

Open Source SW Utilization

(524820-2)

송영상(Youngsang Song)

sw.yssong@dankook.ac.kr

Outline

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

● 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 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

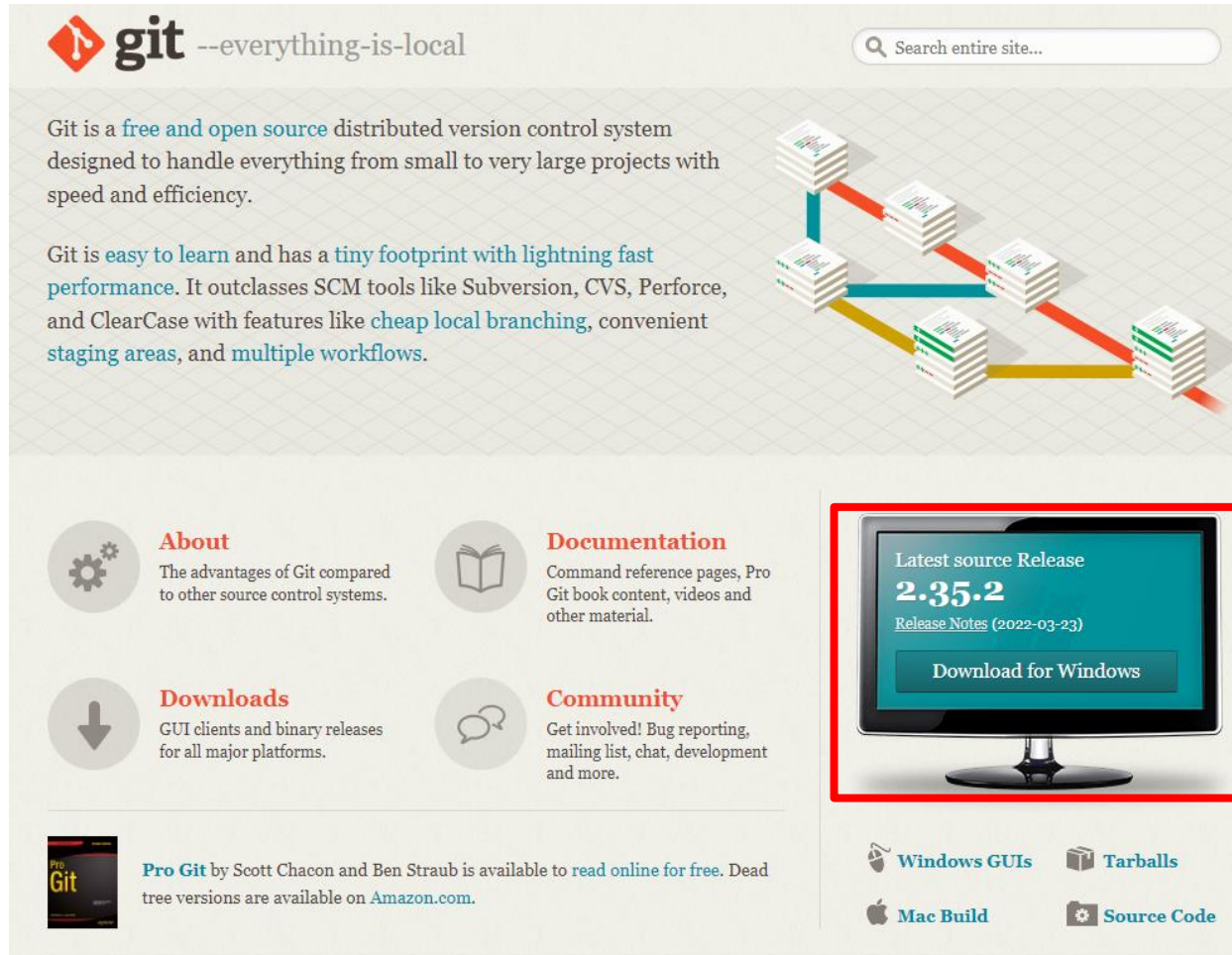
- Shell Setting

- Github 가입

- Github가입
 - 소스 올리기

● Git 설치

■ <http://git-scm.com/>



The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--everything-is-local". A search bar is located in the top right corner. The main content area features two paragraphs describing Git as a free and open source distributed version control system, highlighting its speed, efficiency, ease of learning, and performance. To the right of the text is a diagram illustrating Git's distributed nature with multiple repositories connected by a network. Below the main text are four sections: "About" (advantages of Git), "Documentation" (command reference, Pro Git book, videos), "Downloads" (GUI clients, binary releases), and "Community" (bug reporting, mailing list, chat). A red box highlights the "Latest source Release 2.35.2" section, which includes a "Download for Windows" button. At the bottom, there are links for "Pro Git" (available online for free), "Windows GUIs", "Tarballs", "Mac Build", and "Source Code".

