Welcome to SINAGRAM!

1학년 반가워요!

여러분을 소개해 주세요!

무엇을 좋아하는지, 어디에서 왔는지 등등...

시나그램이 뭐예요?

🔎 시나그램이 뭐예요?

디자인?

웹?

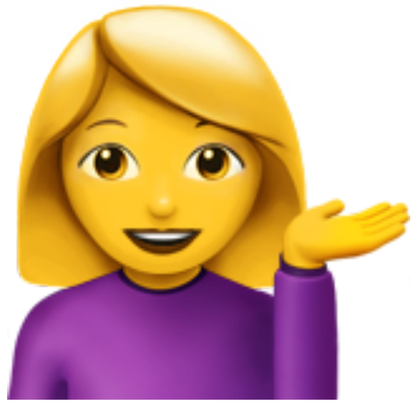
서버?

iOS?

1학년들

안드로이드?

🔎 시나그램이 뭐예요?



2학년들

우리가 알려 줄게!

시나브로? 그램?

🔎 시나브로? 그램?

시나브로 Since 2015

: 모르는 사이 조금씩 조금씩

함께 배우며 모르는 사이 조금씩 조금씩 성장하는
대마고 최고령 WEB 동아리

🔎 시나브로? 그램?

장점 Advantage

1. 멘토링을 통해 쌓는 탄탄한 기초 개념
2. 실무에서 재직 중인 멘토 선배분들
3. 깊게 할 수 있는 웹 공부
4. 주도적으로 진행하는 학년별 프로젝트

🔎 시나브로? 그램?

주요 활동 Activity

1. 선후배 멘토링
2. 학년별 프로젝트
3. 대회 참가
4. 분야별 스터디
5. 라이트닝 토크

🔎 시나브로? 그램?

그램 Since 2017

: GReat Awesome Members

2017년에 시작된
모바일 앱 프로그래밍에 대해 공부하는 동아리

🔎 시나브로? 그램?

장점 Advantage

1. 좋은 사람들이 모여 있는 공간
2. 딱딱한 분위기 NO! 언제나 즐거운 동아리 시간
3. 빵빵한 선배 지원
4. 체계적인 시스템

🔎 시나브로? 그램?

주요 활동 Activity

1. 팀 스터디
2. 그램톡
3. 알고리즘 스터디
4. 동아리 프로젝트
5. 스낵 타임

무엇을 배울 수 있나요?

🔎 무엇을 배울 수 있나요?

Design

: 프론트엔드 개발 전 최종 결과물을 미리 만들어 UI 개발에 도움을 준다

우리는 이래서 재미있어요!

1. 내가 작업한 게 바로 눈에 보인다!
2. 요구사항을 UI로 가시화한다!

시나그램에서는 이걸 해 볼 수 있어요!

나만의 자기소개 페이지 디자인

🔎 무엇을 배울 수 있나요?

WEB Front-End

: 웹 페이지에서 보여지는 것들을 구성하고 개발

우리는 이래서 재미있어요!

1. 코딩하자마자 바로 확인하는 아웃풋
2. 설치 없이 브라우저만으로!
3. 데스크탑, 태블릿, 스마트폰 어디서든

시나그램에서는 이걸 해 볼 수 있어요!

나만의 자기소개 페이지 만들기

🔎 무엇을 배울 수 있나요?

iOS

: 애플의 모바일 운영체제 iOS에서 동작하는 애플리케이션 개발

우리는 이래서 재미있어요!

1. **Swift라는 매력적인 언어**
2. **애플에서 제공하는 XCode라는 깔끔, 쾌적한 통합개발환경**
3. **시뮬레이션을 통해 바로바로 확인하는 아웃풋**

시나그램에서는 이걸 해 볼 수 있어요!

나만의 음악 플레이어 만들기

🔎 무엇을 배울 수 있나요?

Android

: 구글의 모바일 운영체제 안드로이드에서 작동하는 애플리케이션 개발

우리는 이래서 재미있어요!

1. 내가 필요로 하는 앱을 직접 만들 수 있다!
2. 안드로이드 기반의 휴대폰이라면 어디든! 다양한 기종에서 내가 만든 앱을 볼 수 있다

시나그램에서는 이걸 해 볼 수 있어요!

음성 인식 애플리케이션 만들기

🔎 무엇을 배울 수 있나요?

Back-End

: 정보를 저장하거나 요청에 맞게 데이터를 제공하는 등 데이터를 활용

우리는 이래서 재미있어요!

