Open Source SW Utilization

(524820-2)

송영상(Youngsang Song)

sw.yssong@dankook.ac.kr

Outline

- Github
  - 버전 관리(나만의 저장소)
  - 협업(collaboration)
- GitLab

Github

- **Github**
  - 깃허브(Github)는 분산 버전 관리 툴인 깃(Git)를 사용하는 프로젝트를 지원하는 웹호스팅 서비스
  - github는 **버전 관리와 협업**을 위한 코드 **웹 호스팅 플랫폼**으로, 언제, 어디서나 협업 프로젝트를 쉽게 진행할 수 있도록 돕는 역할

Github

- **Github 용어**

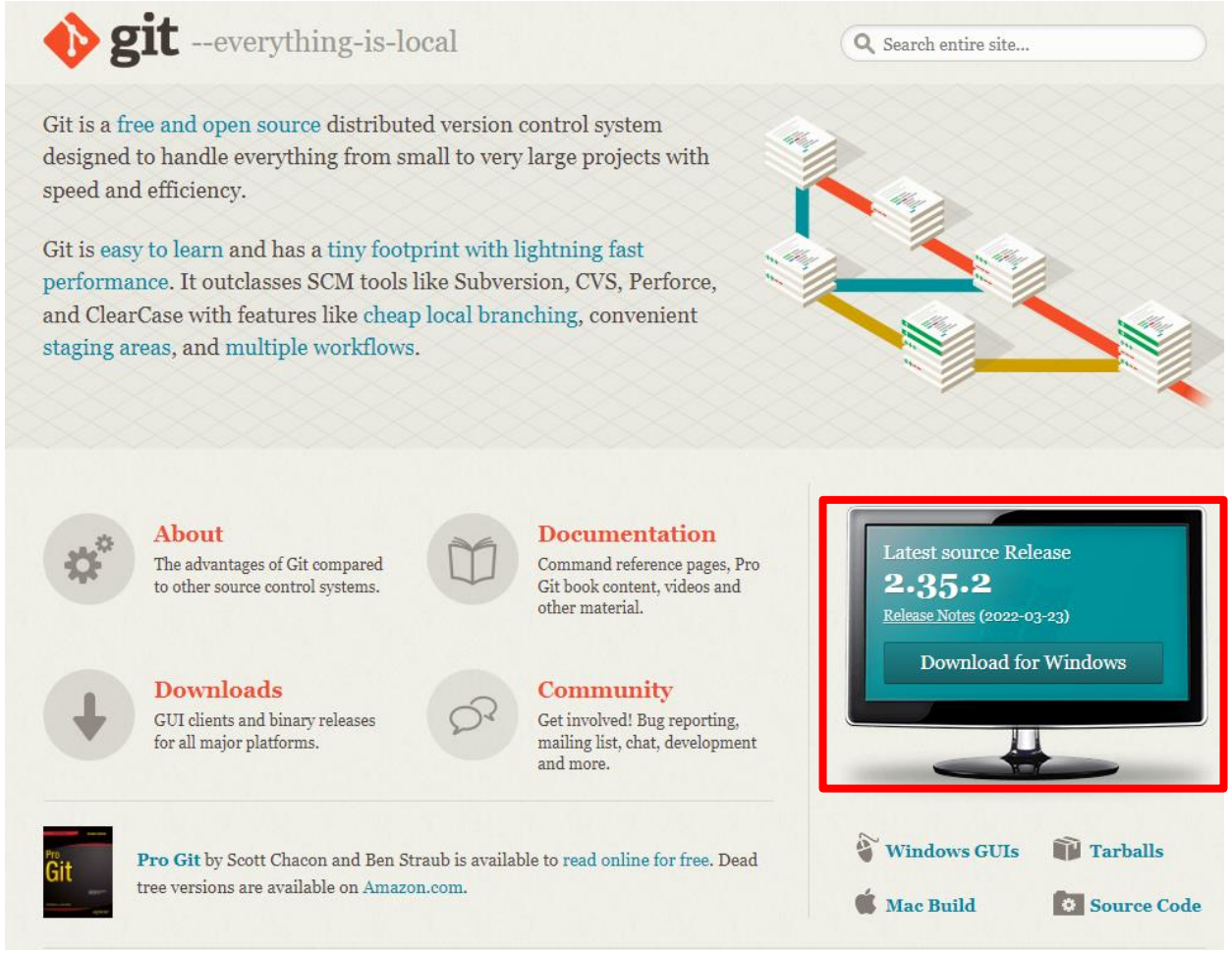
Git	코드 버전 관리 프로그램
리포 (repo, repository)	코드 저장소
브랜치 (branch)	코드의 버전 이름
마스터/메인 (master/main)	메인/디폴트 브랜치
커밋 (commit)	코드 변경사항 제출
PR (pull request)	코드 리뷰/검토 신청
머지 (merge)	변경사항을 메인 브랜치에 반영

Github

- 환경 셋팅
  - Git 설치
    - Git configuration Setting
  - VS code 설치
    - Shell Setting
  - Github 가입
    - Github가입
    - 소스 올리기

Github

- Git 설치
  - http://git-scm.com/



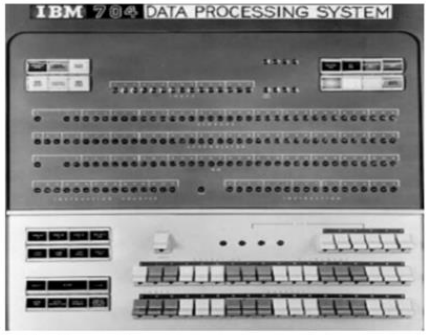
Github

- **Terminal & Shell & Bash**
  - The **terminal** is the GUI window that you see on the screen. It takes commands and shows output
  - The **shell** is the software that interprets and executes the various commands that we type in the terminal.
  - **Bash** is a particular shell.

1. **Terminal** physical input/output connected to a

http://www.istockphoto.com

2. **Console** physical port linked to a



http://www.ibm.com

3. **Mainframe** hardware controlled by

https://en.wikipedia.org

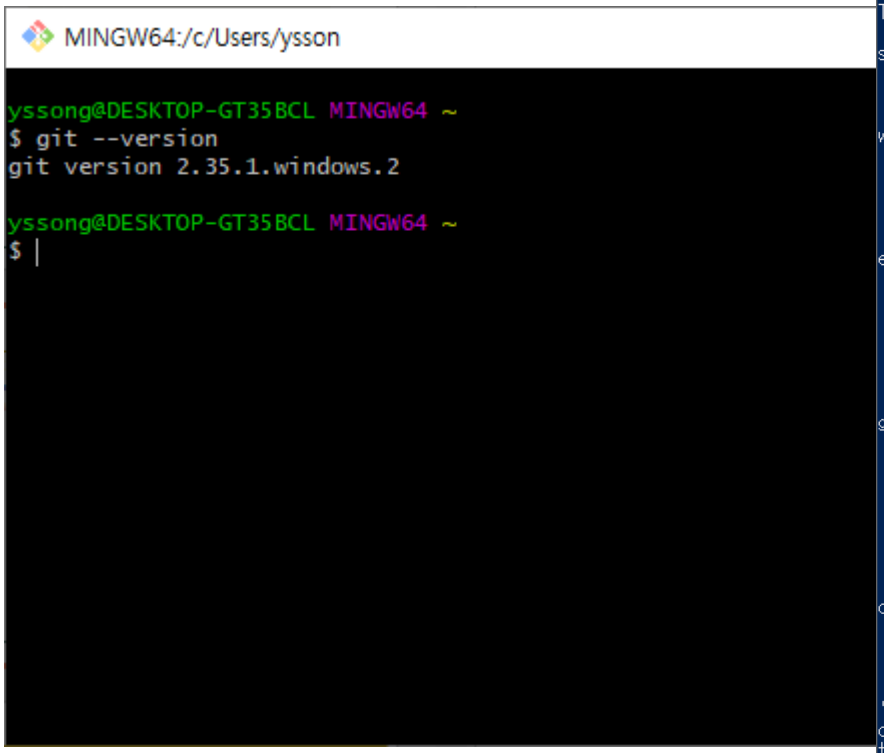
4. **Kernel** backend software accessed by

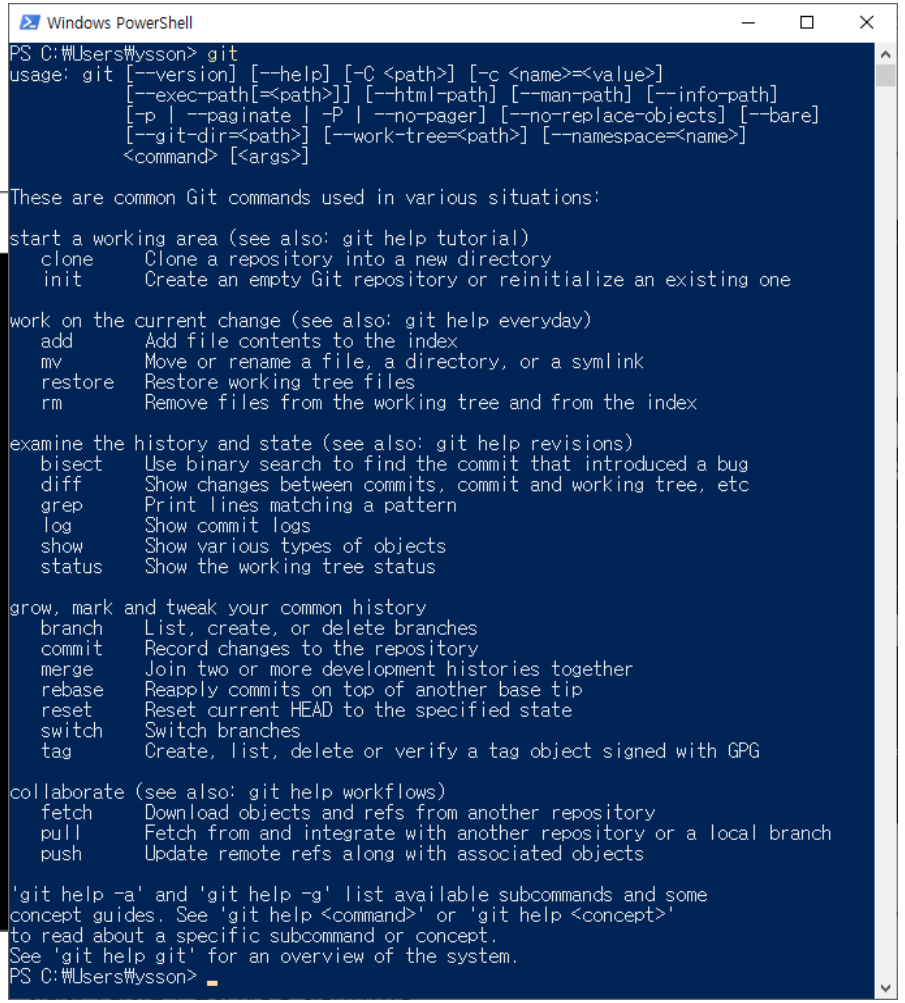
5. **Shell** software for input/output

bash one of the "modern (1989)" shells
(Like Chrome is a modern browser)

Github

- Git 설치 확인
  - Window key + R : Git Bash or PowerShell or cmd
    - git --version



