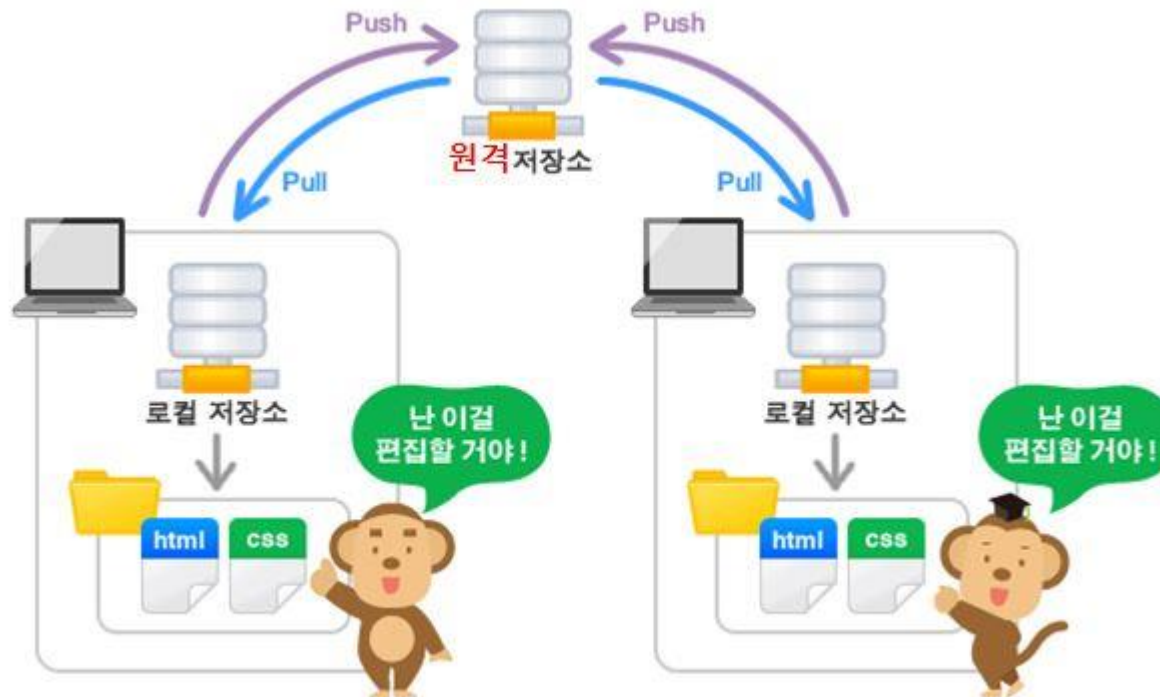


이력을 저장하고 관리하는 저장소(repository)

저장소(Git repository) = 파일이나 폴더를 저장해 두는 곳

Git 저장소의 좋은 점 중 하나는 파일이 변경 이력 별로 구분되어 저장된다는 점입니다. 비슷한 파일이라도 실제 내용 일부 문구가 다르다면 다른 파일로 인식하기 때문에, 파일을 변경 사항 별로 구분해 저장 가능합니다.



원격 저장소 (Remote Repository)	파일이 원격 저장소 전용 서버에서 관리되며, 여러 사람이 함께 공유하기 위한 저장소.
로컬 저장소 (Local Repository)	내 PC 에 파일이 저장되는 개인 전용 저장소.

평소에는 내 PC 의 로컬 저장소에서 작업하다가 작업한 내용을 공개하고 싶을 때에 원격 저장소에 업로드 하고, 원격 저장소에서 다른 사람이 작업한 파일을 로컬 저장소로 가져올 수 있다.