git --everything-is-local

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.35.2
[Release Notes \(2022-03-23\)](#)
[Download for Windows](#)

Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

[Windows GUIs](#) [Tarballs](#)
[Mac Build](#) [Source Code](#)

● Terminal & Shell & Bash

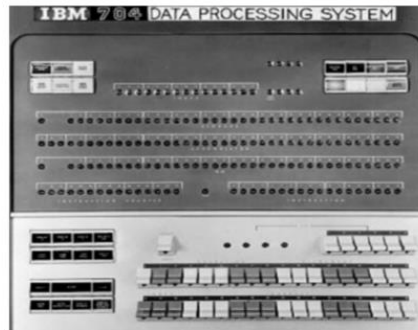
- The **terminal** is the GUI window that you see on the screen. It takes commands and shows output
- The **shell** is the software that interprets and executes the various commands that we type in the terminal.
- **Bash** is a particular shell.

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>

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

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

4. **Kernel** backend software accessed by

● Git 설치 확인

■ Window key + R : **Git Bash** or PowerShell or cmd

● git --version

```
MINGW64:/c/Users/ysson  
  
yssong@DESKTOP-GT35BCL MINGW64 ~  
$ git --version  
git version 2.35.1.windows.2  
  
yssong@DESKTOP-GT35BCL MINGW64 ~  
$ |
```

```
Windows PowerShell  
PS C:\Users\ysson> git  
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]  
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]  
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]  
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]  
        <command> [<args>]  
  
These are common Git commands used in various situations:  
  
start a working area (see also: git help tutorial)  
  clone Clone a repository into a new directory  
  init   Create an empty Git repository or reinitialize an existing one  
  
work on the current change (see also: git help everyday)  
  add    Add file contents to the index  
  mv     Move or rename a file, a directory, or a symlink  
  restore Restore working tree files  
  rm     Remove files from the working tree and from the index  
  
examine the history and state (see also: git help revisions)  
  bisect Use binary search to find the commit that introduced a bug  
  diff   Show changes between commits, commit and working tree, etc  
  grep   Print lines matching a pattern  
  log    Show commit logs  
  show   Show various types of objects  
  status Show the working tree status  
  
grow, mark and tweak your common history  
  branch List, create, or delete branches  
  commit Record changes to the repository  
  merge  Join two or more development histories together  
  rebase Reapply commits on top of another base tip  
  reset  Reset current HEAD to the specified state  
  switch Switch branches  
  tag    Create, list, delete or verify a tag object signed with GPG  
  
collaborate (see also: git help workflows)  
  fetch  Download objects and refs from another repository  
  pull   Fetch from and integrate with another repository or a local branch  
  push   Update remote refs along with associated objects  
  
'git help -a' and 'git help -g' list available subcommands and some  
concept guides. See 'git help <command>' or 'git help <concept>'  
to read about a specific subcommand or concept.  
See 'git help git' for an overview of the system.  
PS C:\Users\ysson>
```


● Git 초기 설정

- 사용자명 등록
- 메일 주소 등록

```
$ git config --global user.name "<사용자명>"  
$ git config --global user.email "<메일 주소>"
```

Windows PowerShell

```
PS C:\Users\ysson> git config --global user.name "syscrytpo"  
PS C:\Users\ysson> git config --global user.email "sw.yssong@dankook.ac.kr"  
PS C:\Users\ysson>
```

● 설정 확인 : VS code Terminal

- git config -list
 - 빠져나오기 :q

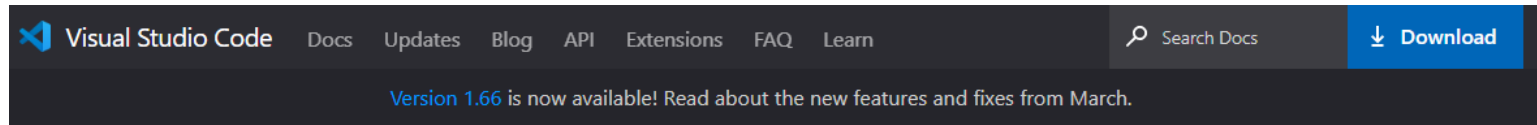
MINGW64:/c/Users/ysson

```
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=true  
pull.rebase=false  
credential.helper=manager-core  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
filter.lfs.clean=git-lfs clean -- %f  
user.name=syscrytpo  
user.email=sw.yssong@dankook.ac.kr
```

```
yssong@DESKTOP-GT35BCL MINGW64 ~  
$
```