1. 자유로운 언어 선택
2. 데이터를 관리하는 로직, 다양한 언어와 DB 등 섭렵해야 할 게 많아서 생각할 게 많다

시나그램에서는 이걸 해 볼 수 있어요!

웹 페이지 서버로 띄워 보기
간단한 api 만들기

본격적으로 활동하기 위한 준비!

🔎 본격적으로 활동하기 위한 준비!

우리는 슬랙으로 소통해요!

🔗 슬랙 URL sinagram.slack.com

🔎 본격적으로 활동하기 위한 준비!

#team_1	#team_2
김예진	장서영
박동행	최아연
윤석훈	정창용
윤준기	정지원

🔎 본격적으로 활동하기 위한 준비!

#team_1

안드로이드
디자인
프론트
백엔드
iOS

#team_2

디자인
안드로이드
프론트
iOS
백엔드

🔎 본격적으로 활동하기 위한 준비!

세미나실 2-1 (그램실)

Android
iOS

3-2교실 (시나브로실)

Front-End
Design
Back-End

🔎 본격적으로 활동하기 위한 준비!

우리는 깃허브를 사용해요!

🔗 깃허브 URL https://github.com/SinaGram-DSM/Tasting

🔎 본격적으로 활동하기 위한 준비!

🔎 본격적으로 활동하기 위한 준비!

VCS

: 버전 관리 시스템

(Version Control System)

🔎 본격적으로 활동하기 위한 준비!

컴퓨터 구조 발표

컴구발표.pptx
컴구발표(1).pptx
컴구발표_최종.pptx
컴구발표_진짜최종.pptx
컴구발표_찐찐찐찐찐찐찐찐최종.pptx

이것도 나름의 버전 관리랍니다!

🔎 본격적으로 활동하기 위한 준비!

git

: VCS 툴

🔎 본격적으로 활동하기 위한 준비!

왜 쓸까요?

1. 잘못된 변경으로부터의 복구
2. 협업할 시 각자의 코드를 하나로 합치기 위함
3. 협업할 시 분업하기 위함
4. 코드의 수정을 누가, 언제 했는지 알기 위함
5. 코드의 변경을 안전하게 하기 위함

🔎 본격적으로 활동하기 위한 준비!

왜 쓸까요?

1. 잘못된 변경으로부터의 복구
2. 협업할 시 각자의 코드를 하나로 합치기 위함 → **Merge(병합)**
3. 협업할 시 분업하기 위함 → **Branch(브랜치)**
4. 코드의 수정을 누가, 언제 했는지 알기 위함
5. 코드의 변경을 안전하게 하기 위함

🔎 본격적으로 활동하기 위한 준비!

GitHub

: git을 이용한 프로젝트를 지원하는 웹 서비스

🔎 본격적으로 활동하기 위한 준비!

관련된 용어는 뭐가 있을까요?

1. 레포지토리(Repository): 저장소
   A. local: 로컬 컴퓨터의 저장소
   B. remote: 서버상의 원격 저장소
2. 클론(clone): remote repository를 로컬로 복사해 오는 것
3. 커밋(commit): 프로젝트의 변경 사항을 repository에 기록
4. 스테이지(stage): 커밋 직전의 상태

🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?

REMOTE	LOCAL		
COMMIT	COMMIT	STAGED	FILE
6월 12일 C	6월 12일 C		
6월 11일 B	6월 11일 B		
6월 10일 A	6월 10일 A		

🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?

REMOTE

COMMIT

6월 12일
C

6월 11일
B

6월 10일
A

LOCAL

COMMIT

6월 12일
C

6월 11일
B

6월 10일
A

STAGED

git add 파일명.확장자

FILE

🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?

