

버전 관리



기획서.pdf



기획서_수정본.pdf



기획서_최종.pdf



기획서_진짜_최종.pdf



기획서_진짜진짜_최종.pdf



기획서_20210503.pdf



starbucks



starbucks 로그인 페이지 개발본



starbucks 스크롤 버그 수정



starbucks 배포용



starbucks 배포 버그 수정본



starbucks 20210503 메뉴 개발



Git(깃)은 컴퓨터 파일의 변경사항을 추적하고
여러 사용자들 간에 해당 파일 작업을 조율하기 위한
대표적인 버전 관리 시스템(VCS)입니다.

SVN :

win+r

cmd

```
$ git --version
```



'git' 명령어는 명령어 라인 개발자 도구가 필요합니다.
도구를 지금 설치하겠습니까?

계속하려면 설치를 선택하십시오. App Store에서 Xcode 및 명령어 라인
개발자 도구를 설치하려면 Xcode 가져오기를 선택하십시오.

지금 안 함

설치



starbucks



index.html



main.css



favicon.png



사용자
(local, 로컬)



starbucks



index.html



main.css



favicon.png

로컬에 설치



 git



사용자
(local, 로컬)

;;;

개행 문자(Newline) 설정

macOS, Linux

\$ git config --global core.autocrlf input

Windows

\$ git config --global core.autocrlf true

사용자 정보

커밋(버전 생성)을 위한 정보 등록

\$ git config --global user.name 'YOUR_NAME'

\$ git config --global user.email 'YOUR_EMAIL'

구성 확인

Q키를 눌러서 종료!

\$ git config --global --list


```
$ git init
```

```
# 현재 프로젝트에서 변경사항 추적(버전 관리)을 시작.
```

master



index.html



main.css



favicon.png



변경사항 추적 중..
(stage)



git



사용자
(local, 로컬)

git

.git

가

```
$ git add index.html
```

변경사항을 추적할 특정 파일(index.html)을 지정.

master



index.html



main.css



favicon.png

working directory

index.html

변경사항 추적 중..
(stage)



git

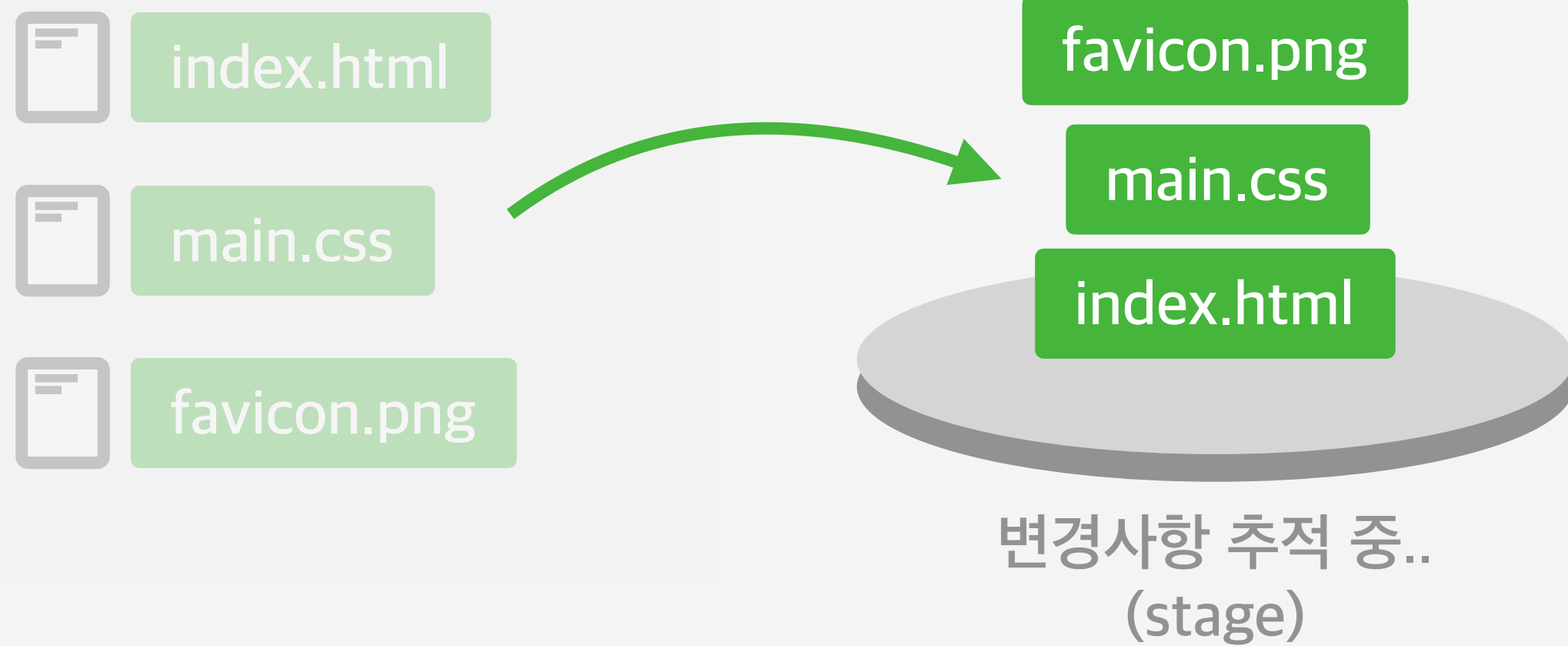


사용자
(local, 로컬)

```
$ git add .
```

모든 파일의 변경사항을 추적하도록 지정.

master



사용자
(local, 로컬)

\$

master



index.html



main.css



favicon.png

favicon.png

main.css

생략

변경사항 추적 중..
(stage)



사용자
(local, 로컬)

```
$ git commit -m '프로젝트 생성'  
# 메시지(-m)와 함께 버전을 생성.
```

master



index.html



main.css



favicon.png

 프로젝트 생성사용자
(local, 로컬)

\$

master



index.html



main.css



favicon.png



main.js

파일 생성



프로젝트 생성



git



사용자
(local, 로컬)

```
$ git add .
```

```
# 모든 파일의 변경사항을 추적하도록 지정.
```

master



index.html



main.css



favicon.png



main.js



프로젝트 생성



git

사용자
(local, 로컬)

```
$ git commit -m 'main.js 추가'
# 메시지(-m)와 함께 버전을 생성.
```

master



index.html



main.css



favicon.png



main.js



main.js 추가



프로젝트 생성



git

사용자
(local, 로컬)

\$

master



index.html

수정함



main.css

수정함



favicon.png



main.js



main.js 추가



프로젝트 생성



git



사용자
(local, 로컬)

```
$ git add .
```

```
# 모든 파일의 변경사항을 추적하도록 지정.
```

master



index.html



main.css



favicon.png



main.js



main.js 추가



프로젝트 생성



git

사용자
(local, 로컬)

```
$ git commit -m 'index.html 수정'
# 메시지(-m)와 함께 버전을 생성.
```

master



index.html



main.css



favicon.png



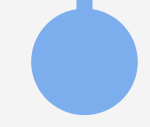
main.js



index.html 수정



main.js 추가



프로젝트 생성



git

사용자
(local, 로컬)

\$

master



index.html



main.css



favicon.png



main.js



사용자
(local, 로컬)

회원 가입



GitHub
(remote, 원격)

\$

master



index.html



main.css



favicon.png



main.js



git



사용자
(local, 로컬)

저장소 생성



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git remote add origin https://github.c...  
# origin이란 별칭으로 원격 저장소를 연결.
```

master

```
git remote -v  
origin https://github.com/passing7by/frontend-study.git (fetch)  
origin https://github.com/passing7by/frontend-study.git (push)
```



index.html



main.css



favicon.png



main.js

index.html 수정
main.js 추가
프로젝트 생성

연결



my-starbucks
(repository, 저장소)



사용자
(local, 로컬)



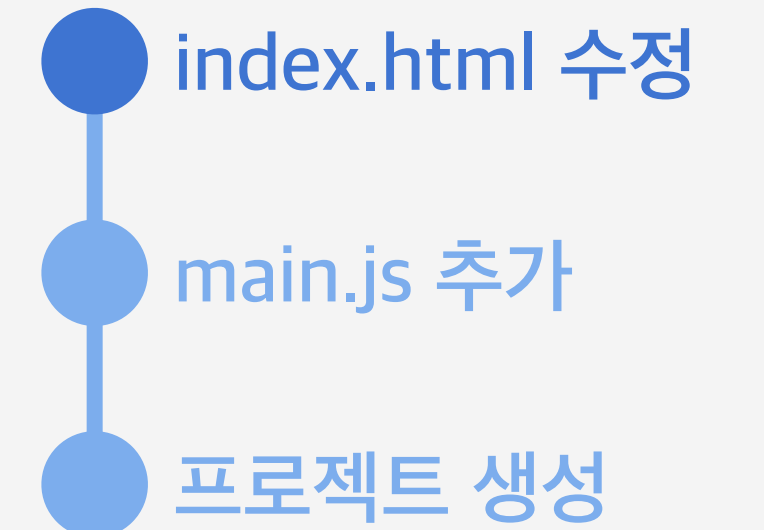
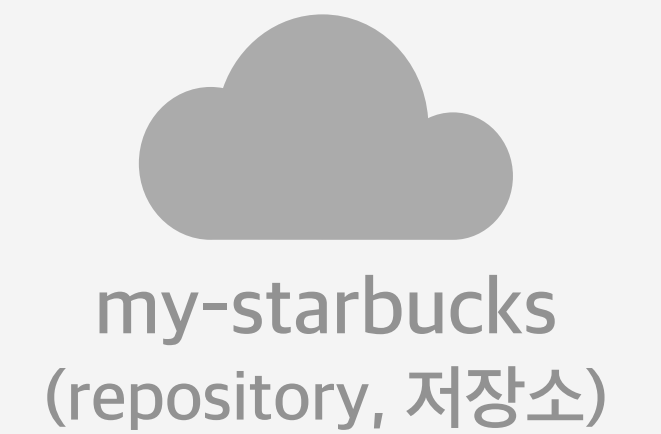
GitHub
(remote, 원격)

~~main~~
\$ git push origin master
origin이란 별칭의 원격 저장소로 버전 내역 전송.

master



전송



사용자
(local, 로컬)



GitHub
(remote, 원격)

```
$ git branch signin  
# 원하는 이름(signin)의 브랜치를 생성.
```

master

signin



index.html



main.css



favicon.png



main.js



index.html 수정



main.js 추가



프로젝트 생성



my-starbucks
(repository, 저장소)



index.html 수정



main.js 추가



프로젝트 생성



사용자
(local, 로컬)



GitHub
(remote, 원격)


```
$ git checkout signin  
# 생성된 브랜치(signin)로 이동.
```

master

signin



index.html



main.css



favicon.png



main.js



index.html 수정



main.js 추가



프로젝트 생성



my-starbucks
(repository, 저장소)



index.html 수정



main.js 추가



프로젝트 생성



사용자
(local, 로컬)



GitHub
(remote, 원격)

\$

master

signin



index.html



main.css

수정함



favicon.png



main.js



signin.html

파일 생성



index.html 수정



main.js 추가



프로젝트 생성



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



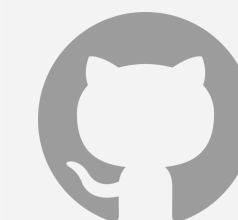
index.html 수정



main.js 추가



프로젝트 생성



GitHub
(remote, 원격)

```
$ git add .
```

모든 파일의 변경사항을 추적하도록 지정.

master

signin



index.html



main.css



favicon.png



main.js



signin.html



index.html 수정



main.js 추가



프로젝트 생성



my-starbucks
(repository, 저장소)



index.html 수정



main.js 추가



프로젝트 생성



사용자
(local, 로컬)



GitHub
(remote, 원격)

```
$ git commit -m '로그인 추가'
# 메시지(-m)와 함께 버전을 생성.
```

master

signin



index.html



main.css



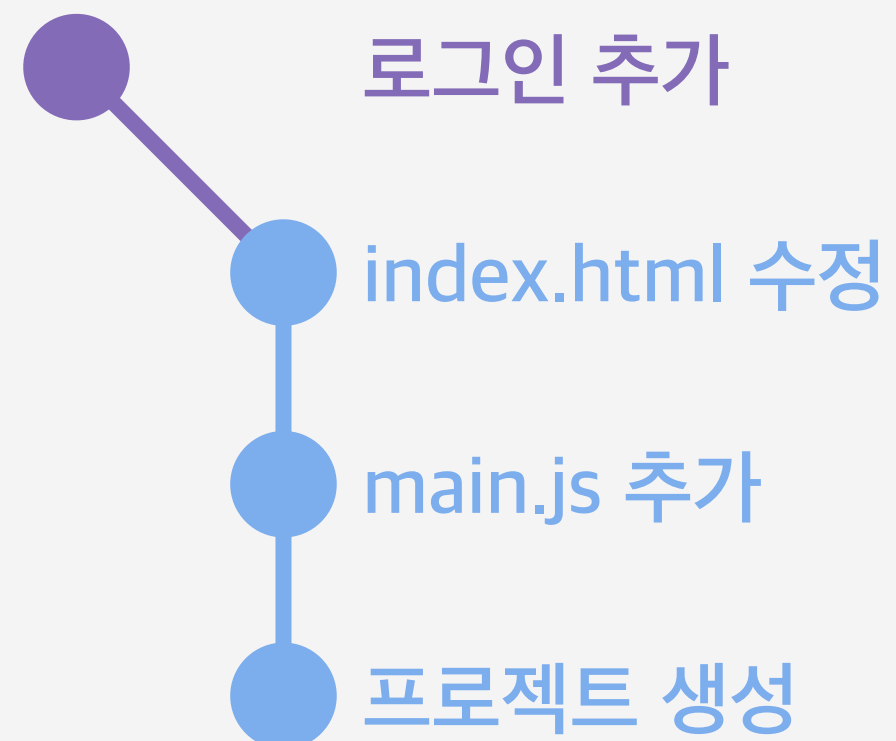
favicon.png



main.js



signin.html



사용자
(local, 로컬)



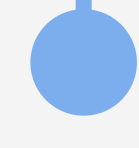
my-starbucks
(repository, 저장소)



index.html 수정



main.js 추가



프로젝트 생성



GitHub
(remote, 원격)