● Visual Studio Code

- <https://code.visualstudio.com/download>



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10, 11

User Installer [64 bit](#) [32 bit](#) [ARM](#)
System Installer [64 bit](#) [32 bit](#) [ARM](#)
.zip [64 bit](#) [32 bit](#) [ARM](#)



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb [64 bit](#) [ARM](#) [ARM 64](#)
.rpm [64 bit](#) [ARM](#) [ARM 64](#)
.tar.gz [64 bit](#) [ARM](#) [ARM 64](#)

[Snap Store](#)

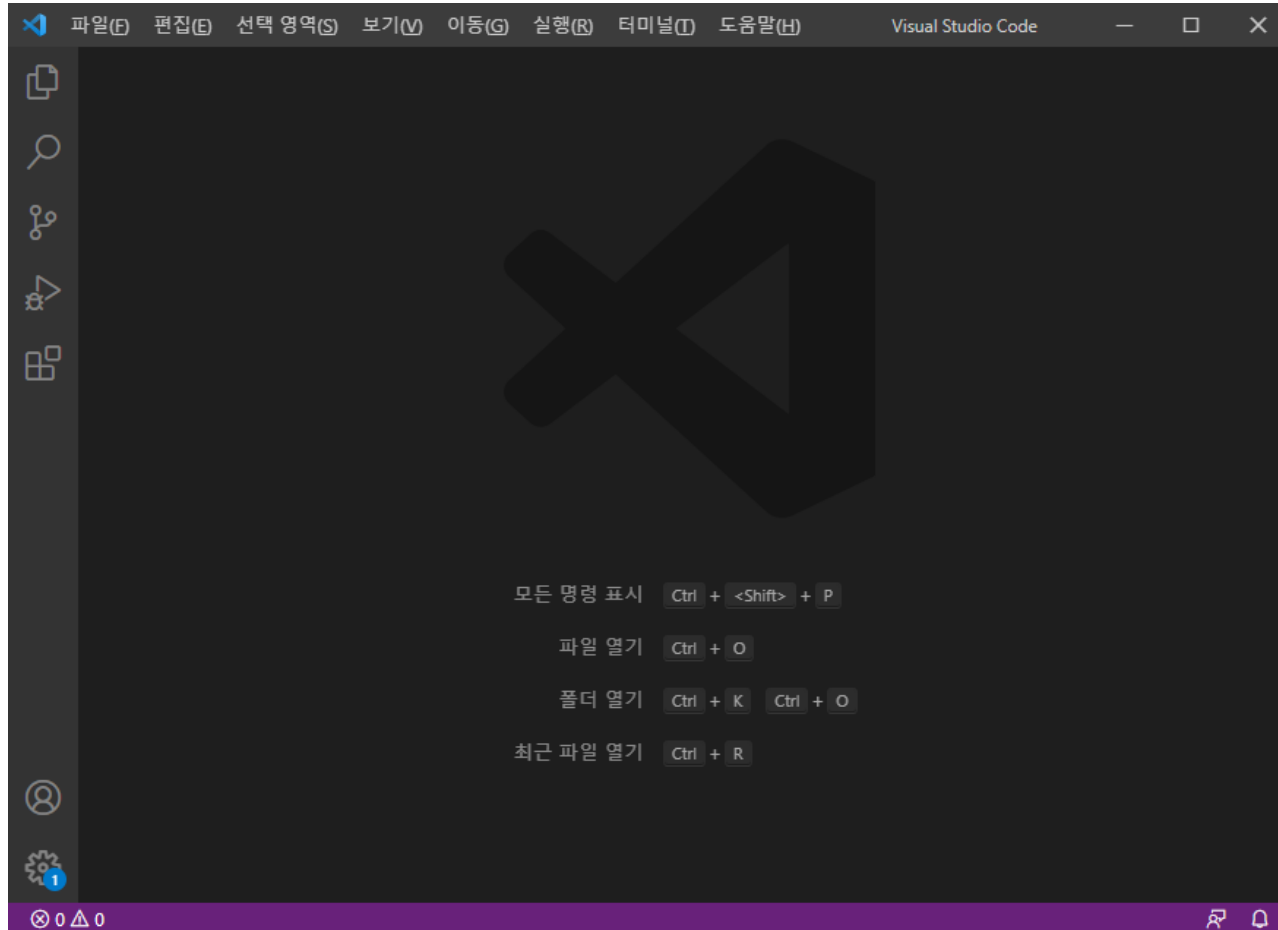


↓ Mac

macOS 10.11+

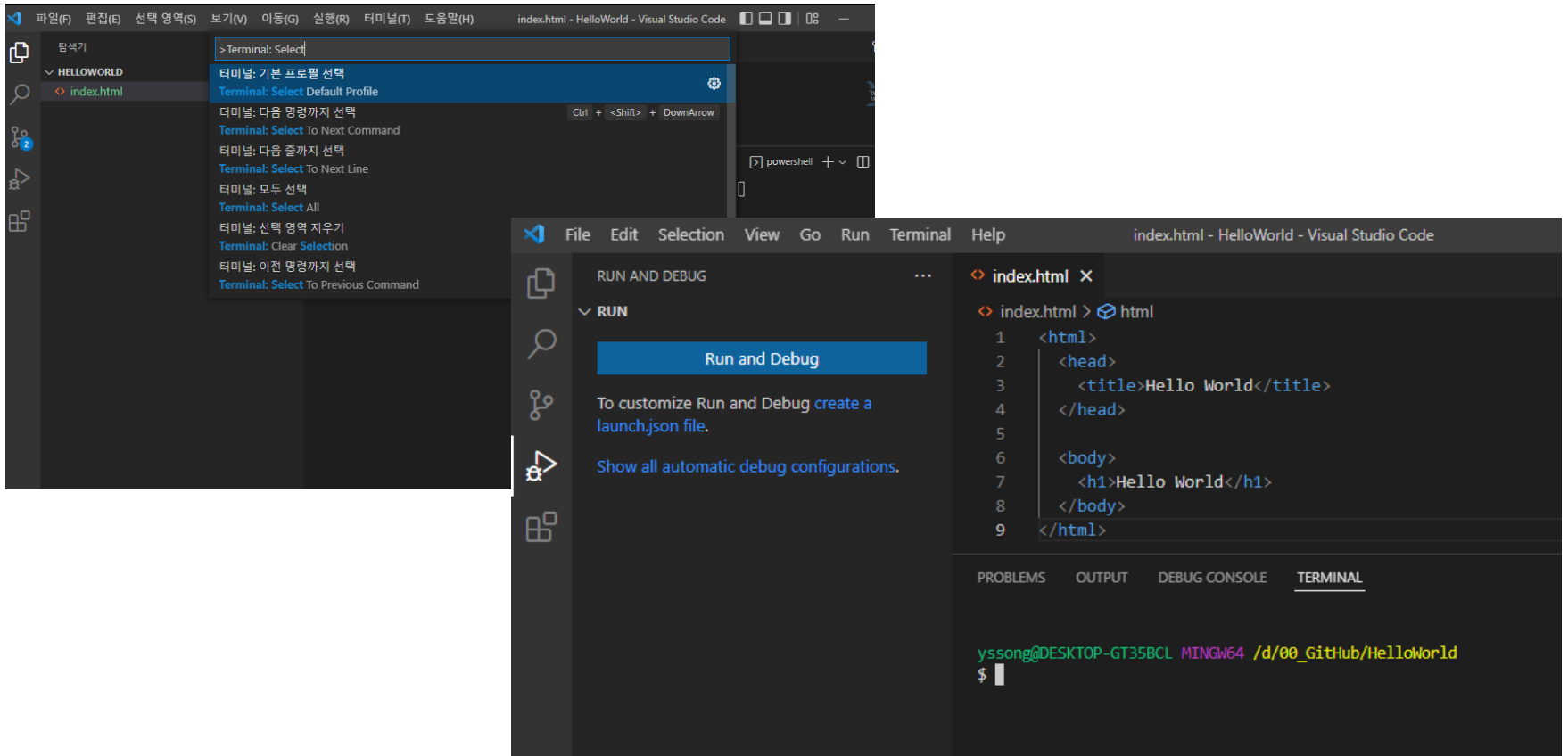
.zip [Universal](#) [Intel Chip](#) [Apple Silicon](#)

- Visual Studio Code 실행
 - CMD창에서 :code .