REMOTE

COMMIT

6월 12일
C

6월 11일
B

6월 10일
A

LOCAL

COMMIT

6월 13일
📂

6월 12일
C

6월 11일
B

6월 10일
A

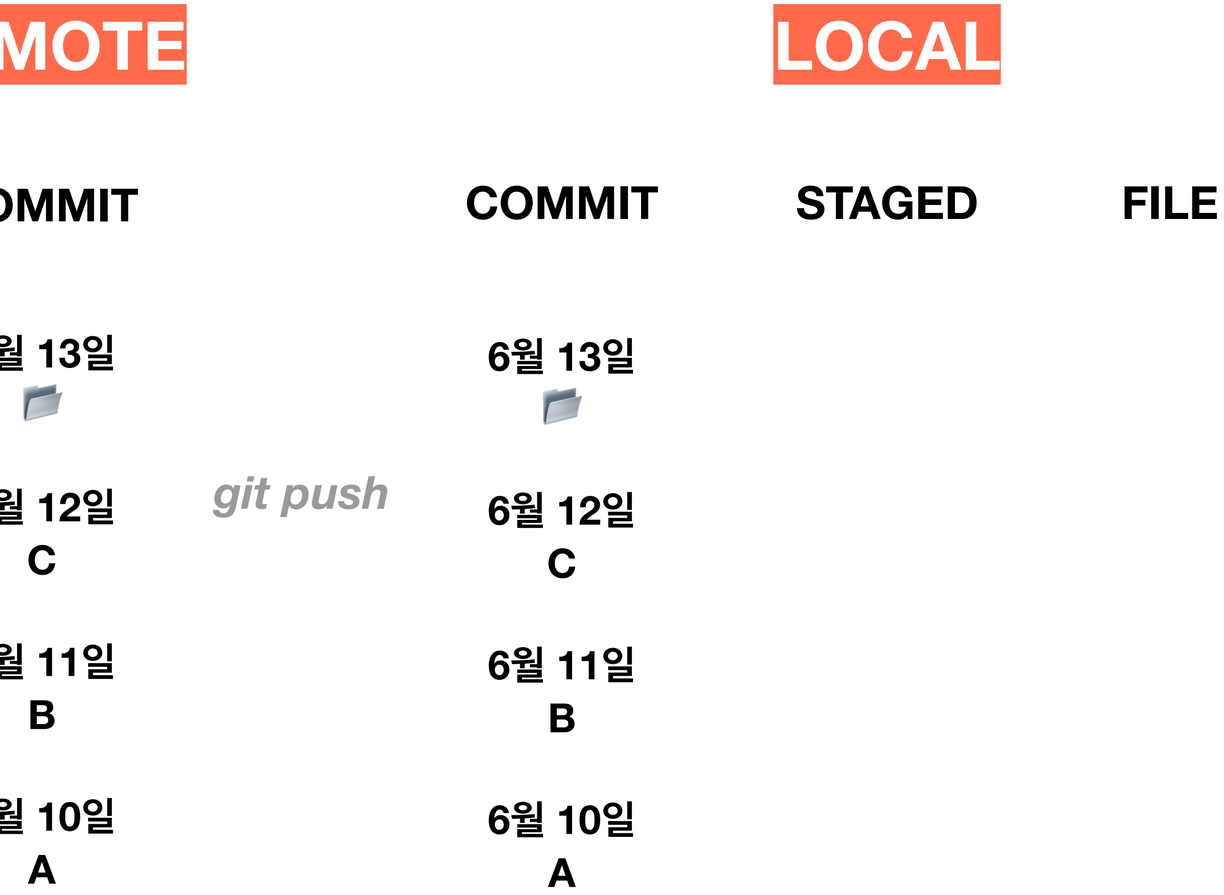
STAGED

FILE

git commit -m “커밋 메시지”

🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?



🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?