```
$ git checkout master  
# master 브랜치로 이동.
```

master

signin



index.html



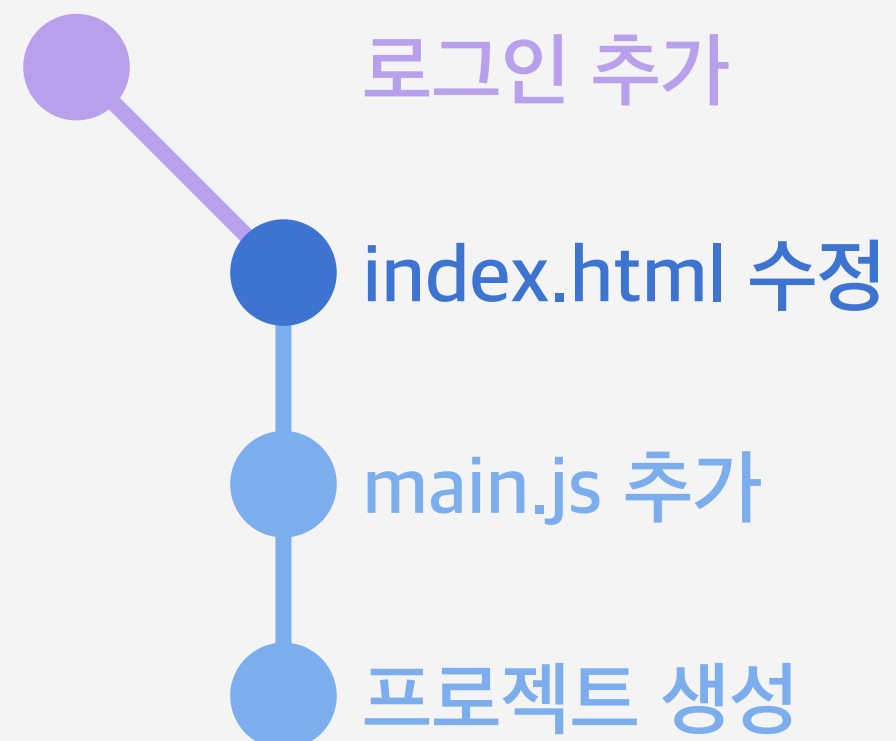
main.css



favicon.png



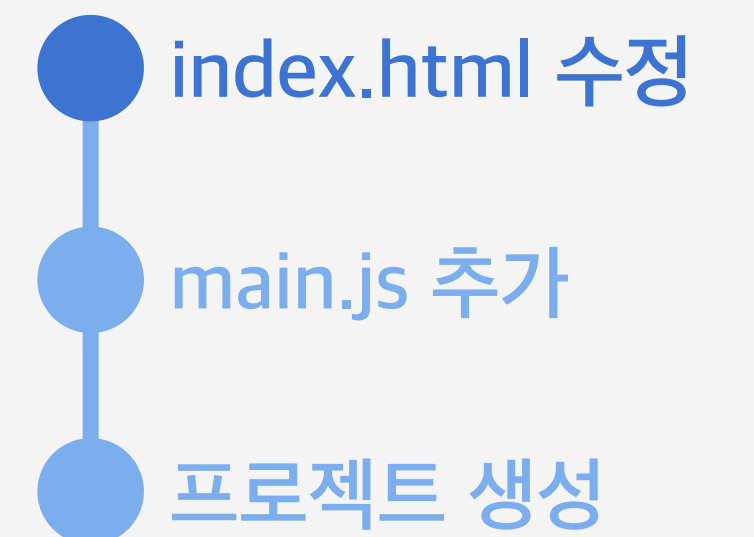
main.js



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

\$

master

signin



index.html



main.css

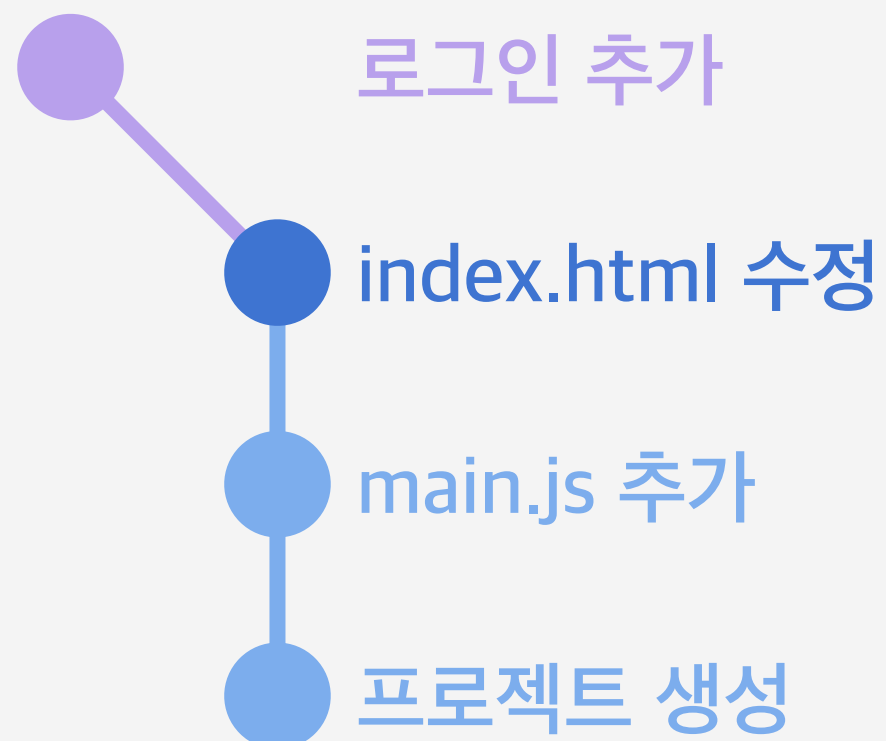
수정함



favicon.png



main.js



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git add .
```

모든 파일의 변경사항을 추적하도록 지정.

master

signin



index.html



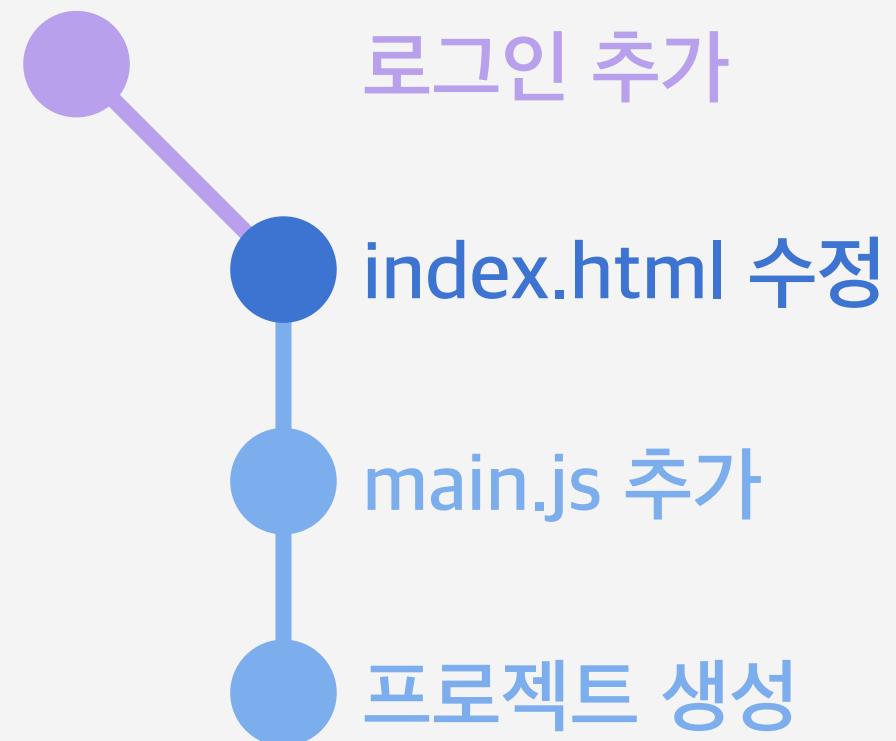
main.css



favicon.png



main.js



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git commit -m '스타일 수정'
# 메시지(-m)와 함께 버전을 생성.
```

master

signin



index.html



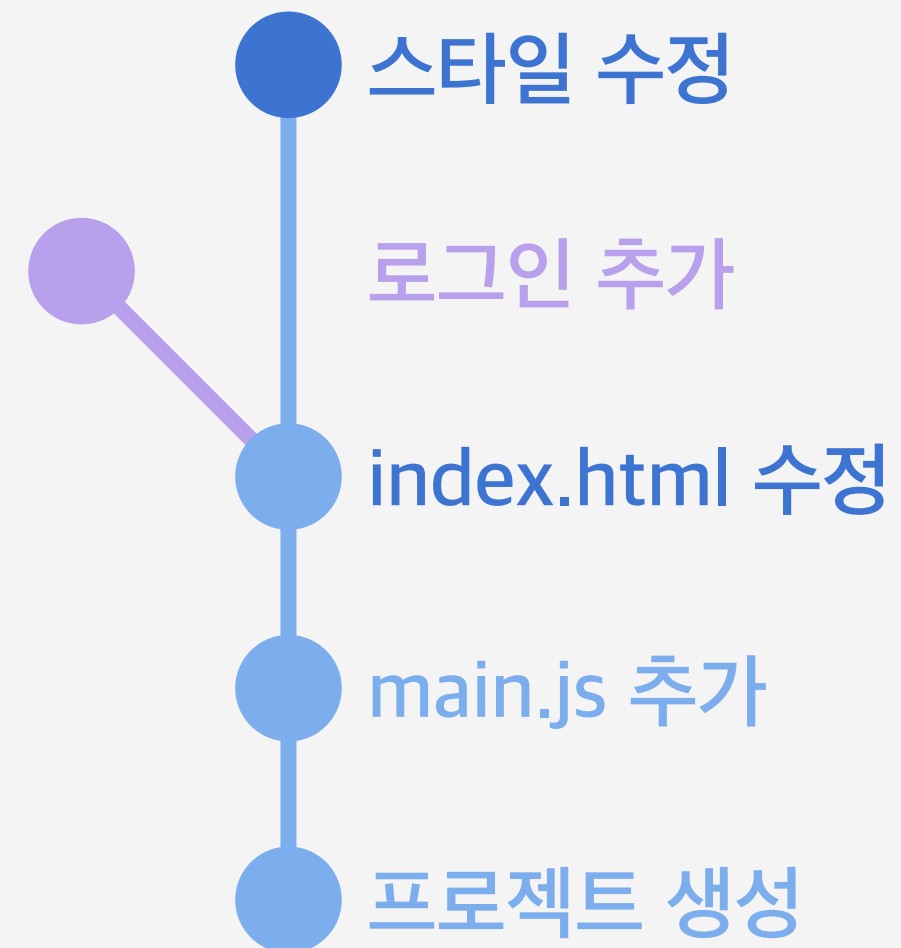
main.css



favicon.png



main.js



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



index.html 수정



main.js 추가



프로젝트 생성



GitHub
(remote, 원격)


```
$ git push origin master
```

origin이란 별칭의 원격 저장소로 버전 내역 전송.

master

signin



index.html



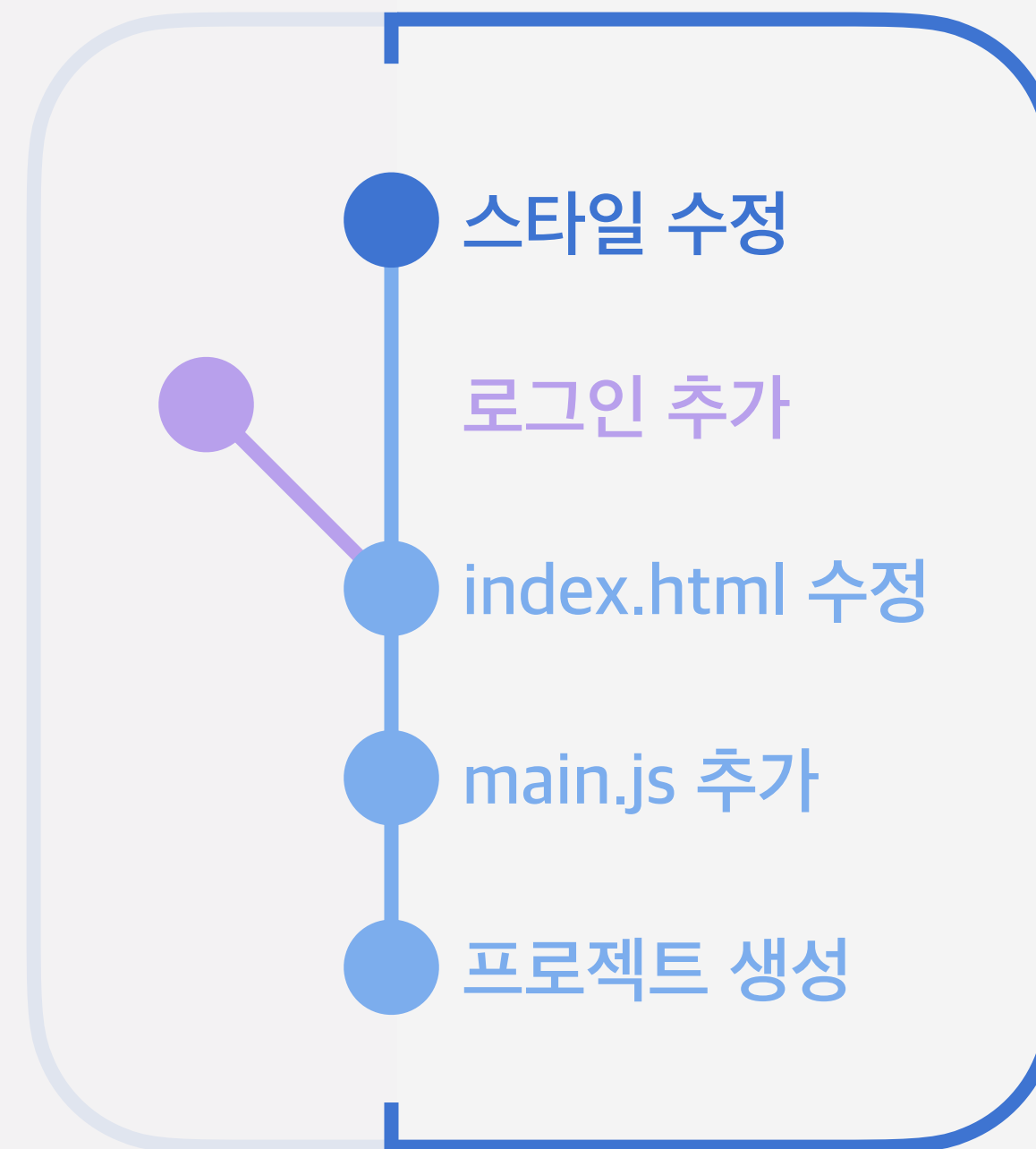
main.css



favicon.png



main.js



전송

my-starbucks
(repository, 저장소)



사용자
(local, 로컬)



GitHub
(remote, 원격)

```
$ git checkout signin  
# signin 브랜치로 이동.
```

master

signin



index.html



main.css



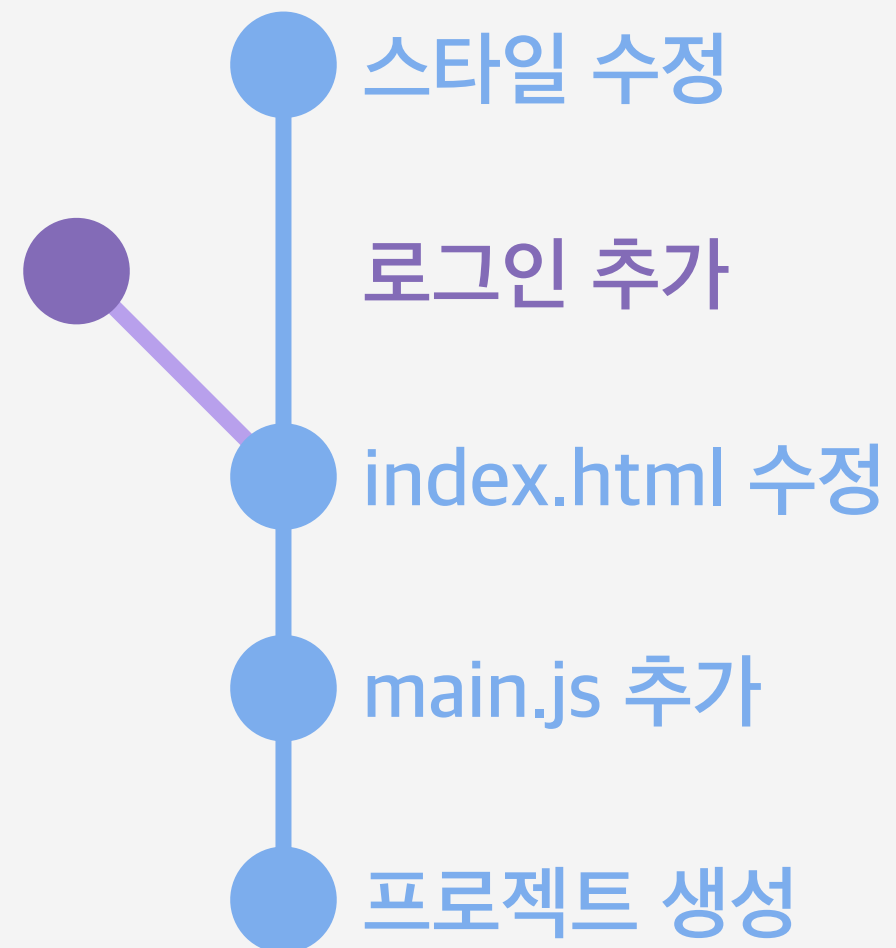
favicon.png



main.js



signin.html



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git push origin signin  
# origin이란 별칭의 원격 저장소로 버전 내역 전송.
```

