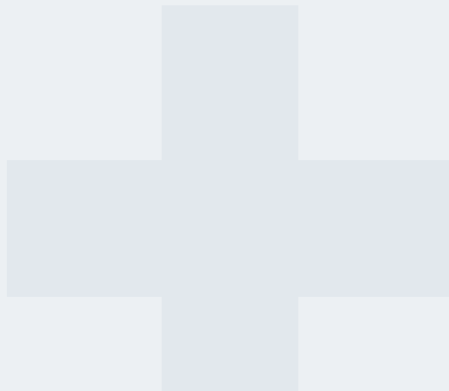


Git & GitHub

깃과 깃허브 기초 및 활용

목차

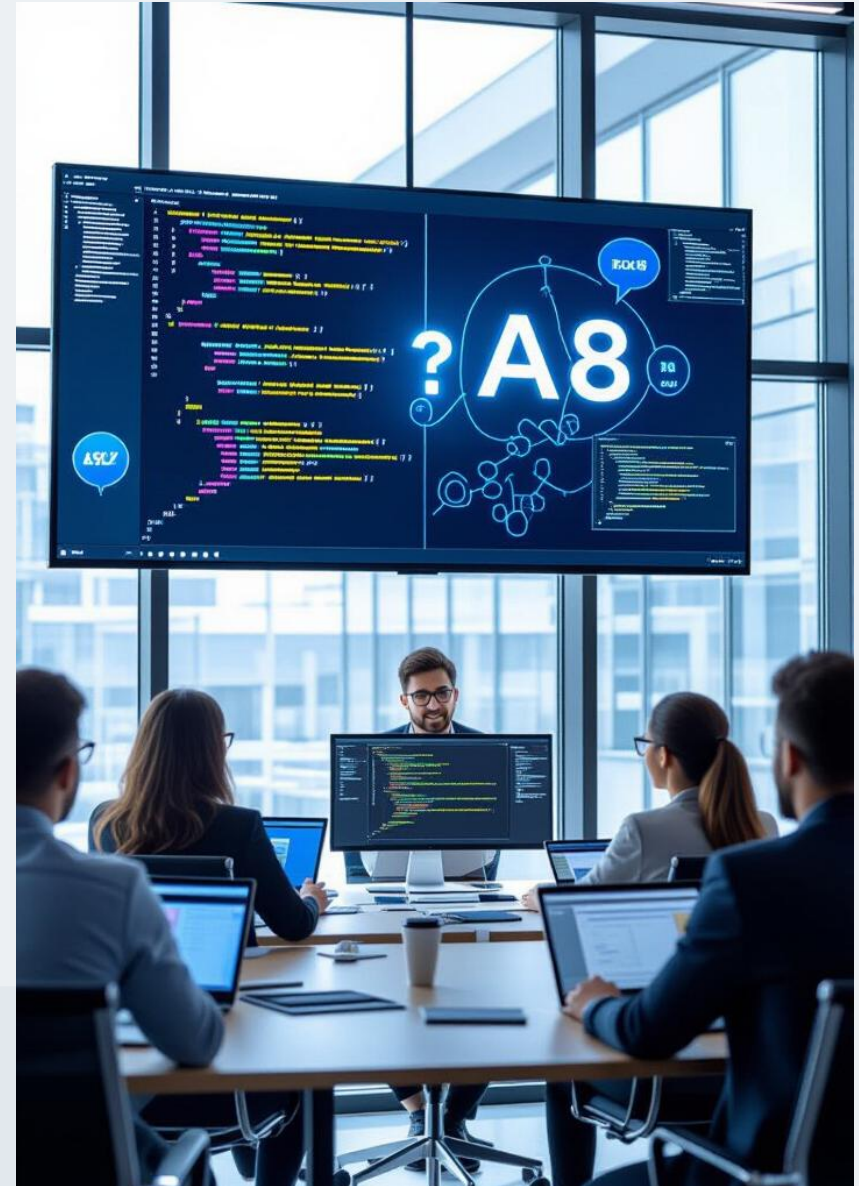
1. 소개: Git & GitHub의 중요성
2. Git 이론: 버전 관리의 시작
3. Git 핵심 개념
4. Git의 기본 명령어 맛보기
5. Git 기본 실습
6. GitHub 이론: 협업의 중심, 원격 저장소
7. GitHub 기본 실습
8. 마무리 및 질의응답



강의 목표

핵심 학습 포인트:

- GitHub를 통해 기본 Git 명령 마스터하기
- 효과적인 협업 전략 이해
- 실습 참여



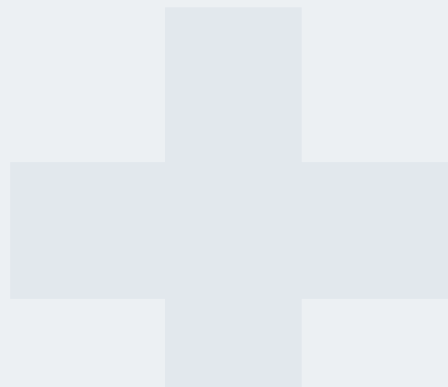
소개: Git & GitHub의 중요성

Git

분산형 버전 관리 시스템

소프트웨어 개발 시 코드 변경 내역을 기록하는 데 핵심 역할

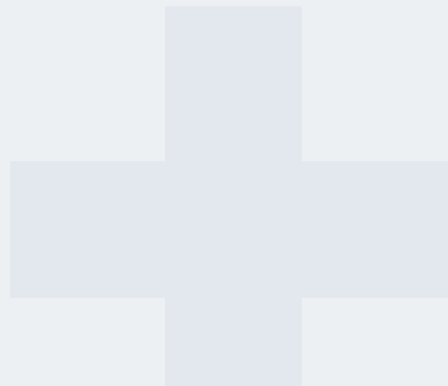
깃허브는 이러한 깃을 바탕으로 한 협업 플랫폼으로,
개발자들이 프로젝트를 공유하고 협력하도록 도와줍니다.