REMOTE

COMMIT

6월 13일
D

6월 13일
📂

6월 12일
C

6월 11일
B

6월 10일
A

LOCAL

COMMIT

6월 13일
📂

6월 12일
C

6월 11일
B

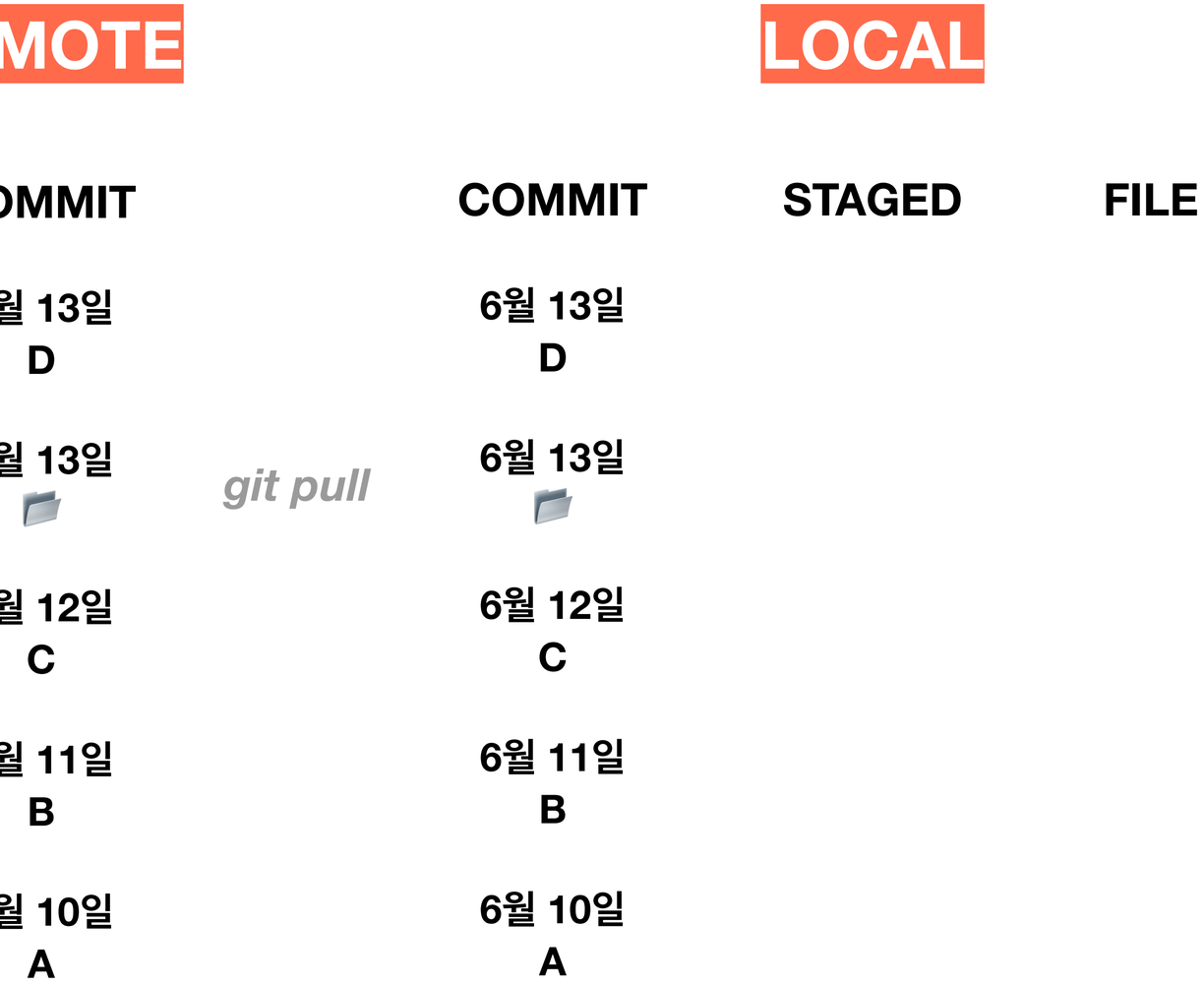
6월 10일
A

STAGED

FILE

🔎 본격적으로 활동하기 위한 준비!

어떻게
동작하는지
알아볼까요?



🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

repository를 생성하는 방법은 두 가지가 있습니다!

로컬에 생성하기

원격 저장소를 생성한 뒤
클론하기

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

repository를 생성하는 방법은 두 가지가 있습니다!

로컬에 생성하기

원격 저장소를 생성한 뒤
클론하기

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

💡

잠깐! git을 사용하기 전에,
앞으로 local repository들을 모아 둘 디렉토리를 하나 만드는 걸 추천해요!

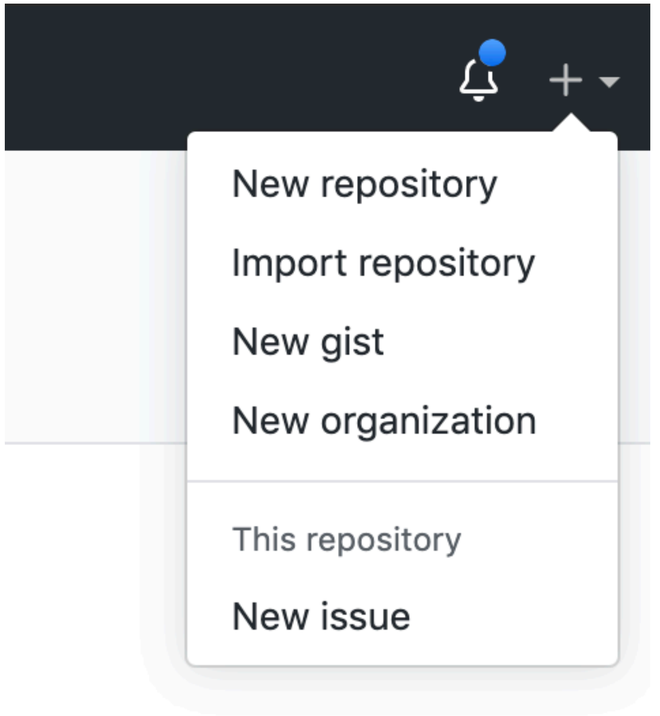
여러분이 편한 곳에  을 생성해 주세요!