로컬 저장소(Local Repository) 생성하기

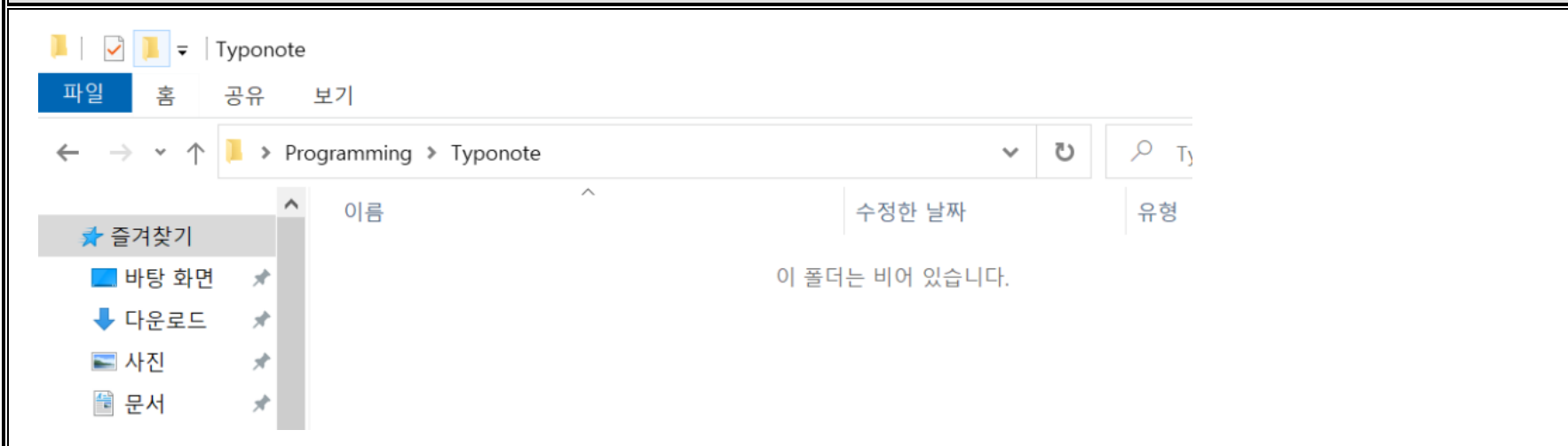
기본적으로 Git 이 설치되어 있어야 한다.

<https://git-scm.com/downloads> 에서 깃을 다운받는다.

Window10 Git 설치

1. Git 설치 Git 은 패키지를 설치하거나 별도의 인스톨러로 설치할 수 있다.

1. 로컬저장소를 만들기 희망하는 폴더 위치로 간다.



2. 오른쪽 마우스를 누르고 Git Bash Here 를 클릭



3. Git bash 에서 명령어 입력

```
권민정 @DESKTOP-SNS4I6A MINGW64 ~/Desktop/Programming/Typonote
$ git init
Initialized empty Git repository in C:/Users/권민정/Desktop/Programming/Typonote/.git/

권민정 @DESKTOP-SNS4I6A MINGW64 ~/Desktop/Programming/Typonote (master)
$ .....
```

```

C:\Users\김민준\Desktop\Programming\Typonote
$ git init
Initialized empty Git repository in C:/Users/김민준/Desktop/Programming/Typonote/.git/

C:\Users\김민준\Desktop\Programming\Typonote (master)
$ ls -al
total 9
drwxr-xr-x 1 김민준 197609 0 Mar 29 20:52 ./
drwxr-xr-x 1 김민준 197609 0 Mar 29 20:48 ../
drwxr-xr-x 1 김민준 197609 0 Mar 29 20:52 .git/
-rw-r--r-- 1 김민준 197609 83 Mar 29 20:50 READ.txt

C:\Users\김민준\Desktop\Programming\Typonote (master)
$ |

```

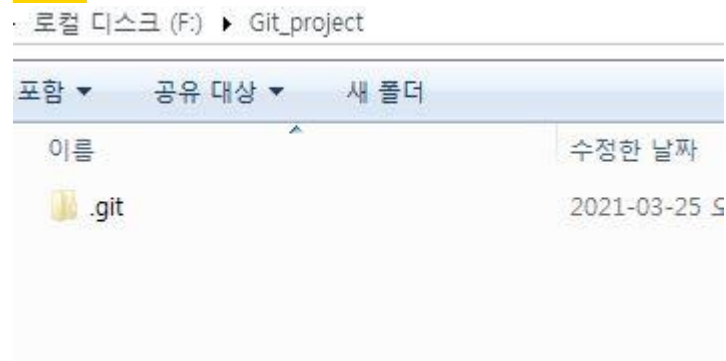
git init 과 **ls -al** 를 차례로 입력한다.