master

signin



index.html



main.css



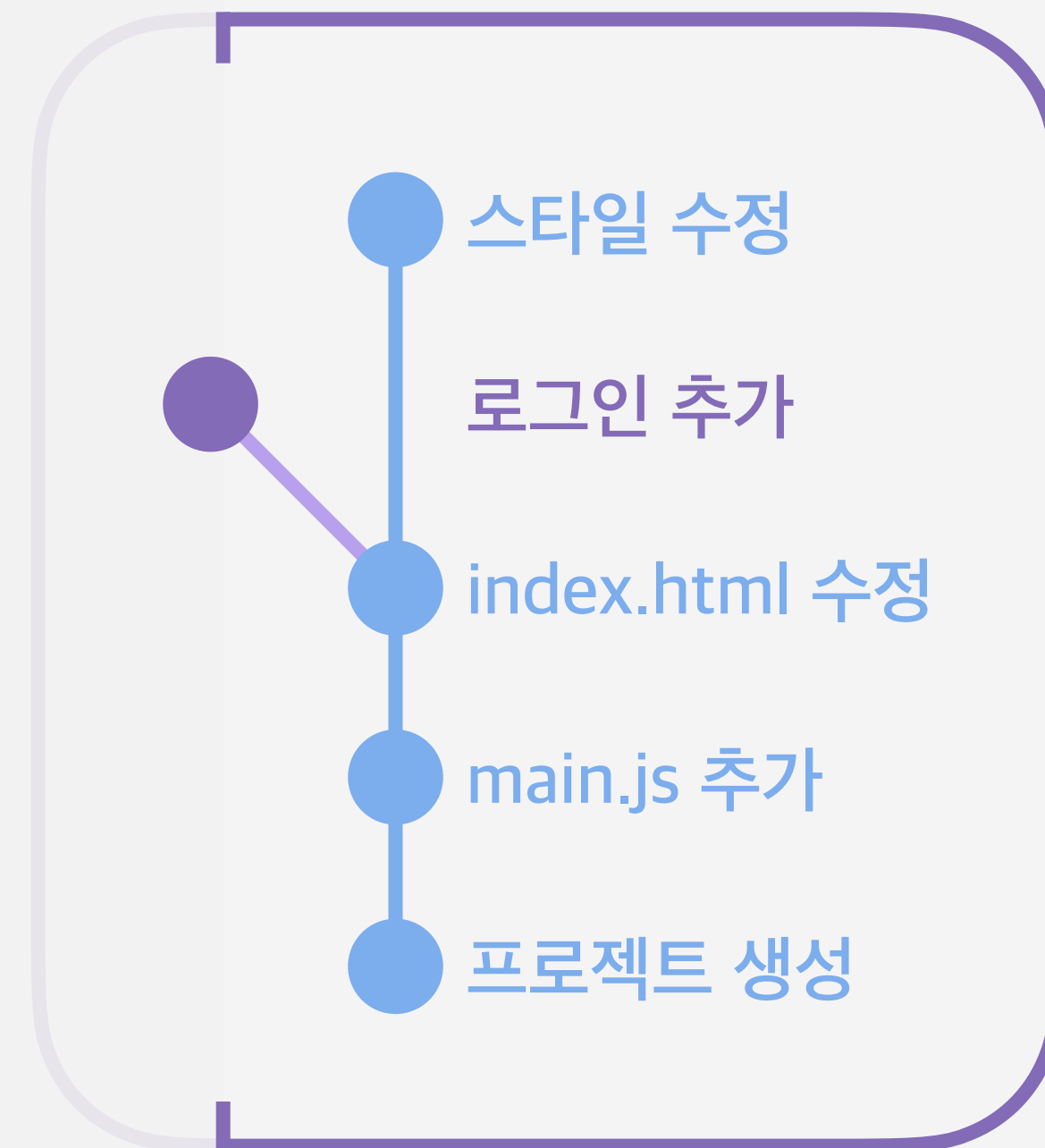
favicon.png



main.js

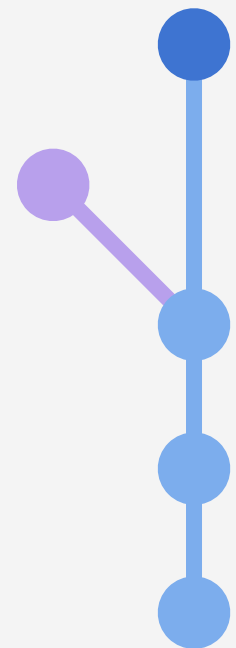


signin.html



전송

my-starbucks
(repository, 저장소)



사용자
(local, 로컬)



GitHub
(remote, 원격)

\$



바탕화면



git

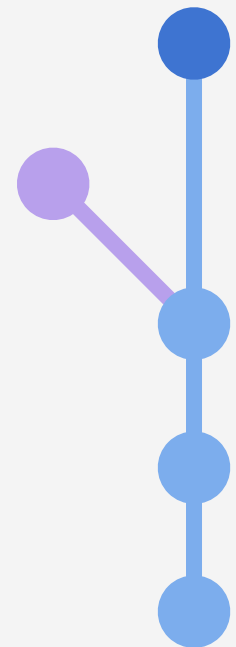


사용자
(local, 로컬)

버전 관리(VCS)



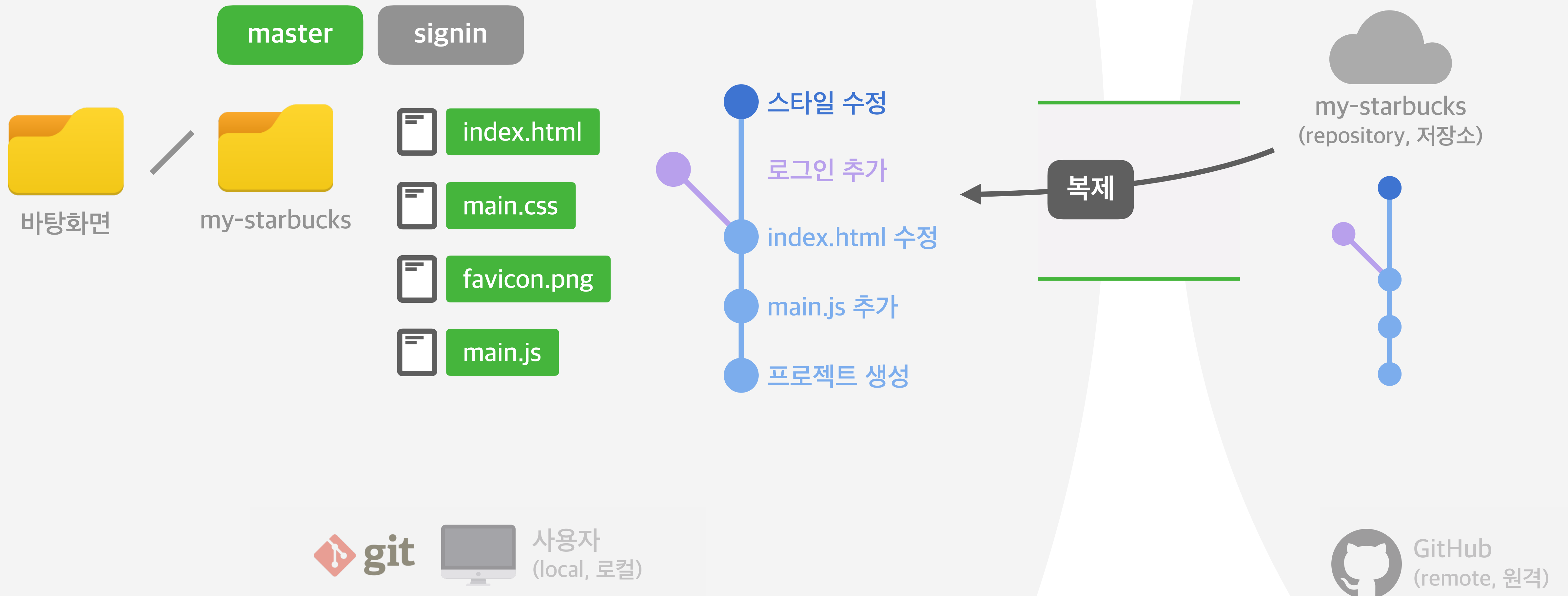
my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git clone https://github.com/ParkYoun...
```

원격 저장소에서 프로젝트 가져오기(복제)



```
$ git checkout signin  
# signin 브랜치로 이동.
```

master

signin



index.html



main.css



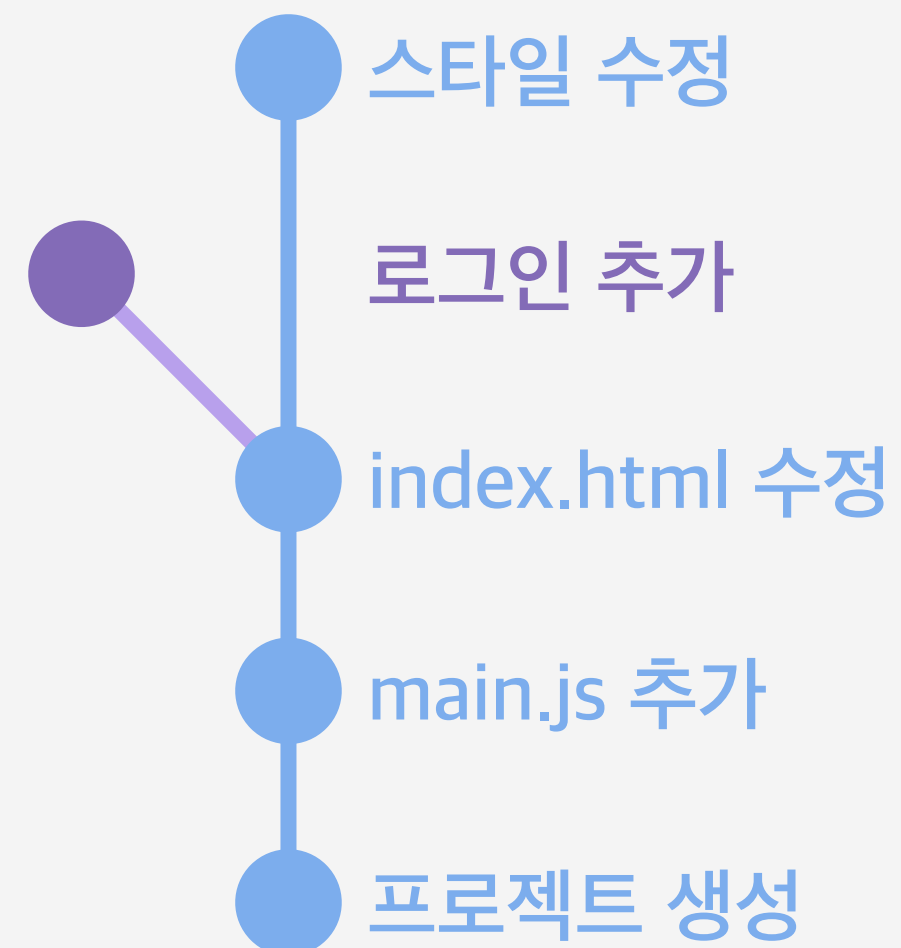
favicon.png



main.js



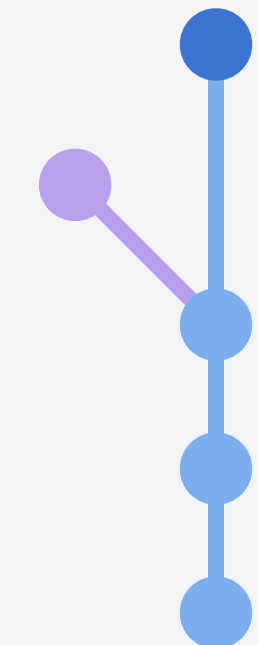
signin.html



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



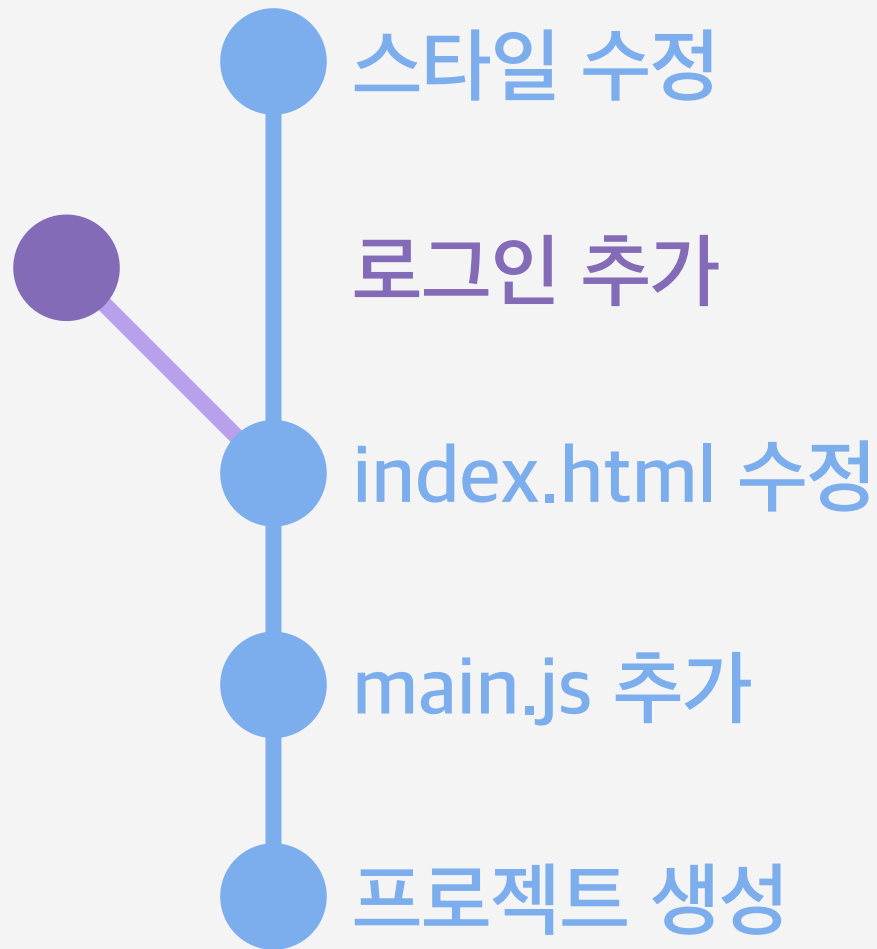
GitHub
(remote, 원격)

```
$
```

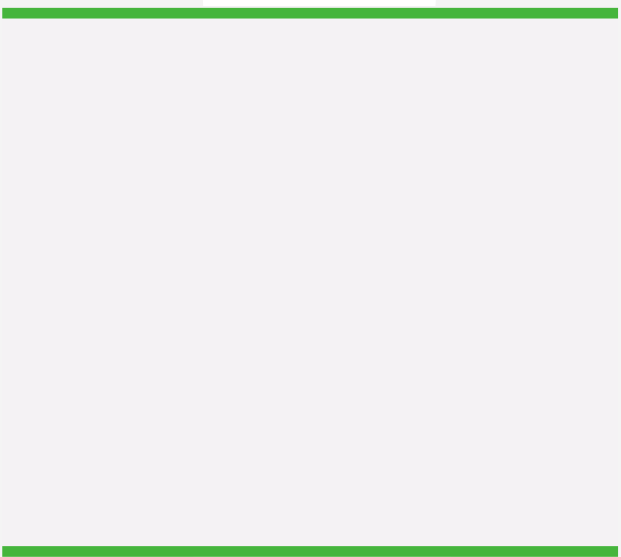
master **signin**



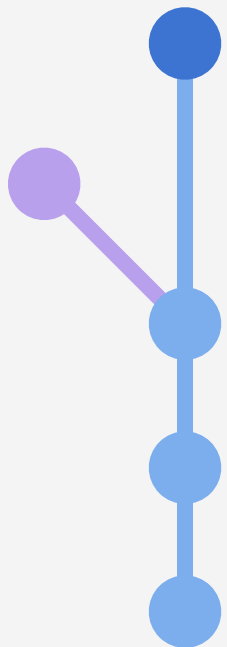
-  index.html
-  main.css
-  favicon.png
-  **main.js** **수정함**
-  signin.html



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git add .
```

모든 파일의 변경사항을 추적하도록 지정.