GIT

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

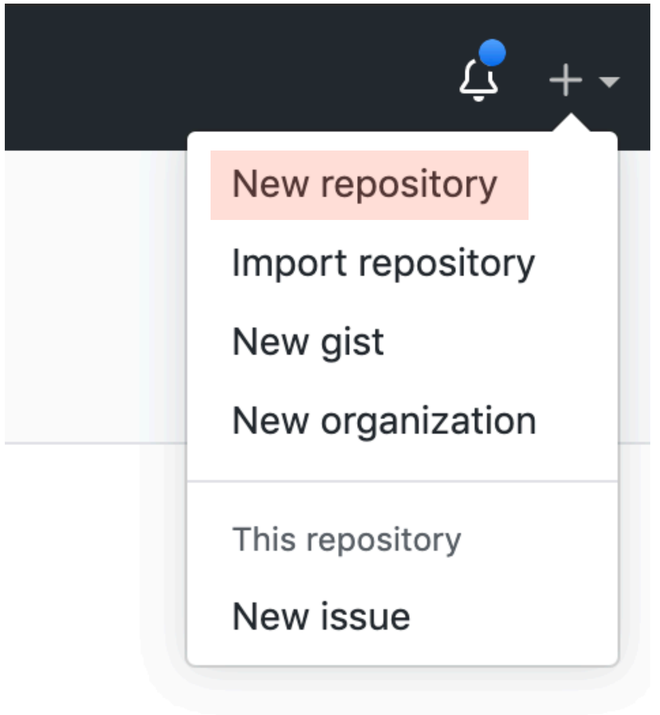
GitHub의 오른쪽 상단 + 버튼을 누릅니다.



🔎 본격적으로 활동하기 위한 준비!

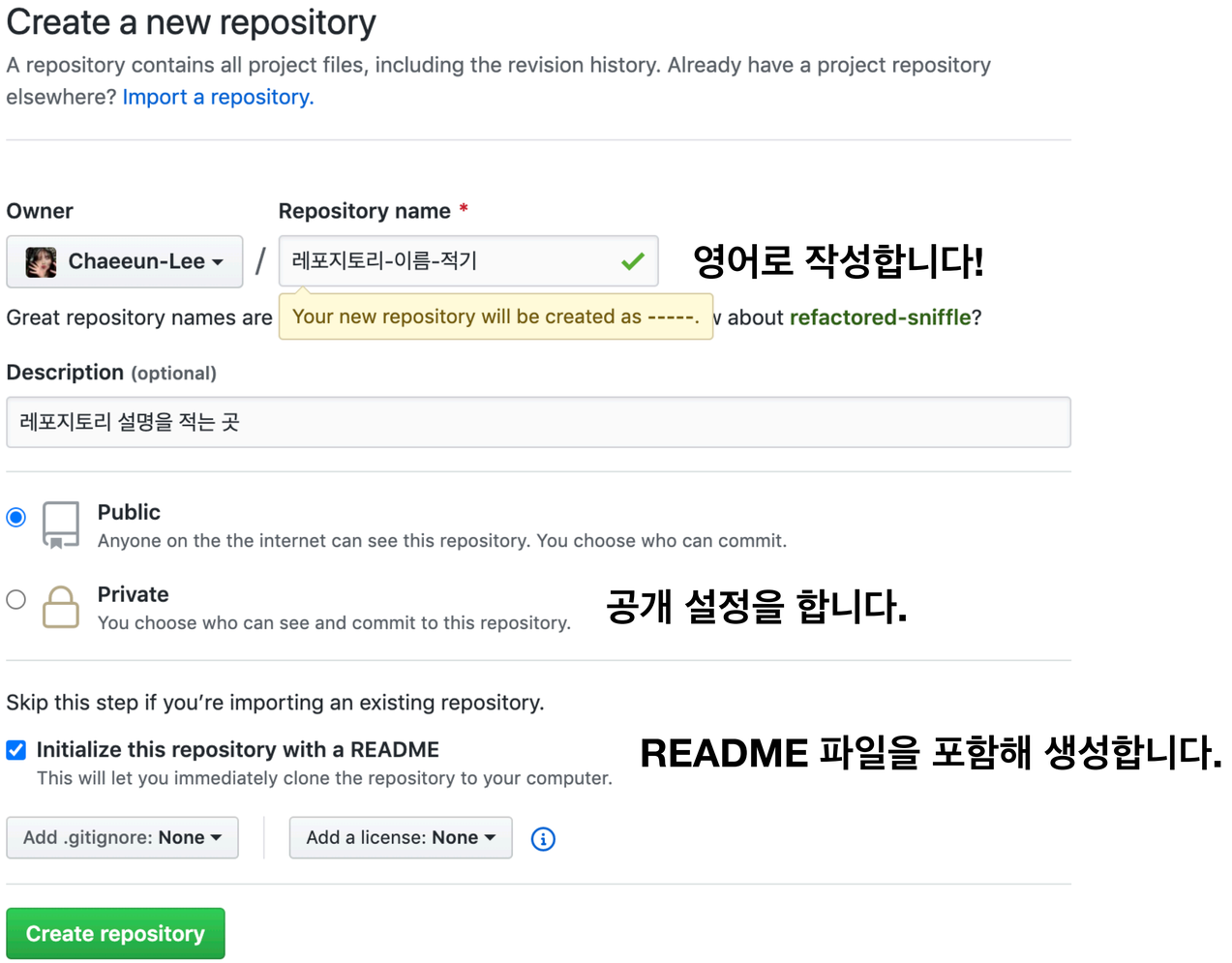
우리도
한번
해 볼까요?