소개: Git & GitHub의 중요성

버전 관리의 필요성

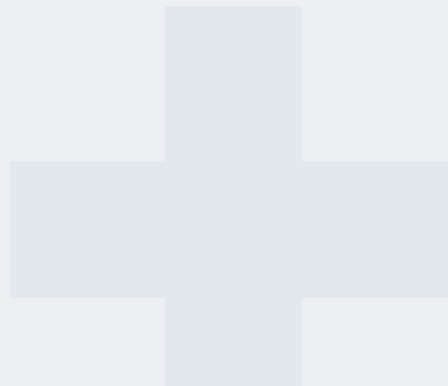
- 코드 변경 이력 관리
 - 이전 버전으로 복구 가능
 - 대규모 개발팀이 동시에 작업 가능
- > 코드 품질 유지 및 오류 줄임



소개: Git & GitHub의 중요성

협업 도구로서 깃허브의 역할

GitHub는 코드 저장소이자 프로젝트 관리 도구
이슈 트래킹, 풀 리퀘스트 및
팀원의 작업을 통합하여
프로젝트의 협업을 더욱 원활하게 지원함



Git 이론: 버전 관리의 시작

VCS(Version Control System)

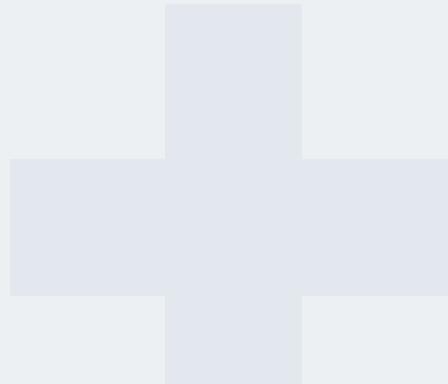
= 버전 관리 시스템

소프트웨어 프로젝트의 모든 변경 이력을
기록하고 관리하는 도구

코드의 복잡성을 줄이고 팀워크를 개선하기 위한
필수적 요소

코드의 중복, 충돌 및 이전 버전으로 복구 문제
해결

각 변경 사항을 기록하여 개발자가 이전 상태로
쉽게 전환할 수 있도록



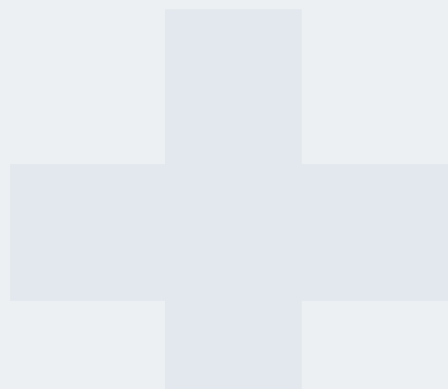
Git 이론

VCS의 종류 및 특징

로컬 VCS: 간단하지만 협업 불가

중앙집중식 VCS: 서버 필요, 단일 장애점

분산형 VCS: 서버 장애 시에도 작업 가능, 로컬에 전체 이력 저장



Git 이론

Git, 어떻게 동작할까?

Repository(저장소)

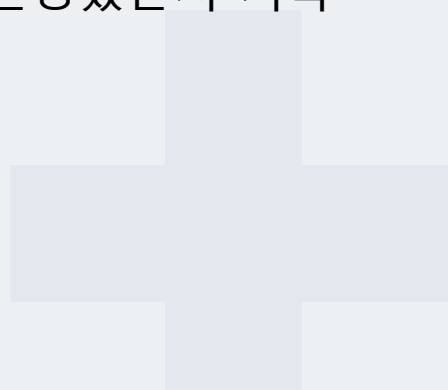
로컬 저장소(Local Repository): 내 컴퓨터에 있는 작업 공간 및 버전 이력 저장 공간

원격 저장소(Remote Repository): GitHub와 같이 외부에 있는 저장소, 협업의 중심

Commit(커밋)

“버전 저장”의 단위, 특정 시점의 스냅샷

커밋 메시지의 중요성: 변경 내용 요약, 누가, 왜 변경했는지 기록



Git 이론

Git, 어떻게 동작할까?

Branch(브랜치)

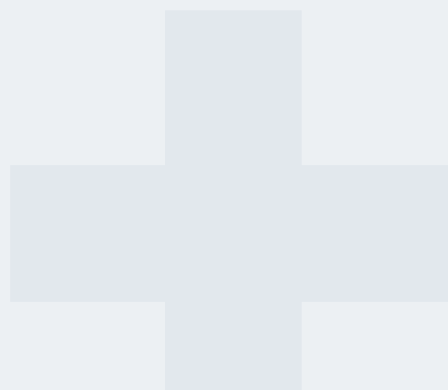
“나뉘어가지”처럼 독립적인 작업 공간

메인(main/master) 브랜치와 분리하여 기능 개발, 버그 수정 등

Merge(머지)

분리된 브랜치들을 하나로 합치는 작업

충돌 발생 가능성 있음



Git 이론

Git, 어떻게 동작할까?

Working Directory:

실제 파일 수정이 일어나는 공간, 변경 사항을 깃이 추적하지만 관리 영역으로는 들어오지 않음

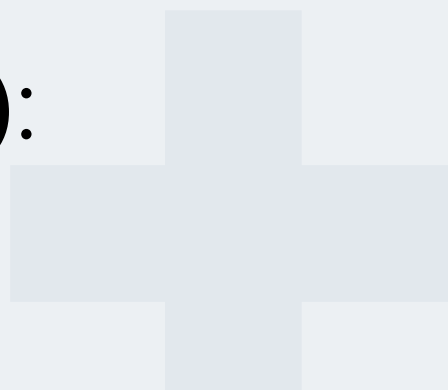
Staging Area(Index):

커밋할 준비가 된 파일들이 대기하는 공간

add 명령을 통해 Working Directory의 변경 사항을 가져옴

Local Repository(Head):

커밋된 버전들이 저장되는 공간



Git의 기본 명령어 맛보기

git init: 새로운 git 저장소 초기화

git status: 현재 저장소 상태 확인

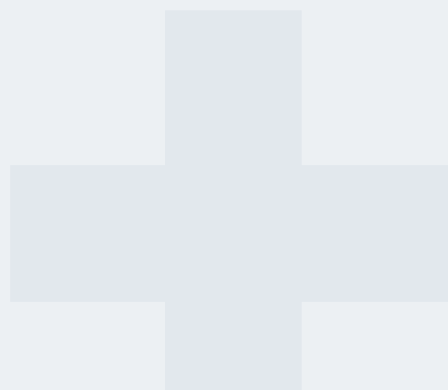
git add <파일이름> / git add: 변경된 파일을 Staging area에 추가

git commit -m "메시지": Staging area의 파일을 Local repository에 커밋

git log: 커밋 이력 확인

git diff: 변경 사항 비교

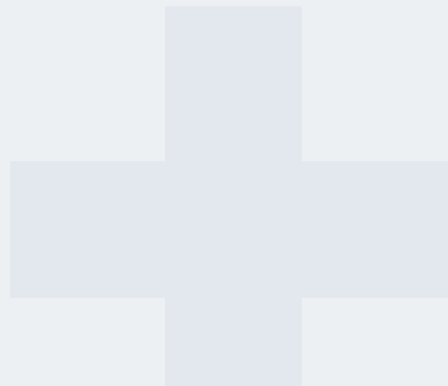
git revert: 특정 커밋의 변경 사항 되돌리기



Git 기본 실습

Git 설치하기 -> <https://git-scm.com/downloads>

실습 전, 워밍업! -> <https://youtu.be/YFNQwo7iTnc?si=vXHSJhM2PWVnbswL>



Git 기본 실습

1. git 버전 확인

>> git --version

```
C:\Users\82109>git --version  
git version 2.51.0.windows.1
```