※ **git init**: 해당 폴더에 저장소 디렉토리를 생성하는 명령어. → 저장소를 만듦.

※ **ls -al**: 디렉토리 안의 내용을 조회하는 명령어.

지정한 폴더에서 숨김 파일을 확인하면,

.git 이라는 폴더가 생성된 것을 확인할 수 있다.



.git 폴더 생성

커밋(commit)이란 무엇인가?



Git hub 의 많은 개발자들이 실천하고 있는 '1 일 1 커밋 운동'
국제 개발자는 이를 '잔디'라고도 부른다..

커밋 (Commit)

파일 및 폴더의 추가/변경 사항을 저장소에 기록하도록 명령하는 것.
→ git 에게 **git init** 로 지정된 폴더의 파일들의 상태를 하나의 version 으로써 기억하게 만든다.
(이전 커밋 상태부터 현재 상태까지의 변경 이력이 기록된다.)



언제든지 원하는 커밋한 시점으로 되돌아 갈 수 있다.

**메세지
(message)**

커밋을 실행하는 경우, 이력 관리를 위해 커밋 메시지를 필수로 입력해야 한다.
(메시지가 없으면 커밋이 실행되지 않는다.)
→ 메시지는 명료하고 이해하기 쉽게 남겨야
커밋 이력을 확인하기 쉽다.

< 실습 준비 >

바탕화면에 "Programming" 폴더를 생성하고,
안에 본인의 닉네임으로 폴더를 또 하나 생성한다.

/Desktop/Programming/Typonote

→ 이후에 READ.txt 를 생성한다.

READ.txt

1. 내 정보를 등록하기

각 버전을 누가 만들었는지 표시하기 위해,
사용자의 정보를 git bash 에 등록시켜놔야 한다.

```
git config --global user.email  
git -config user.name
```

"나의 메일 주소(되도록 깃허브 가입 메일)"
" 닉네임이나 깃허브 계정 "

2. 현재 파일의 상태를 확인하기(Status)

git status 명령어를 통해 파일의 현재 **깃 상태**를 확인한다.

→ 커밋할 파일이 있는지 없는지, 아직 커밋을 안했는지 등을 Check

- on branch master : 현재 마스터(master)라는 브랜치(branch)에 있다.
- No commits yet : 지금까지 커밋한 파일이 없다.
- nothing to commit : 커밋할 파일이 없다.
- Untracked files : 추적되지 않는 파일이 있다.

3. 파일을 커밋 대상에 추가하기(Add)

git add 명령어를 통해 파일을 커밋 대상으로 지정한다.

```
git add + 파일명.확장자
```

4. 커밋(commit)을 실행하고, 상세설명 적기(-m): 첫 번째 버전 생성

커밋 시에, 상세 설명을 잘 적어놔야 나중에 코딩 할 때 매우 수월하다.

(설명을 작 적는 습관을 기르도록 하자.)

```
git commit + -m "상세설명을 적는 부분"
```

→ m 은 message 의 약자이다.

5. 파일을 수정하고 커밋하기 : 두 번째 버전 생성

READ.txt 를 수정한 뒤, 2 번(add)과 3 번(commit) 과정을 거쳐 두 번째 커밋을 생성한다.

텍스트 수정

두 번째 커밋 생성.

6. 지금까지 만든 커밋을 확인하기 (log)

git log 명령어를 통해 지금까지 생성한 커밋의 목록을 확인한다.

- commit + c5bdde8... (두 번째 커밋 아이디)
- commit + 87501a9... (첫 번째 커밋 아이디)

7. 다른 커밋으로 버전 이동하기 (checkout)

git checkout 명령어를 통해 다른 커밋의 버전으로 시간 여행을 한다.

```
@DESKTOP-SNS4I6A MINGW64 ~/Desktop/Programming/Typonote (master)
$ git checkout 87501a9
Note: switching to '87501a9'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 87501a9 역 사 적 인 첫 커밋 수 행

READ.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

안녕하세요?

즐거로운 안드로이드 생활에 오신걸 환영합니다.

< 두 번째 커밋에서 첫 번째 커밋으로 이동한 경우 >

- git checkout + 커밋 아이디 : 해당 커밋으로 이동