● Terminal Shell 변경

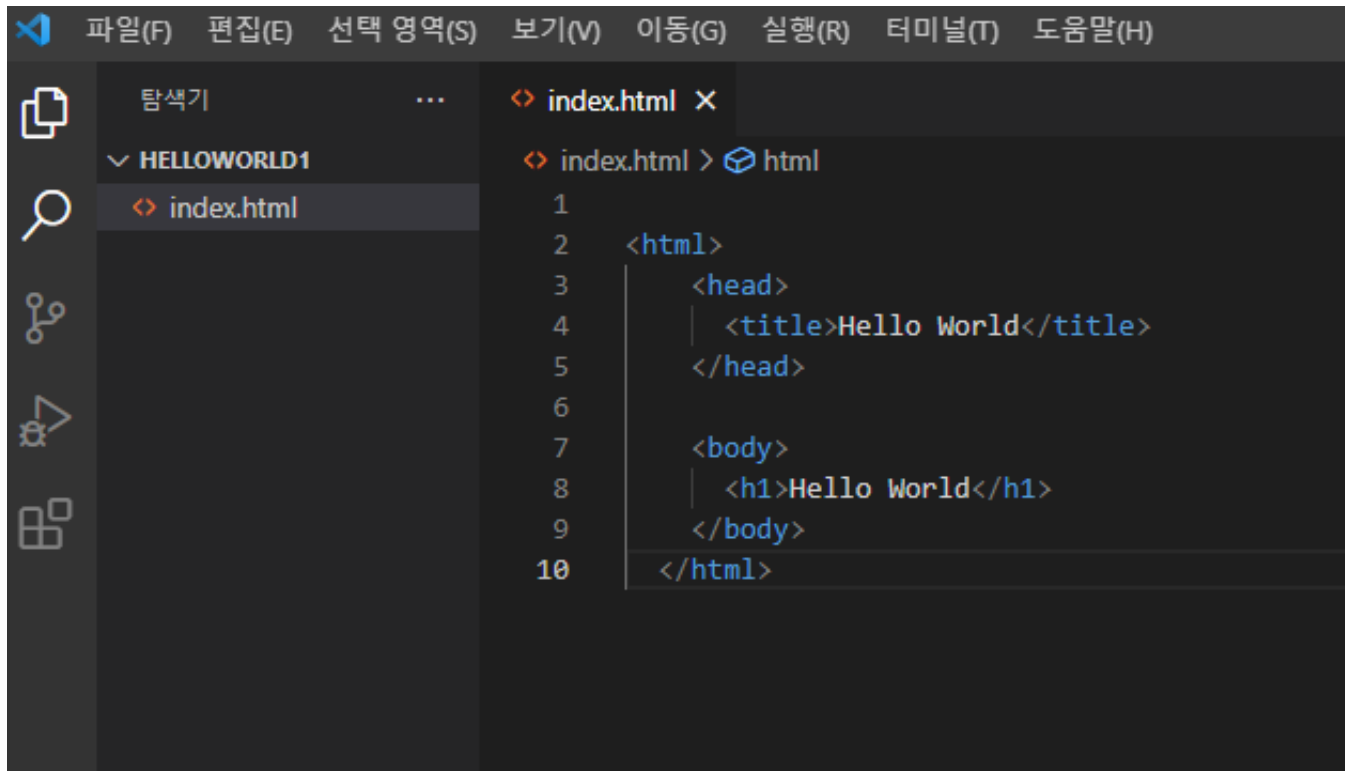
- Ctrl + Shift + P
- Command Palette -> Terminal: Select Default Profile