2. 사용자 정보 설정

>> git config --global user.name "Your Name"

>> git config --global user.email your.email@example.com

```
C:\Users\82109>git config --global user.name "Chaeuny"
```

```
C:\Users\82109>git config --global user.email "jade98904@naver.com"
```

3. 설정 확인

>> git config --global user.name

>> git config --global user.email

>> git config --global --list

```
C:\Users\82109>git config --global user.name  
Chaeuny
```

```
C:\Users\82109>git config --global user.email  
jade98904@naver.com
```

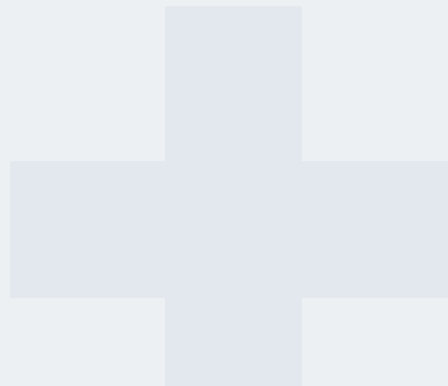
Git 기본 실습

4. 작업 디렉토리 생성

>> cd "바탕화면 폴더 경로"



```
C:\Users\82109> cd C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny  
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>
```



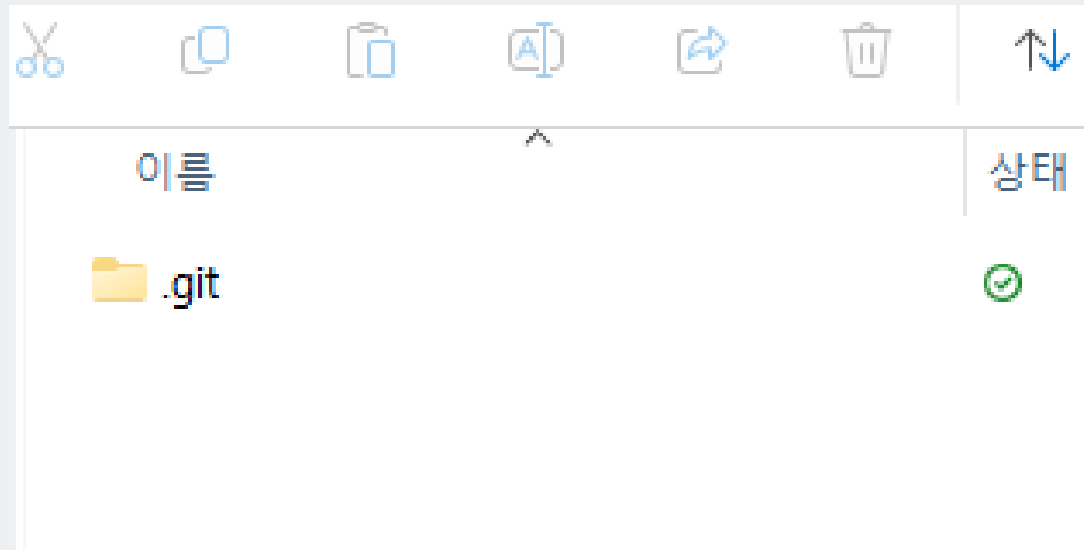
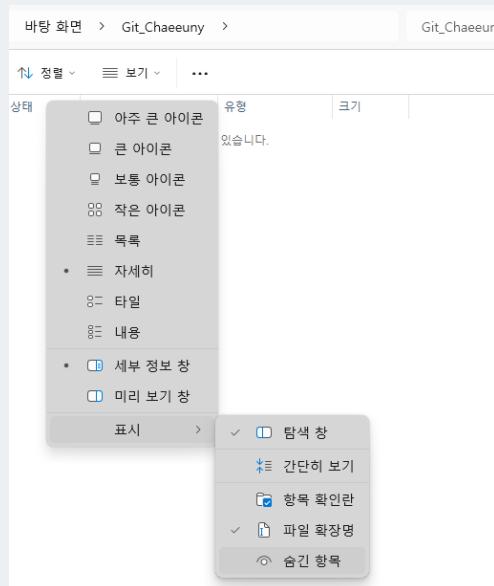
Git 기본 실습

5. git 저장소 초기화: 새로운 깃 저장소 시작할 때 사용, 현재 디렉토리가 git으로 관리되기 시작

>>git init

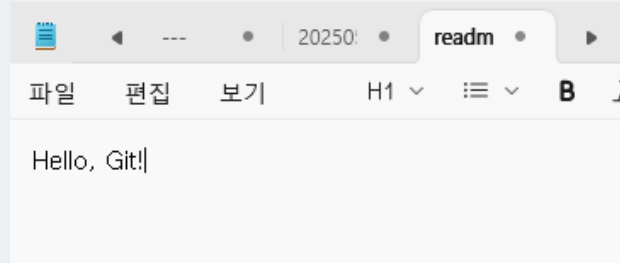
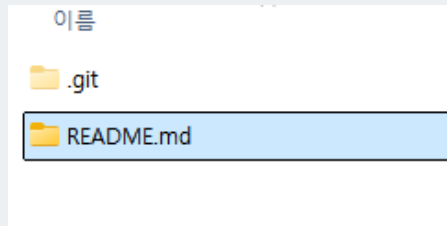
➔ 해당 폴더에 .git이라는 숨김 폴더 생성되었는지 확인

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git init
Initialized empty Git repository in C:/Users/82109/OneDrive/바탕 화면/Git_Chaeuny/.git/
```



Git 기본 실습

6. 파일 생성 및 수정



7. 현재 상태 확인: Working directory와 Staging Area의 상태 확인,
어떤 파일이 변경됐는지, 커밋 준비가 됐는지 확인

>>git status (결과: untracked files: README.md -> 아직 git이 추적하지 않는 파일)

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.md/

nothing added to commit but untracked files present (use "git add" to track)
```