```
@DESKTOP-SNS4I6A MINGW64 ~/Desktop/Programming/Typonote ((87501a9...))
$ git checkout -
Previous HEAD position was 87501a9 역 사 적 인 첫 커밋 수 행
Switched to branch 'master'
```

READ.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

안녕하세요?

즐거로운 안드로이드 생활에 오신걸 환영합니다.

행복하세요~~

< 첫 번째 커밋에서 두 번째 커밋으로 이동한 경우 >

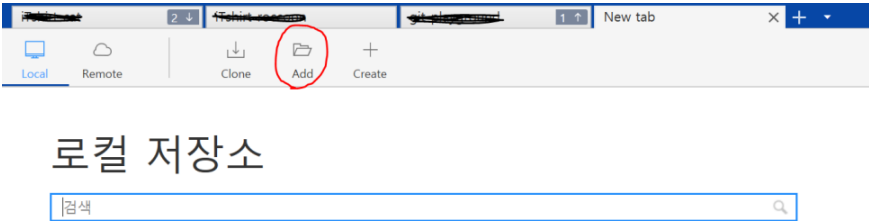
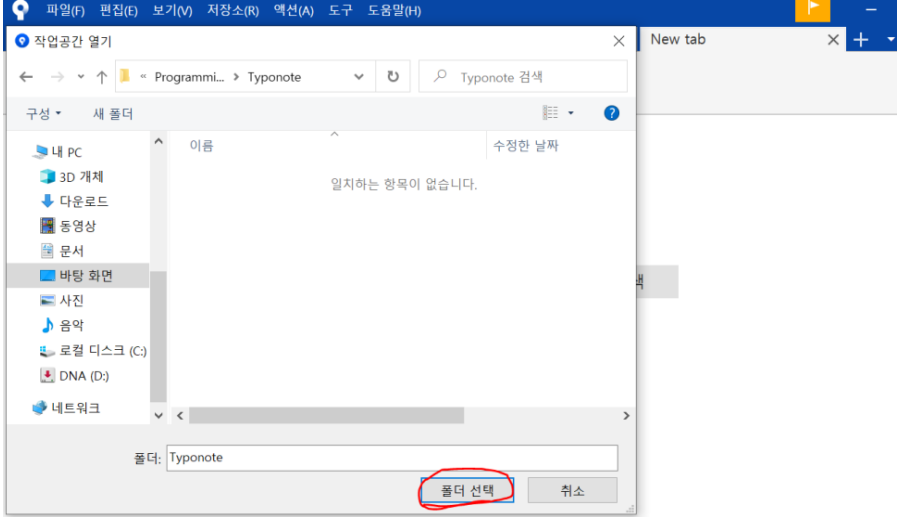
- git checkout + - : 최신 커밋으로 이동

소스트리에 폴더를 불러오는 법은 크게 **두 가지**가 있다.

1. 이미 로컬저장소로 지정한 폴더를 소스트리에 불러오기.
2. 순수한 폴더를 소스트리를 이용하여 로컬저장소로 지정하기.

1-1. 이미 로컬저장소로 지정한 폴더를 소스트리에 불러오기

지난 시간 git bash 로 이미 로컬저장소로 지정한 폴더를 소스트리로 가져오는 작업입니다.
(이미 폴더에 [.git]이 숨긴 파일로 존재하는 폴더)

	
<p>새로운 탭에서 생성(Create)이 아닌 추가(Add)를 누른다.</p>	<p>탐색 버튼을 누르고 로컬저장소로 이미 만든 폴더를 선택한다.</p>

+

커밋

↓

Pull

↑

Push

↻

패치

↕

브랜치

↕

병합

☰

스태시

↺

폐기

🏷

태그

🔗

깃 플로우...

🌐

원격

📄

터미널

🔍

탐색기

⚙

설정