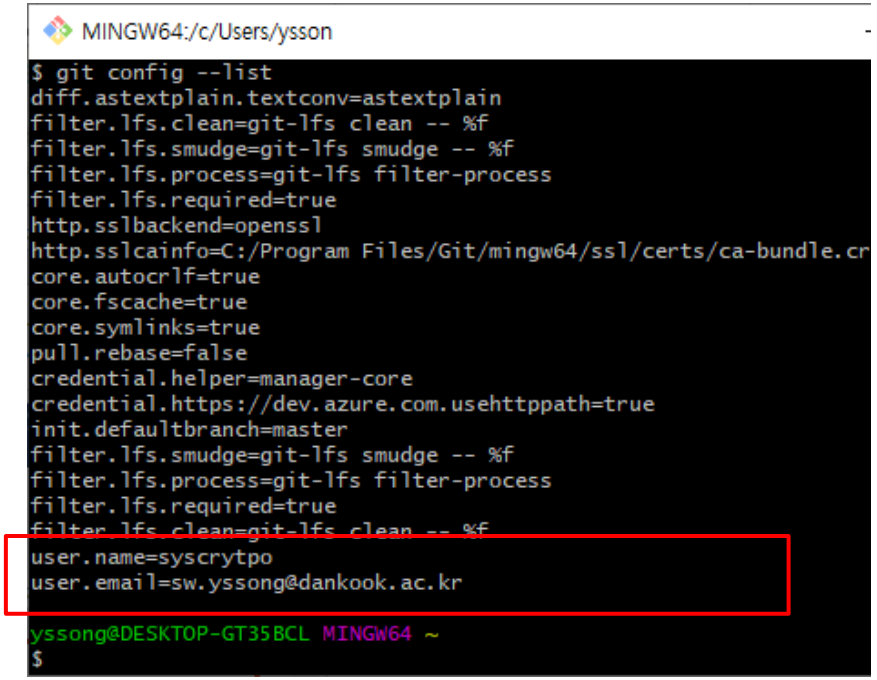


Github

- Git 초기 설정
  - 사용자명 등록
  - 메일 주소 등록

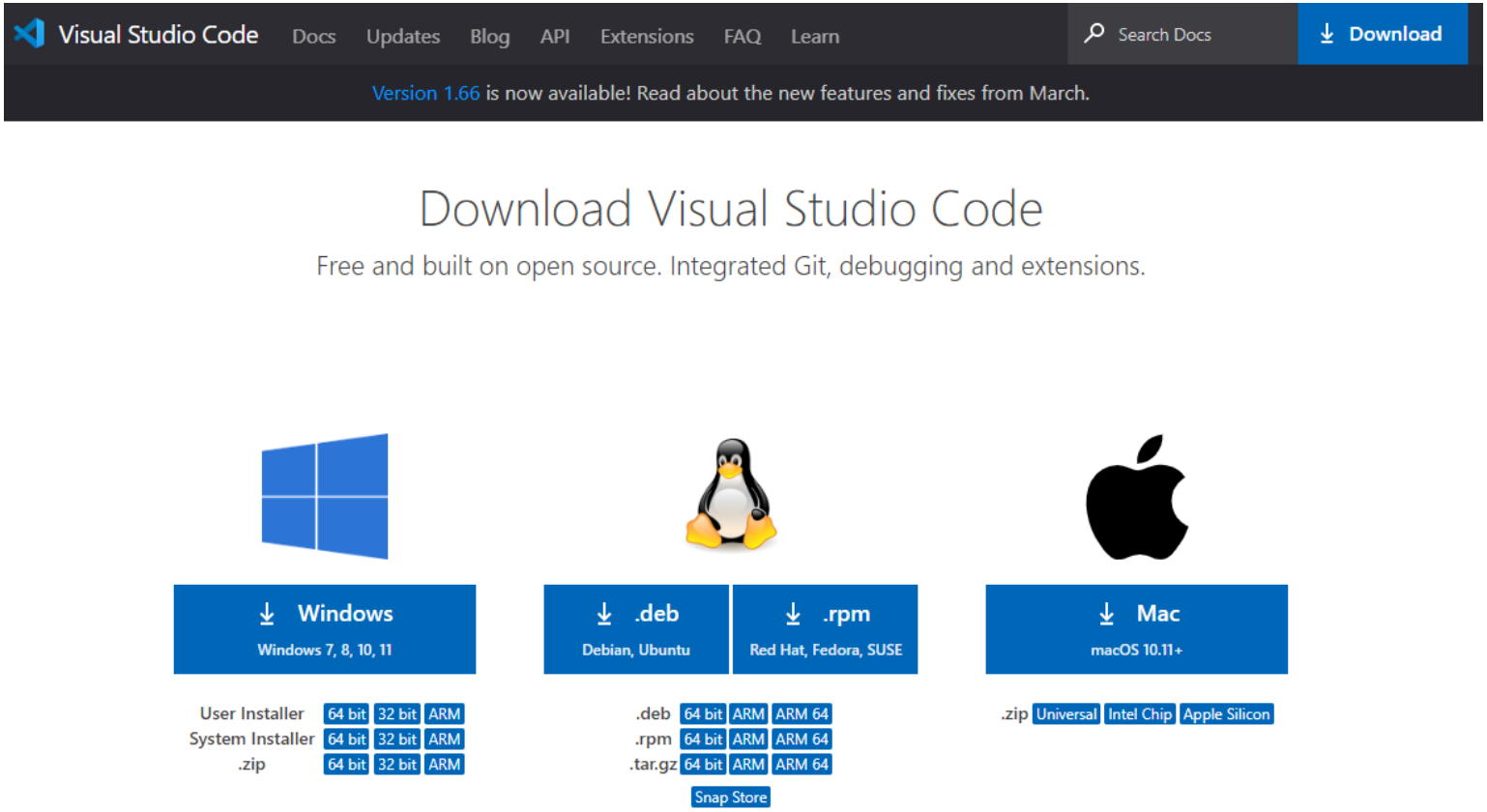
```
$ git config --global user.name "〈사용자명〉"
$ git config --global user.email "〈메일 주소〉"
```

- 설정 확인 : VS code Terminal
  - git config –list
    - 빠져나오기 :q



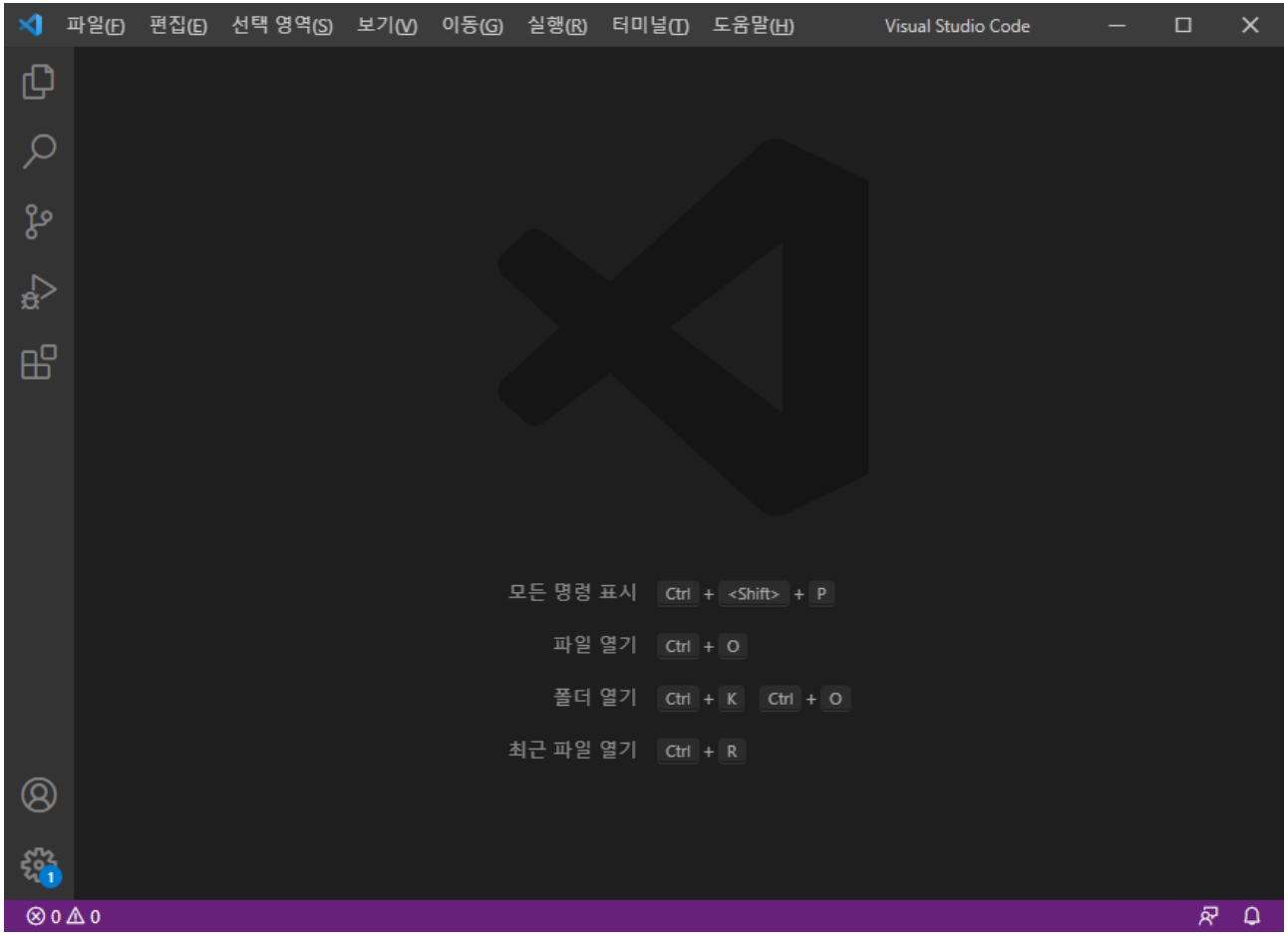
Github

- **Visual Studio Code**
  - **https://code.visualstudio.com/download**



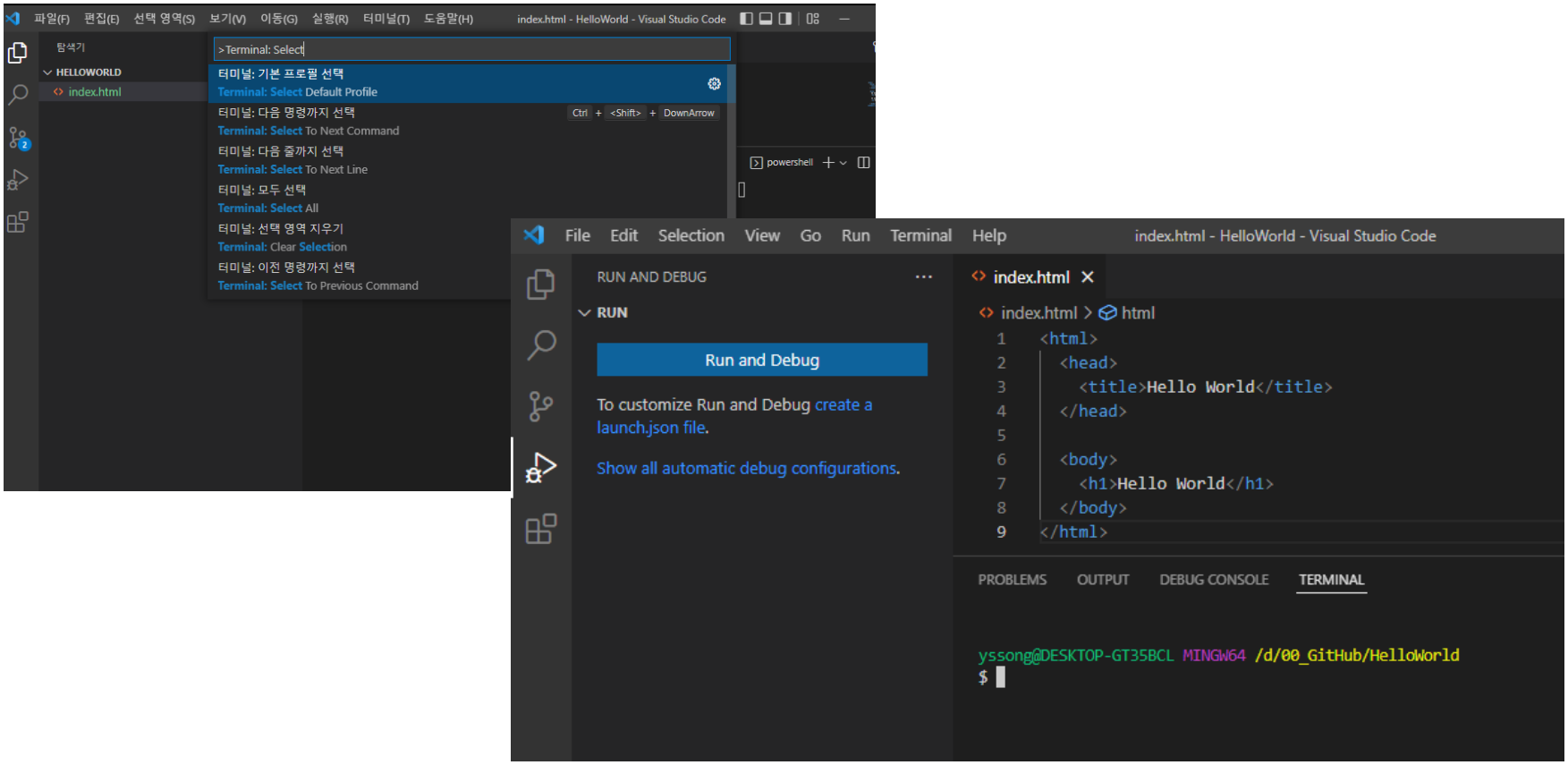
Github

- **Visual Studio Code 실행**
  - **CMD창에서 : code .**



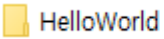
Github

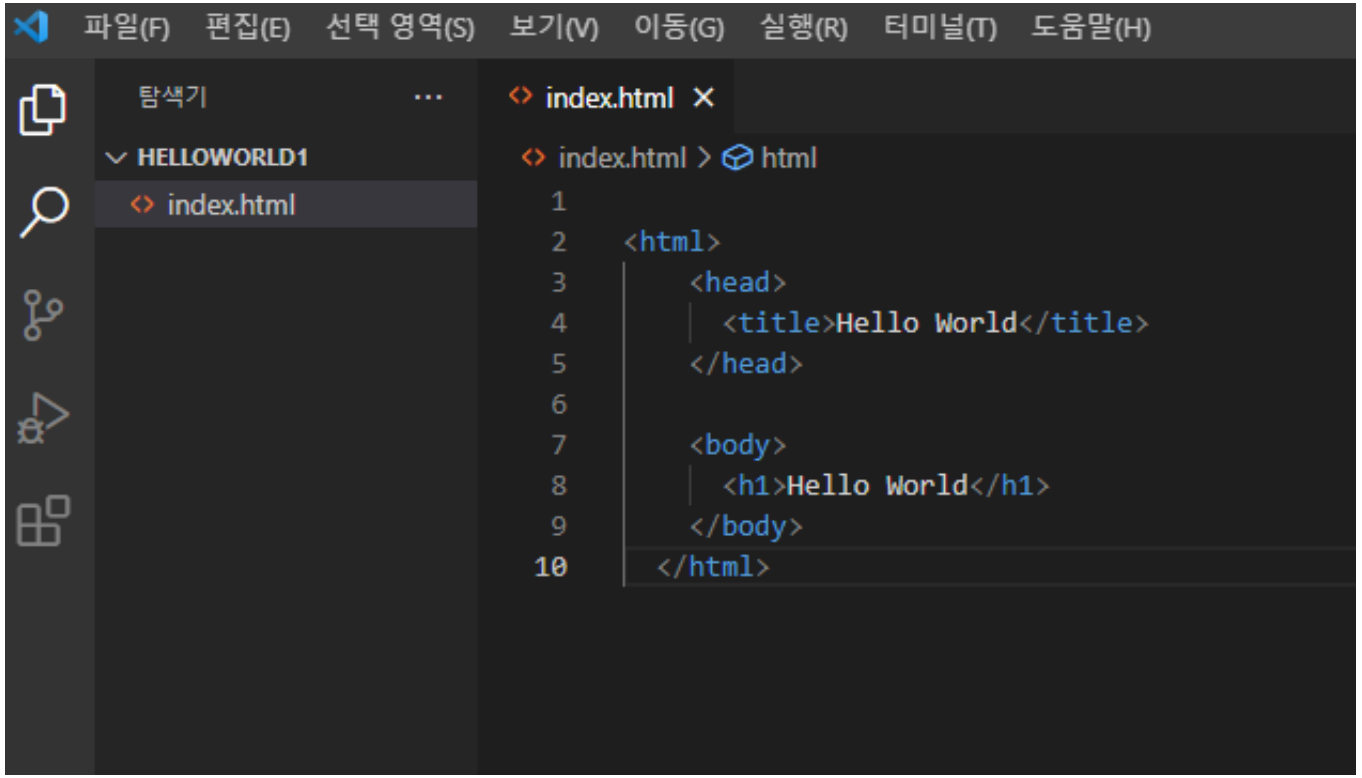
- **Terminal Shell 변경**
  - **Ctrl + Shift + P**
  - **Command Palette -> Terminal: Select Default Profile**