master

signin



index.html



main.css



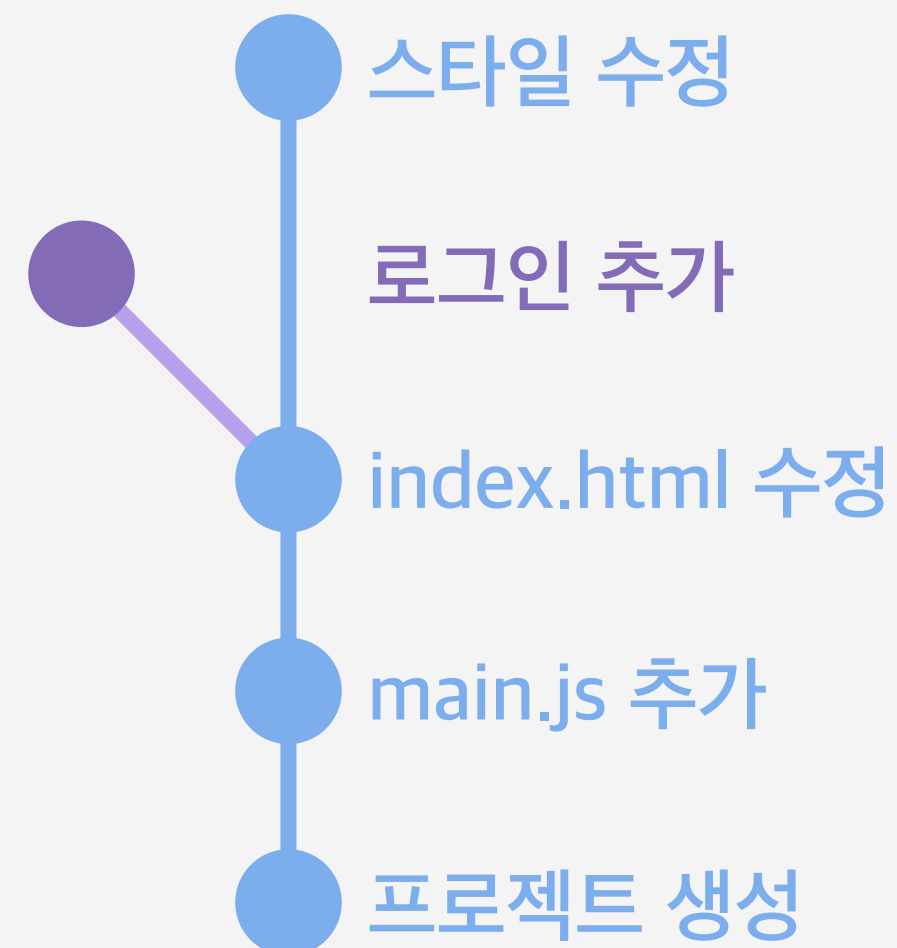
favicon.png



main.js



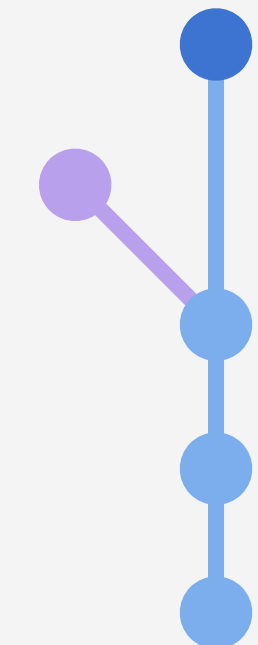
signin.html



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)


```
$ git commit -m '로직 수정'
# 메시지(-m)와 함께 버전을 생성.
```

master

signin



index.html



main.css



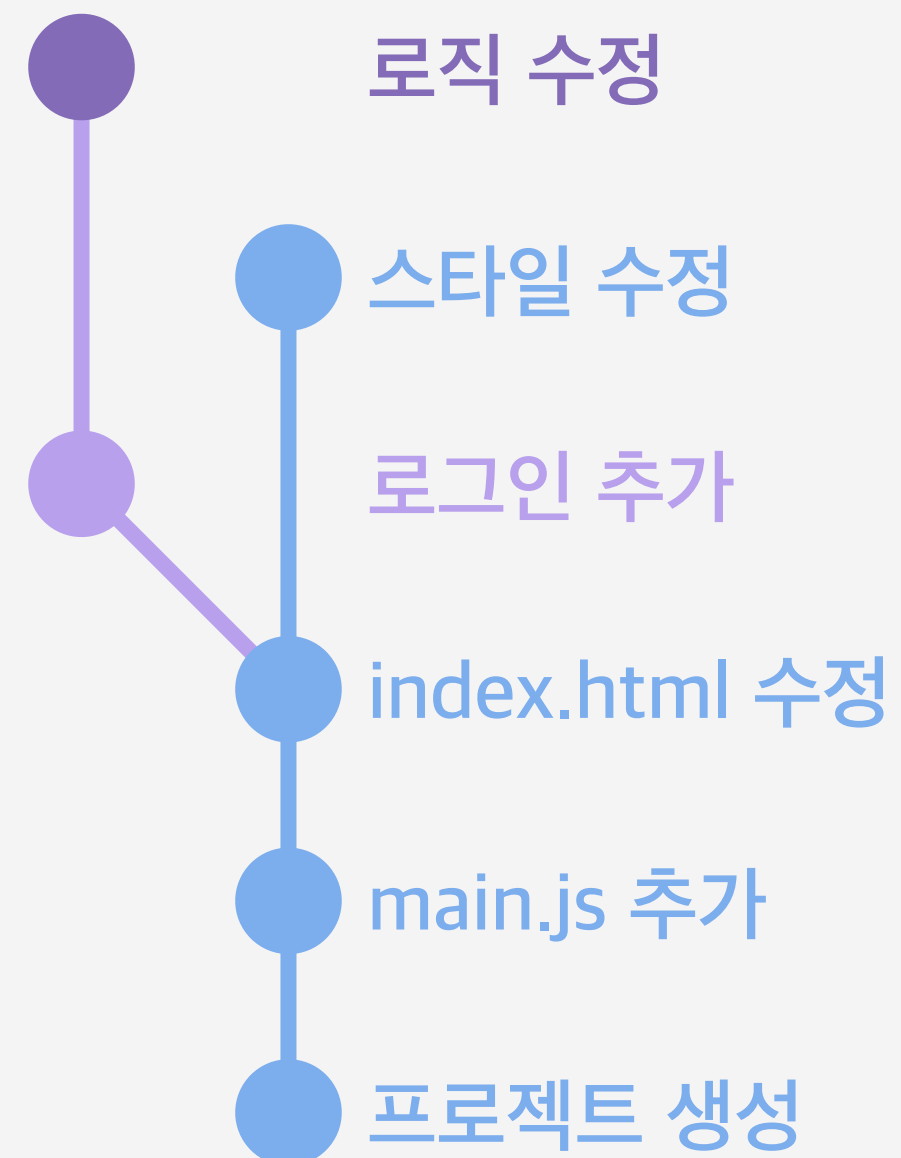
favicon.png



main.js



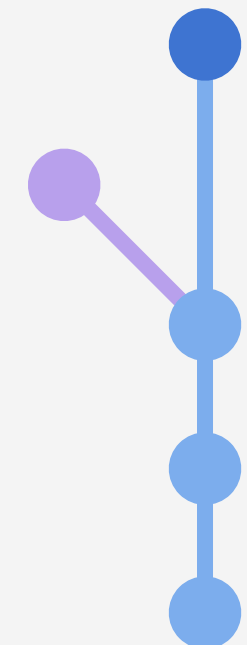
signin.html



사용자
(local, 로컬)



my-starbucks
(repository, 저장소)



GitHub
(remote, 원격)

```
$ git push origin signin
```

origin이란 별칭의 원격 저장소로 버전 내역 전송.

master

signin



index.html



main.css



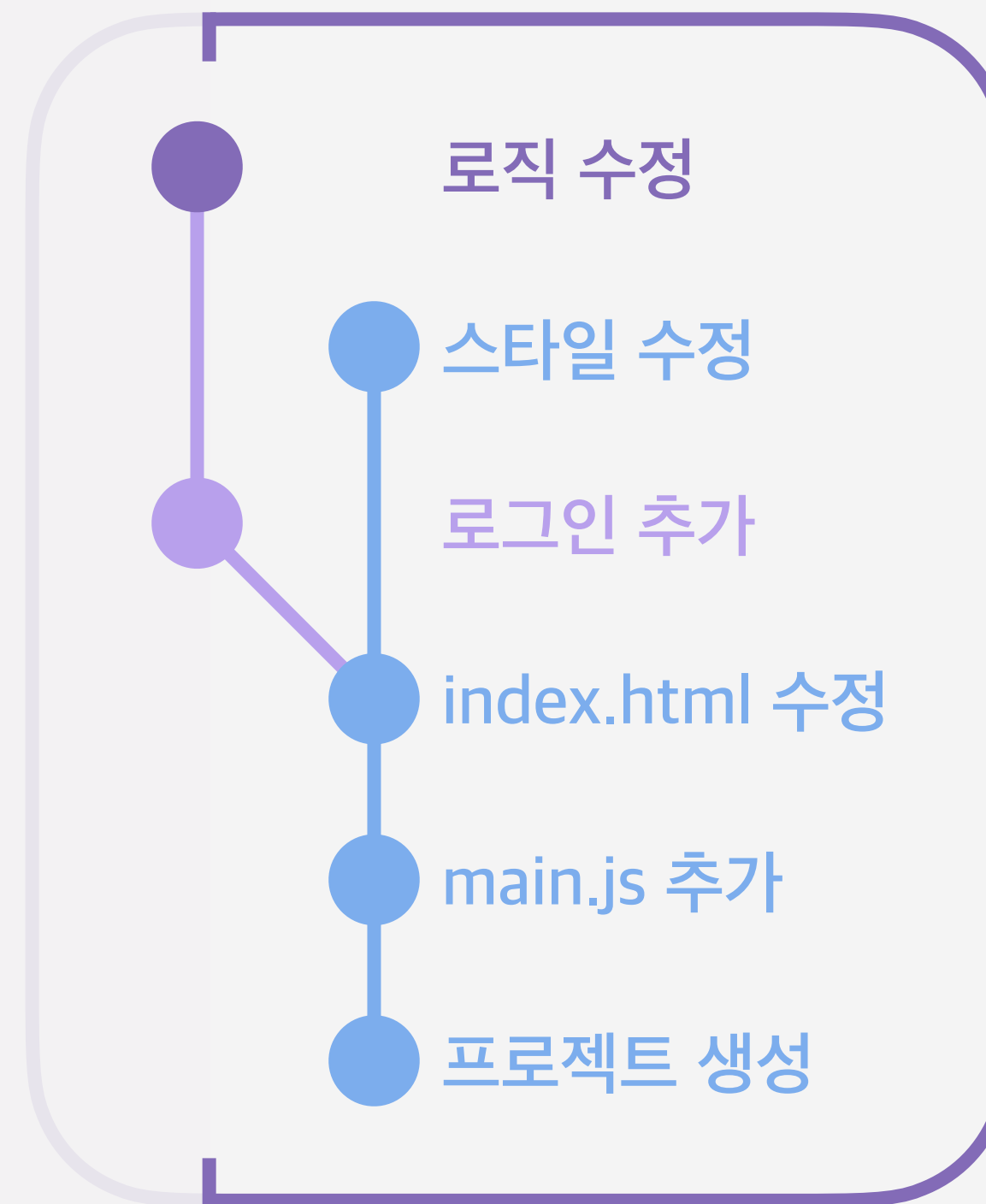
favicon.png



main.js



signin.html

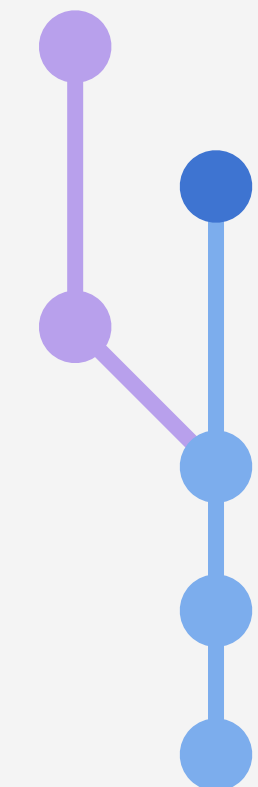


전송

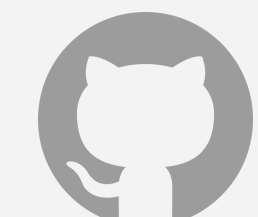
버전 관리(VCS)



my-starbucks
(repository, 저장소)



사용자
(local, 로컬)



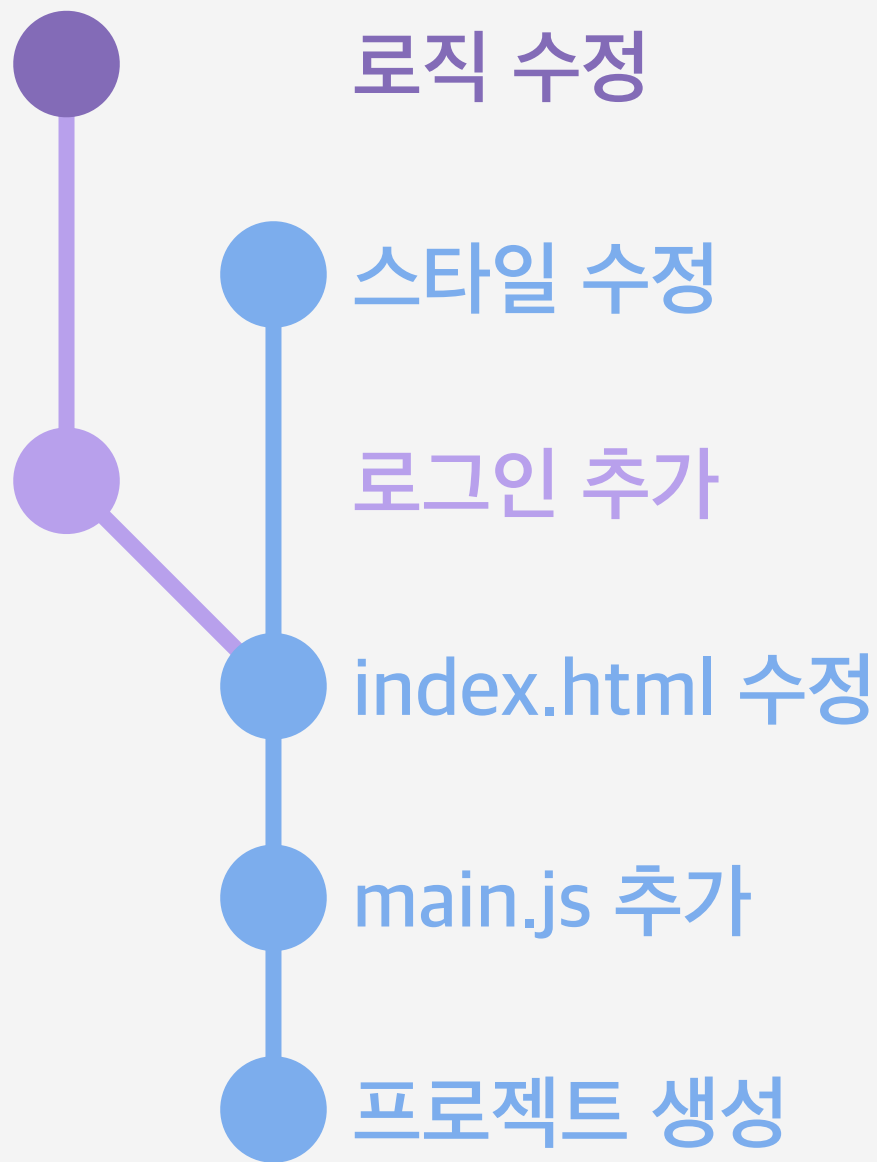
GitHub
(remote, 원격)

```
$
```

master **signin**



- index.html
- main.css
- favicon.png
- main.js
- signin.html

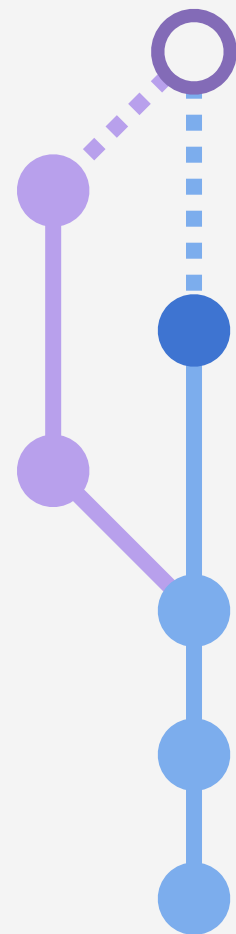


사용자
(local, 로컬)



my-starbucks
(repository, 저장소)

Pull Request 생성



GitHub
(remote, 원격)

\$

master

signin



index.html



main.css



favicon.png



main.js



signin.html



로직 수정

스타일 수정

로그인 추가

index.html 수정

main.js 추가

프로젝트 생성



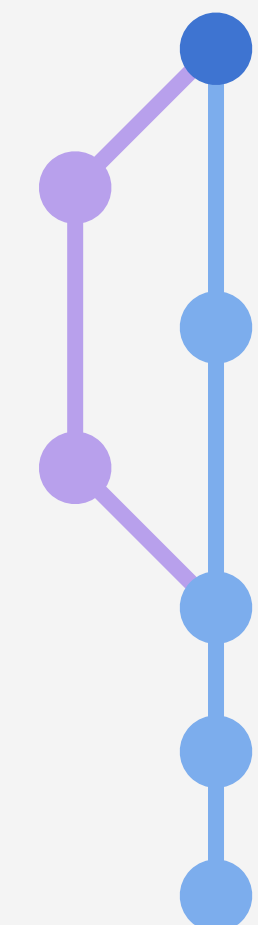
사용자
(local, 로컬)

버전 관리(VCS)



my-starbucks
(repository, 저장소)

병합(Merge)



