??

추가한 새 기능을 프로그램에 적용시키고 싶어 원본 레포 (BOAZ 팀의 repo)에 pr을 날린다

[BOAZ/say-hello](#)

main(master)

기준이 되는 브랜치

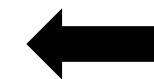
```
print("이상암")
print("hello!!")
```

[augustinLib/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("0|상암")
print("hello!!")
print("have a nice day")
```



나 이번에 이런거 개발함
내 기능 반영해줘

Fork와 Branch??

BOAZ팀은 검토 후 흡족해하며 해당 기능을 반영하였다.

[BOAZ/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("0상임")
print("hello!!")
print("have a nice day")
```

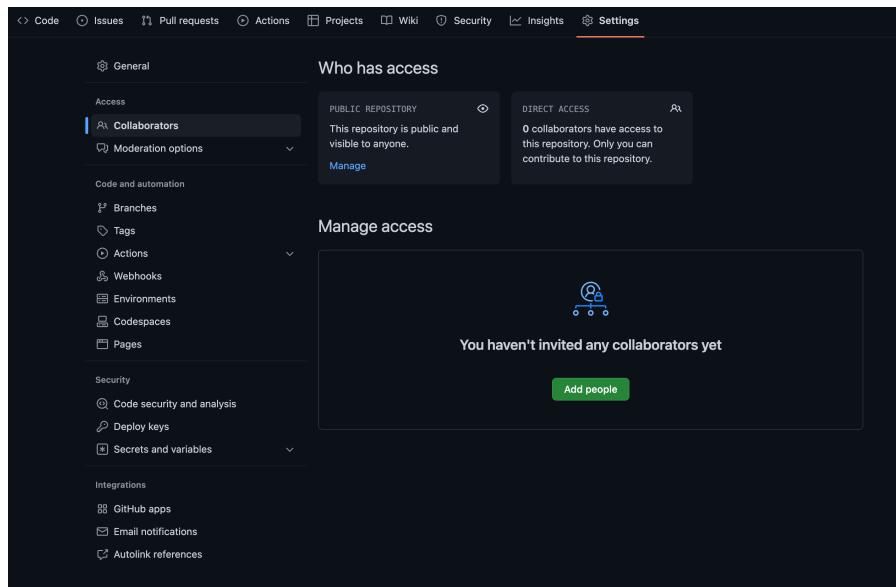
[augustinLib/say-hello](#)

main(master)

기준이 되는 브랜치

```
print("0|상임")
print("hello!!")
print("have a nice day")
```

Fork와 Branch??



정리하자면

Fork와 Branch 둘 다 **분기를 나누는 것이다.**

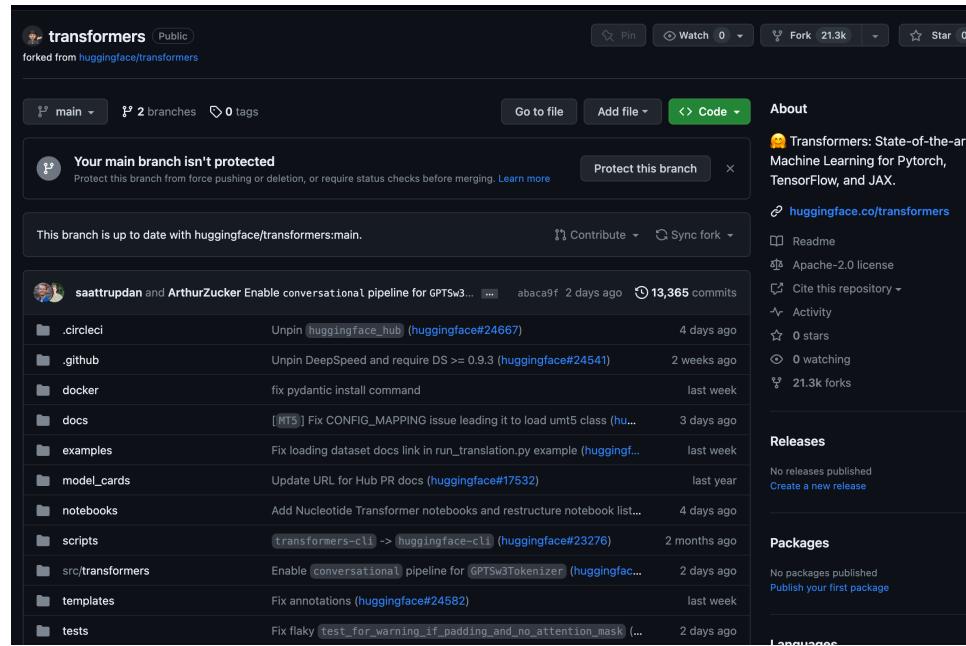
다만, Branch는 레포 안에서, Fork는 레포 바깥에서 분기를 나누는 것이다.

**Fork는 여려 개의 원격 저장소(깃허브의 레포지토리)를 파서 분기를 나누고,
Branch는 하나의 원본 저장소 안에서 분기를 나누는 개념이다.**

이러한 특성 때문에 **branch는 local에서 나눌 수 있고** (물론 깃허브에서도 나눌 수 있습니다)
Fork는 local에서 불가능하고 깃허브에서만 가능하다.