Github

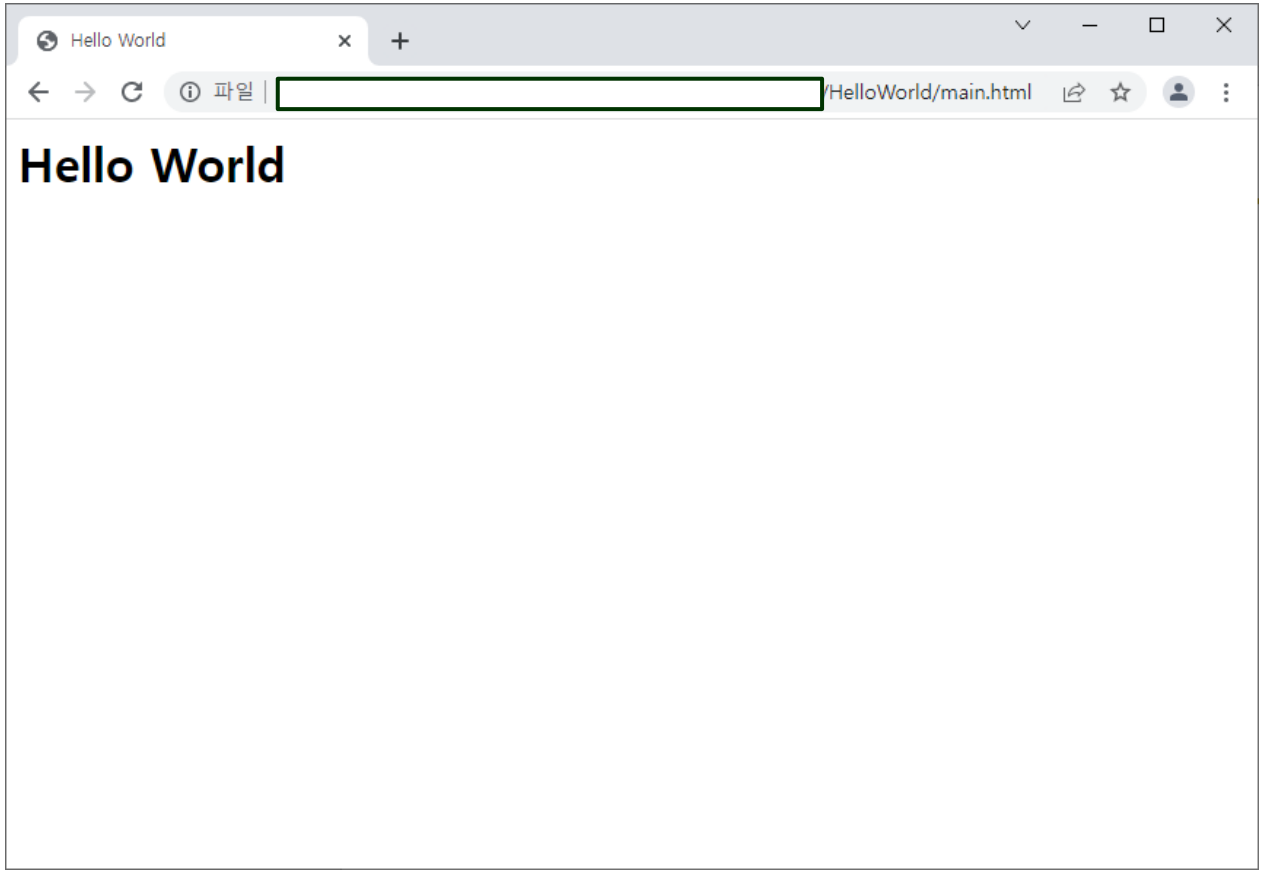
- **HelloWrold Project**
  - **Project실행을 위한 폴더 생성**
  - **VS code 파일 – 폴더열기 : 생성한 폴더 지정**
  - **파일 만들기 : index.html**





