



git

Git과 GitHub

2023 상반기 CMD 정규세션

발표자 전은지



목차

- Git을 이용한 협업의 중요성
- Git 개념, 용어 정리
- GitHub 소개
- GitHub를 사용하면 어떤 장점이 있을까
- Git, GitHub, GitHub Desktop 사용법 (실습)
- GitHub를 사용한 협업방법 - issue, branch
- 현업 개발자처럼 git 사용하기

- JavaScript/HTML/CSS에 대한 활용 능력과 내부 동작 원리에 대한 관심이 있으신 분
- **Git을 이용한 코드 형상관리와 코드 리뷰 경험이 있으신 분**
- 컴퓨터 과학의 기초 지식(데이터베이스/네트워크 등)을 이해하고 있으신 분

- **git/github 등의 협업 툴을 통한 동료와의 협업 개발에 어려움이 없으신 분**
- 긍정적이며 자기 주도적인 성장이 가능한 분
- **코드 리뷰에 적극적이신 분**
- 팀원들과 열린 마음으로 소통하며 개발을 진행할 수 있으신 분
- 새로운 프로그래밍 언어나 플랫폼에 대해서 도전적인 유연한 개발자

- 유닛 테스트 및 UI 테스트 작성 경험이 있으신 분
- 코드 리팩토링 경험이 있으신 분
- **페어 코딩 및 동료 코드 리뷰 경험이 있으신 분**
- 배포 자동화 경험이 있으신 분
- 웹사이트 성능 측정 및 최적화 경험이 있으신 분
- 웹사이트 보안에 대한 이해가 있으신 분
- **Git, Jira, Wiki 사용 경험이 있으신 분**
- 프로젝트 구축과 설정, 배포를 포함하여 프로젝트 시작부터 운영까지 경험이 있으신 분

- JavaScript 에 대한 이해가 있으신 분
- HTML, CSS 등 마크업에 대한 충분한 이해도가 있으신 분
- React, Vue, Angular 등 프론트엔드 라이브러리 또는 프레임워크에 대한 이해가 있으신 분
- 반응형 웹 개발에 대한 이해가 있으신 분
- HTTP 통신에 대해 기본적인 이해가 있으신 분
- **Git과 Github에 대한 기본적인 사용법을 알고 있고, Git Flow를 이해하고 계신 분**

- React, Vue, Angular 등 SPA 프레임워크 사용에 능숙하신 분을 찾습니다
- HTML, CSS, JavaScript에 대한 이해가 깊은 분을 찾습니다.
- UI/UX에 대해 높은 가치를 두고 있는 분을 찾습니다.
- **Git 등의 분산 버전 관리 시스템 이용에 능숙한 분을 찾습니다**



Git은

버전 관리 시스템(VCS)이다.

무엇을?

git 레포지터리(디렉토리)를 관리한다.

어떻게?

각 커밋(버전)별로 파일의 스냅샷을 찍는다.

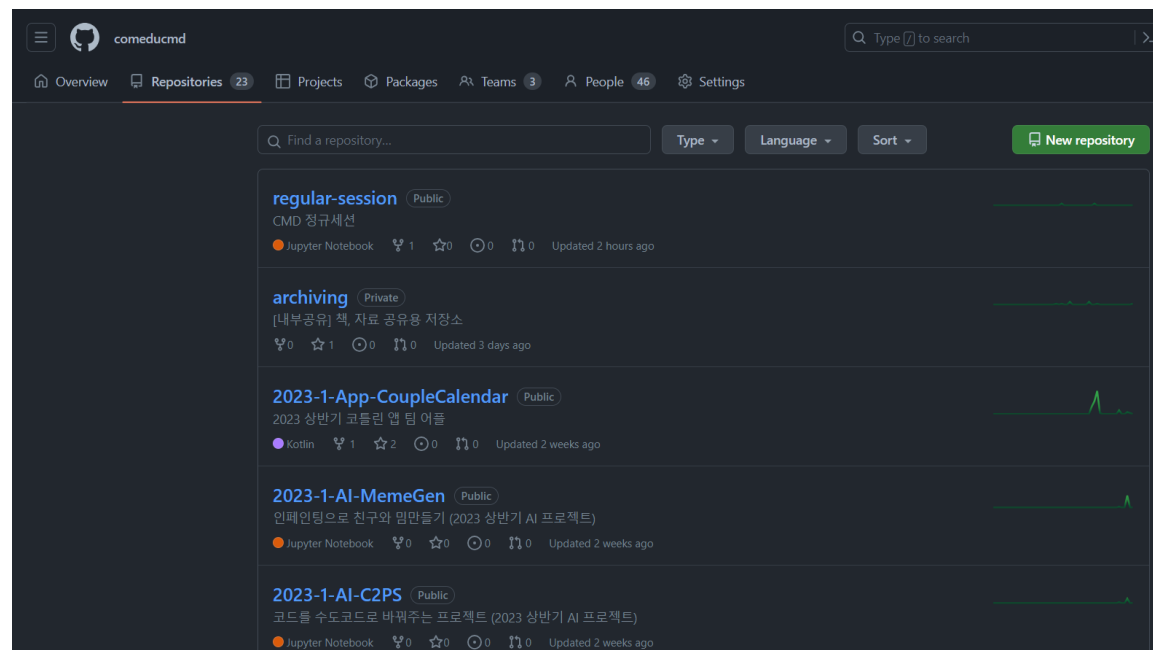
용어 1. 레포지터리 (repository)