New repository를 선택합니다.



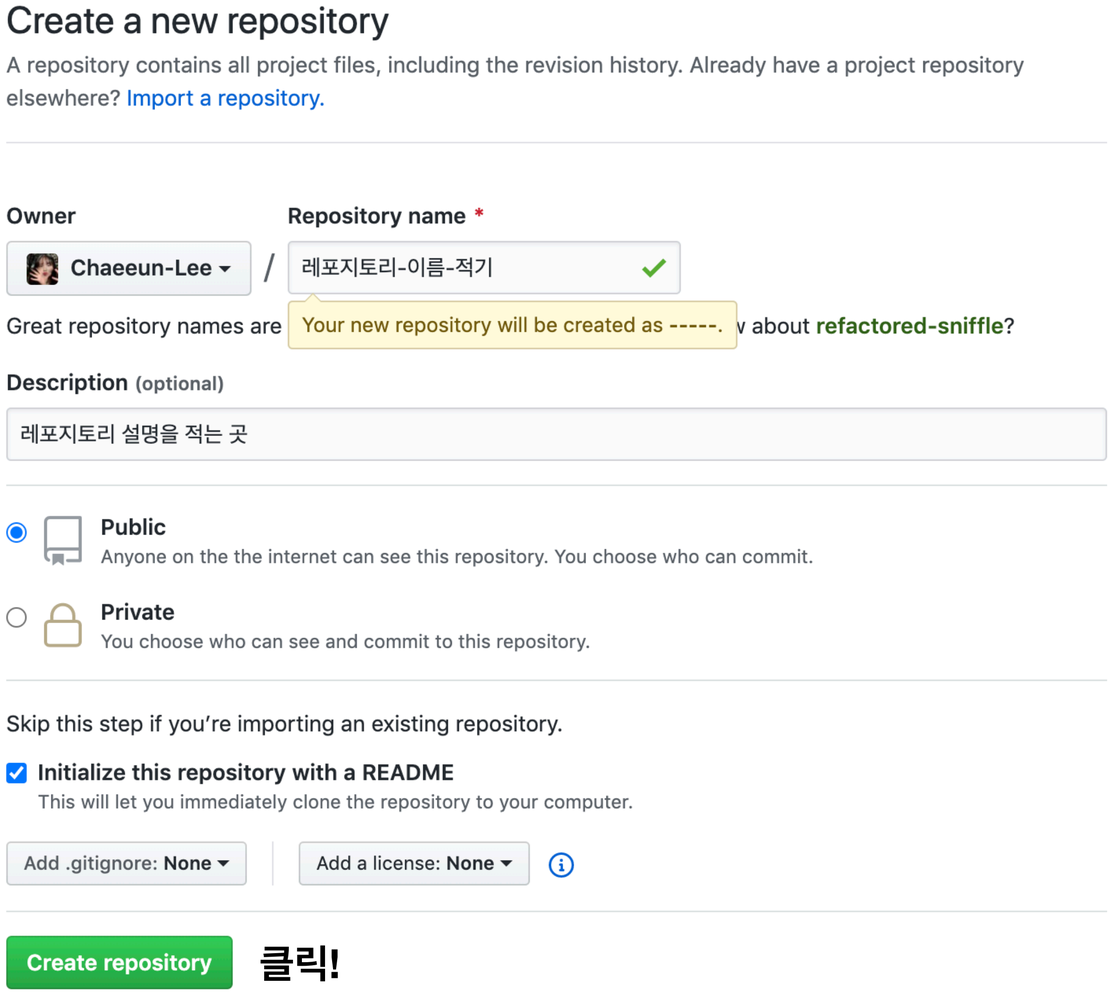
🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?