GitHub
(remote, 원격)

```
$ git checkout master  
# master 브랜치로 이동.
```

master

signin



index.html



main.css



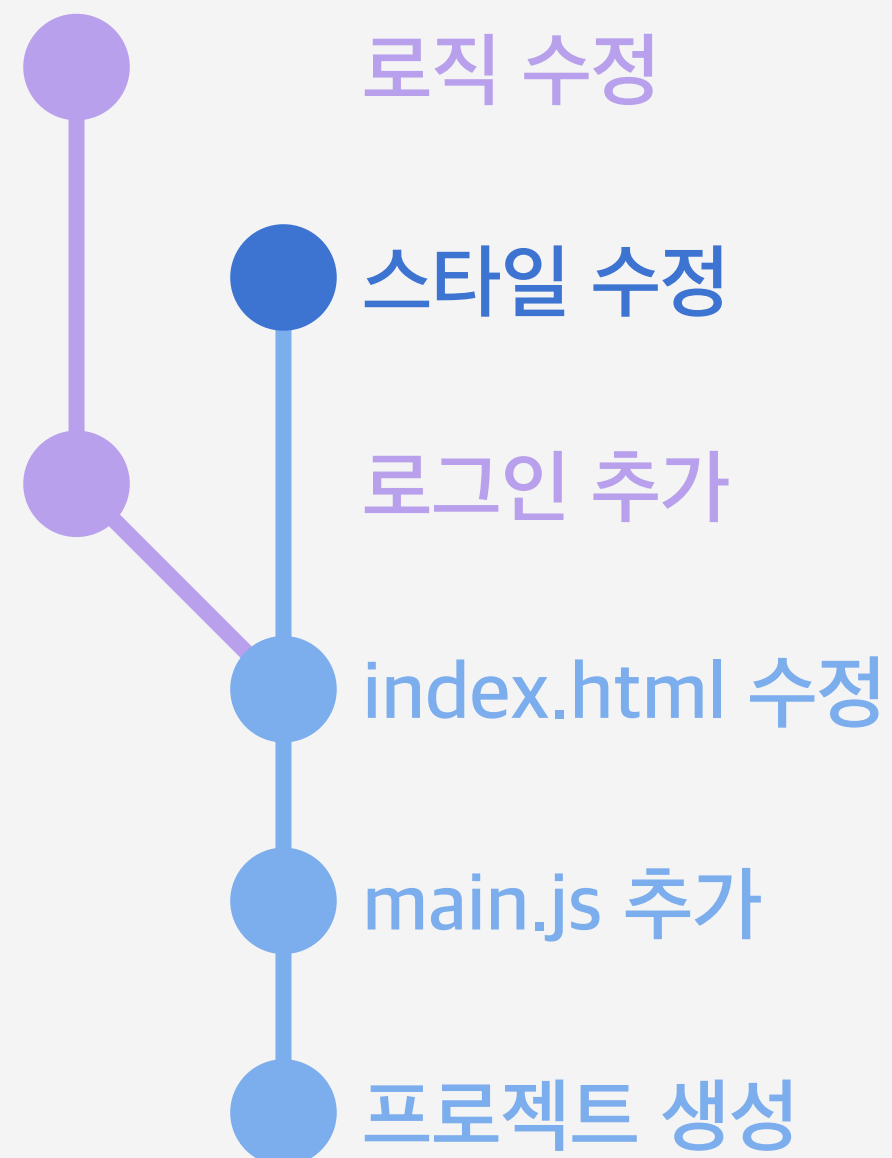
favicon.png



main.js



signin.html



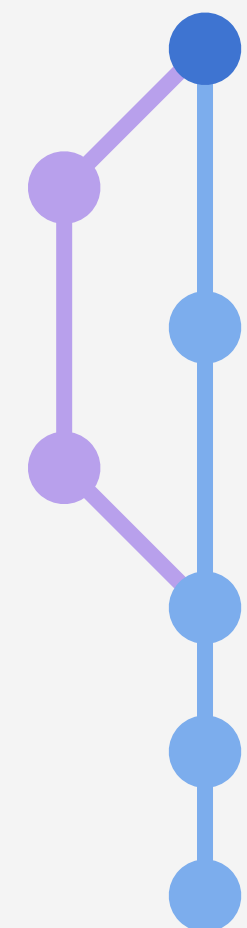
사용자
(local, 로컬)

버전 관리(VCS)



my-starbucks
(repository, 저장소)

병합(Merge)



GitHub
(remote, 원격)

```
$ git pull origin master  
# 원격 저장소의 master 브랜치 내역 당겨오기.
```

master

signin



index.html



main.css



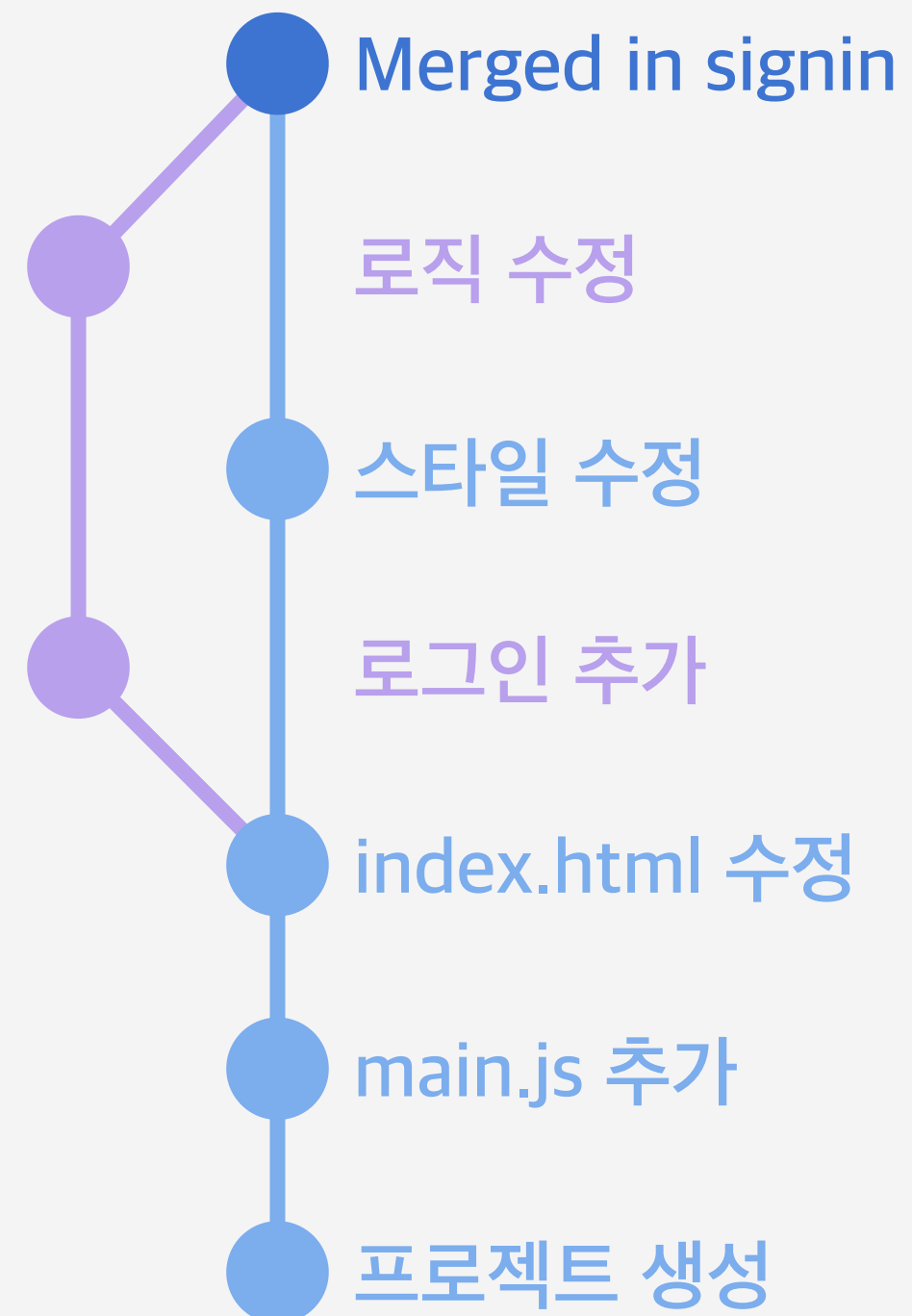
favicon.png



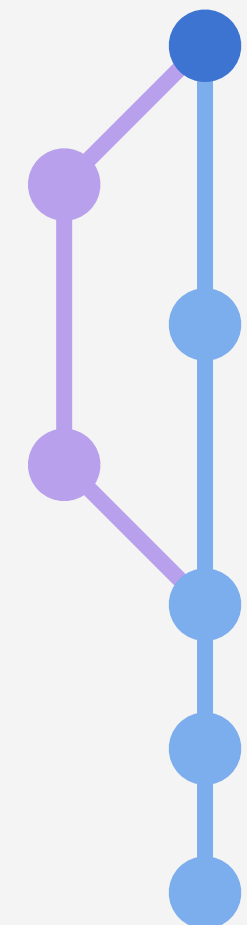
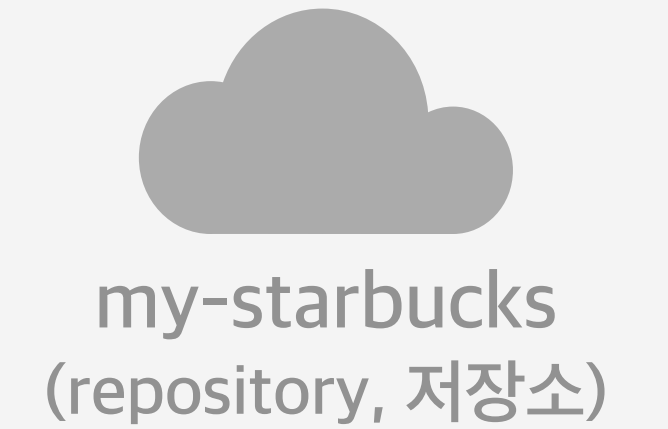
main.js



signin.html



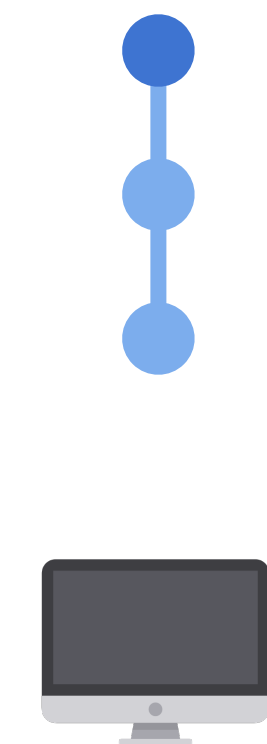
당겨오기



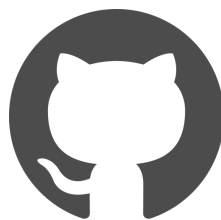
사용자
(local, 로컬)



GitHub
(remote, 원격)



A 개발자



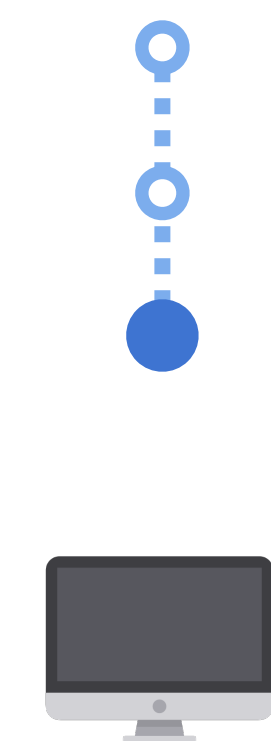
원격 저장소



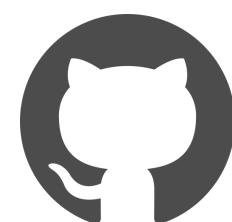
B 개발자

\$

\$



A 개발자



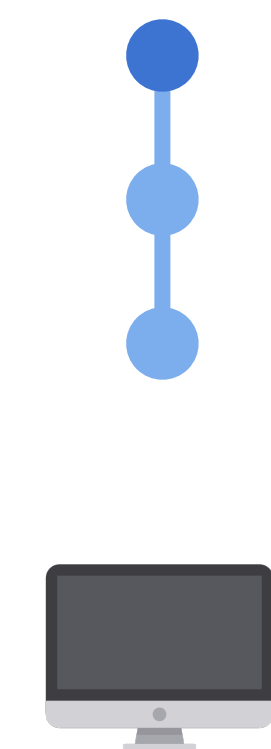
원격 저장소



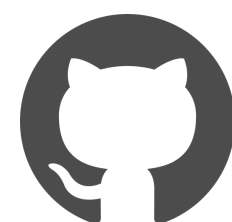
B 개발자

```
$ git reset --hard HEAD~2
```

```
$
```

A 개발자



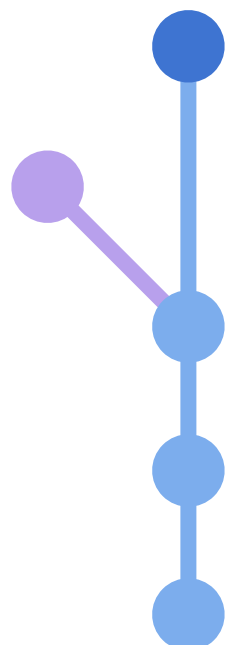
원격 저장소



B 개발자

```
$ git reset --hard ORIG_HEAD
```

```
$
```



A 개발자



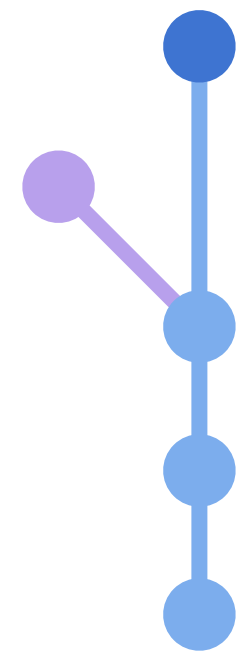
원격 저장소



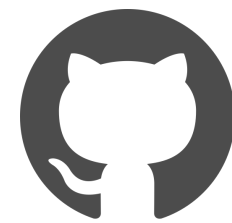
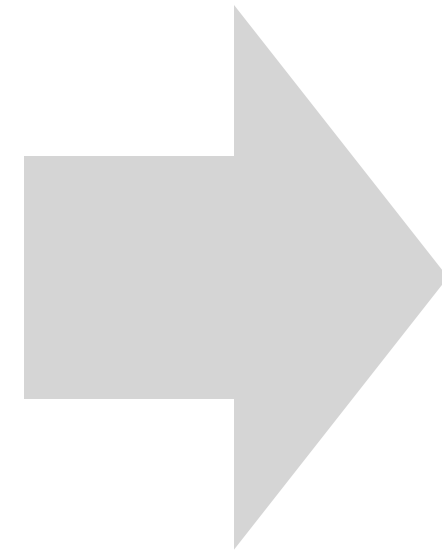
B 개발자

\$

\$



A 개발자



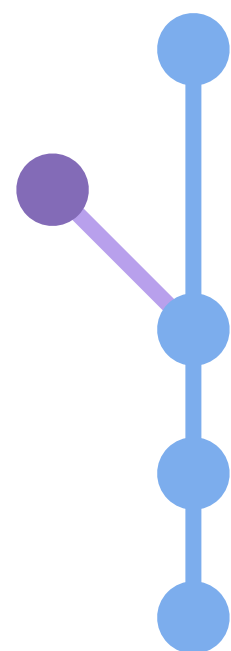
원격 저장소



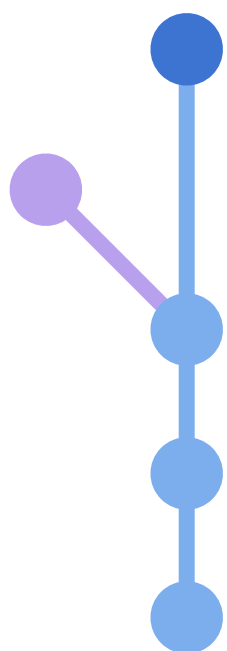
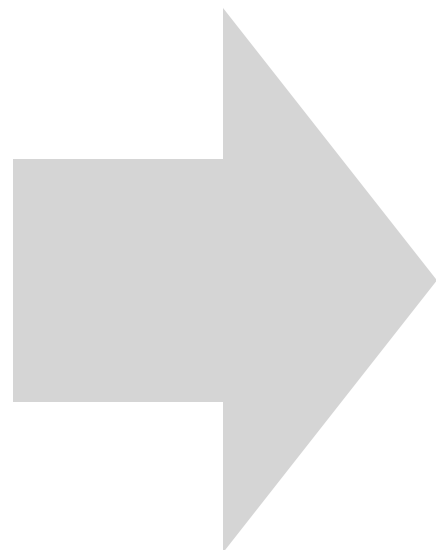
B 개발자

```
$ git push origin master
```

```
$
```