Git 기본 실습

8. 파일을 Staging Area에 추가: add 명령어로

“이 파일을 다음 커밋에 포함할 준비가 됐다”고 깃에게 알림

```
>>git add README.md
```

```
>>git add .
```

```
>>git status
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git add README.md
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git add .
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   README.md/readme.txt
```

Git 기본 실습

9. 변경 사항 커밋: Staging Area에 있는 변경 사항들을 Local Repository에 영구적으로 저장

>>git commit -m "첫 README 파일 추가"

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git commit -m "첫 README 파일 추가"
[master (root-commit) 253a20d] 첫 README 파일 추가
1 file changed, 1 insertion(+)
create mode 100644 README.md/readme.txt
```

10. 커밋 이력 확인

>>git log

>>git log --oneline (한줄 요약)

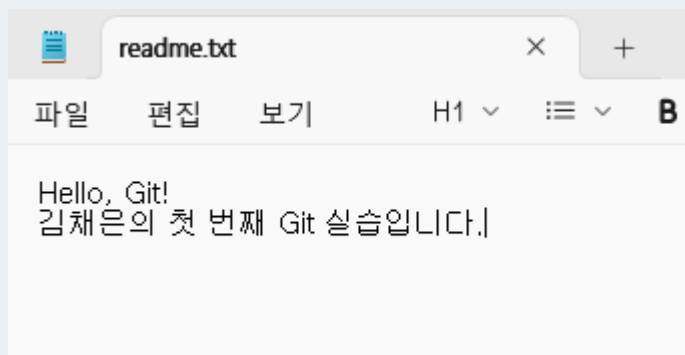
```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git log
commit 253a20db89838abf2f40689dc82c6926a73e7060 (HEAD -> master)
Author: Chaeuny <jade98904@naver.com>
Date: Wed Aug 20 10:06:18 2025 +0900
```

첫 README 파일 추가

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git log --oneline
253a20d (HEAD -> master) 첫 README 파일 추가
```

Git 기본 실습

11. 파일 수정 후 저장 및 두 번째 커밋



>>git status

>>git add .

>>git commit -m "README 내용 추가"

>>git log

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git log
commit 3487b03446d27d8122adcbf4a84baee9011661fe (HEAD -> master)
Author: Chaeuny <jade98904@naver.com>
Date:   Wed Aug 20 10:19:19 2025 +0900
```

README 내용 추가

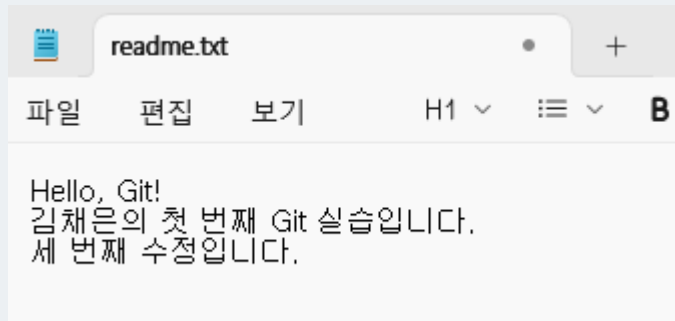
```
commit 253a20db89838abf2f40689dc82c6926a73e7060
Author: Chaeuny <jade98904@naver.com>
Date:   Wed Aug 20 10:06:18 2025 +0900
```

첫 README 파일 추가

Git 기본 실습

12. 변경 사항 비교: 아직 add하지 않은 Working Directory의 변경 사항과 마지막 커밋 간의 차이를 보여줌

>>git diff .



```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git diff .
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git diff --staged
diff --git a/README.md/readme.txt b/README.md/readme.txt
index 65a4b12..91a6639 100644
--- a/README.md/readme.txt
+++ b/README.md/readme.txt
@@ -1,2 +1,3 @@
 Hello, Git!
-김채은의 첫 번째 Git 실습입니다.
 \ No newline at end of file
+김채은의 첫 번째 Git 실습입니다.
+세 번째 수정입니다.
 \ No newline at end of file
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git log
commit 3487b03446d27d8122adcbf4a84baee9011661fe (HEAD -> master)
Author: Chaeuny <jade98904@naver.com>
Date: Wed Aug 20 10:19:19 2025 +0900
```

README 내용 추가

```
commit 253a20db89838abf2f40689dc82c6926a73e7060
Author: Chaeuny <jade98904@naver.com>
Date: Wed Aug 20 10:06:18 2025 +0900
```

첫 README 파일 추가

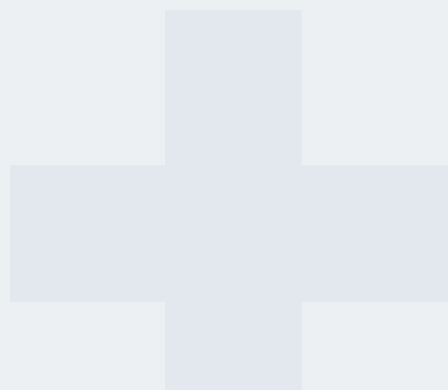
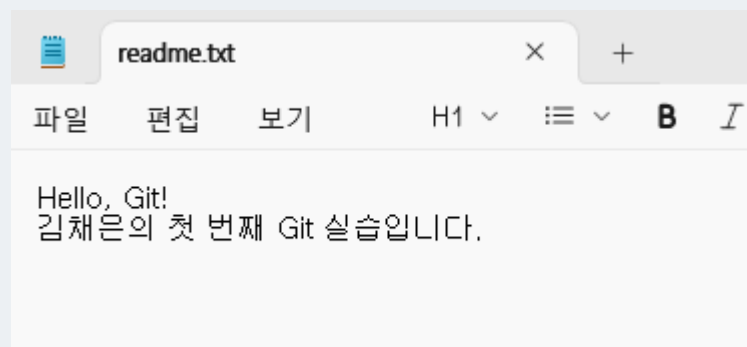
Git 기본 실습

13. 특정 커밋의 변경 사항 되돌리기

>>git revert <되돌릴 커밋 ID 앞 7자리 or 전체>

>>git log

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeun>git revert 9a03cfcc813c5ccb5ced88181a144c0ae5415ed7
[master 874976a] Revert "세 번째 수정 내용 추가"
1 file changed, 1 insertion(+), 2 deletions(-)
```



Git 기본 실습

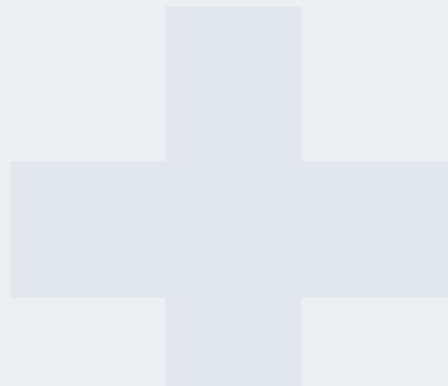
13. 특정 커밋으로 되돌리기(주의)

```
>>git reset --hard <커밋 ID 앞 7자리 or 전체>
```

```
>>git log
```

지정된 커밋 상태로 Working Directory와 Local Repository를 **강제**로 되돌림.