Git으로 관리되는 디렉토리 == 저장소, 프로젝트, 폴더

folder > DS_study

이름	수정한 날짜	유형	크기
.git	2023-01-29 오후 8:29	파일 폴더	
그래프	2023-01-29 오후 8:29	파일 폴더	
스택	2023-01-05 오전 1:33	파일 폴더	
연결리스트	2023-01-12 오후 1:04	파일 폴더	
큐	2023-01-21 오후 10:41	파일 폴더	
트리	2023-01-21 오후 10:40	파일 폴더	

로컬 레포

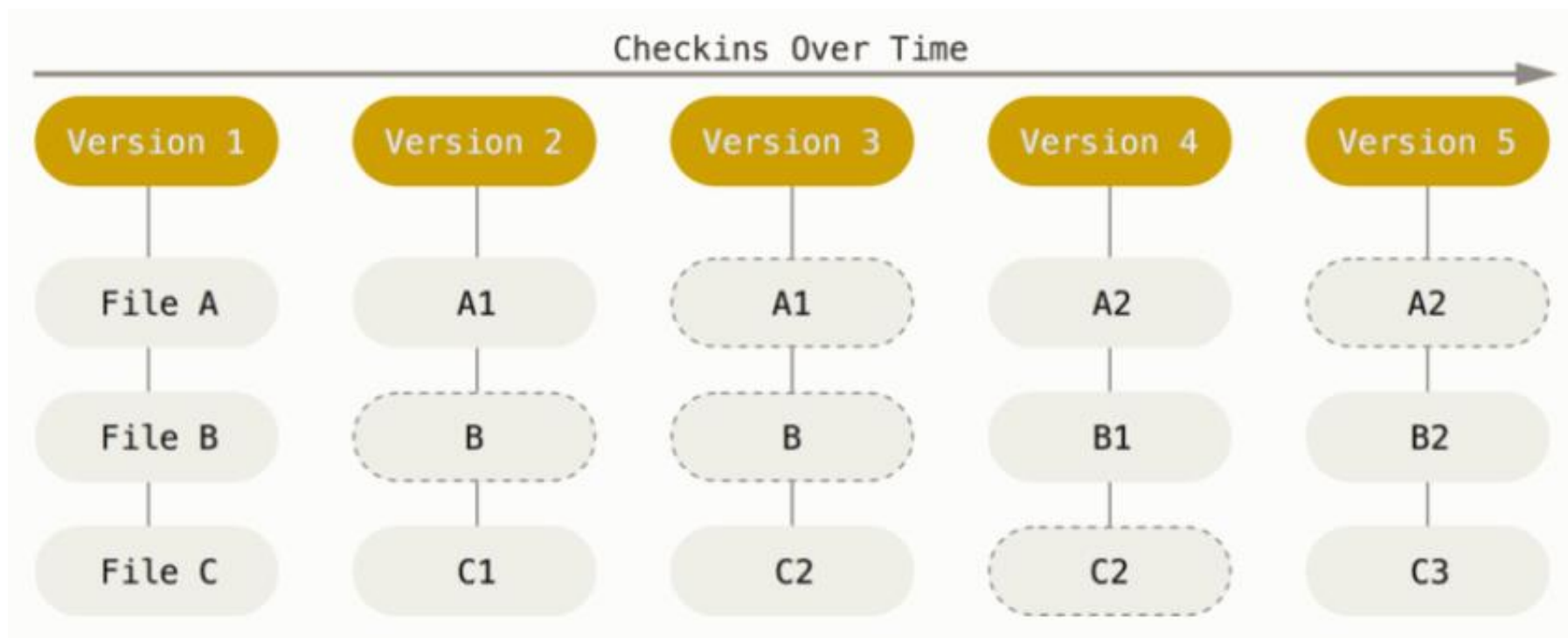


원격 레포

용어 2. 커밋 (commit)