A 개발자



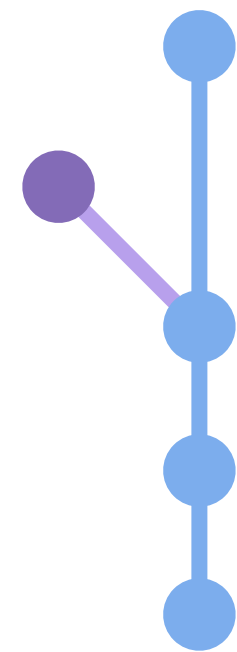
원격 저장소



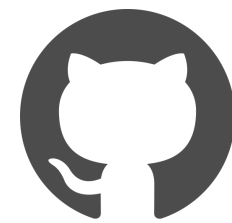
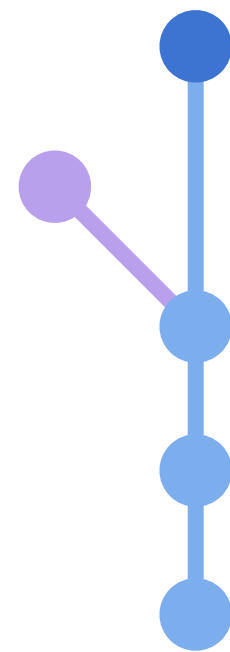
B 개발자

```
$ git push origin purple
```

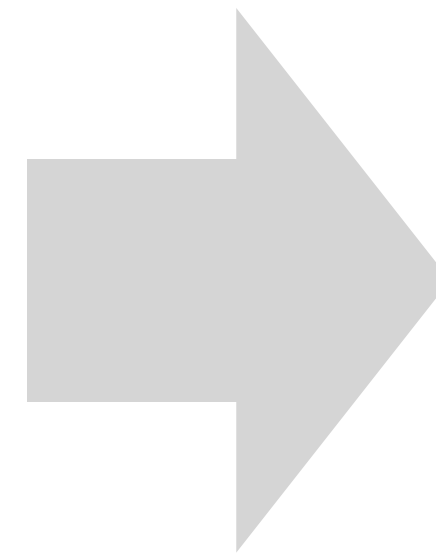
```
$
```



A 개발자



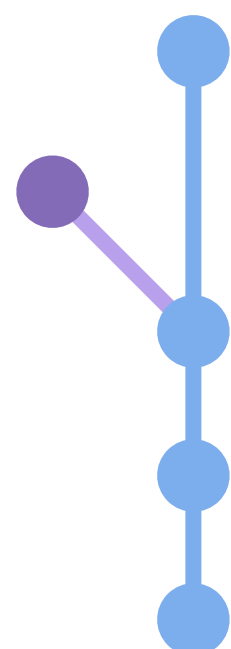
원격 저장소



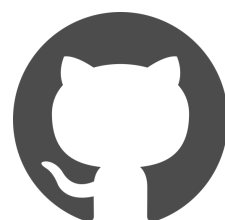
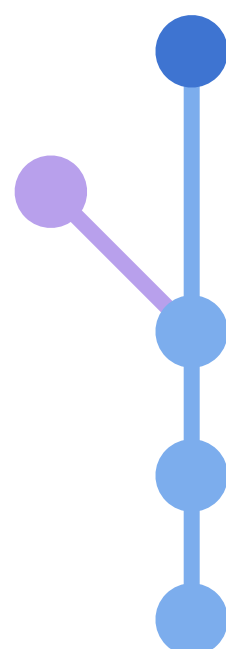
B 개발자

\$

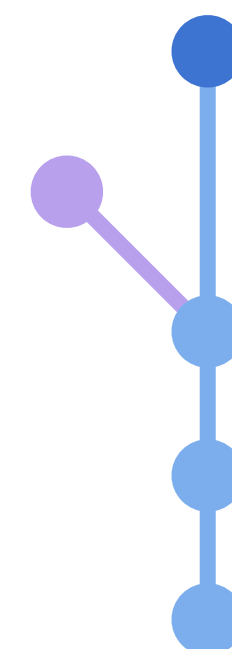
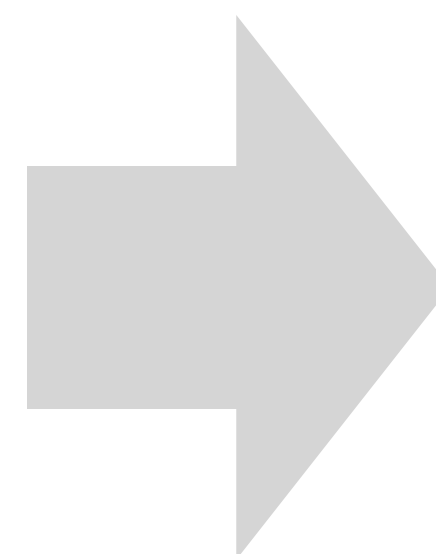
```
$ git clone https://github.c...
```



A 개발자



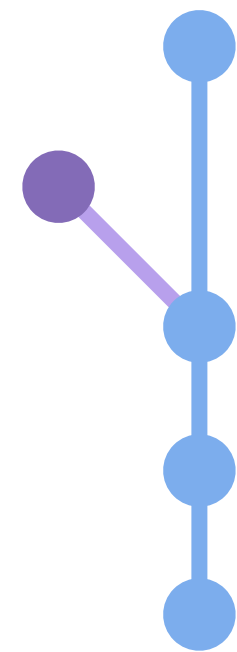
원격 저장소



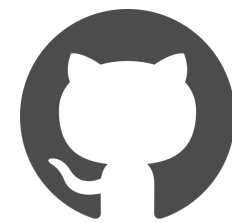
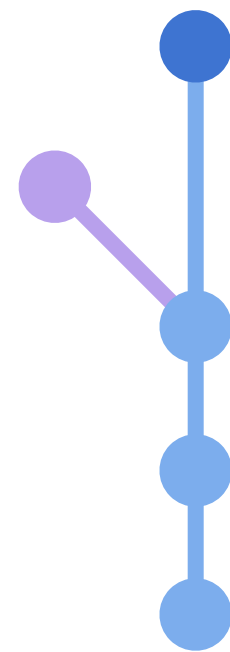
B 개발자

\$

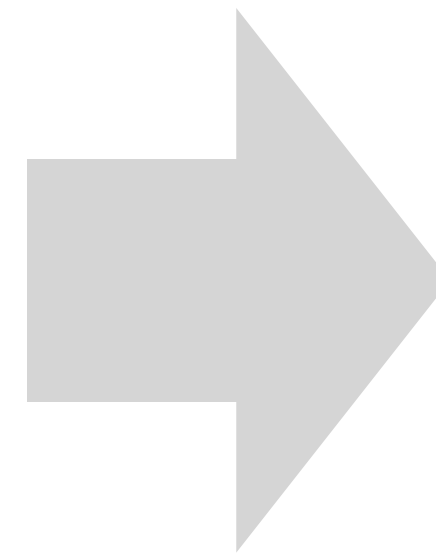
```
$ git checkout -t origin/purple
```



A 개발자



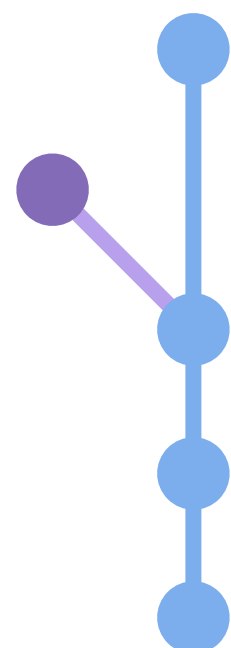
원격 저장소



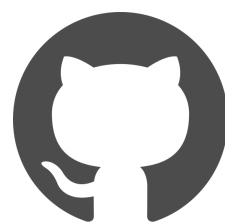
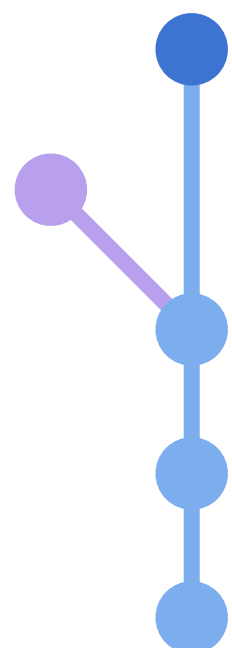
B 개발자

\$

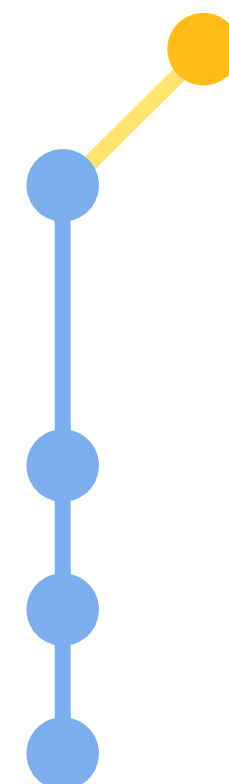
```
$ git branch -d purple
```



A 개발자



원격 저장소



B 개발자

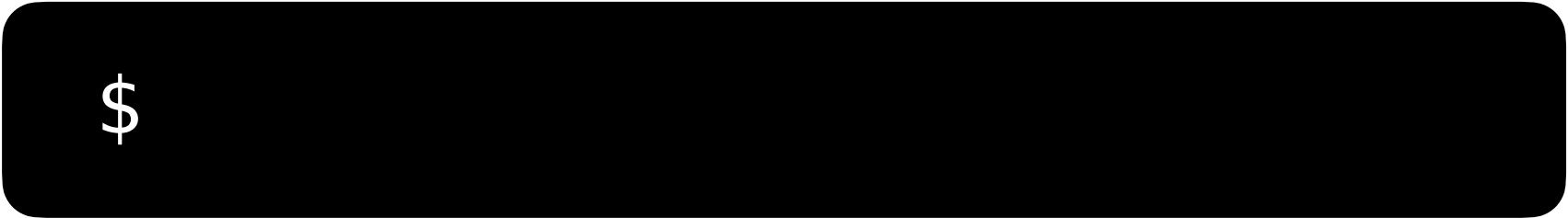
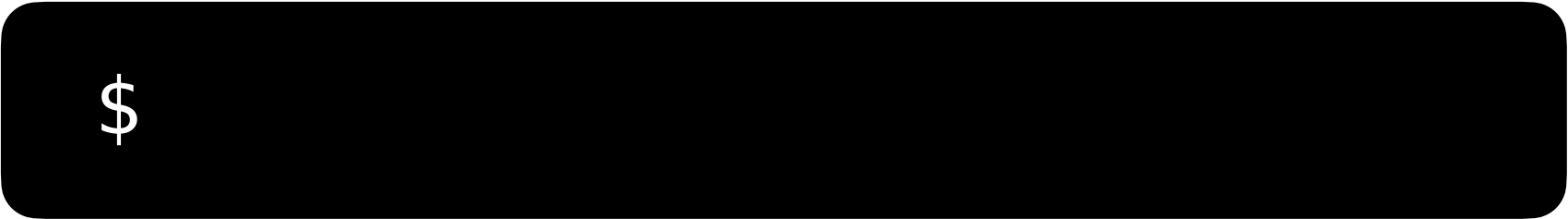
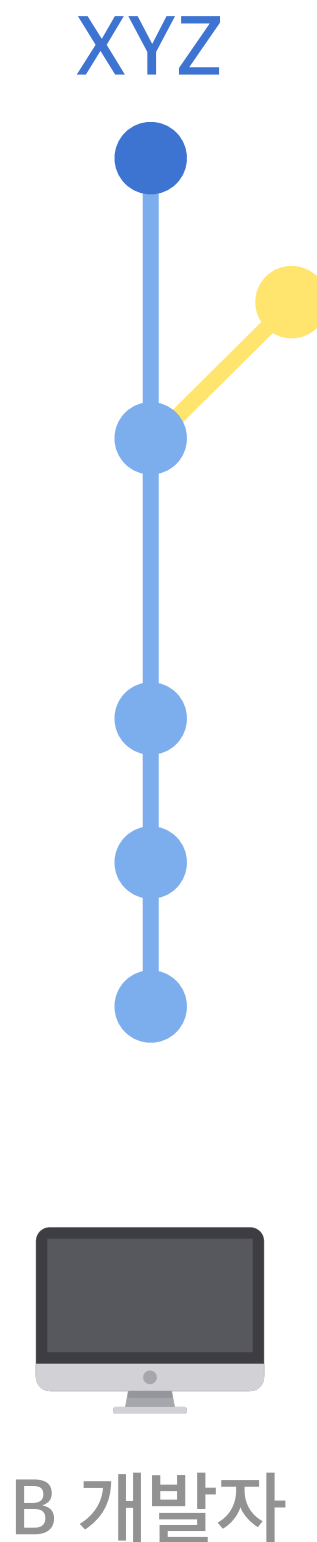
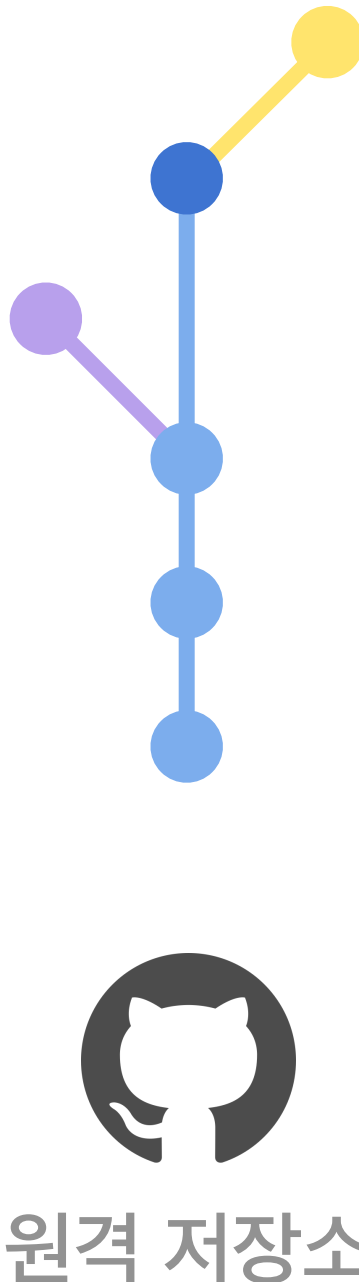
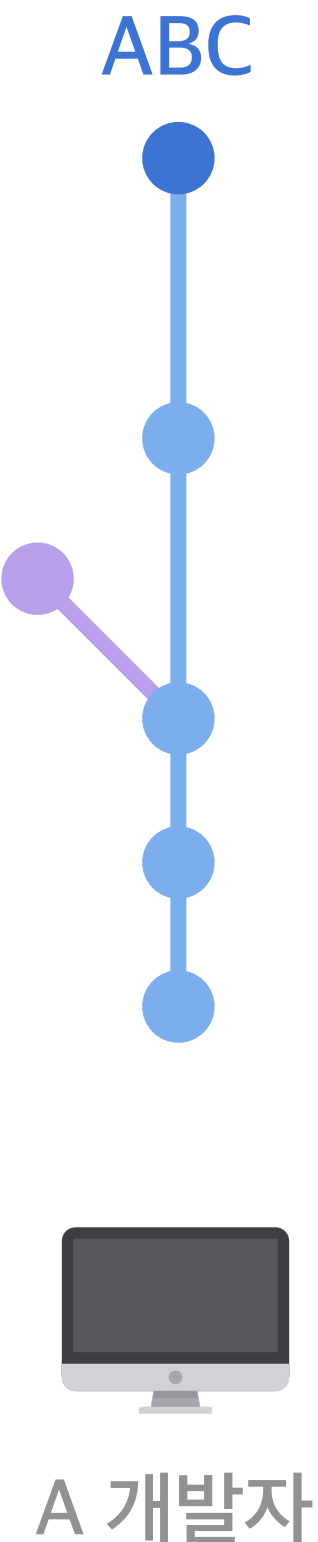
\$