● HelloWorld Project

- Project 실행을 위한 폴더 생성
- VS code 파일 – 폴더 열기 : 생성한 폴더 지정
- 파일 만들기 : index.html

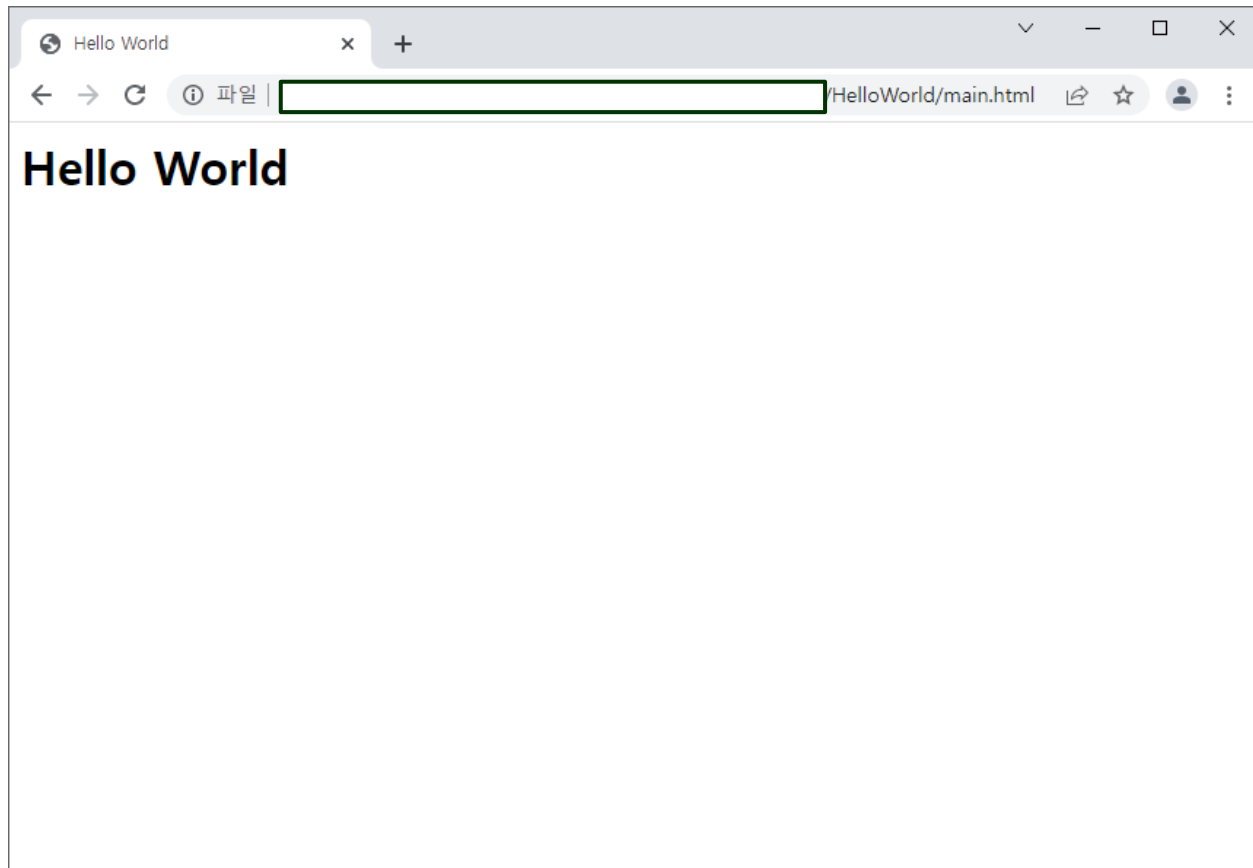
HelloWorld



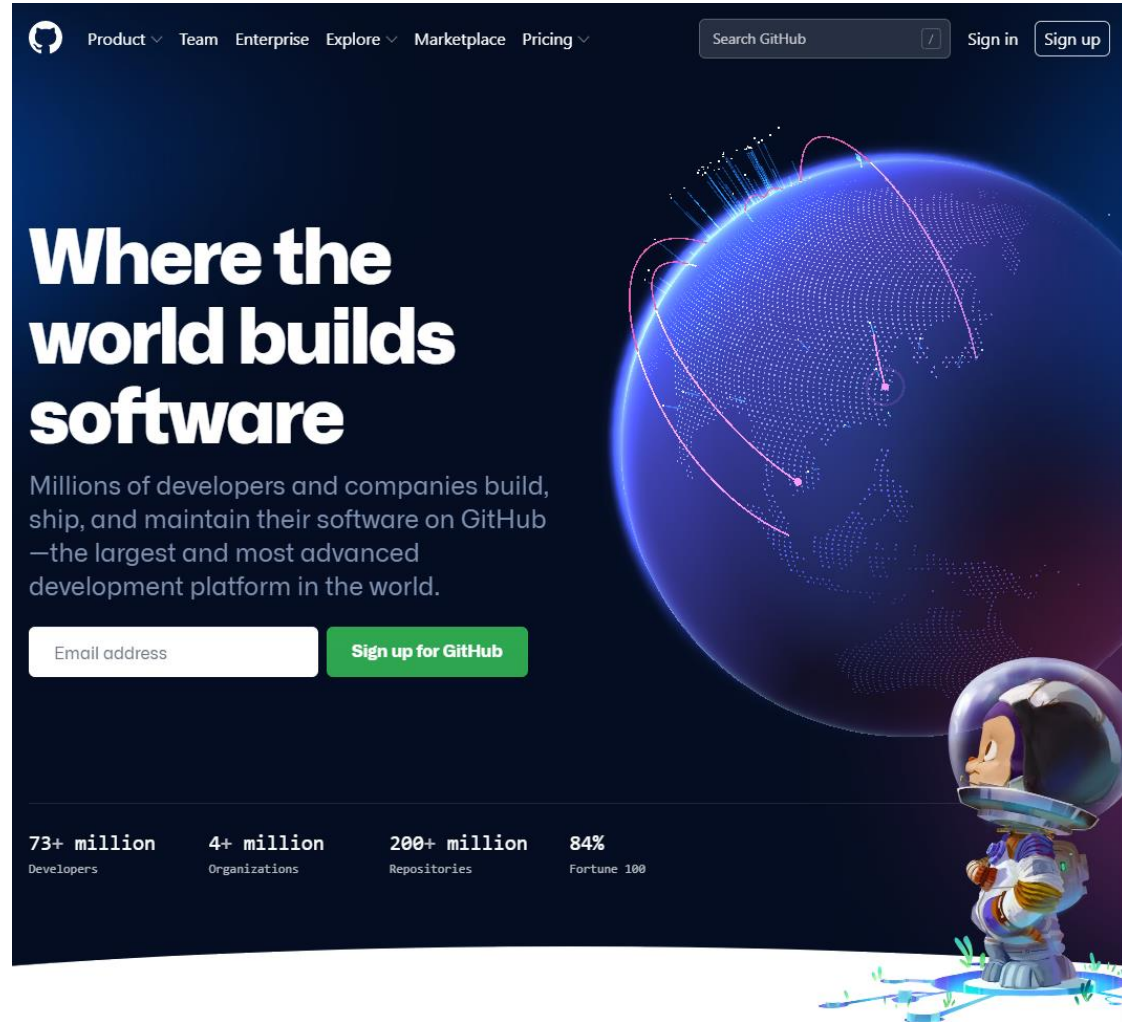
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a folder named 'HELLOWORLD1' with a file 'index.html' inside. The main editor area displays the content of 'index.html', which is a basic HTML document. The code is as follows:

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

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



● Github가입

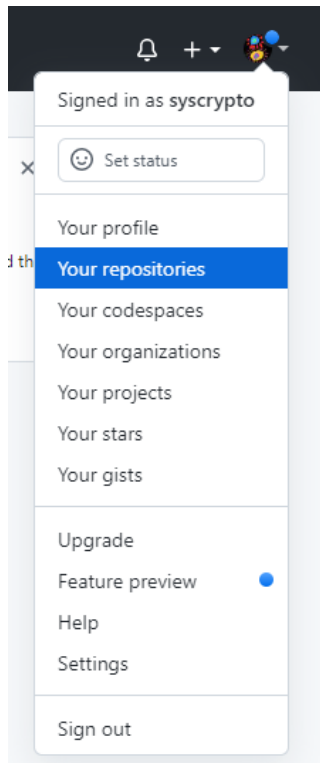


● Repository 생성

■ Your repositories

■ New

■ Create repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

syscrypto ▾

Repository name *

HelloWorld1 ✓

Great repository names are short and memorable. Need inspiration? How about [miniature-disco?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:


Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

● 첫 번째 Repository 화면

Quick setup — if you've done this kind of thing before

 Set up in Desktop or HTTPS SSH `https://github.com/syscrypto/HelloWorld1.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# HelloWorld1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/syscrypto/HelloWorld1.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/syscrypto/HelloWorld1.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

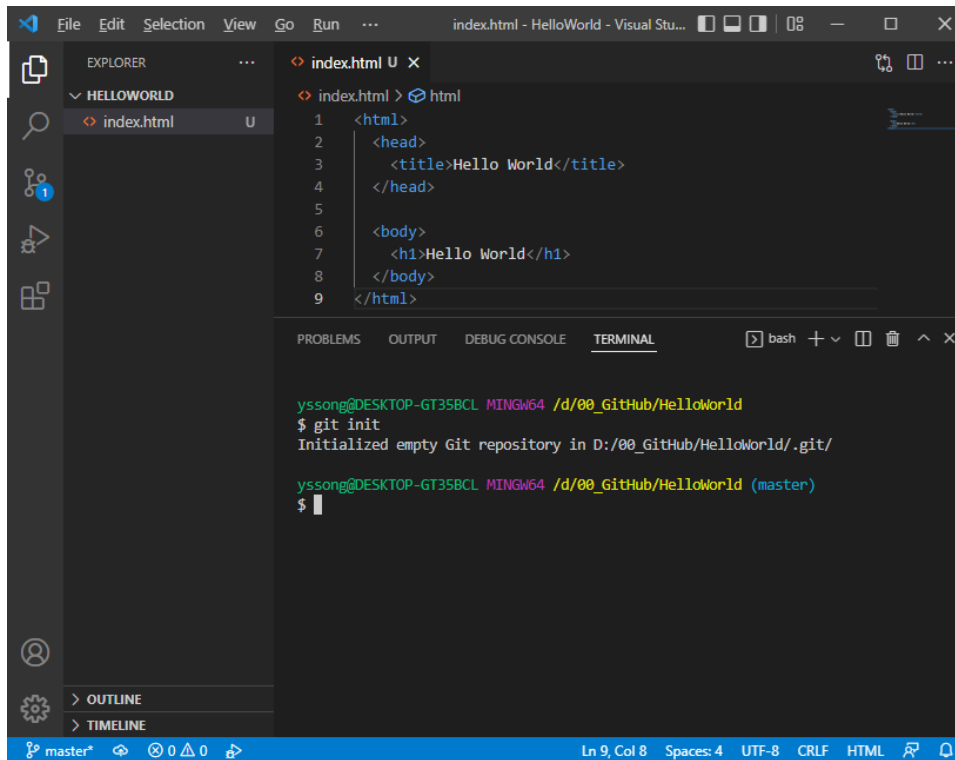
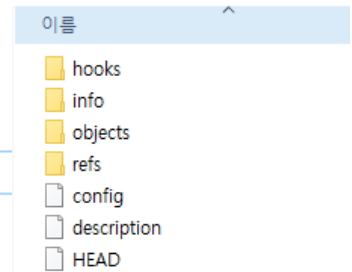
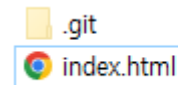
● 초기화