WORKSPACE

파일 상태

History

Search

브랜치

master

태그

원격

스태시

모든 브랜치

원격 브랜치

날짜순으로 정렬

그래프	설명	날짜	작성자	SHA-1
●	master	29 3 2021 22:53	Typonote <kimradium@naver.com>	48412d0f5f211e45f45f2a222816b9d8af58b8c0
●	맨 마지막 줄 수정	29 3 2021 22:31	Typonote <kimradium@naver.com>	8c412d0f5f211e45f45f2a222816b9d8af58b8c0
●	역사적인 첫 커밋 수행	29 3 2021 22:30	Typonote <kimradium@naver.com>	95f211e45f45f2a222816b9d8af58b8c0

파일 상태순 정렬

READ.txt

1 부분 : 1-4 줄

코드문치 되돌리

커밋: 48412d0f5f211e45f45f2a222816b9d8af58b8c0

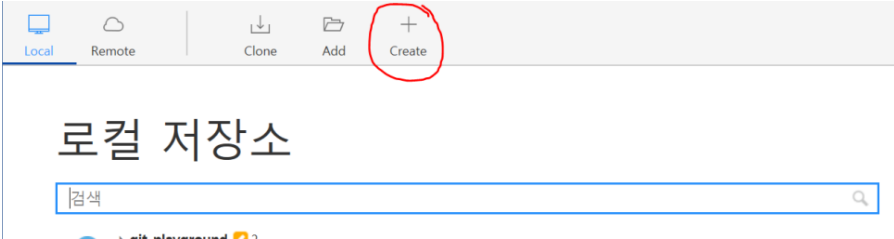
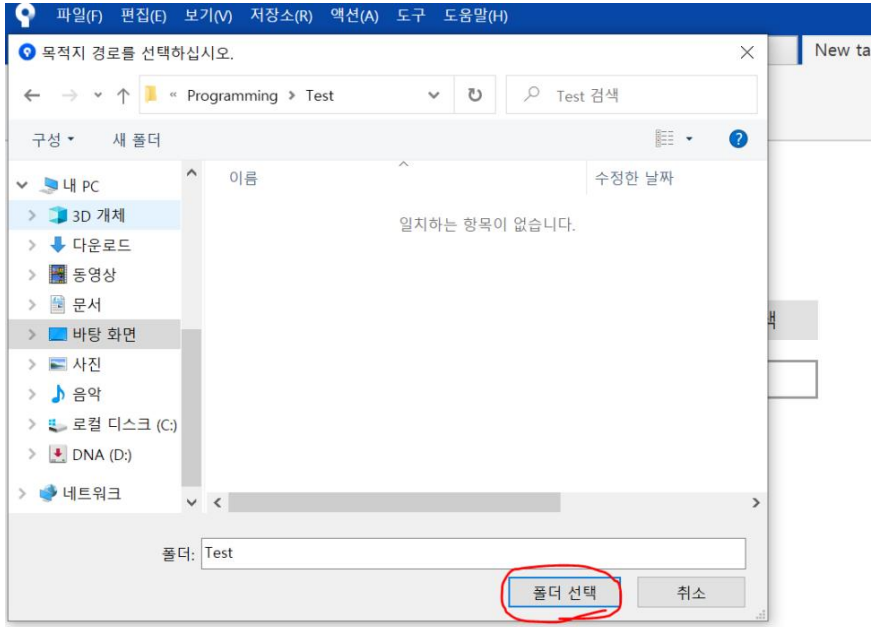
[48412d0]

그동안 해온 작업이 그래픽으로 좀 더 직관적으로 이해할 수 있다.

1-2. 순수한 폴더를 소스트리를 이용하여 로컬저장소로 지정하기


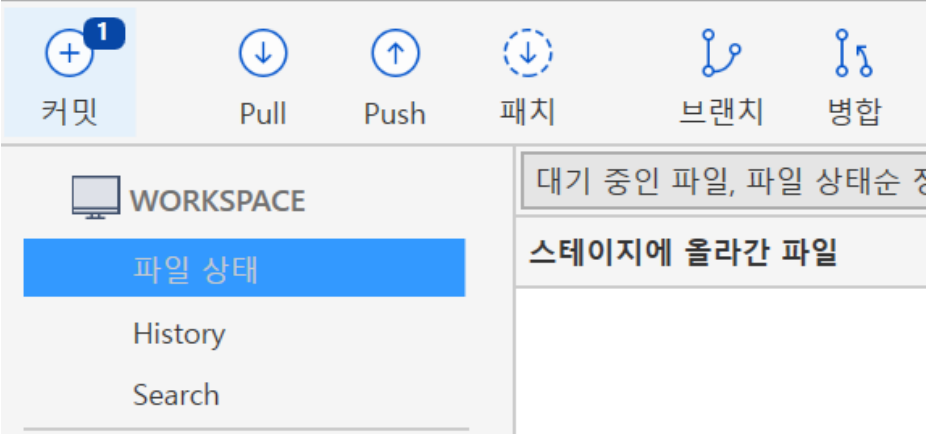
로컬저장소로 지정한적 없는 순수한 폴더를 소스트리에서 로컬저장소로 지정하는 작업입니다..

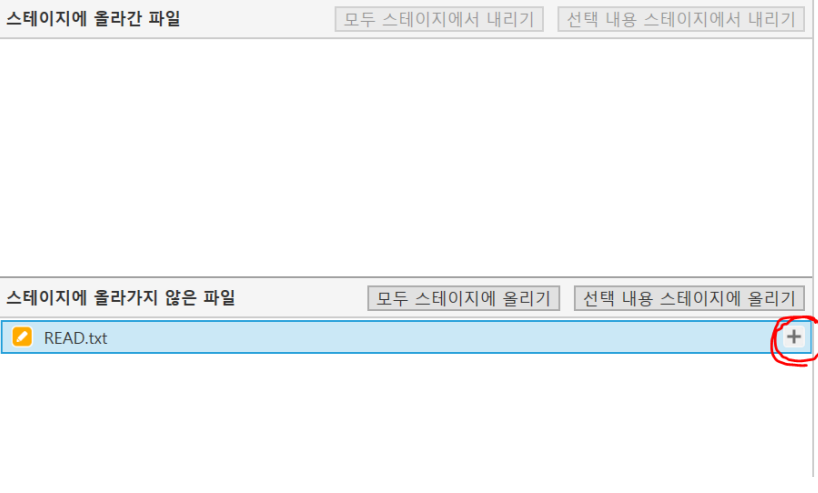
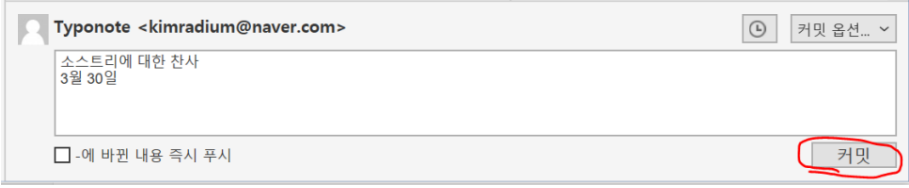
(폴더에 [.git]이 숨긴 파일로도 존재하지 않는 경우)