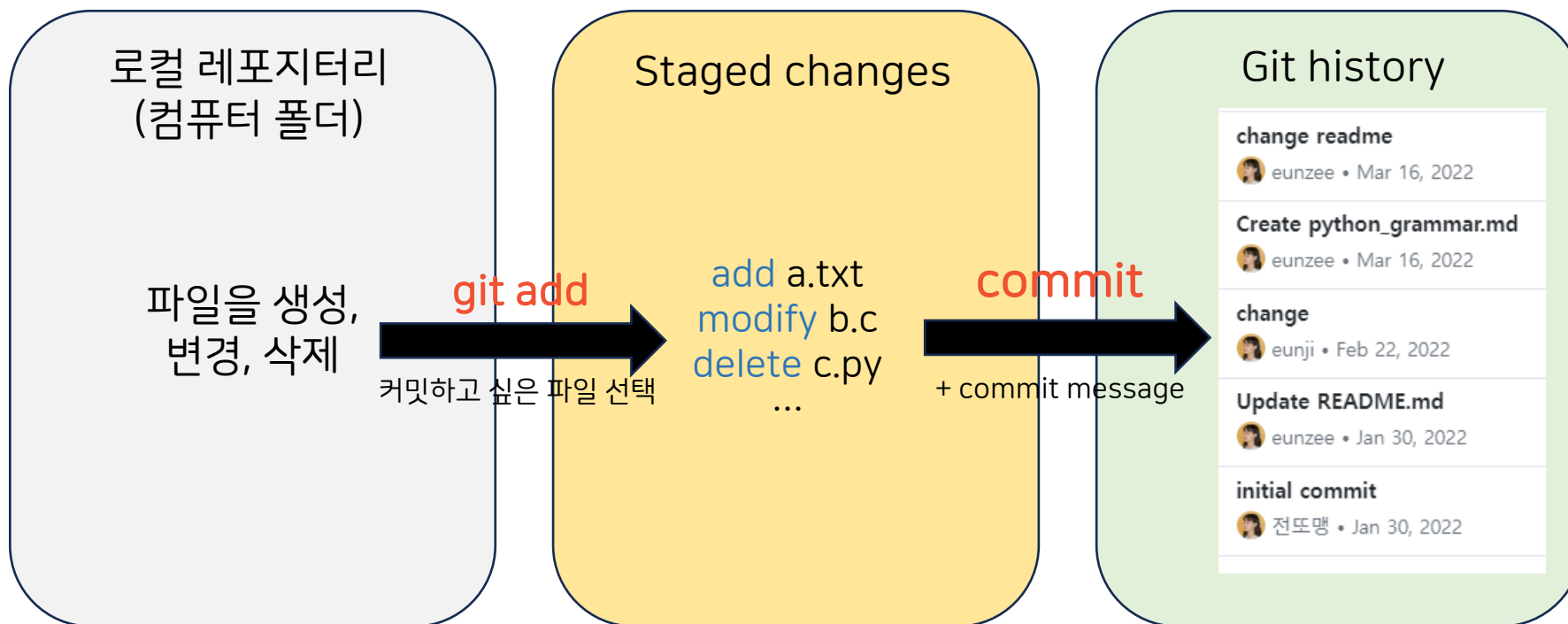
- 대부분의 작업이 로컬에서 진행
- 초초초 빠름

레포지터리의 스냅샷 == 버전, 체크포인트

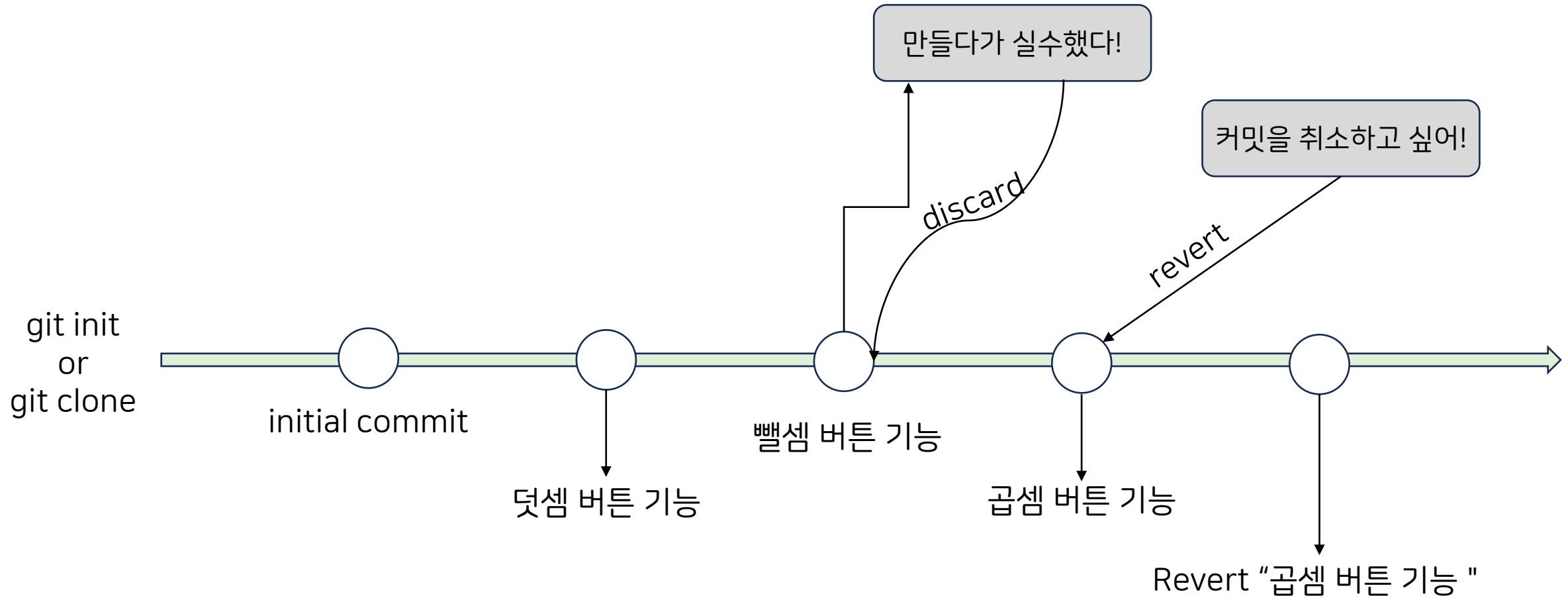


용어 2. 커밋 (commit)