🔎 본격적으로 활동하기 위한 준비!

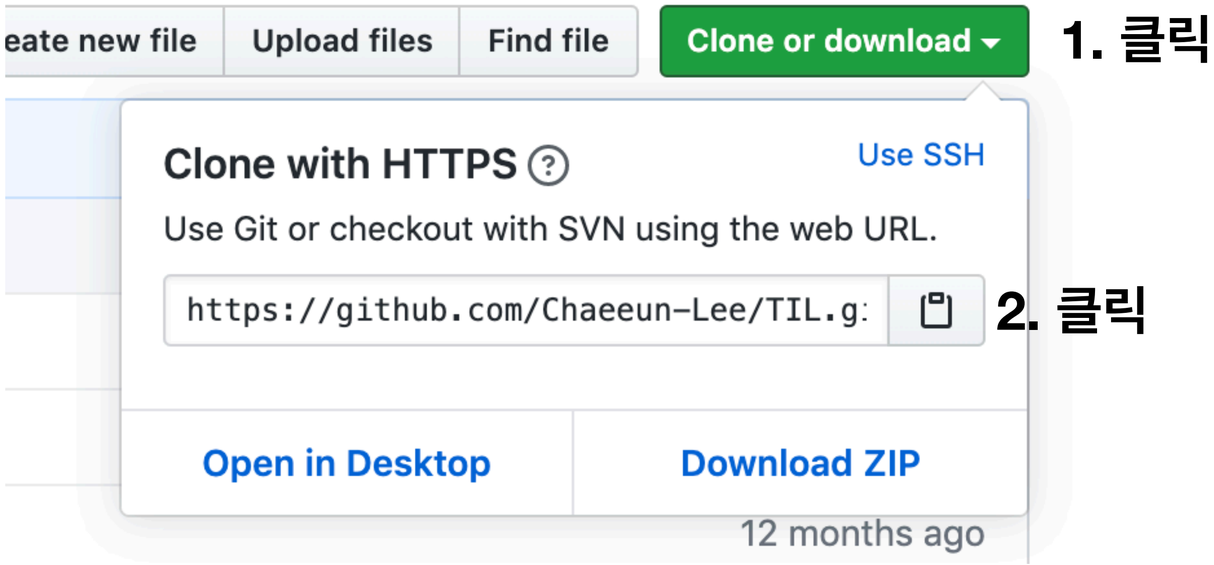
우리도
한번
해 볼까요?



🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

방금 만든 repository에 들어가

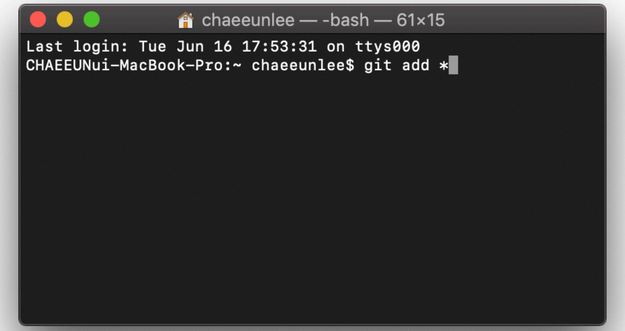


🔎 본격적으로 활동하기 위한 준비!

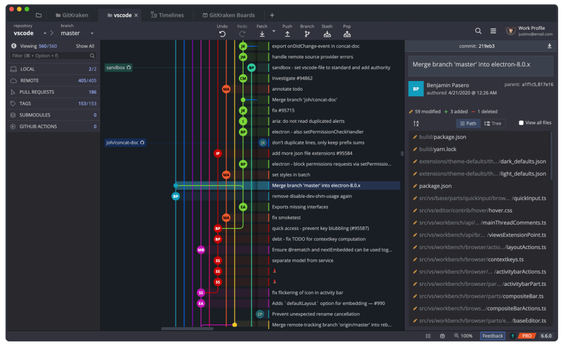
우리도
한번
해 볼까요?

git을 사용하는 방법은 두 가지가 있습니다!