이후의 커밋 이력이 사라짐



Git 기본 실습

브랜치와 머지 실습

14. 현재 작업 폴더로 이동

>>cd "깃 파일 경로"

```
C:\Users\82109>cd C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny  
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>
```

15. 현재 브랜치와 상태 확인

>>git status

>>git branch

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git status  
On branch master  
Your branch is up to date with 'origin/master'.  
  
nothing to commit, working tree clean  
  
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git branch  
* master  
  
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>
```


Git 기본 실습

16. 새로운 기능 브랜치 생성 및 이동

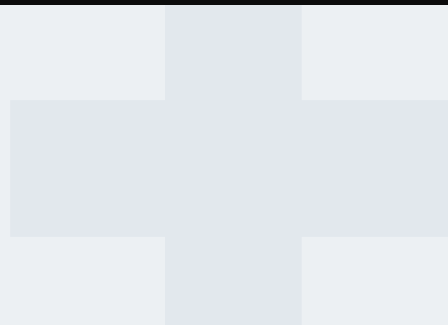
>>git switch -c "브랜치 이름"

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git switch -c newbranch  
Switched to a new branch 'newbranch'
```

17. 새로운 브랜치 확인

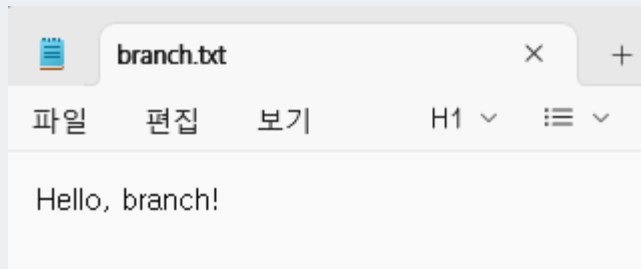
>>git branch

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git branch  
master  
* newbranch
```



Git 기본 실습

18. 브랜치 텍스트파일 생성



19. 변경된 파일 staging(스테이징) 후 커밋 메시지 남기기

>>git add .

>>git commit -m "커밋 메시지 자유롭게 추가"

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git add .
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git commit -m "새로운 브랜치 생성"
```

```
[newbranch 7a3e5d7] 새로운 브랜치 생성
```

```
2 files changed, 2 insertions(+)
```

```
create mode 100644 README.md/branch.txt
```

Git 기본 실습

20. 결과 확인

>>git log // >>git log --oneline

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git log --oneline
7a3e5d7 (HEAD -> newbranch) 새로운 브랜치 생성
ca31507 (origin/master, master) Update readme.txt
f997faf Revert "그 전 내용으로 되돌리기"
874976a Revert "세 번째 수정 내용 추가"
9a03cfc 세 번째 수정 내용 추가
3487b03 README 내용 추가
253a20d 첫 README 파일 추가
```

Git 기본 실습

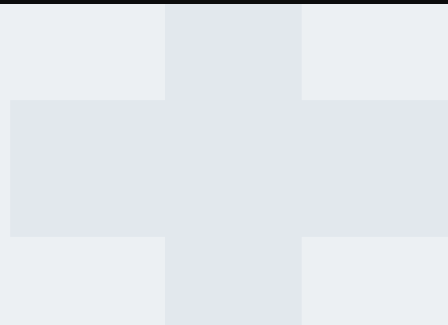
21. 메인(마스터) 브랜치로 돌아오기

>>git switch main // git switch master

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git branch
master
* newbranch

C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git branch
* master
newbranch
```



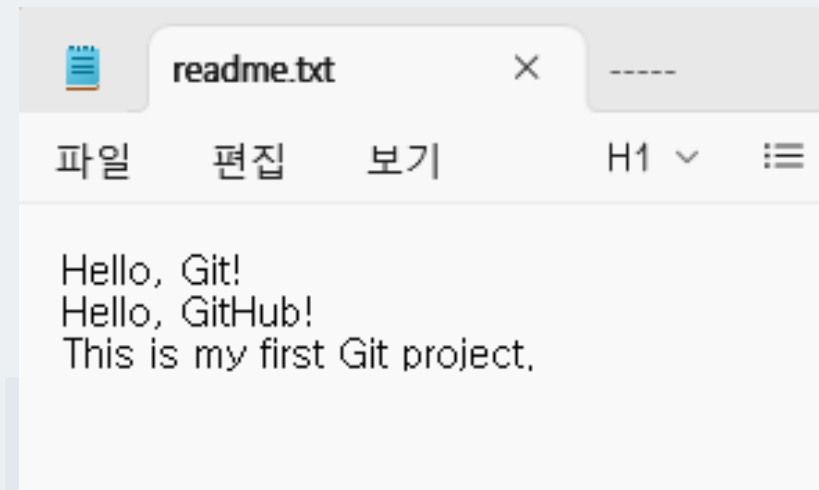
Git 기본 실습

22. 새로 생성한 브랜치에서 작업한 내용을 메인(마스터) 브랜치에 병합(merge)