커밋의 과정 (로컬에서)



Git 사용 flow ex) 계산기 프로그램



GitHub 개념



Git 저장소 호스팅 서비스

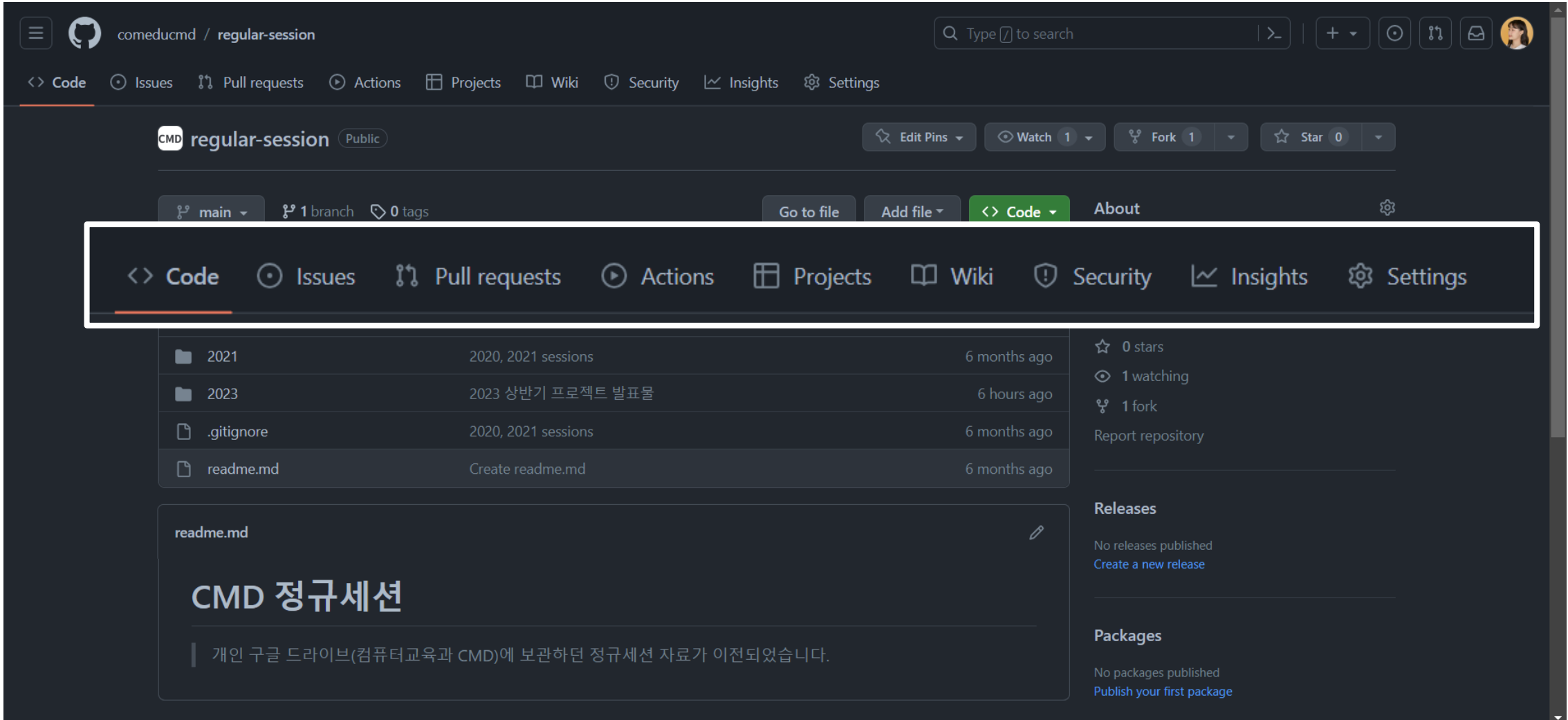
= 원격(remote) git 레포를 만들 수 있다.