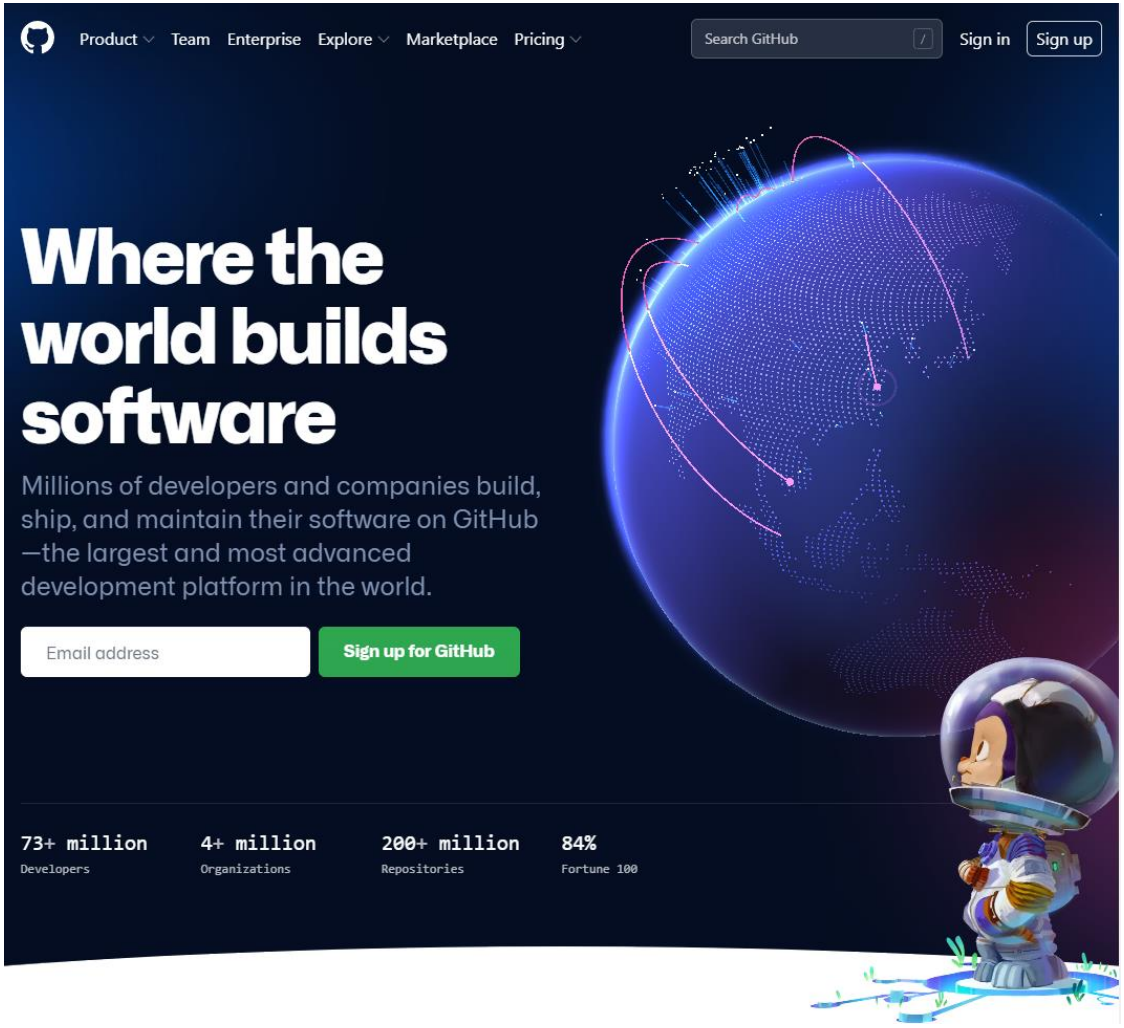
Github

- HelloWorld 실행
  - 실행 – 디버깅없이 실행 (Ctrl + F5)
  - Chrome 선택



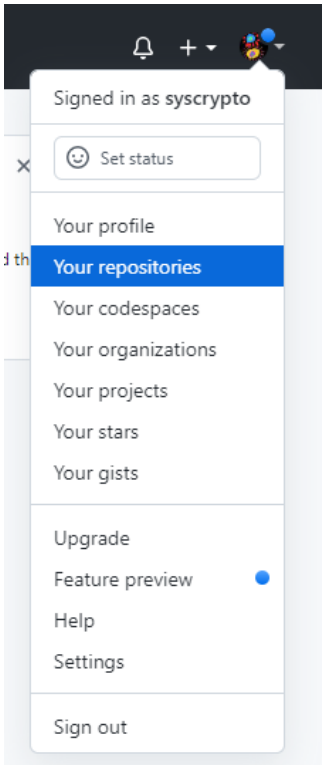
Github

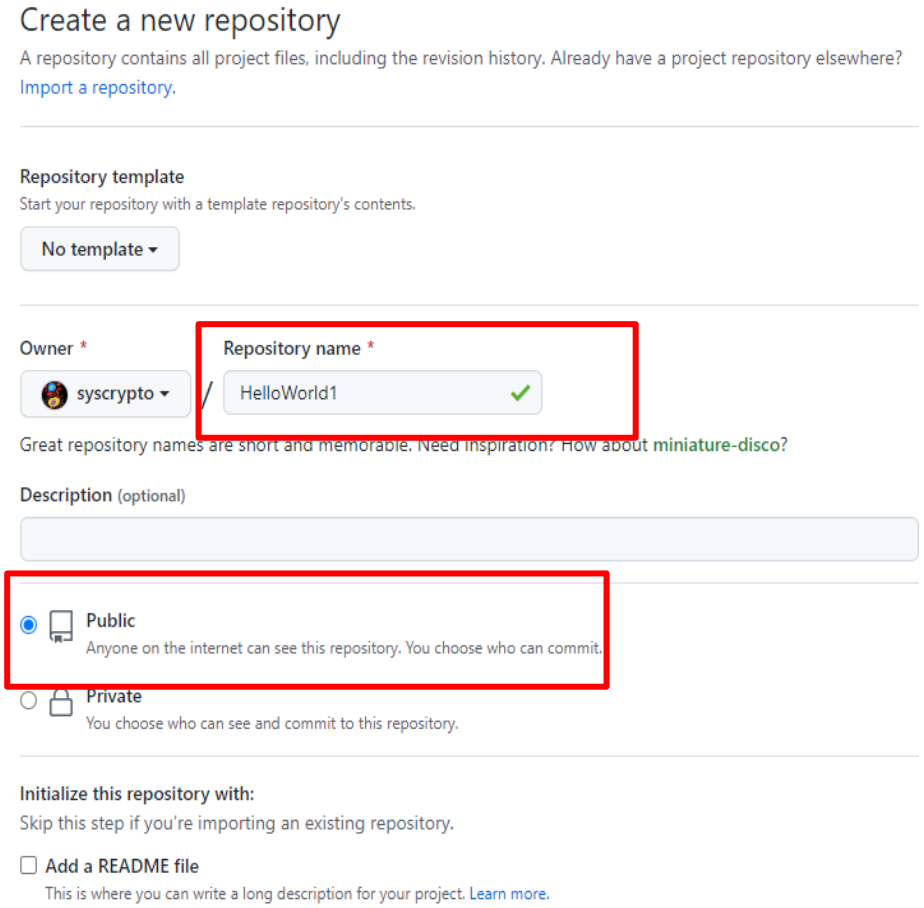
- Github가입



Github

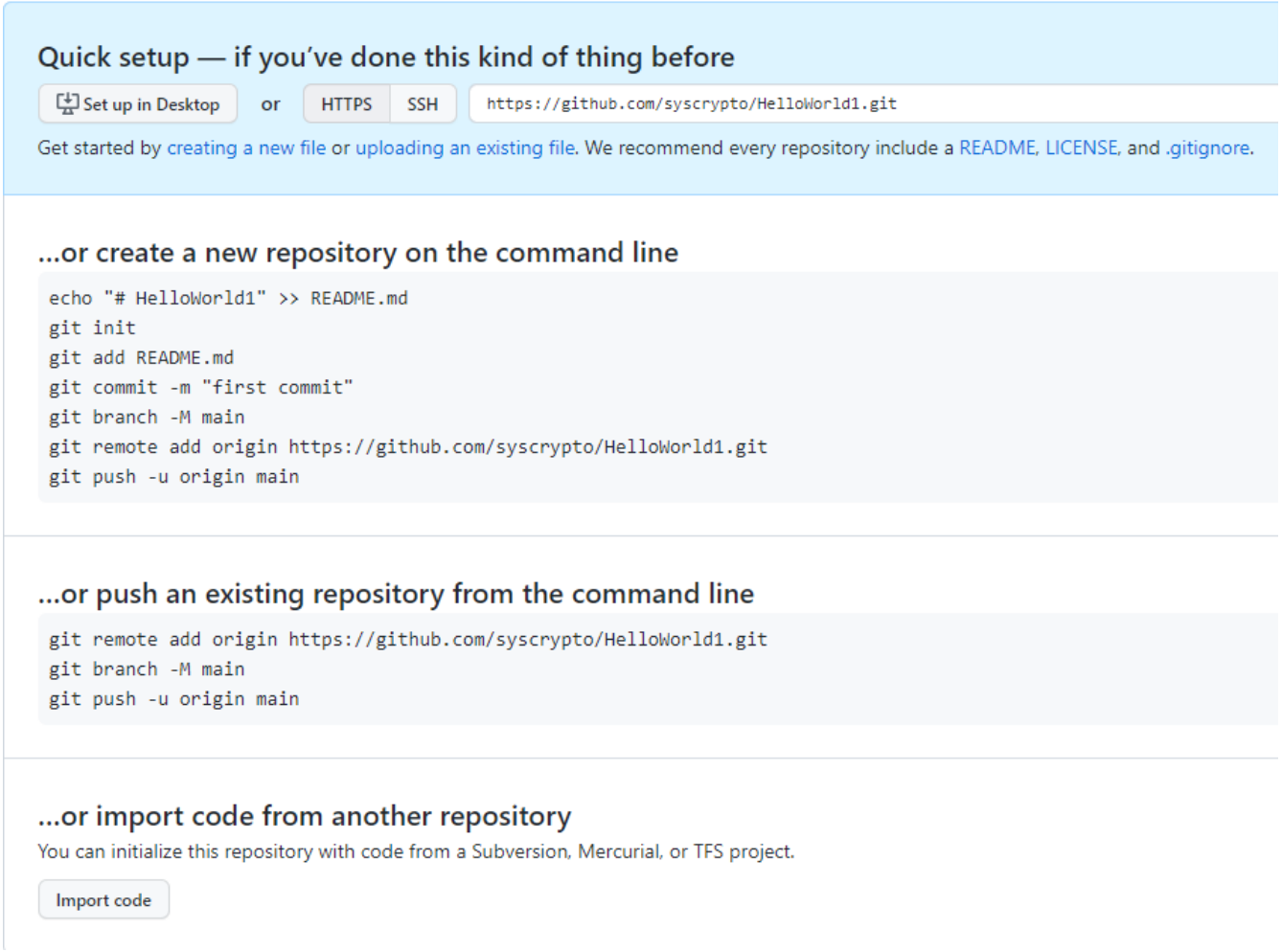
- **Repository 생성**
  - **Your repositories**
  - **New**
  - **Create repository**





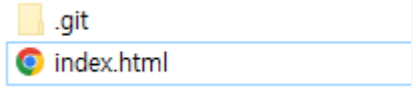
Github

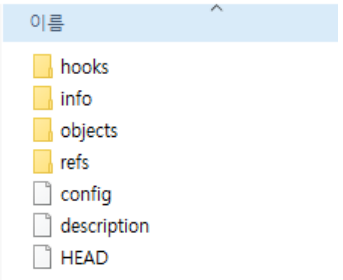
- 첫 번째 Repository 화면

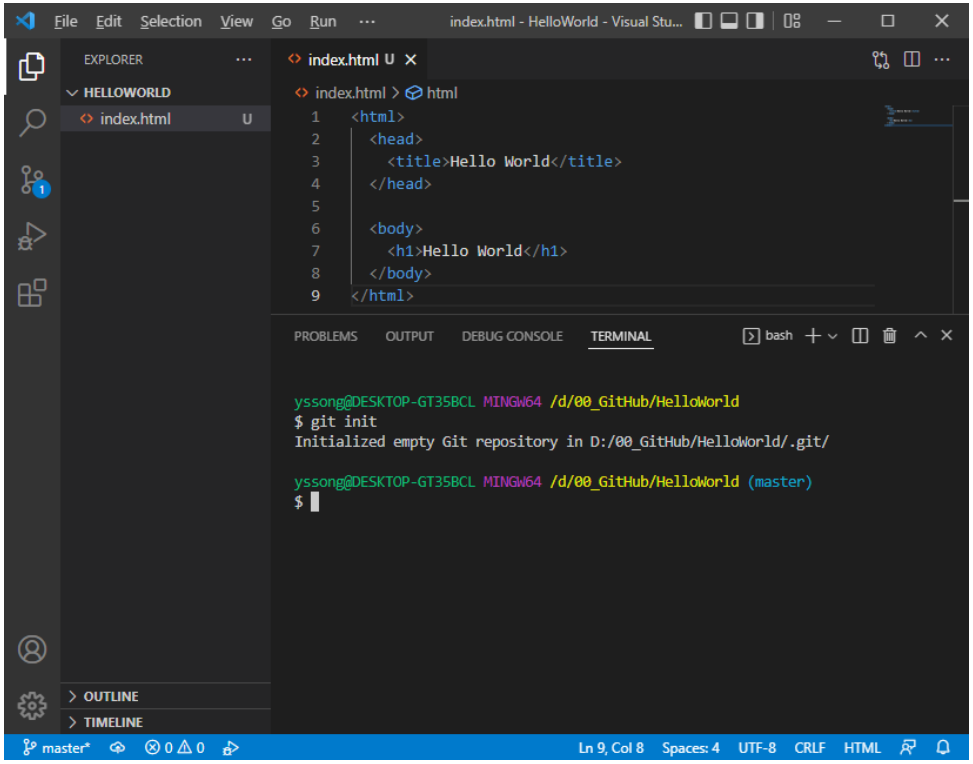


Github

- 초기화
  - 소스코드 폴더를 버전 관리를 위해 initialize함
    - git init
  - Project하위 폴더에 .git폴더 생성
    - 버전관리 정보가 저장

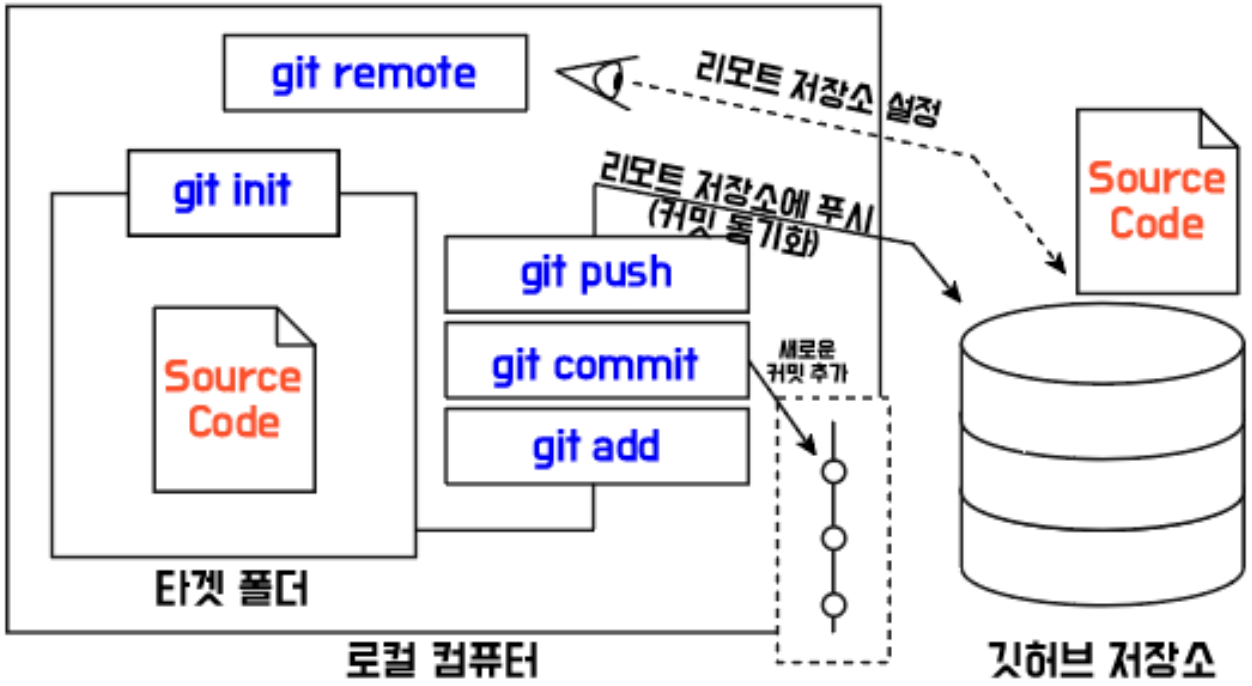






Github

- **Github 소스코드 올리기**
  - **add – commit –(github와 연결) push**
    - add 를 통해 폴더에서 모든 변경사항을 깃이 체크
    - commit 을 통해 변경사항을 새로운 commit으로 저장
    - push를 통해 github 저장소와 동기화(update 된 commit)



Github

- Github 소스코드 올리기
  - git add . : 모든 파일 올리기
    - git status : 상태 확인
  - git commit : history 만들기
    - git commit –m “history 내용”
  - Github와 연결 -> Next Page
  - git push origin master : 파일 보내기

```
$ git add .
$ git commit -m "커밋에 대한 간단한 설명 메세지"
$ git push origin master
```

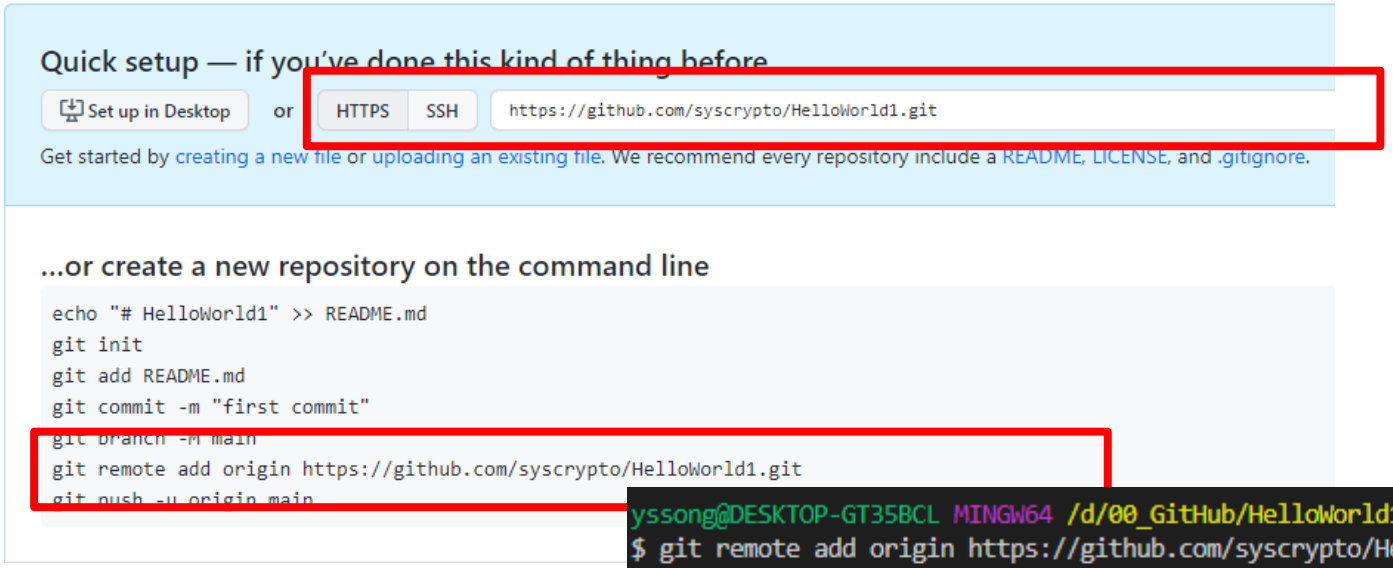
```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWo
$ git add .

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWo
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
```

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/Hello
$ git commit -m "first commit"
[master (root-commit) 3174985] first commit
 1 file changed, 9 insertions(+)
 create mode 100644 index.html
```

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (mas
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 285.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/syscrypto/HelloWorld1.git
 * [new branch]      master -> master
```