>>git merge newbranch

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git merge newbranch
Updating ca31507..7a3e5d7
Fast-forward
 README.md/branch.txt | 1 +
 README.md/readme.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 README.md/branch.txt
```

23. readme.md 파일 오픈 후, 병합되었는지 확인



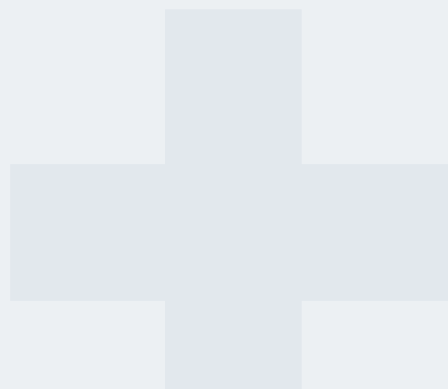
GitHub 이론

Git과 GitHub의 관계?

Git: 버전 관리 도구 // GitHub: Git 저장소를 호스팅하고 협업 기능을 제공하는 웹 서비스

GitHub의 역할

- 코드 공유 및 백업: 로컬 저장소의 코드를 온라인에 안전하게 보관
- 협업: 여러 개발자가 함께 하나의 프로젝트를 효율적으로 개발
- 프로젝트 관리: Issue Tracker, Pull Request 등으로 개발 프로세스 관리
- 오픈 소스 생태계: 전 세계 개발자들과 코드를 공유하고 기여



GitHub 이론

GitHub 핵심 기능 살펴보기

- **Repository(원격 저장소):** GitHub에 생성되는 프로젝트 단위의 저장소, 공개 및 비공개
- **Pull Request(PR):** 다른 브랜치로 자신의 변경 사항을 병합(Merge)해달라고 요청하는 기능, 코드 리뷰 및 토론 공간
- **Issue:** 버그 보고, 기능 요청, 개선 사항 등 프로젝트 관련 논의 및 할 일 관리
- **Fork:** 다른 사용자의 공개 저장소를 내 계정으로 통째로 복사해 오는 기능, 주로 오픈소스 기여 시 사용
- **Clone:** 원격 저장소의 내용을 내 로컬 컴퓨터로 복제하는 기능

GitHub 기본 실습

Git과 GitHub 연동하기

>>git remote: 원격 저장소 관리

>>git remote add origin <URL>: 원격 저장소 연결

>>git remote -v: 연결된 원격 저장소 확인

Create a new repository [Preview](#) [Switch back to classic experience](#)

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * **Repository name ***

Git_Chaeun is available.

Great repository names are short and memorable. How about [literate-octo-meme](#)?

Description

24 / 350 characters

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

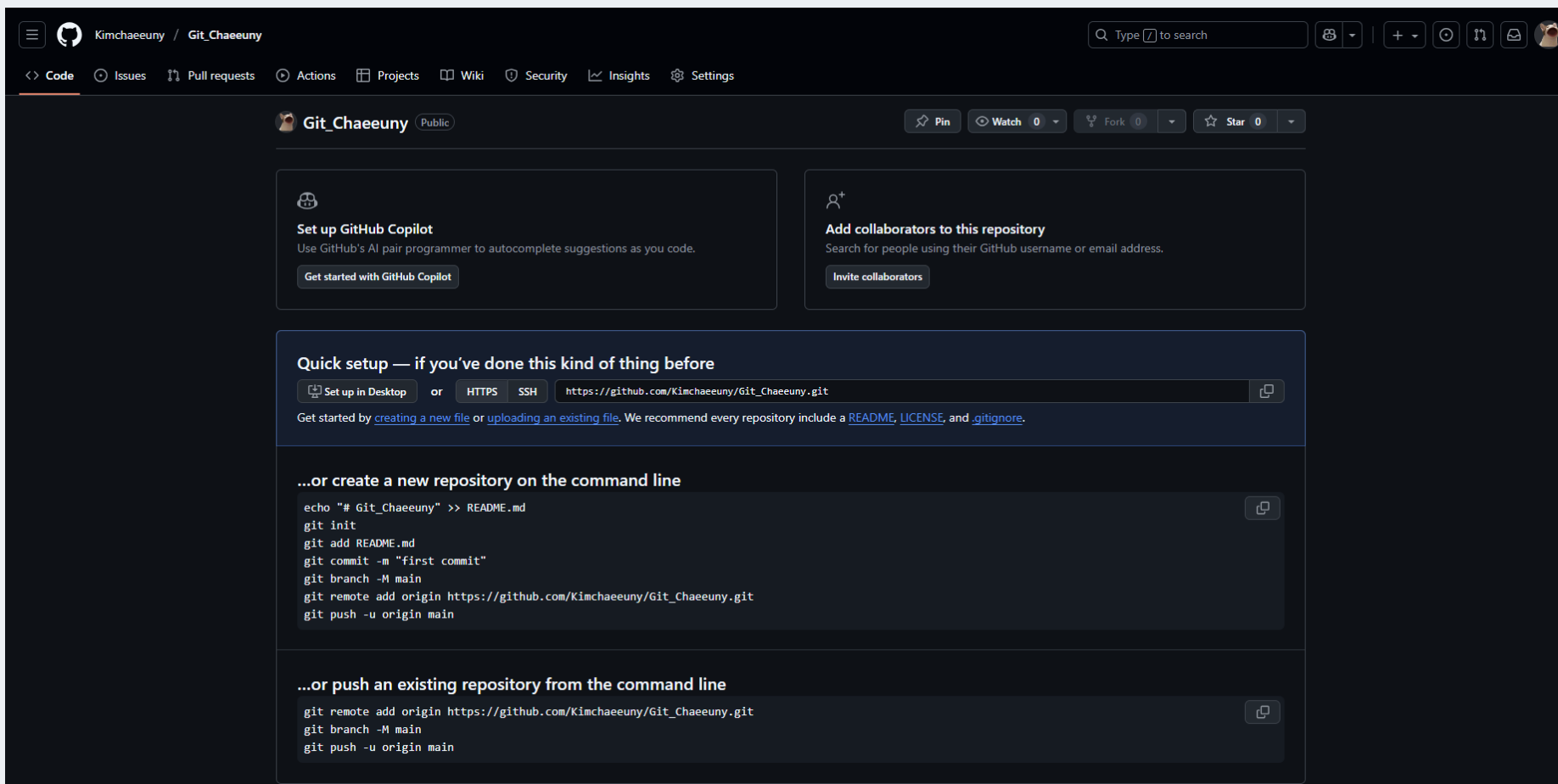
Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

[Create repository](#)

GitHub 기본 실습

1. 새로운 저장소 생성 확인



GitHub 기본 실습

2. 로컬 Git 저장소 폴더로 이동

터미널에서 만들어뒀던 git 폴더로 이동

```
>>cd "폴더 경로"
```

3. 원격 저장소(remote) 추가

```
>>git remote add origin "깃허브 메인 주소"
```

4. 연결 확인

```
>>git remote -v
```

```
C:\Users\82109>cd C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git remote add origin https://github.com/Kimchaeuny/Git_Chaeuny
```

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git remote -v
```

```
origin https://github.com/Kimchaeuny/Git_Chaeuny (fetch)
```

```
origin https://github.com/Kimchaeuny/Git_Chaeuny (push)
```