= 깃허브 서버에 git 레포를 만들 수 있다.

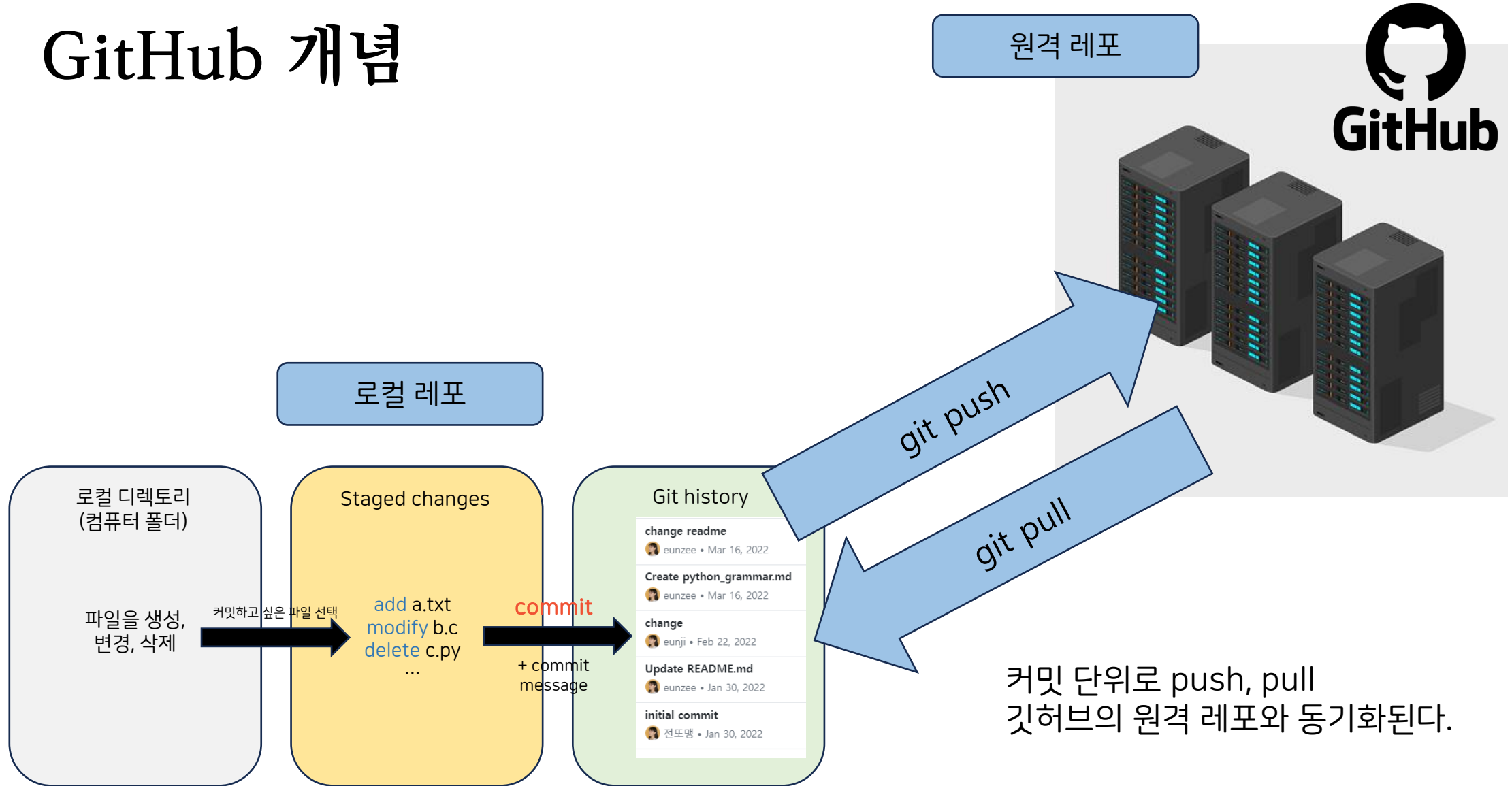
원격 레포와 내 컴퓨터 로컬 레포의 연동이 가능하다.