Github

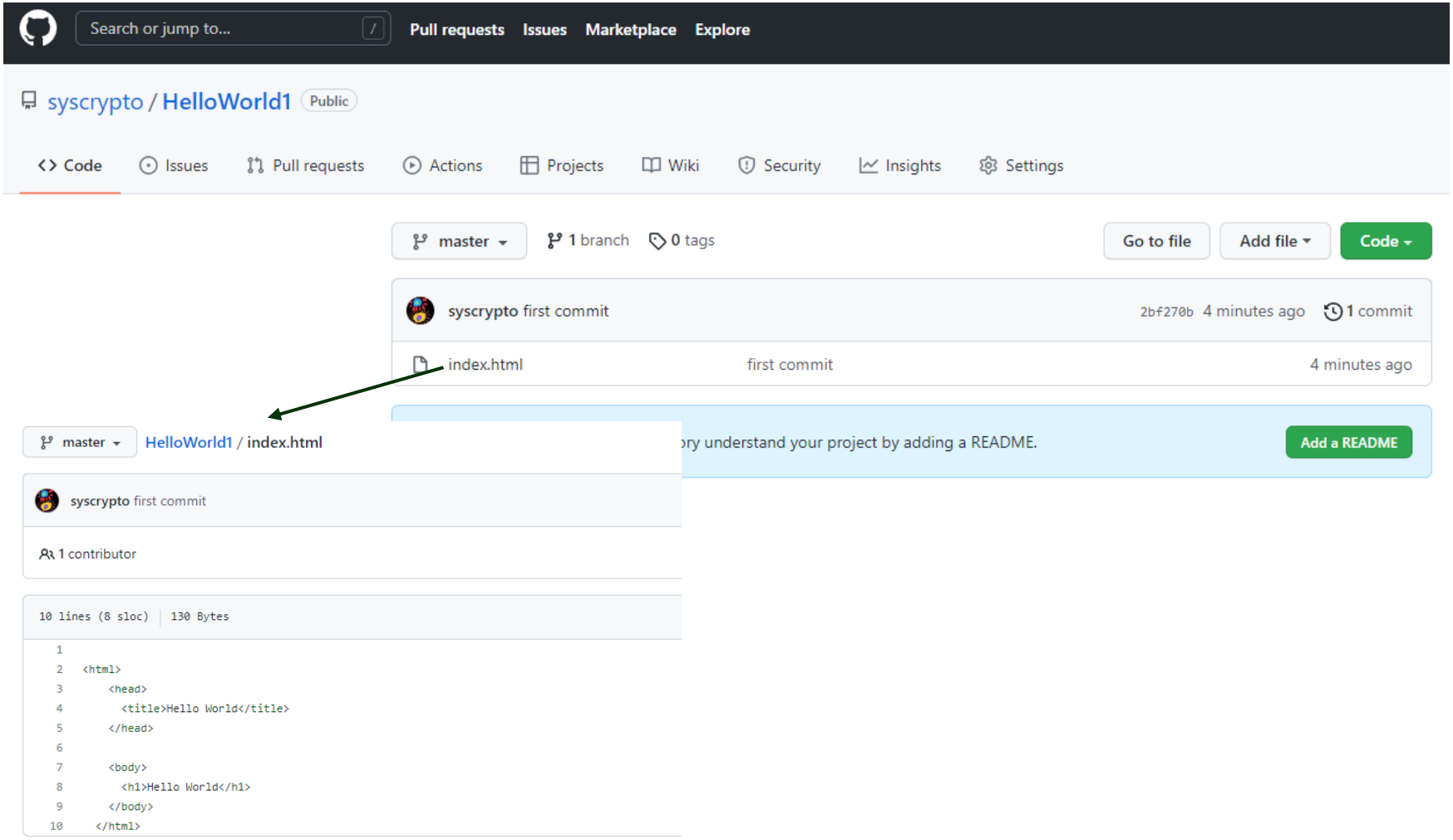
- **Github저장소와 연결**
  - 폴더와 github 저장소의 remote 주소를 연동
    - git remote add *origin* https://github.com/ ...(복사한 주소를 붙여넣기)
    - **git remote –v : 연결 확인**
  - origin : github저장소에 업로드, 다른 주소로도 등록 가능




```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git remote add origin https://github.com/syscrypto/HelloWorld1.git

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git remote -v
origin  https://github.com/syscrypto/HelloWorld1.git (fetch)
origin  https://github.com/syscrypto/HelloWorld1.git (push)
```

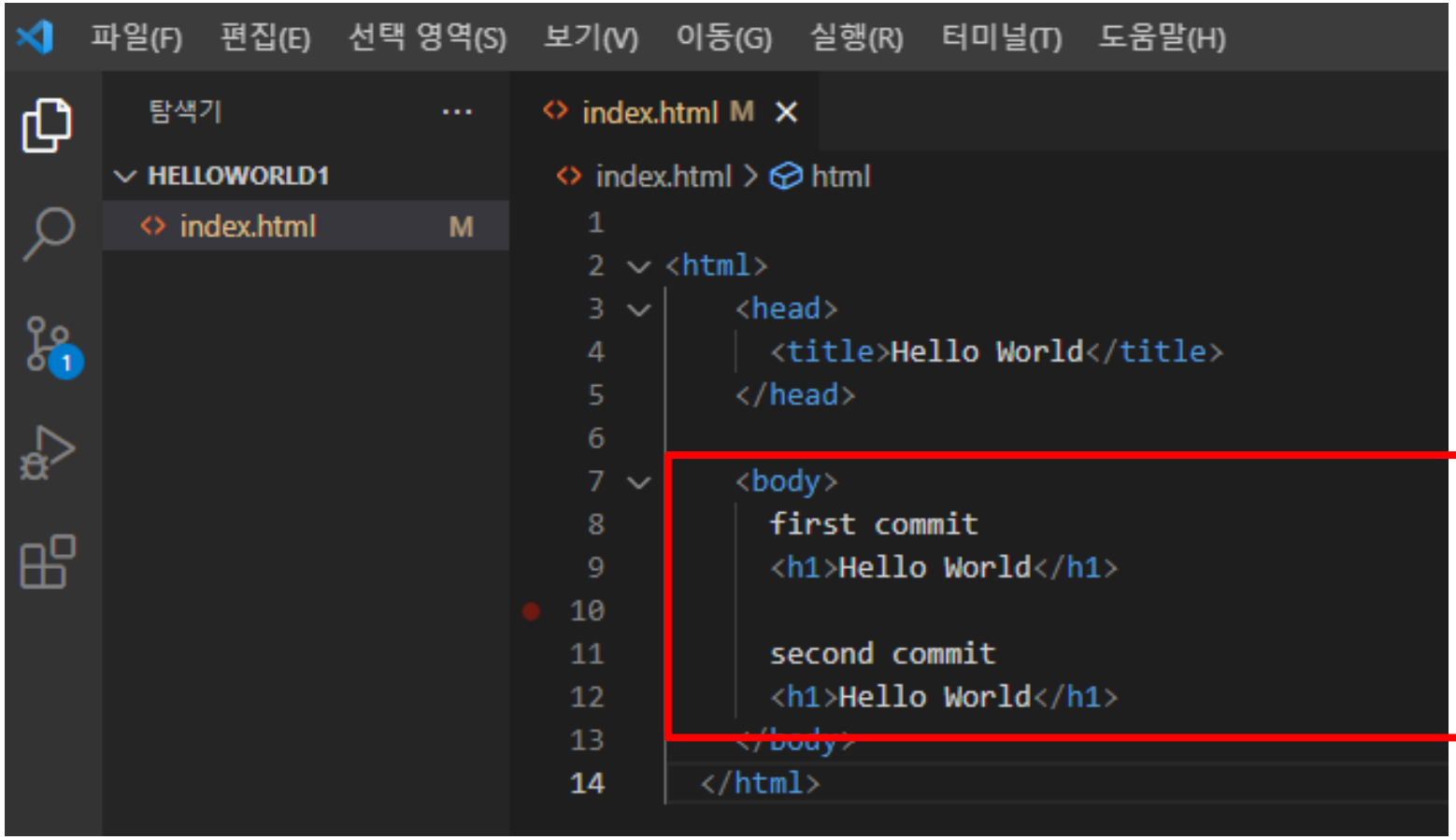
Github

- **Github Repository 확인**



Github

- 파일 수정


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
탐색기
HELLOWORLD1
index.html M
index.html M
index.html > html
1
2 <html>
3 <head>
4 <title>Hello World</title>
5 </head>
6
7 <body>
8 first commit
9 <h1>Hello World</h1>
10
11 second commit
12 <h1>Hello World</h1>
13 </body>
14 </html>

Github

- 수정 파일 적용
  - git add .

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git add .

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html
```

  - git commit –m “second commit”

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/He
$ git commit -m "second commit"
[master f6bbbf0] second commit
 1 file changed, 4 insertions(+)
```

  - git push origin master

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/syscrypto/HelloWorld1.git
   2bf270b..f6bbbf0  master -> master
```

Github

- 적용 확인

master ▾ HelloWorld1 / index.html

syscrypto first commit

1 contributor

10 lines (8 sloc) | 130 Bytes

```
1
2   <html>
3       <head>
4         <title>Hello World</title>
5       </head>
6
7       <body>
8         <h1>Hello World</h1>
9       </body>
10    </html>
```

master ▾ 1 branch 0 tags

syscrypto second commit

index.html second commit

master ▾ HelloWorld1 / index.html

syscrypto second commit

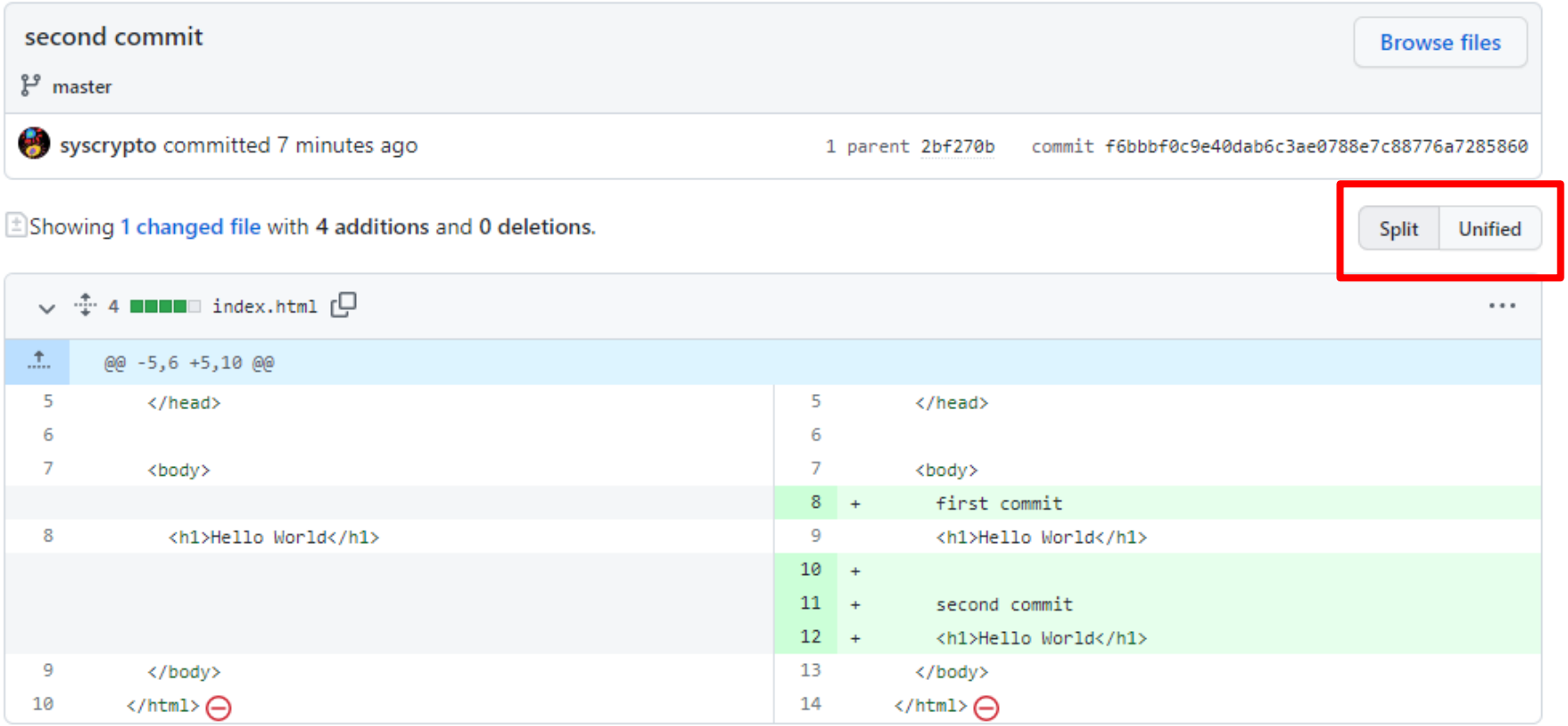
1 contributor

14 lines (11 sloc) | 204 Bytes

```
1
2   <html>
3       <head>
4         <title>Hello World</title>
5       </head>
6
7       <body>
8         first commit
9         <h1>Hello World</h1>
10
11        second commit
12        <h1>Hello World</h1>
13      </body>
14    </html>
```

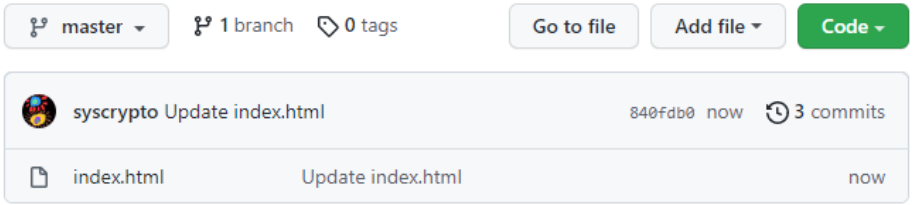
Github

- 변경사항 표시
  - second commit click -> Split/Unified



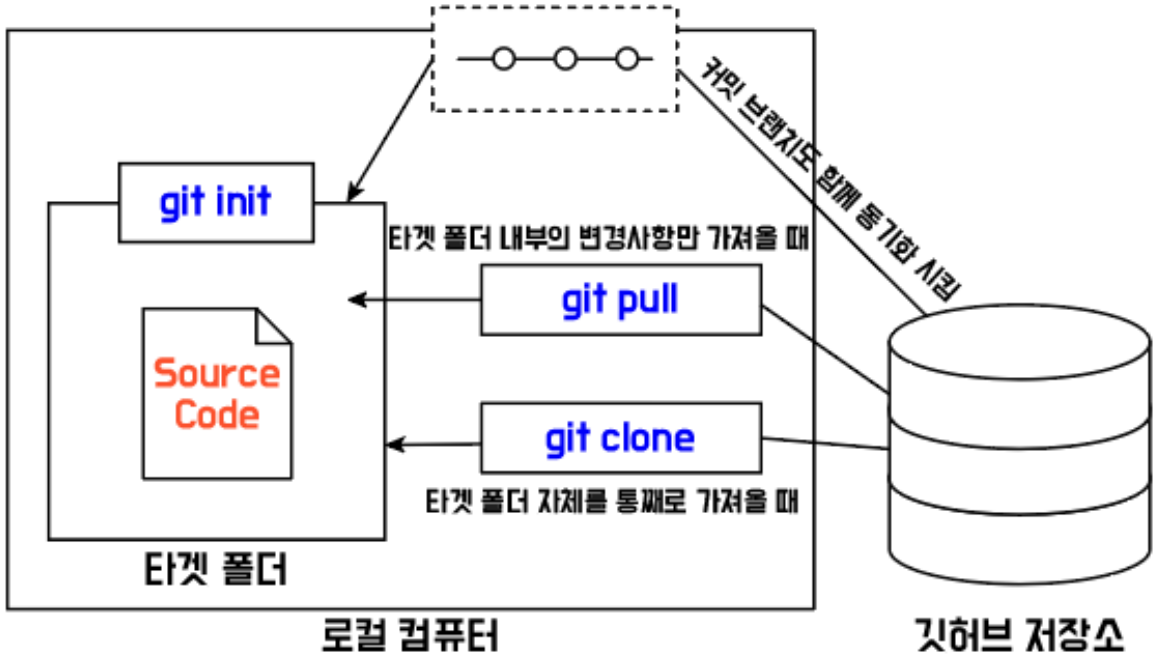
Github

- Github에서 소스 수정



Github

- 소스 가져오기
  - clone : 모든 소스를 가져오기
    - git clone 주소(remote 주소)
  - pull : 수정된 사항만 가져오기
    - git pull origin master



Github

- 수정된 파일 갖고 오기
  - git pull origin master

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 698 bytes | 6.00 KiB/s, done.
From https://github.com/syscrypto/HelloWorld1
 * branch            master     -> FETCH_HEAD
   f6bbbf0..840fdb0  master     -> origin/master
Updating f6bbbf0..840fdb0
Fast-forward
 index.html | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)
```

index.html > ...

```
1
2  <html>
3      <head>
4          <title>Hello World</title>
5      </head>
6
7      <body>
8          first commit
9          <h1>Hello World</h1>
10
11         second commit
12         <h1>Hello World</h1>
13
14         Modify line
15         <h1>Hello World</h1>
16
17     </body>
18 </html>
19
```