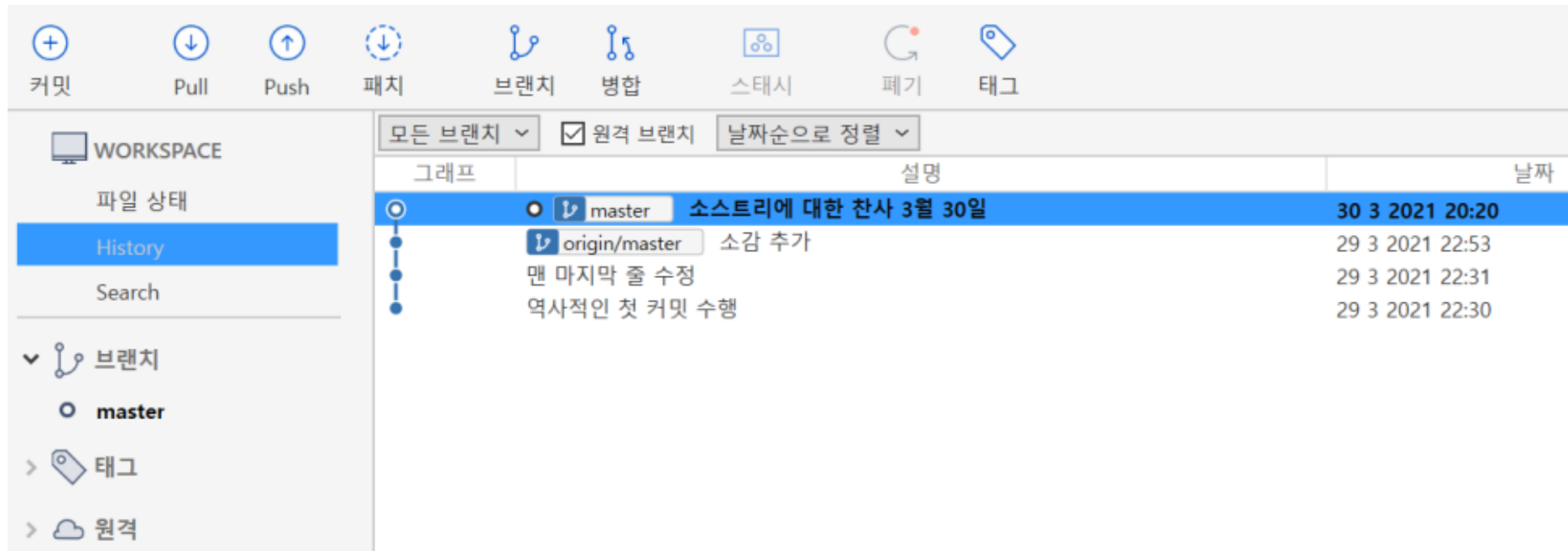
	
<p>새로운 탭에서 추가(Add)가 아닌 생성(Create)을 누른다.</p>	<p>탐색 버튼을 누르고 로컬저장소로 이미 만든 폴더를 선택한다.</p>

2. 소스트리로 커밋을 실행하기

git bash 와 수행했던 과정과 엇비슷하다.

<p> *READ.txt - Windows 메모장</p> <p>파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)</p> <p>안녕하세요? 슬기로운 안드로이드 생활에 오신걸 환영합니다. 행복하세요~~ 오늘 소감: 시작이 반이다. 소스트리 정말 좋은 프로그램.</p>	 <p>The image shows the Git GUI interface. At the top, there are icons for commit (+), pull (down arrow), push (up arrow), patch (dashed down arrow), branch (fork), and merge (two arrows). Below these is a 'WORKSPACE' section with a list of files. The '파일 상태' (File Status) button is highlighted in blue. To the right, there are sections for '대기 중인 파일, 파일 상태순 정' (Staged files, sorted by status) and '스테이지에 올라간 파일' (Files staged).</p>
<p>로컬저장소로 지정한 폴더의 READ.txt 에 한 줄 추가한다.</p>	<p>소스트리로 돌아와보면, 왼쪽 상단 커밋에 알림이 뜬다.</p>

	