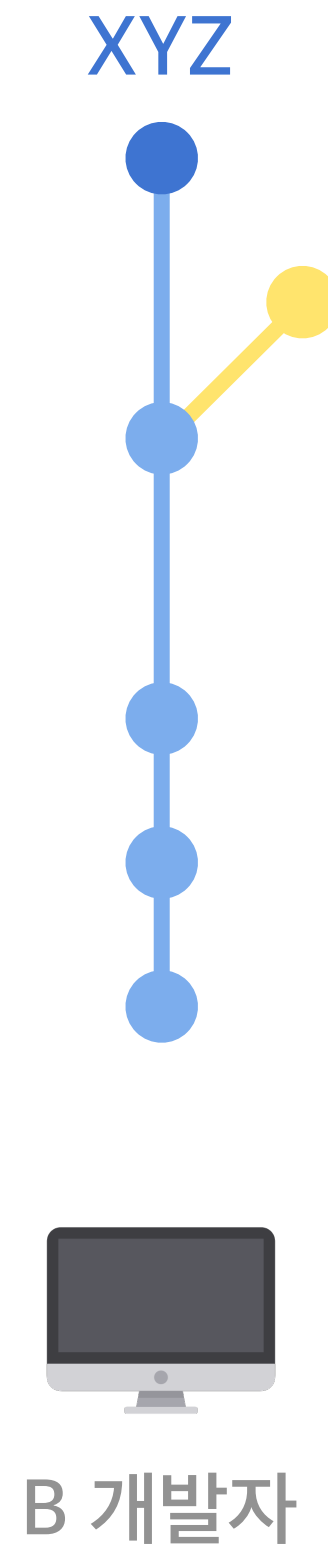
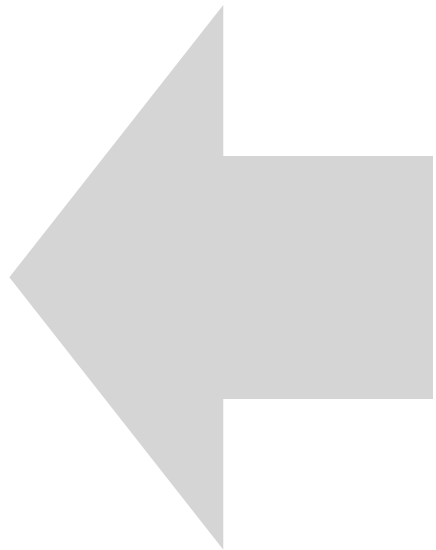
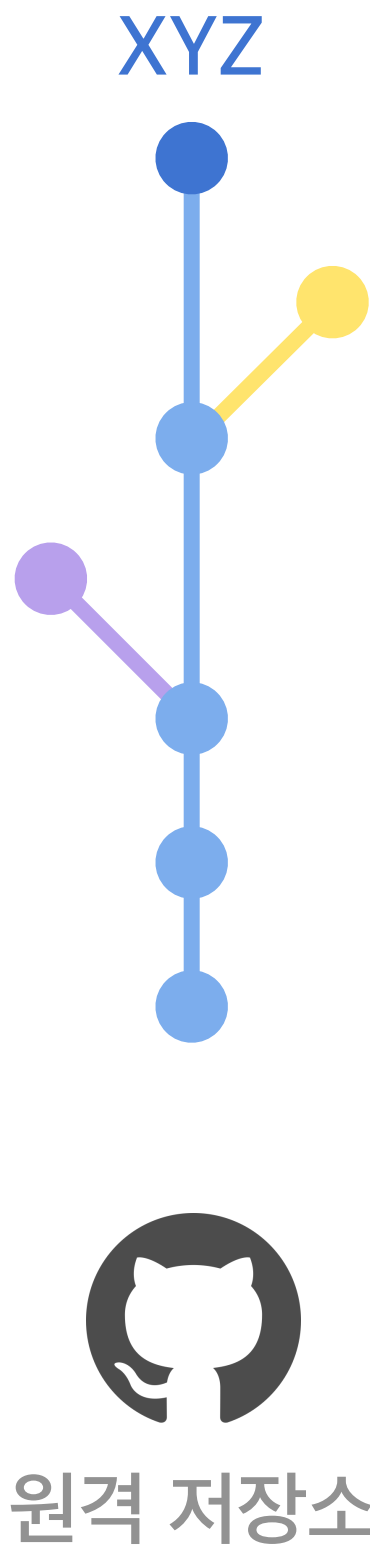
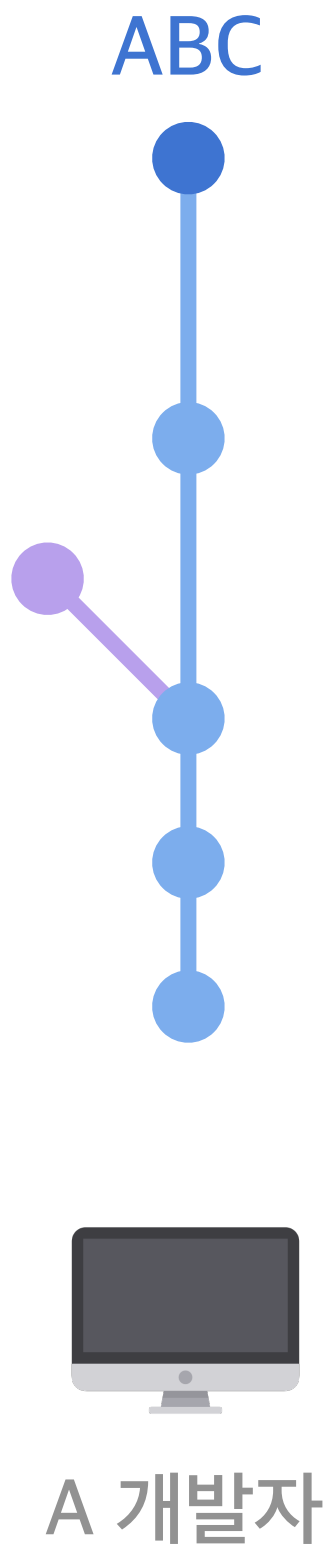
```
$ git checkout -b yellow
```




\$

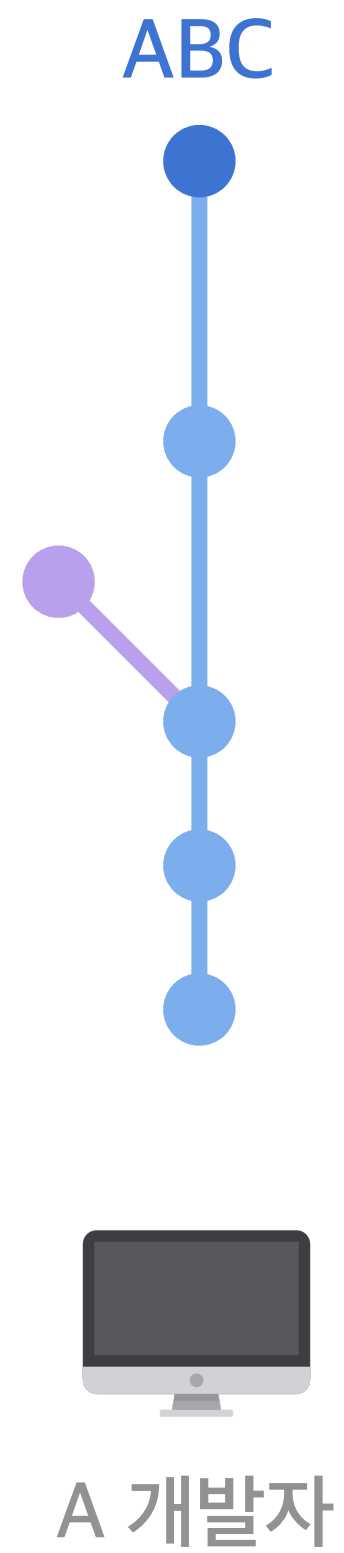
```
$ git push origin yellow
```



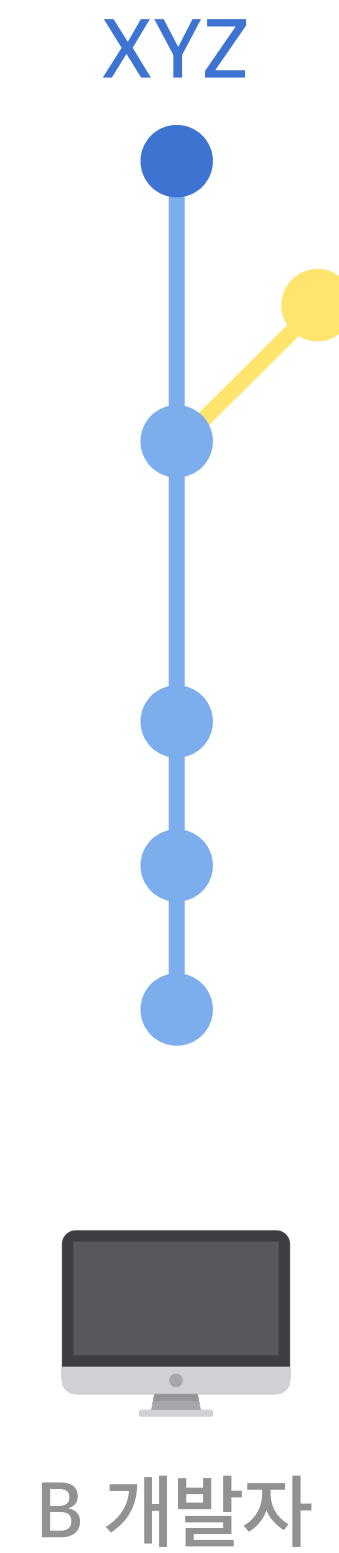
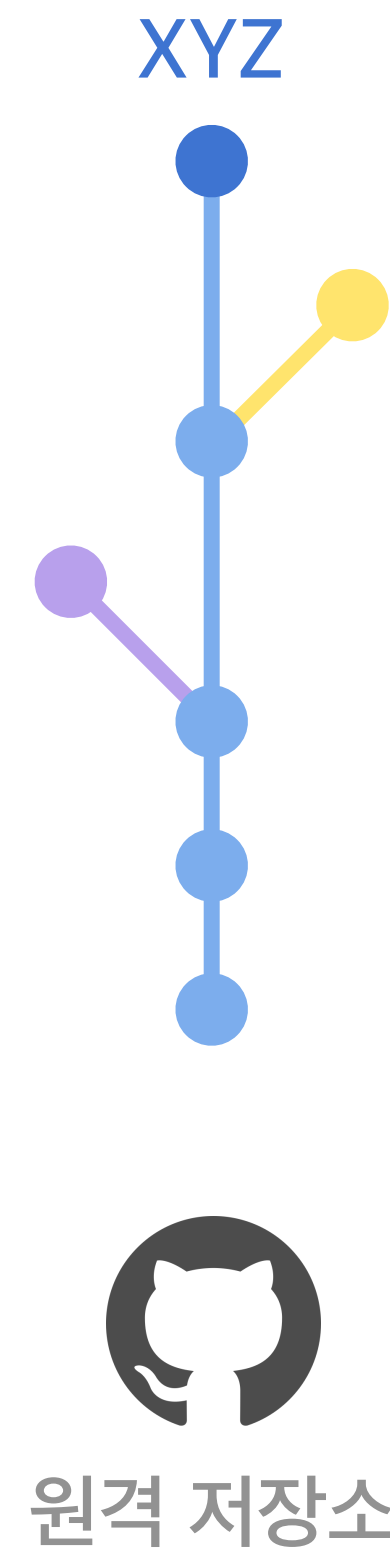
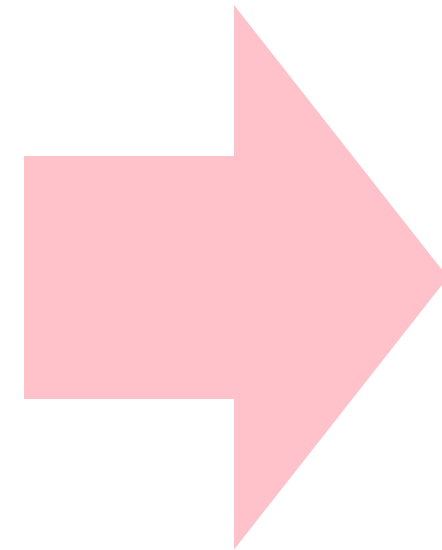


```
$
```

```
$ git push origin master
```



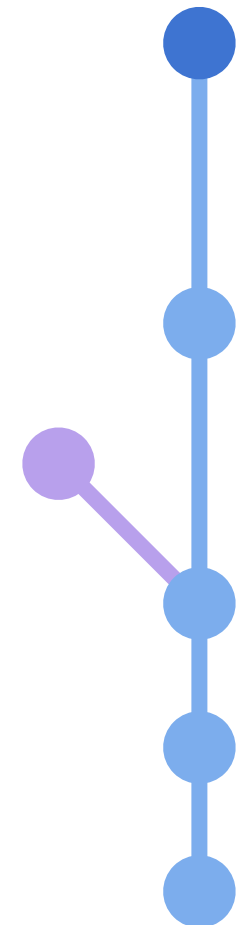
충돌
Conflict



```
$ git push origin master
```

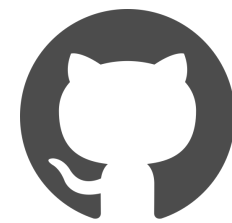
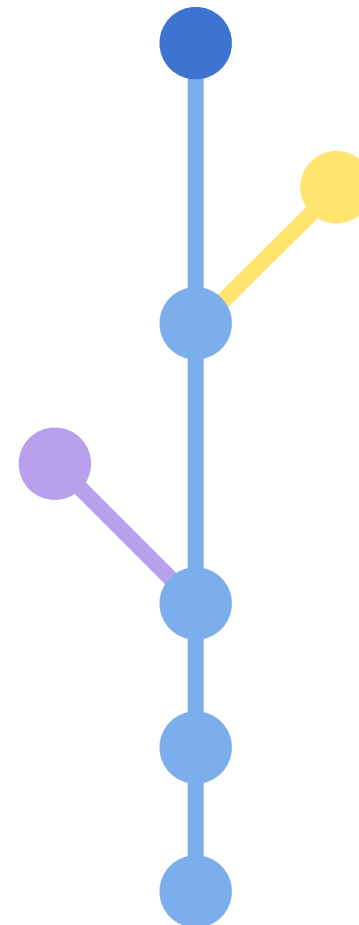
```
$
```