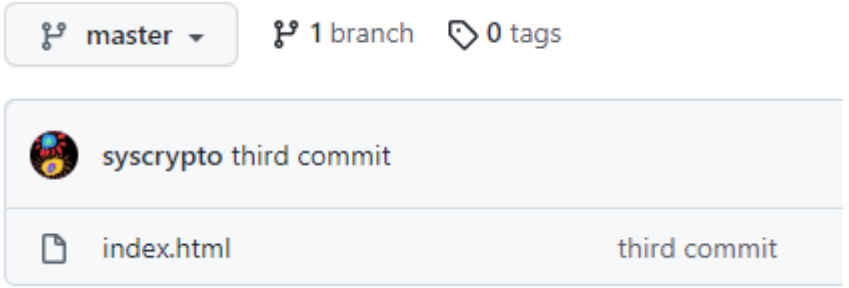
Github

- **Third commit**

index.html > html > body

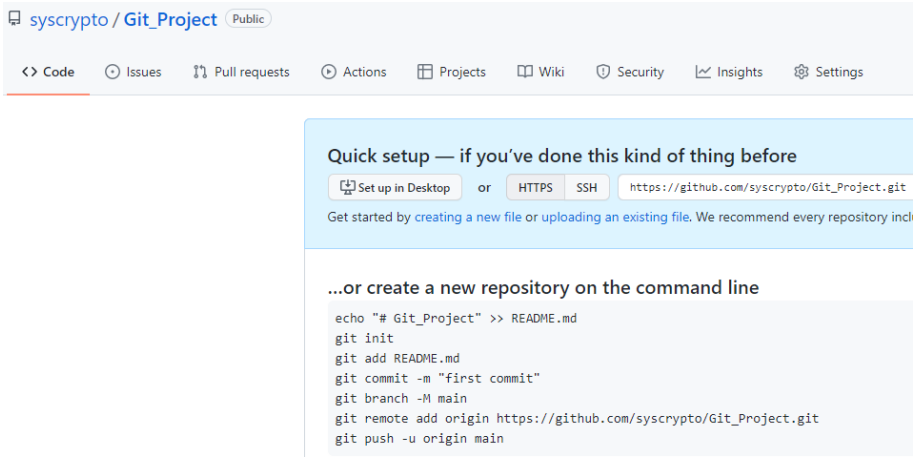
```
1
2   <html>
3       <head>
4         <title>Hello World</title>
5       </head>
6
7       <body>
8         first commit
9         <h1>Hello World</h1>
10
11        second commit
12        <h1>Hello World</h1>
13
14        Modify line
15        <h1>Hello World</h1>
16
17        third commit
18        <h1>Hello World</h1>
19
20      </body>
21    </html>
```

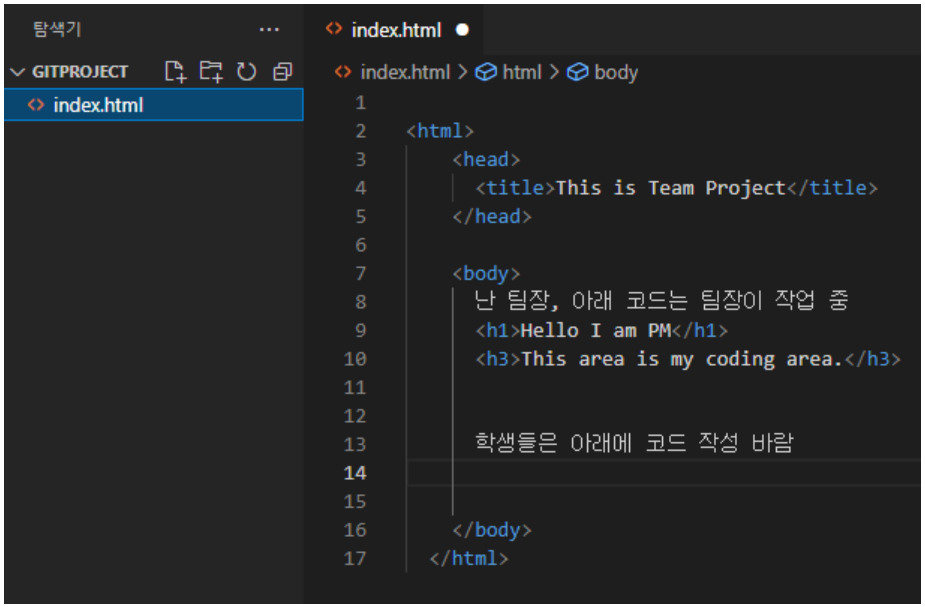
```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git add .
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html


yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git commit -m "third commit"
[master f1cb1f0] third commit
 1 file changed, 3 insertions(+)

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/HelloWorld1 (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/syscrypto/HelloWorld1.git
   840fdb0..f1cb1f0  master -> master
```



Github

- 협업(Collaboration)
  - Github Repository 생성
    - Repository name : Git_Project
    - Public
  - VS code Project
    - GitProject
      - » Index.html





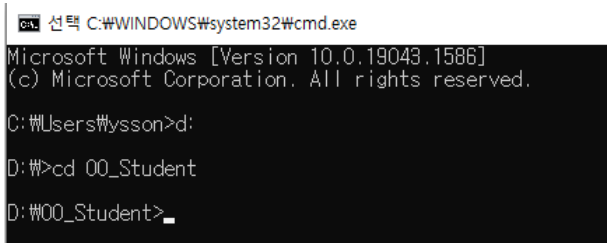
Github

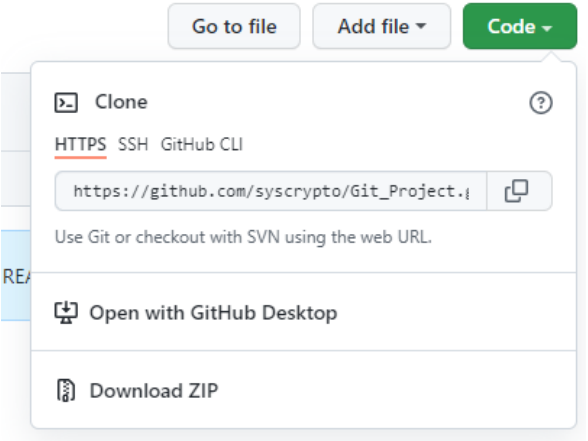
- Github파일 올리기
  - git add .
  - git commit –m “first commit”
  - git remote add origin 복사 주소
  - git push origin master

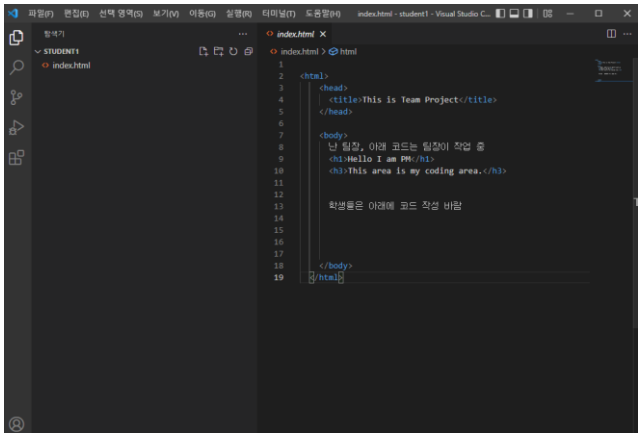
```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git remote add origin https://github.com/syscrypto/Git_Project.git

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 425 bytes | 425.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/syscrypto/Git_Project.git
 * [new branch]      master -> master
```

Github

- 학생과 협업
  - 학생이 Github에서 코드 가져오기
    - Local 컴퓨터에 받을 위치 선택
      - » Window key + R, 층
    - github 주소에서 주소 복사
      - » https://github.com/Github ID/Git_Project
    - 내 컴퓨터로 clone
      - » **Git clone 복사한 주소 저장할 폴더**
    - 코드 확인
      - » 저장한 폴더로 이동하여 **code .** 타이핑







```
D:₩OO_Student>git clone https://github.com/syscrypto/Git_Project.git student1
Cloning into 'student1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

D:₩OO_Student>
```

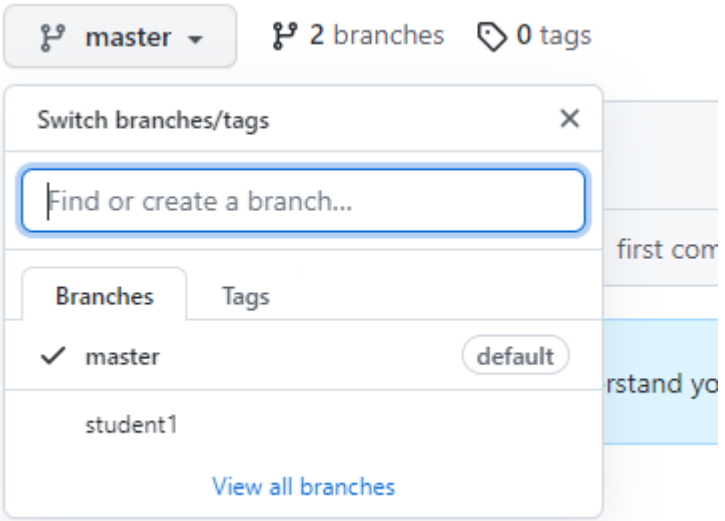
Github

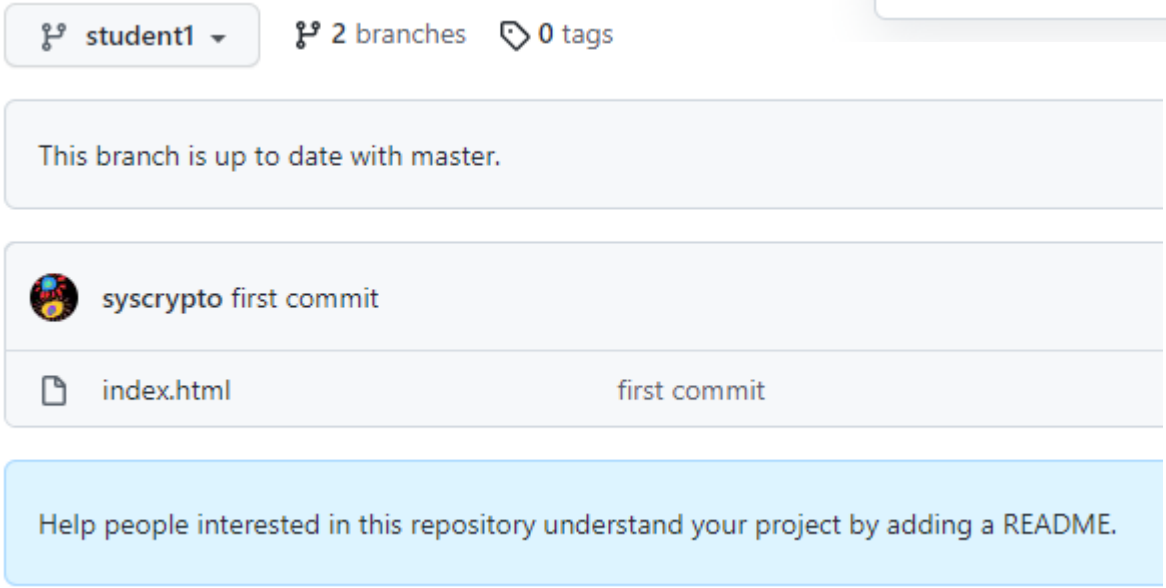
- Student1의 작업
  - Code 추가 작업
  - Terminal에서 push 하기
    - git add .
    - git commit –m “student1 first commit”
    - Branch 생성 : git checkout –b student1
    - git push origin student1

```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_Student/student1 (master)
$ git add .

yssong@DESKTOP-GT35BCL MINGW64 /d/00_Student/student1 (master)
$ git commit -m "student1 first commit"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

yssong@DESKTOP-GT35BCL MINGW64 /d/00_Student/student1 (master)
$ git checkout -b student1
Switched to a new branch 'student1'

yssong@DESKTOP-GT35BCL MINGW64 /d/00_Student/student1 (student1)
$ git push origin student1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'student1' on GitHub by visiting:
remote:      https://github.com/syscrypto/Git_Project/pull/new/student1
remote:
To https://github.com/syscrypto/Git_Project.git
 * [new branch]      student1 -> student1
```