**보통, 해당 레포에 접근 권한이 있는 사람들(해당 프로젝트 팀원 등)은 branch를 이용하고,
접근 권한은 없는데 해당 프로젝트에 기여(contribute)하고 싶은 사람들은 fork로 contribute한다.**

Fork와 Branch??



정리하자면

Fork와 Branch 둘 다 **분기를 나누는 것이다.**

다만, Branch는 레포 안에서, Fork는 레포 바깥에서 분기를 나누는 것이다.

Fork는 여려 개의 원격 저장소(깃허브의 레포지토리)를 파서 분기를 나누고,
Branch는 하나의 원본 저장소 안에서 분기를 나누는 개념이다.

이러한 특성 때문에 **branch는 local에서 나눌 수 있고** (물론 깃허브에서도 나눌 수 있습니다)
Fork는 local에서 불가능하고 깃허브에서만 가능하다.

보통, 해당 레포에 접근 권한이 있는 사람들(해당 프로젝트 팀원 등)은 branch를 이용하고,
접근 권한은 없는데 해당 프로젝트에 기여(contribute)하고 싶은 사람들은 fork로 contribute한다.

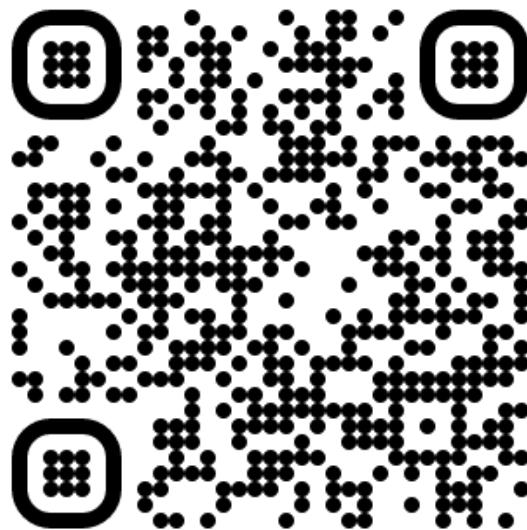


Fork와 Branch??

The screenshot shows the GitHub repository page for 'transformers'. At the top, there are buttons for 'Pin', 'Watch 0', 'Fork 21.3k', and 'Star 0'. Below that, the repository name 'transformers' is shown as 'Public' and 'forked from [huggingface/transformers](#)'. The main navigation bar includes 'main' (selected), '2 branches', '0 tags', 'Go to file', 'Add file', and 'Code'. A prominent message 'Your main branch isn't protected' with a note to 'Protect this branch' is displayed. Below this, it says 'This branch is up to date with [huggingface/transformers:main](#)'. On the right side, there's an 'About' section with a brief description of the project: 'Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.' It also lists links to 'huggingface.co/transformers', 'Readme', 'Apache-2.0 license', 'Cite this repository', 'Activity', '0 stars', '0 watching', and '21.3k forks'. The central part of the page displays a list of recent commits from various contributors, such as 'saatrupdan' and 'ArthurZucker'. At the bottom, sections for 'Releases', 'Packages', and 'Languages' are visible.

그리고 원본 레포랑 포크한 레포랑 자동 동기화되는건 아니기에,
Sync fork를 잊지 말자!

SSH



번거로운 GitHub token 대신 ssh로 인증해요

<https://goddaehee.tistory.com/254>



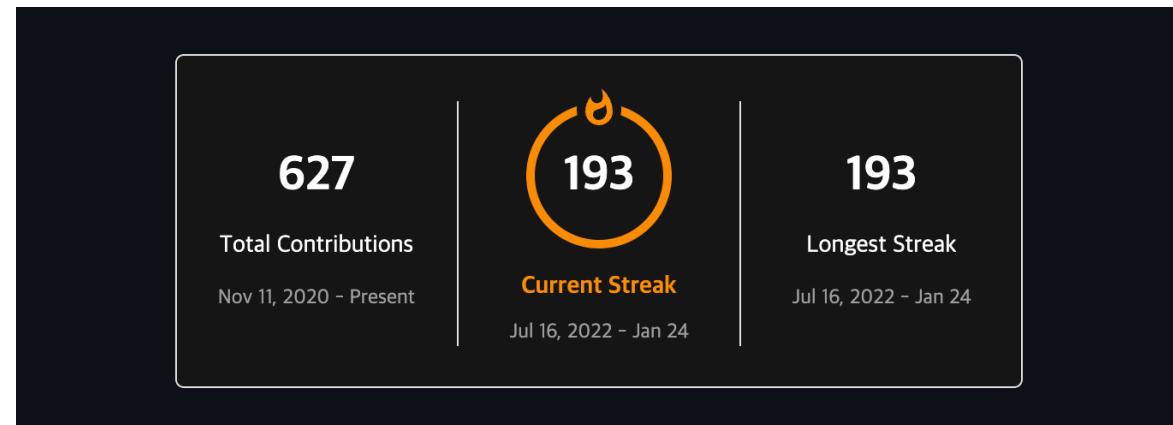
잔디심기

Contribution하면 심을 수 있는 잔디,
1일 1잔디 어떠신가요?



잔디심기

Contribution하면 심을 수 있는 잔디,
1일 1잔디 어떠신가요?



 감사합니다