GitHub 기본 실습

5. 로컬 커밋을 GitHub로 Push

>>git push -u origin master

+ git branch 명령어를 통해 로컬 기본 브랜치 이름을 미리 확인! main일 경우에는 main으로 지정

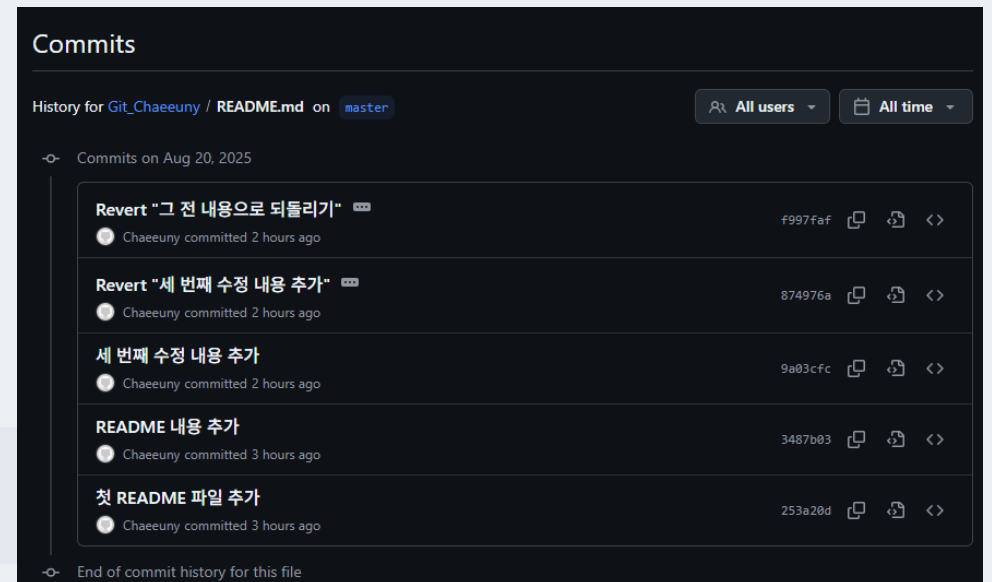
+ origin: 설정한 원격 저장소의 별명

아이디, 비밀번호를 입력하여 인증

GitHub 사이트 접속하여 Repository 새로고침 후 확인

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git branch
* master

C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (14/14), 1.22 KiB | 313.00 KiB/s, done.
Total 14 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Kimchaeuny/Git_Chaeuny
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```



GitHub 기본 실습

5. GitHub의 변경 사항을 로컬로 가져오기

GitHub에서 파일 수정(원격 변경 만들기)



- Repository에서 파일 클릭
- 파일 편집하고 Commit changes


```
master Git_Chaeuny / README.md / readme.txt  
Kimchaeuny Update readme.txt  
2 lines (2 loc) · 27 Bytes  
Code Blame  
1 Hello, Git!  
2 Hello, GitHub!
```

GitHub 기본 실습

6. GitHub의 변경 사항 가져오기(pull)

>>git pull origin master

 master  **Git_Chaeun** / **REA**

 **Kimchaeun** Update readme.txt

2 lines (2 loc) · 27 Bytes

Code Blame

1 Hello, Git!

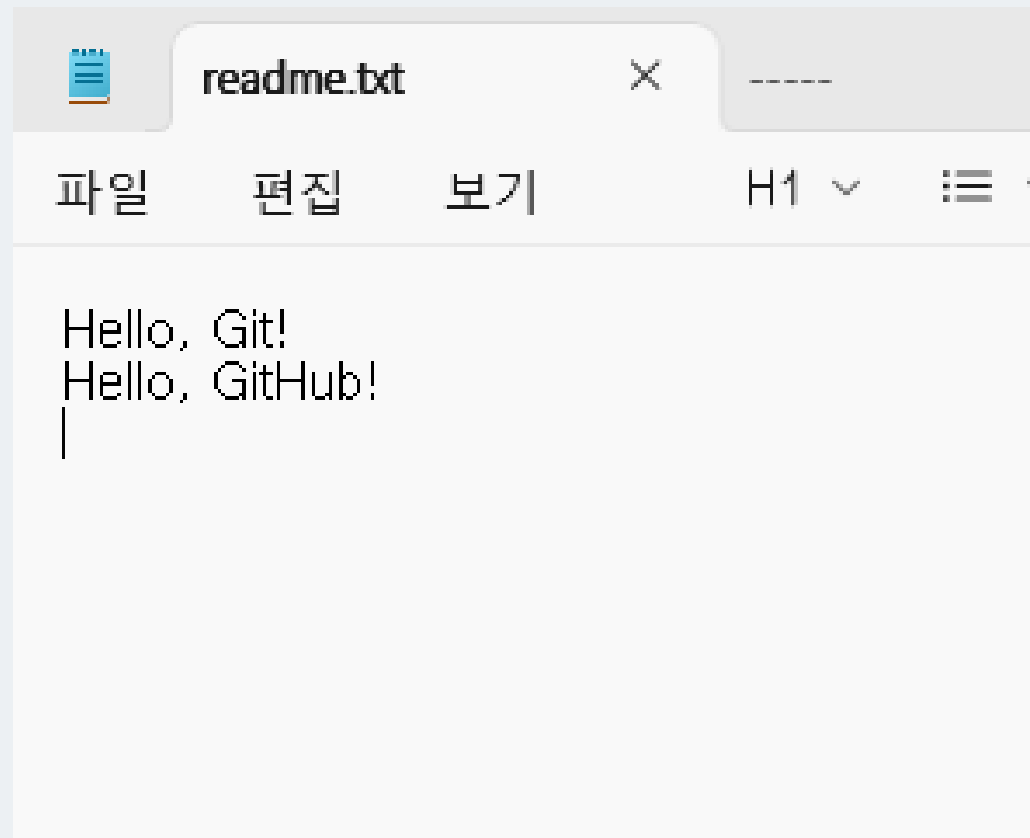
2 Hello, GitHub!