깃허브 말고도 GitLab 등의 저장소 호스팅 서비스가 있지만,
명실상부 깃허브가 가장 유명하다!

GitHub 개념



GitHub 개념



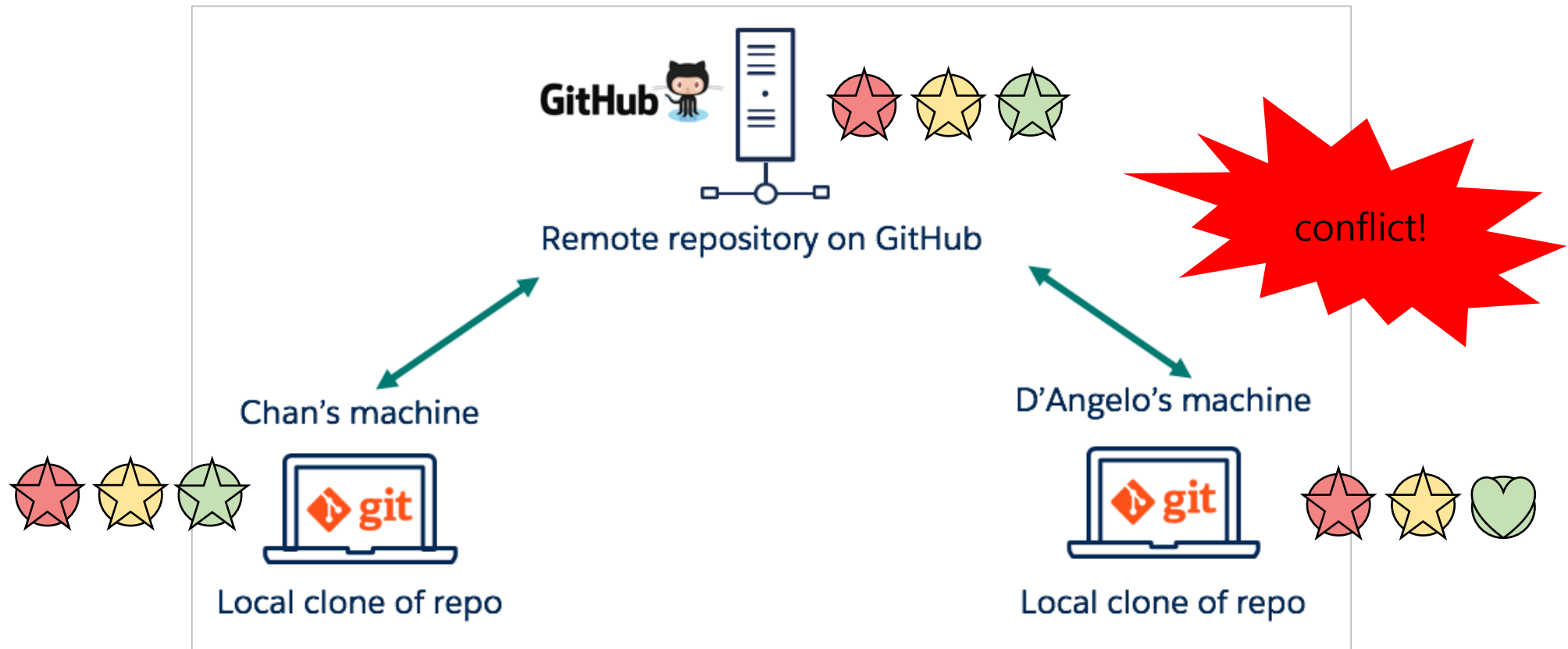
GitHub를 사용하면 어떤 장점이 있을까?