CLI
(Command Line Interface)



GUI
(Graphic User Interface)

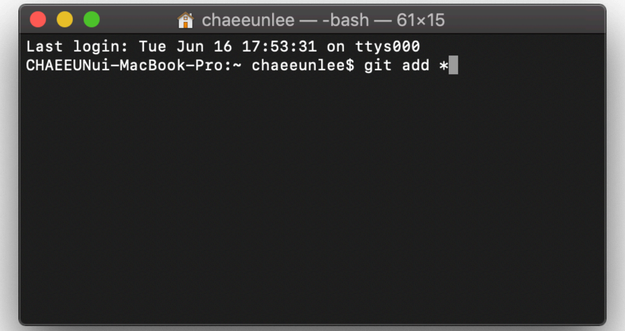


🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

git을 사용하는 방법은 두 가지가 있습니다!

CLI
(Command Line Interface)



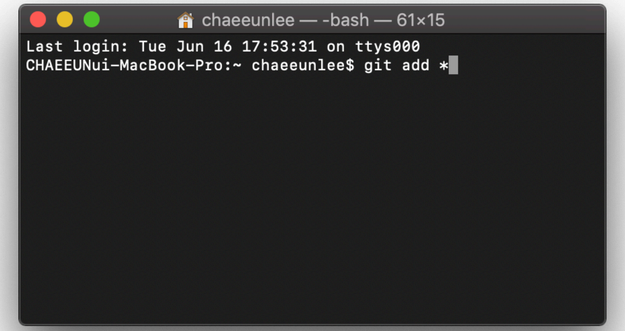
GUI
(Graphic User Interface)

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

git을 사용하는 방법은 두 가지가 있습니다!

CLI
(Command Line Interface)



GUI
(Graphic User Interface)

툴은 자신에게 맞는 것을 사용하는 걸 추천!

하지만 오늘 OT에서는 CLI를 채택했습니다

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

터미널을 켠 뒤, 아까 생성했던 repository에 들어가 주세요.

```
cd GIT/아까 생성한 repository 이름
```

cd는 터미널에서 디렉토리에 접근할 수 있는 명령어입니다.
cd 폴더 이름을 통해 접근할 수 있고,
cd ../를 통해 이전 폴더에도 접근할 수 있습니다.

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

터미널을 켜고 아래의 명령어를 입력하세요!

```
git clone 아까 복사해 둔 주소
```

원격에 생성했던 repository를 로컬로 복사해 올 수 있습니다.

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

아래의 폴더가 생성되었나요?

아까 생성한 repository 이름

축하합니다!

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

폴더 안에 있는 **README.md** 파일을 수정해 봅니다.

저장하는 것을 잊지 마세요!

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

터미널을 켜고 아래의 명령어를 입력하세요!

```
git add README.md
```

변경된 README.md 파일을 STAGE 상태로 변경합니다.

`git add *` 도 가능합니다! Repository 내의 모든 변경사항을 스테이징합니다.

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

아래의 명령어를 입력하세요!

```
git commit -m “Edit README.md”
```

STAGE 되어 있는 변경 사항들을 모두 기록합니다.
커밋은 항상 메시지가 동반되어야 합니다.

🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

터미널에 아래의 명령어를 입력합니다.

```
git push
```

로컬에 기록된 커밋을 원격 저장소에도 적용시켜 줍니다.

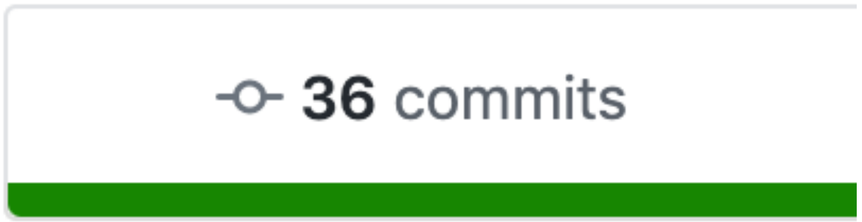
🔎 본격적으로 활동하기 위한 준비!

우리도
한번
해 볼까요?

원격 저장소에 가서 확인해 보세요!

README.md가 변경되어 있나요?

💡



을 클릭하면 이 repository에서 기록된 모든 커밋들을 볼 수 있습니다.

여러분은 이제 모든 준비가 끝났습니다!

여러분을 도울 선배들이 있다는 것을 잊지 말아 주세요!