ABC vs XYZ



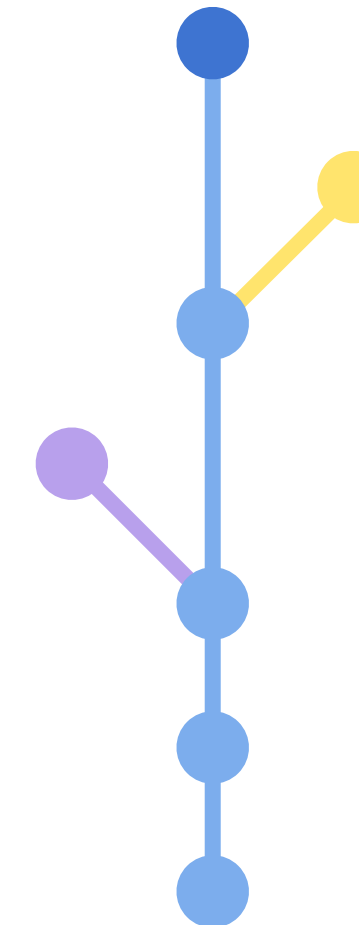
A 개발자

XYZ



원격 저장소

XYZ

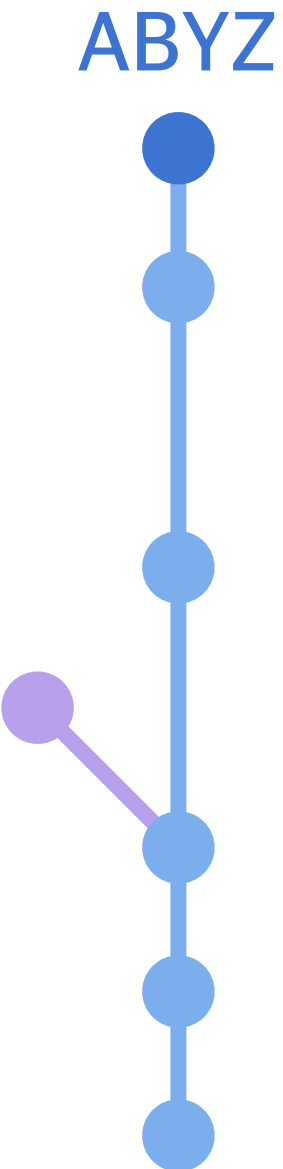


B 개발자

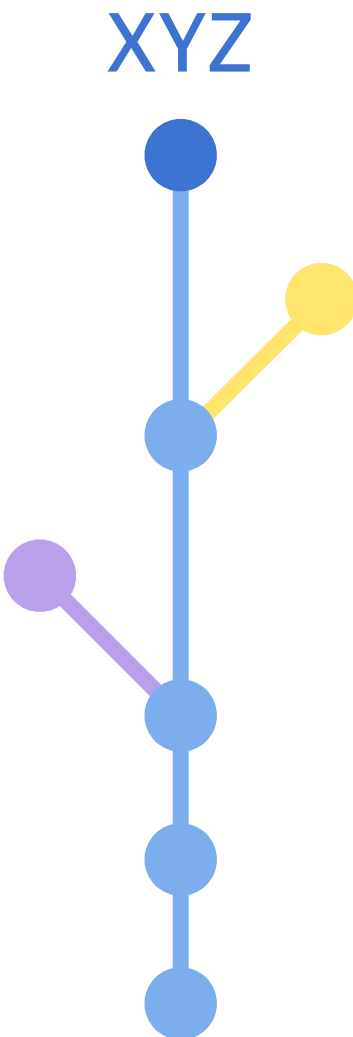


```
$ git pull origin master
```

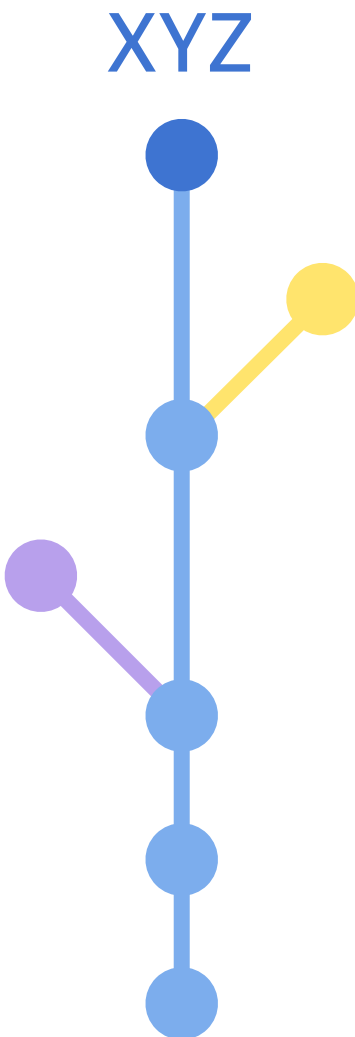
```
$
```



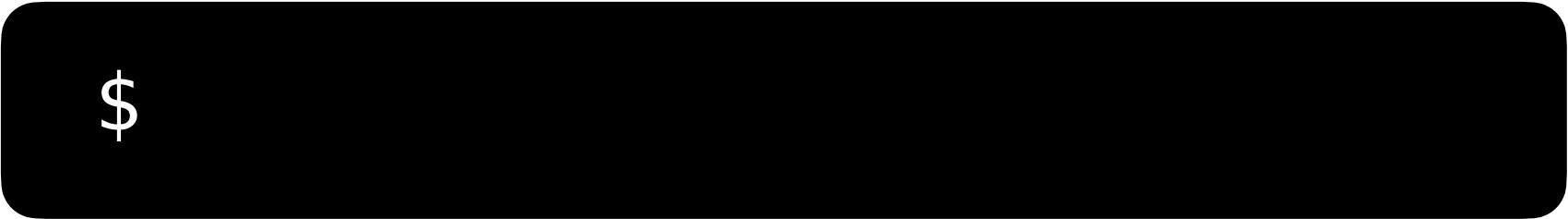
A 개발자

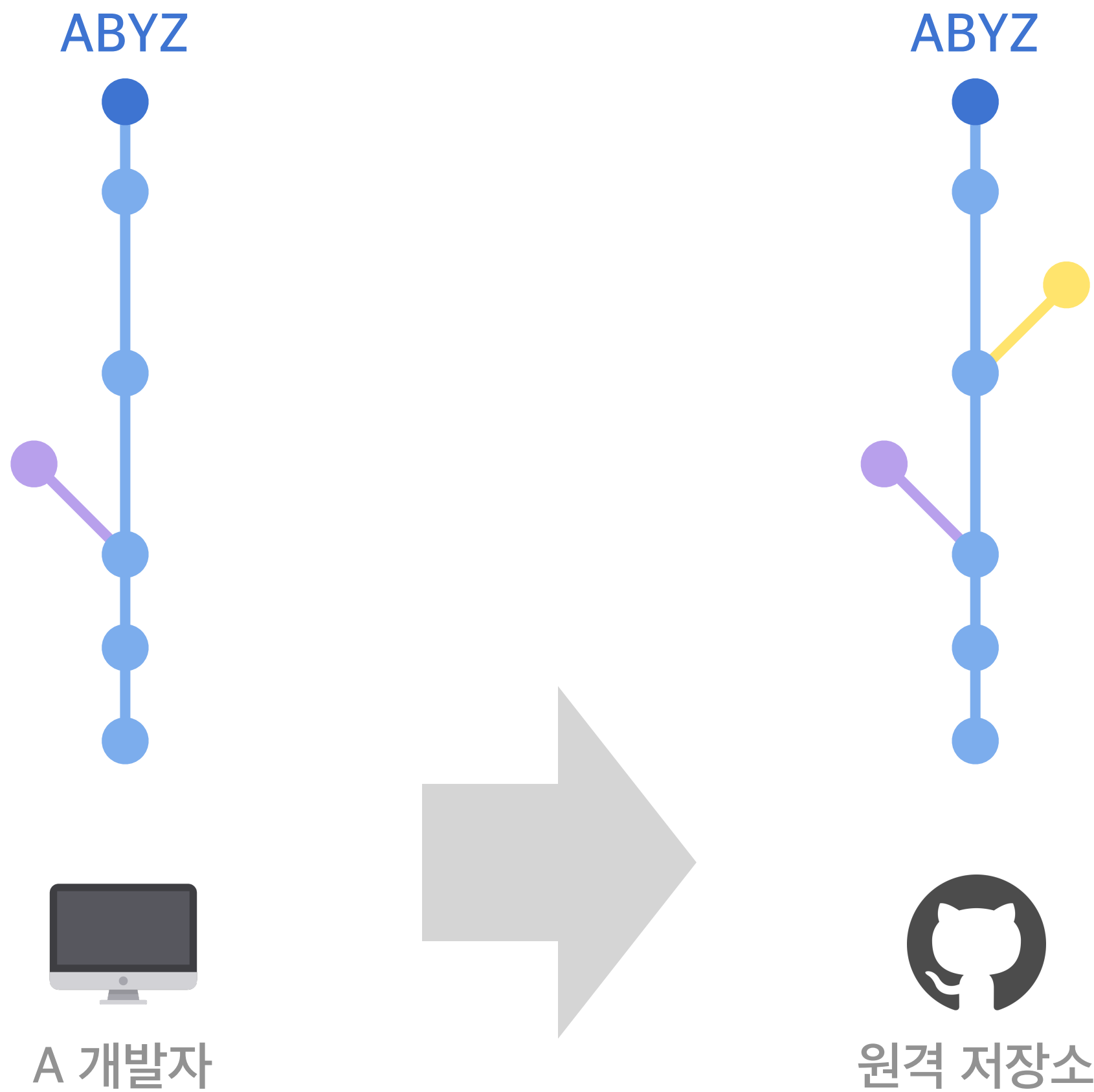


원격 저장소

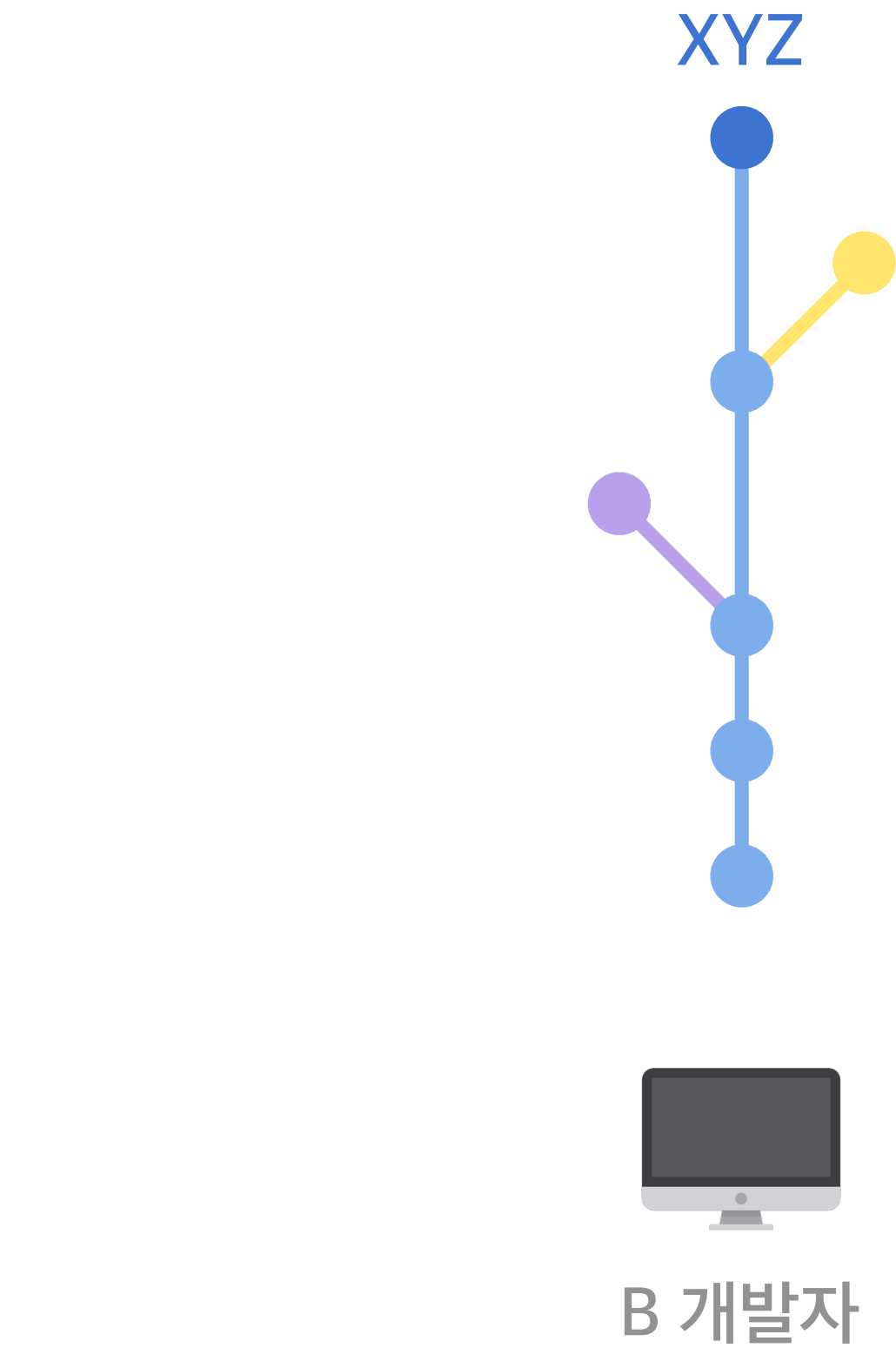


B 개발자

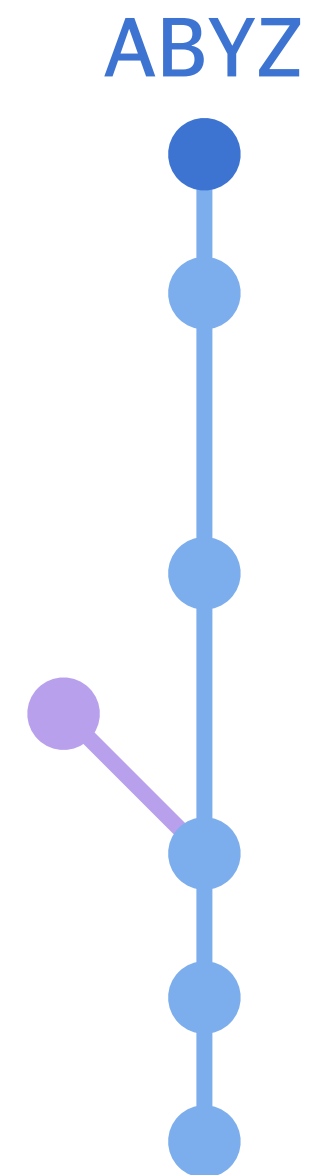




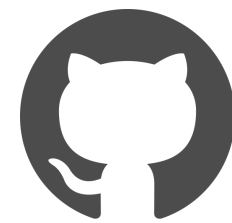
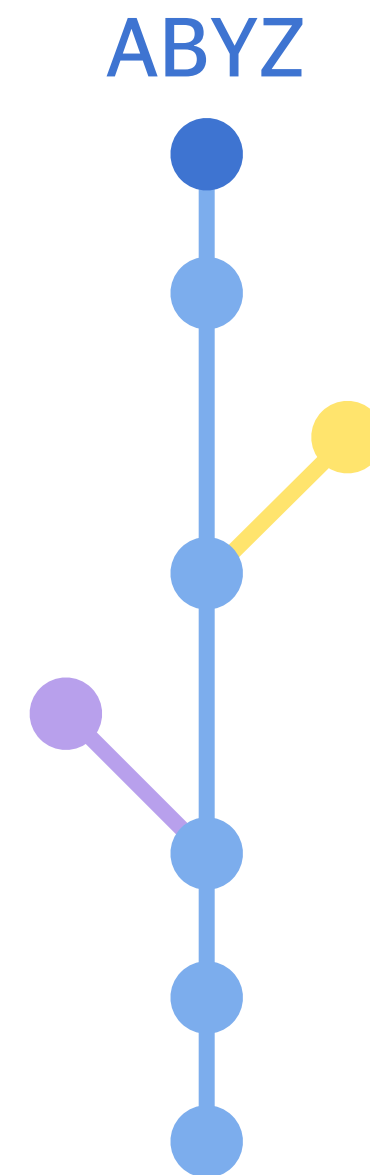
```
$ git push origin master
```



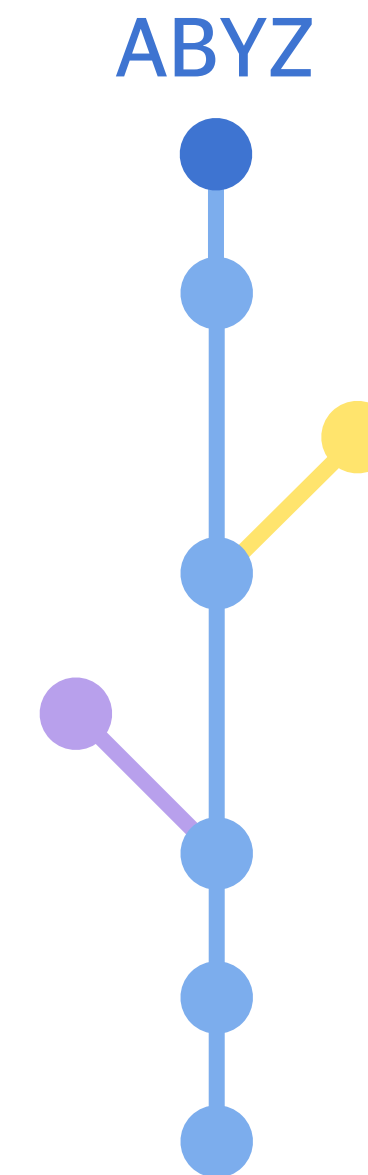
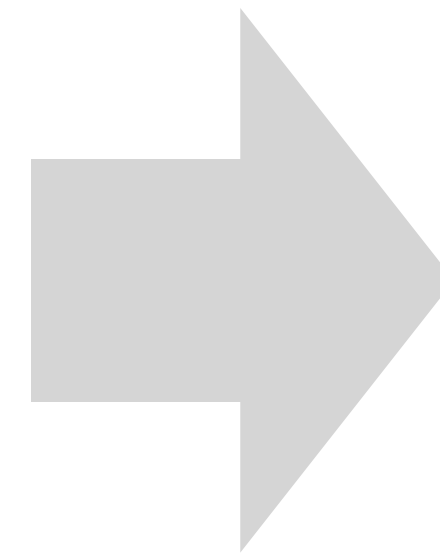
```
$
```



A 개발자



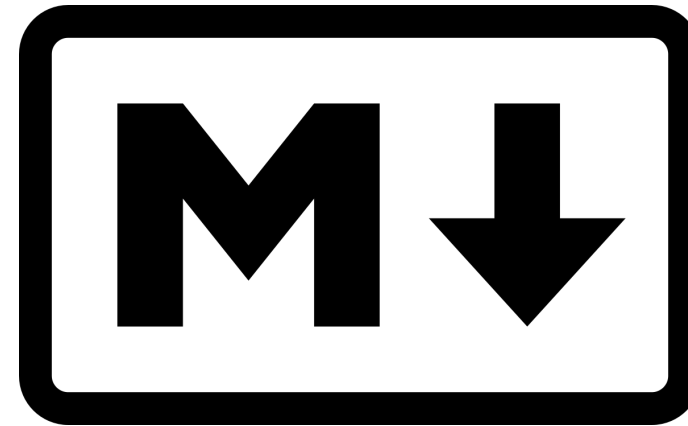
원격 저장소



B 개발자

\$

```
$ git pull origin master
```

Markdown

장점

문법이 쉽고 간결하다!

관리가 쉽다!

지원 가능한 플랫폼과 프로그램이 다양하다!

단점

표준이 없다!

모든 HTML 마크업을 대신하지 못한다!