- 쉬운 코드 공유 (웹페이지를 보여주면 됨)
- 포트폴리오 관리
- 드라이브처럼 사용 가능
- 언제 어디서든지 원격 레포에 접근 가능

- 팀원과 협업 가능 (추후 설명)
- 오픈소스 프로젝트에 참여 가능
- 다양한 기능(이슈, 풀리퀘스트, 위키, 깃허브 액션, 서드파티 기능 등)

팀원과 협업 ★ ★ ★

하나의 원격 레포를 여러 사람이 수정 가능



팀원과 협업 ☆☆☆

be: delete average in /predct api

 nimod7890 • Dec 14, 2022

Merge branch 'main' of https://github.com...

 nimod7890 • Dec 14, 2022

be: predict result 반환 api 작성

 nimod7890 • Dec 14, 2022

[ADD] Submit box

 하유진 • Dec 14, 2022

[ADD] commit button

 하유진 • Dec 14, 2022

[FIX] gradient bug

 jason-choi-skku • Dec 14, 2022

[Mod] React useEffect Pattern

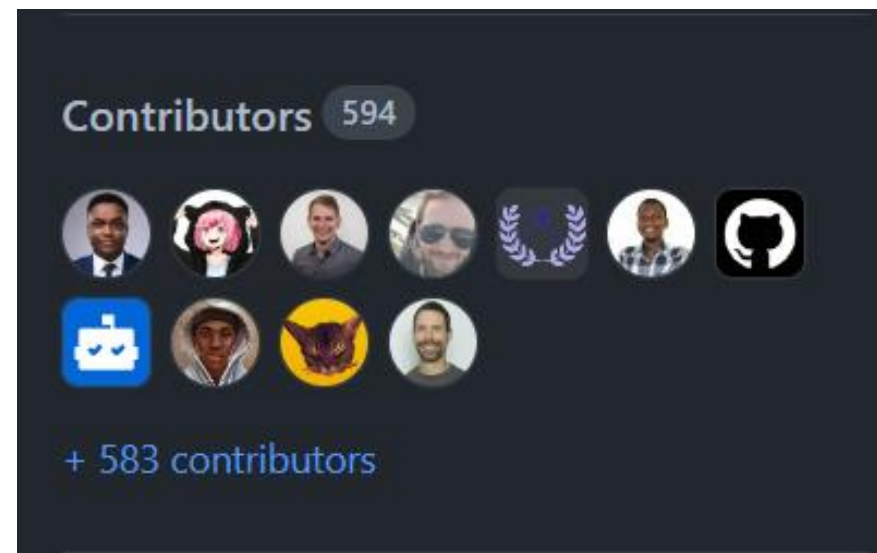
 jason-choi-skku • Dec 14, 2022

[Add] AreaChart, refactor codes

 jason-choi-skku • Dec 14, 2022

add model

 eunzee • Dec 13, 2022



오픈소스 컨트리뷰션

여러분이 github을 사용하게 될 방법

1-1. 폴더를 로컬 레포로(git init) -> 원격 레포 발행(publish)

1-2. git clone: 원격 레포를 내 컴퓨터로 복사

2. 코드 변경 => 저장 => 커밋

3. 깃허브 원격 레포로 pull/push

git, github, github desktop 실습!!

직접 해봅시다!

0. 깃 사용자 설정

```
$ git config --global user.name "Your Name"  
$ git config --global user.email you@example.com
```

1. github desktop 접속 (필요하다면 로그인)
2. clone repository 선택, [주소] 입력, 폴더 위치 선택
3. 생긴 폴더 확인, [이름].txt 파일 생성, 자기소개 입력, 저장
4. github desktop에서 커밋, 푸시

git bash로 하는 방법은

```
$ git clone [주소]
```

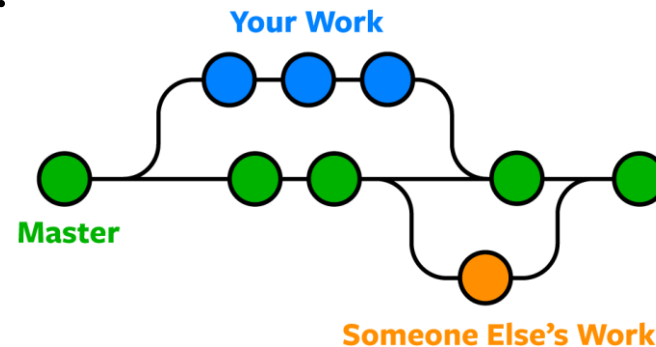

github를 사용한 협업방법 (advanced)

1. issue

- 이슈는 게시판같은 기능
- 업무분배(feature), 에러기록(bug)
- 이슈는 open 되었다가 해결되면 closed 됨.