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeun>git pull origin master
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 958 bytes | 87.00 KiB/s, done.
From https://github.com/Kimchaeun/Git_Chaeun
* branch                master      -> FETCH_HEAD
  f997faf..ca31507      master      -> origin/master
Updating f997faf..ca31507
Fast-forward
 README.md/readme.txt | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

GitHub 기본 실습

6. GitHub의 변경 사항 가져오기(pull)

>>git pull origin master



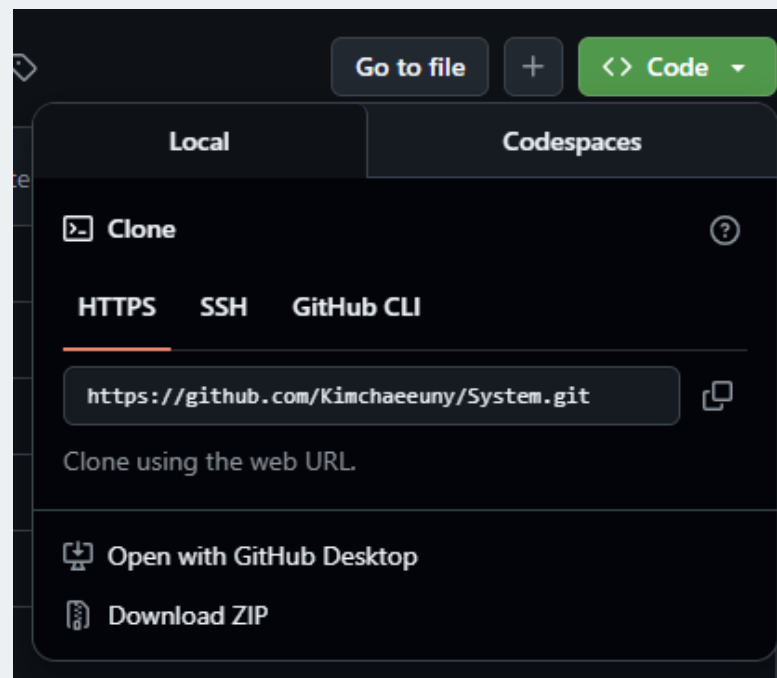
GitHub 기본 실습

7. 다른 사람 저장소 복제하기(clone)

➔ 새로운 프로젝트를 시작할 때, 다른 사람의 기존 프로젝트를 내 컴퓨터로 가져와 작업하고 싶을 때 사용

8. 임의의 다른 저장소 생성 후 이동 or 다른 사람의 저장소로 이동

9. 초록색 '<> Code'버튼 클릭 후 URL 복사



GitHub 기본 실습

10. 새로운 프로젝트를 받을 빈 폴더 만들기



11. 새로 생성한 곳에서 터미널 열기

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>cd C:\Users\82109\OneDrive\바탕 화면\GitHub_Chaeuny
C:\Users\82109\OneDrive\바탕 화면\GitHub_Chaeuny>
```

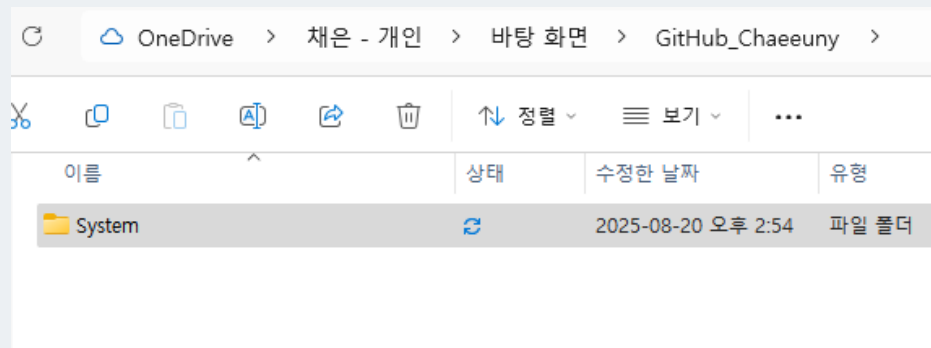
12. 복사할 깃허브 주소 붙여넣어 clone 명령어 진행

>>git clone "복사한 저장소 URL"

```
C:\Users\82109\OneDrive\바탕 화면\Git_Chaeuny>git clone https://github.com/Kimchaeuny/System.git
Cloning into 'System'...
remote: Enumerating objects: 333, done.
remote: Counting objects: 100% (158/158), done.
remote: Compressing objects: 100% (136/136), done.
remote: Total 333 (delta 72), reused 3 (delta 3), pack-reused 175 (from 1)
Receiving objects: 100% (333/333), 102.84 KiB | 2.02 MiB/s, done.
Resolving deltas: 100% (130/130), done.
```


GitHub 기본 실습

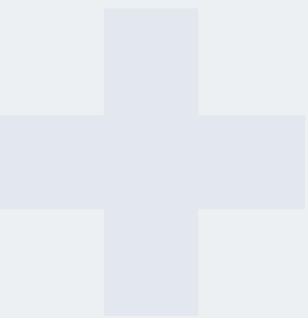
13. 결과 확인



A screenshot of the OneDrive web interface showing a list of folders and a README.md file. The breadcrumb path is 'OneDrive > 채은 - 개인 > 바탕 화면 > GitHub_Chaeeyun > System >'. The file list shows folders named .git, 0317, 0319, 0326, 0402, 0409, 0416, 0430, 0507, 0514, 0521, 0528, and a file named README.md. The table has columns: 이름, 상태, 수정한 날짜, 유형, and 크기.

이름	상태	수정한 날짜	유형	크기
.git		2025-08-20 오후 2:54	파일 폴더	
0317		2025-08-20 오후 2:54	파일 폴더	
0319		2025-08-20 오후 2:54	파일 폴더	
0326		2025-08-20 오후 2:54	파일 폴더	
0402		2025-08-20 오후 2:54	파일 폴더	
0409		2025-08-20 오후 2:54	파일 폴더	
0416		2025-08-20 오후 2:54	파일 폴더	
0430		2025-08-20 오후 2:54	파일 폴더	
0507		2025-08-20 오후 2:54	파일 폴더	
0514		2025-08-20 오후 2:54	파일 폴더	
0521		2025-08-20 오후 2:54	파일 폴더	
0528		2025-08-20 오후 2:54	파일 폴더	
README.md		2025-08-20 오후 2:54	MD 파일	

질의응답



참고자료

<https://git-scm.com/book/ko/v2>

<https://www.youtube.com/watch?v=Z6oBzXWxa8Q>

<https://sseozytank.tistory.com/41>

https://spartacodingclub.kr/blog/github_guide

<https://youtu.be/Fley6lFhlC8?si=6GvN4iD1AQNijzSq>

<https://youtu.be/HXsKNIz0VRk?si=FtmLra1qeVq2FX2v>