Github

- **Branch**
  - **master**
  - **student1 (student1이 생성)**





Github

- **Student1의 작업 push 후**

master Git_Project / index.html

syscrypto first commit

1 contributor

19 lines (11 sloc) | 315 Bytes

```
1
2   <html>
3       <head>
4         <title>This is Team Project</title>
5       </head>
6
7       <body>
8         난 팀장, 아래 코드는 팀장이 작업 중
9         <h1>Hello I am PM</h1>
10        <h3>This area is my coding area.</h3>
11
12
13        학생들은 아래에 코드 작성 바람
14
15
16
17
18      </body>
19    </html>
```

student1 Git_Project / index.html

syscrypto student1 second commit

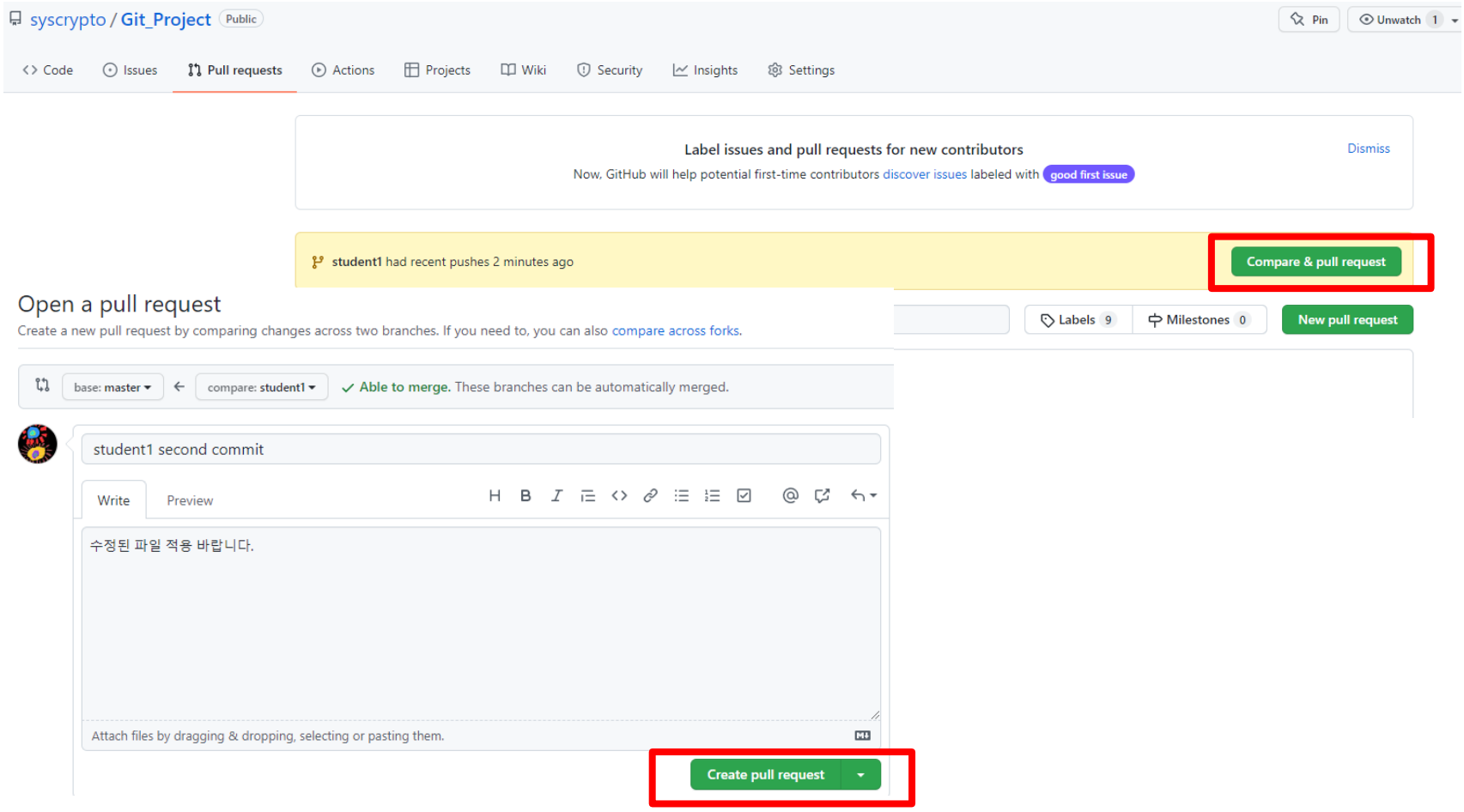
1 contributor

21 lines (14 sloc) | 424 Bytes

```
1
2   <html>
3       <head>
4         <title>This is Team Project</title>
5       </head>
6
7       <body>
8         난 팀장, 아래 코드는 팀장이 작업 중
9         <h1>Hello I am PM</h1>
10        <h3>This area is my coding area.</h3>
11
12
13        학생들은 아래에 코드 작성 바람
14
15        student1이 작업합니다.
16        <h4>Hello I am student1</h4>
17        무슨 작업을 해야할까요....^^
18
19
20      </body>
21    </html>
```