2. branch (git 기능이다)

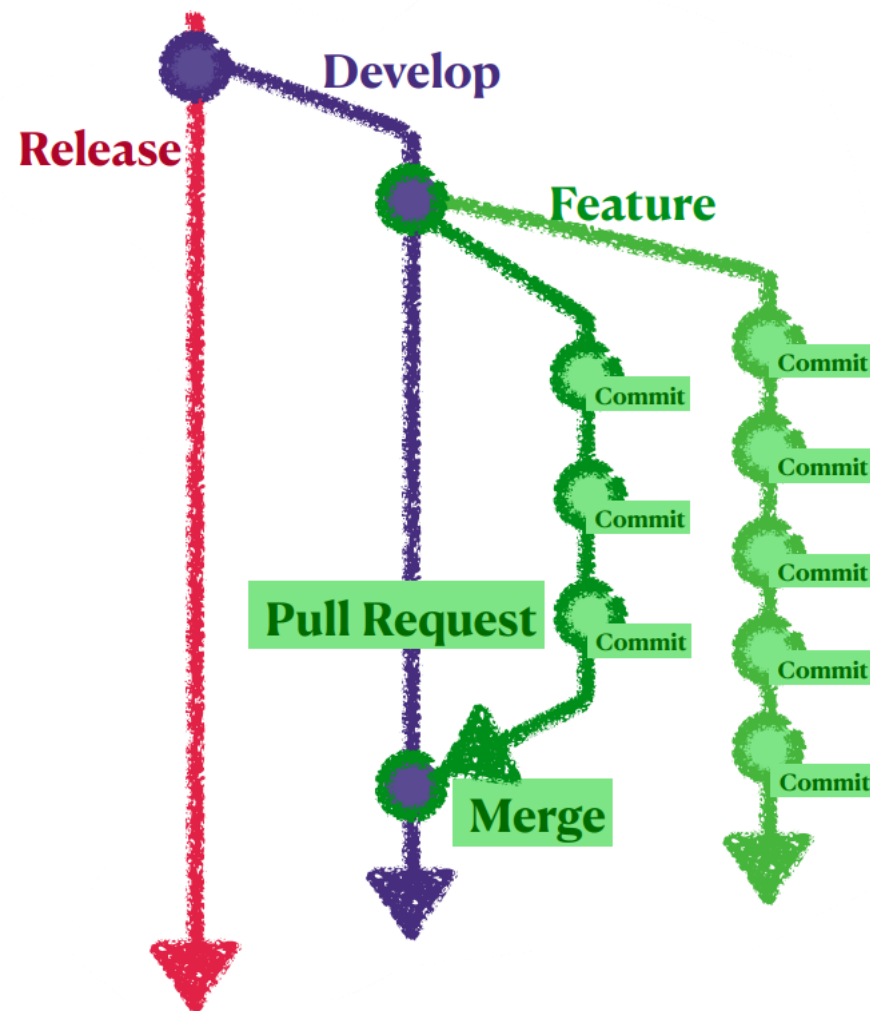
- 브랜치는 commit의 분기를 나누는 것.
- 브랜치끼리 merge 시킬 수 있다.



현업 개발자처럼 git 사용하기

작은 프로젝트에서는 오히려 오버헤드가 발생할 수 있으니 참고만 하기!

1. 메인 브랜치(릴리즈 브랜치)에 직접 커밋 x
 - 브랜치를 feature(기능) 별로 관리
2. 이슈 사용
 - 이슈로 업무분배, 그에 따라 브랜치 파기
 - 이슈 템플릿에 맞춰 자세하게 기록
3. pull request, code review
 - develop에 반영 시 다른 팀원들의 리뷰 후 반영



comeducmd 계정 소개

cmd에서 진행한 프로젝트, 스터디 / 유용한 정보 아카이빙이 진행되는 계정

이슈 탭을 이용해 OB 선배들에게 질문 가능!

(현업에서 일하고 계신 분들이 시간내서 답을 해주십니다... 😊)

<https://github.com/comeducmd>

감사합니다!

궁금한 부분 / conflict가 났다 언제든지 도와드립니다 연락주세요!