<p>스테이지에 올라가지 않은 파일 목록에서 선택하여 + 버튼을 누른다. (커밋 대상으로 지정하는 과정 = <code>git add</code>)</p>	<p>커밋에 대한 상세 설명을 적고, 커밋을 실행한다. (커밋을 실행하는 과정 = <code>git commit + -m</code>)</p>



History에서 방금 수행한 커밋이 잘 수행되었는지 확인한다.

[master] 브랜치는 현재 내 컴퓨터의 로컬저장소의 버전(위치)을 의미한다.

[origin/master] 브랜치는 원격저장소의 버전(위치)을 의미한다.

(지난 시간 origin이란 이름으로 원격저장소를 추가함.)

↓

원격저장소와 로컬저장소의 버전이 일치하지 않는다.

3. 소스트리로 커밋을 원격저장소에 커밋 올리기

<p>커밋을 완료한 후 , 왼쪽 상단의 Push 버튼을 누른다.</p>	<p>원격 저장소의 주소를 확인한 후 , Push 를 진행할 브랜치를 선택하고 Push 버튼을 누른다.</p>

= git bash 에서 `git push origin master` 명령어를 수행한 것과 같다.

그래프	설명	날짜
○ master origin/master	소스트리에 대한 찬사 3월 30일	30 3 2021 20:20
●	소감 추가	29 3 2021 22:53
●	맨 마지막 줄 수정	29 3 2021 22:31
●	역사적인 첫 커밋 수행	29 3 2021 22:30

원격저장소에 Push 한 후, History 를 확인하면

[master] 브랜치 태그와 [origin/master] 브랜치 태그가 같은 위치에 있는 것을 볼 수 있다.

↓

원격저장소와 로컬저장소의 버전이 일치한다.