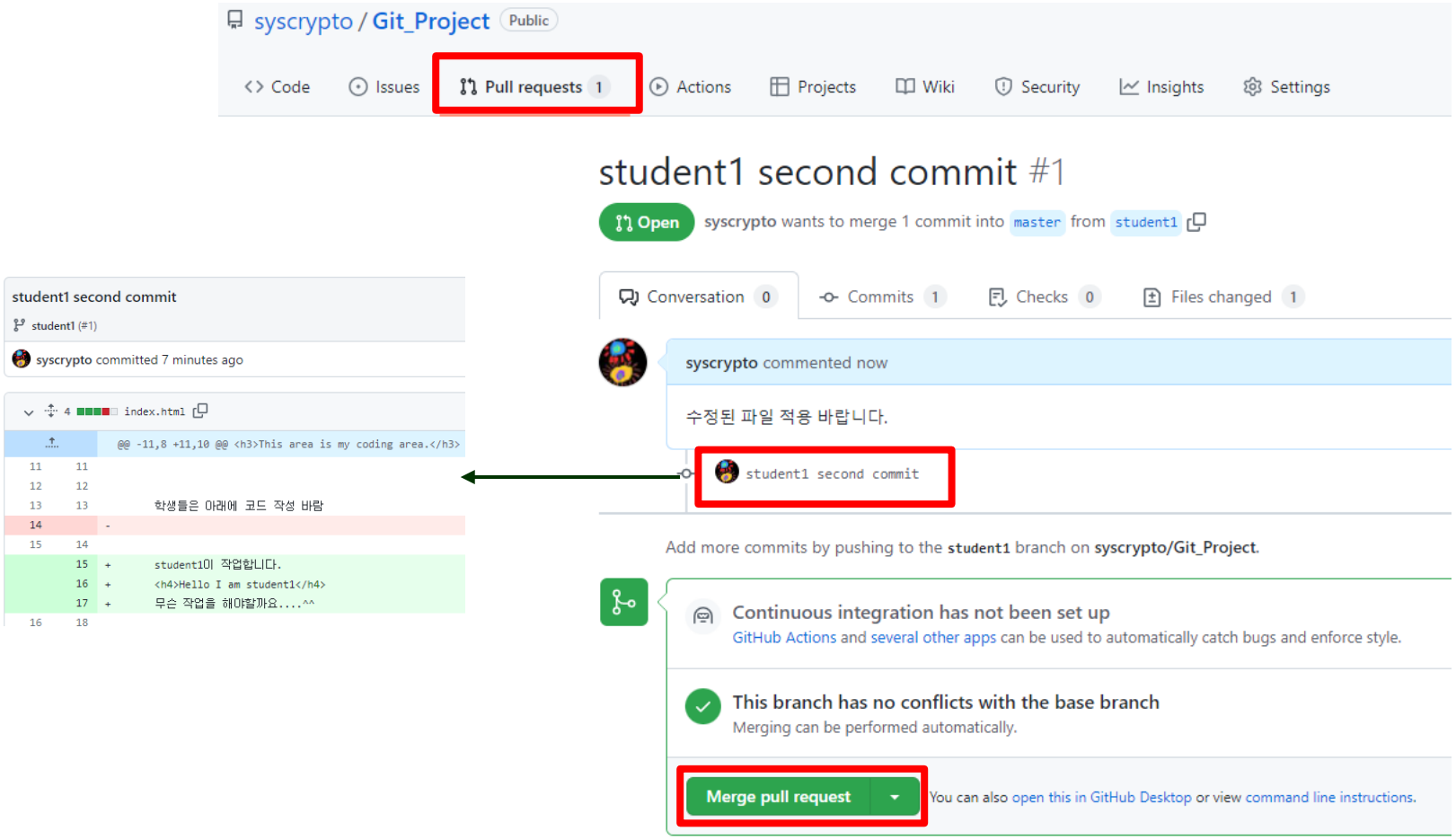
Github

- **Pull requests 요청**
  - **기존 소스에 merge 요청**



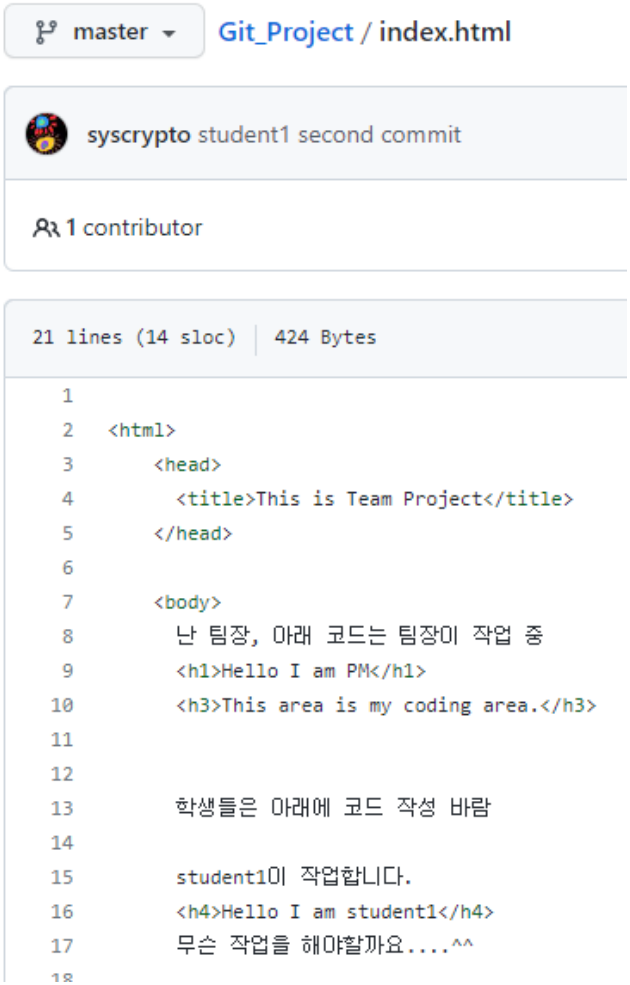
Github

- **Merge**
  - **Main source에 합쳐지기 때문에 신중하게 선택**



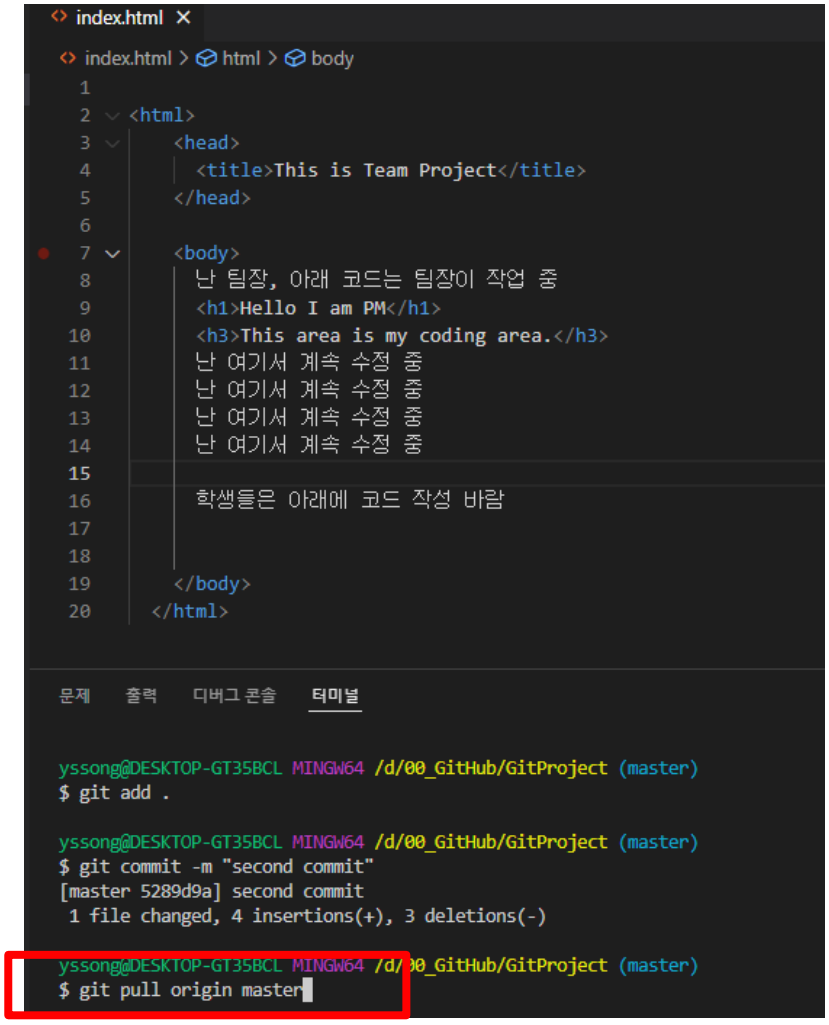
Github

- **Merge후 master code에 적용 화면**

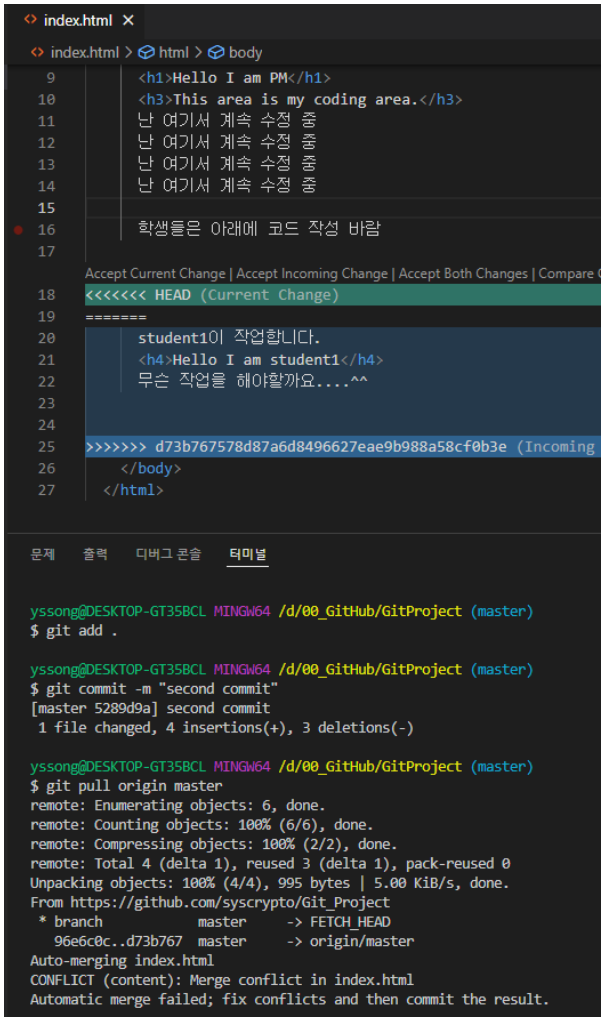


Github

- PM의 Vs_code와 github 수정되 파일 동기화 : pull



<동기화 전 화면>

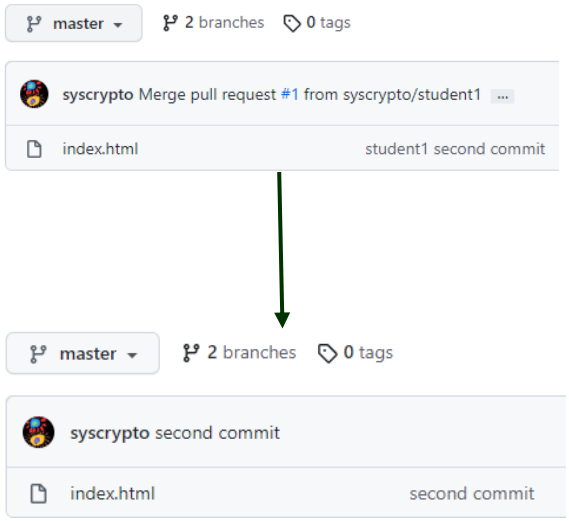


<동기화 후 화면>

Github

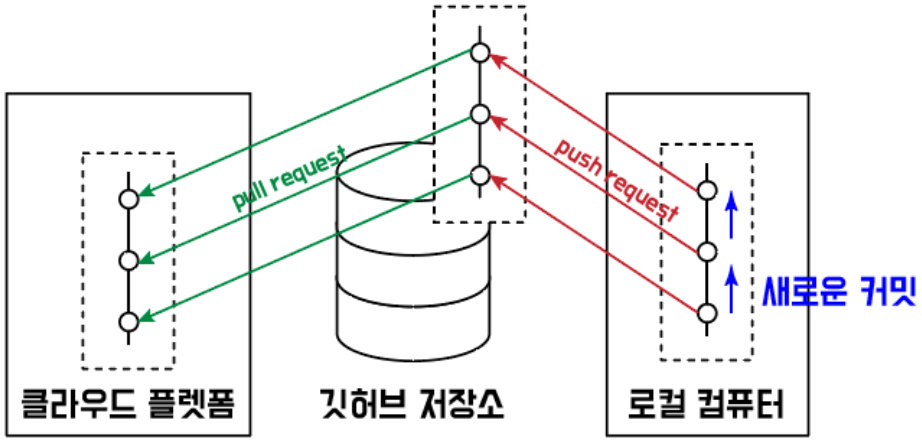
- Master 작업 파일 push
  - 동기화된 파일 저장 후
  - git add .
  - git commit –m “second commit”
  - git push origin master

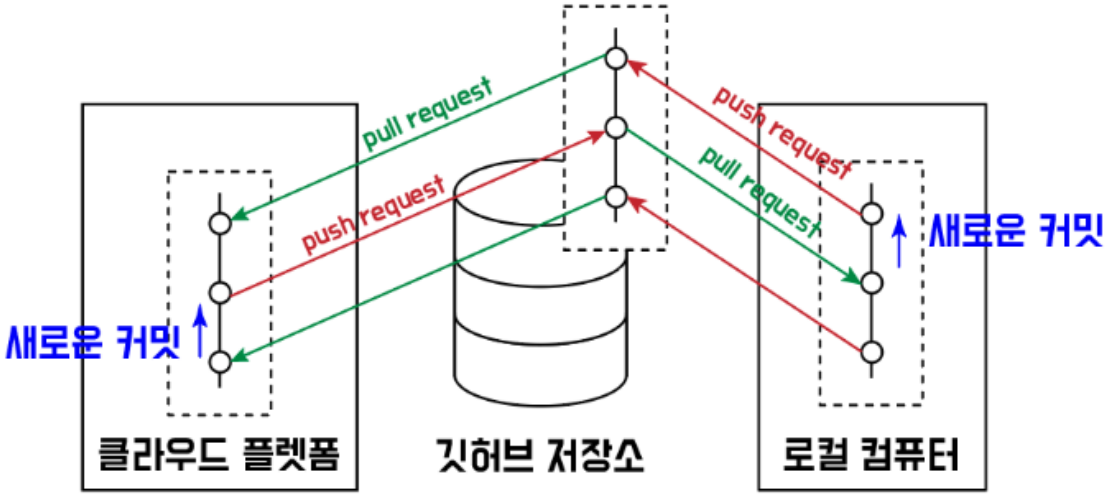
```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master|MERGING)
$ git add .

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master|MERGING)
$ git commit -m "second commit"
[master 952b09f] second commit

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 677 bytes | 677.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/syscrypto/Git_Project.git
   d73b767..952b09f  master -> master
```



Github

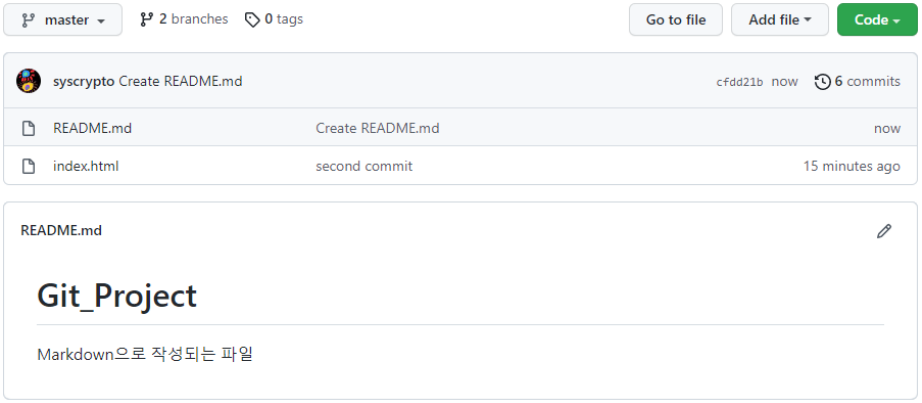
- 여러 장소에서 개발, Github활용





Github

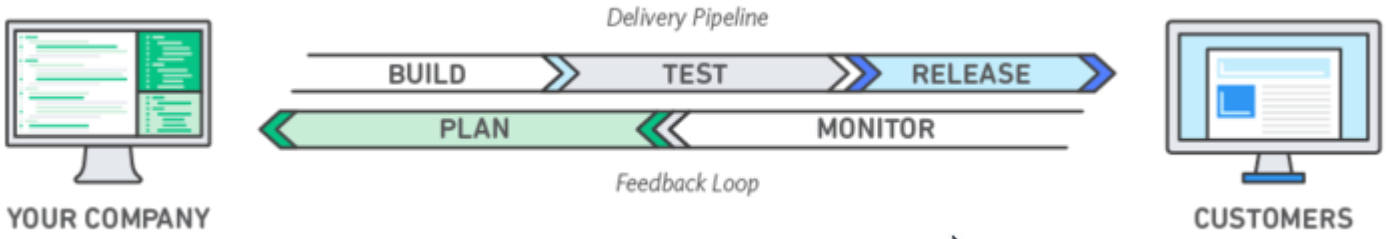
- **Markdown**
  - 텍스트 기반의 마크업언어
  - 2004년 존그루버에 의해 만들어졌으며 쉽게 쓰고 읽을 수 있으며 HTML로 변환이 가능
  - 특수기호와 문자를 이용한 매우 간단한 구조의 문법을 사용하여 웹에서도 보다 빠르게 컨텐츠를 작성하고 보다 직관적으로 인식할 수 있다
  - 마크다운이 최근 각광받기 시작한 이유는 github 덕분
  - github의 저장소Repository에 관한 정보를 기록하는 README.md는 github을 사용하는 사람이라면 누구나 가장 먼저 접하게 되는 마크다운 문서이다.

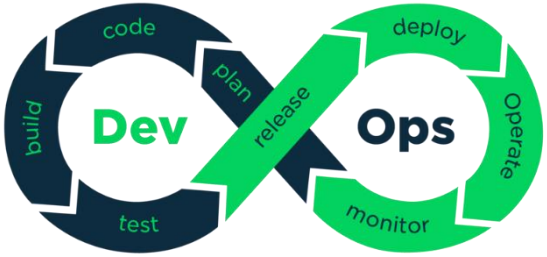


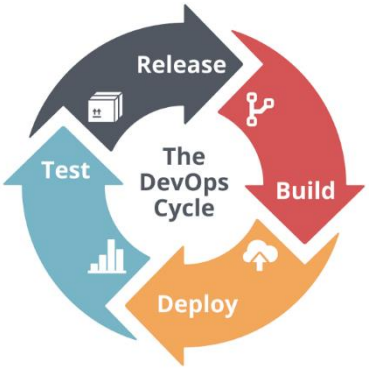
- **GitLab**
  - **깃랩**(GitLab)은 깃랩 사(GitLab Inc.)가 개발한 깃 저장소 및 CI/CD, 이슈 추적, 보안성 테스트 등의 기능을 갖춘 웹 기반의 데브옵스 플랫폼
  - 오픈 소스 라이선스 및 사유 소프트웨어 라이선스를 사용
  - 깃 저장소와 이슈 추적 기능을 갖춘 유일한 단일 어플리케이션의 (Single Application) 데브옵스 솔루션
  - 시중에 유통되고 있는 많은 데브옵스 솔루션들은 자신들의 특화된 영역 이외는 API를 이용한 연동 만을 제공하지만 깃랩은 단일 어플리케이션으로써 데브옵스의 전 영역의 기능들을 모두 제공하고 있어 소스코드 작성부터 배포까지 모든 과정을 깃랩 하나로 할 수 있다.

GitLab

- **DevOps**
  - 소프트웨어의 개발(Development)과 운영(Operations)의 합성어
  - 소프트웨어 개발자와 정보기술 전문가 간의 소통, 협업 및 통합을 강조하는 개발 환경이나 문화를 말함
  - 데브옵스는 소프트웨어 개발조직과 운영조직간의 상호 의존적 대응이며 조직이 소프트웨어 제품과 서비스를 빠른 시간에 개발 및 배포하는 것을 목적으로 함







Summary

Git 

Git-based

- GitHub 
- Gitlab 
- BitBucket 

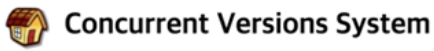
SVN 

Mercurial 

Dimensions CM 

Perforce 

Bazaar

CVS 

Reference

- https://docs.github.com/en/get-started/quickstart/hello-world
- https://medium.com/@kjunha77/%EA%B0%95%EC%9D%98%EB%85%B8%ED%8A%B8-11-%EB%B2%88%EC%99%B8-%EA%B9%83%ED%97%88%EB%B8%8C-%EC%82%AC%EC%9A%A9%EB%B2%95-d8d57f794f5
- https://medium.com/@krish.raghuram/terminal-shell-and-bash-3e76218c8865
- https://www.youtube.com/watch?v=lelVripbt2M
- https://gist.github.com/ihoneymon/652be052a0727ad59601
- https://bskyvision.com/1140
- https://blog.gaerae.com/2015/01/bash-hello-world.html
- https://about.gitlab.com/
- https://aws.amazon.com/ko/devops/what-is-devops/
- https://ko.wikipedia.org/wiki/%EB%8D%B0%EB%B8%8C%EC%98%B5%EC%8A%A4

중간고사 대체 과제

- <Github>
  - ‘Mid-Term-Project’ Repository를 생성해서 작업
  - 최소 10회이상 Commit

- <내용>
  - 자기 소개를 위한 개인 홈페이지 작성
  - 최소 이미지 2개 이상
  - 동영상 또는 동영상 링크 추가 (옵션)