■ 소스코드 폴더를 버전 관리를 위해 initialize함

- git init

■ Project하위 폴더에 .git폴더 생성

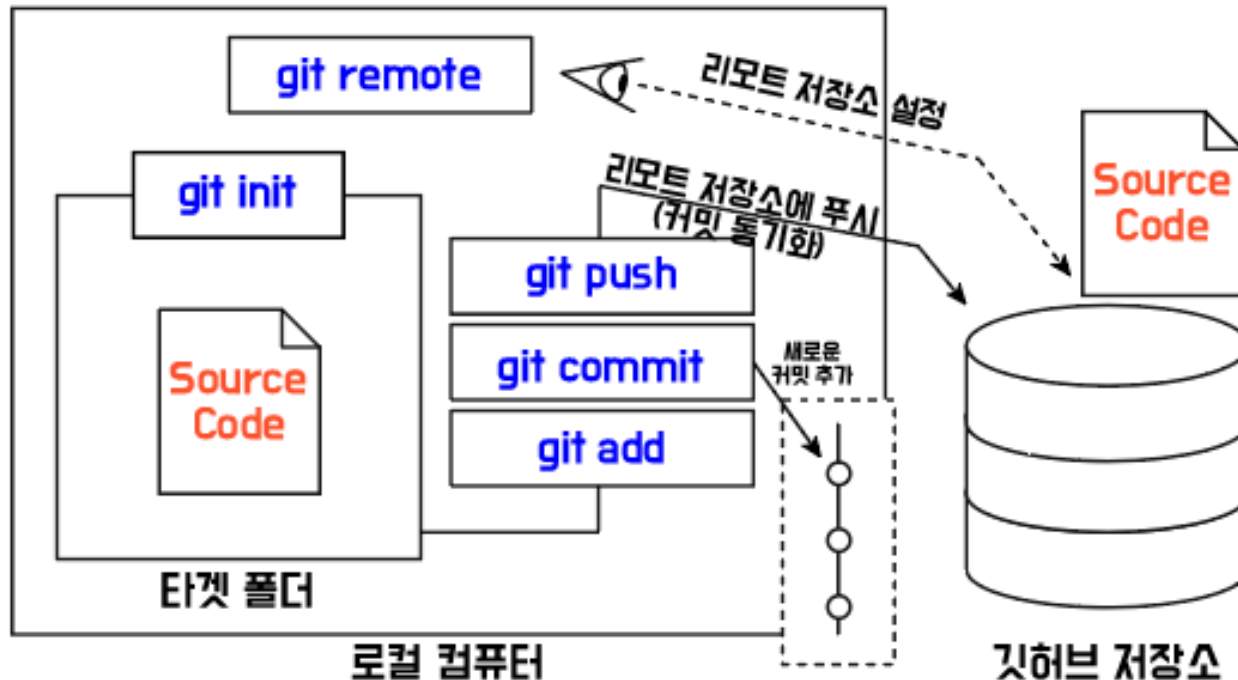
- 버전관리 정보가 저장



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

● 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

Quick setup — if you've done this kind of thing before

or `https://github.com/syscrypto/HelloWorld1.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# HelloWorld1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/syscrypto/HelloWorld1.git
git push -u origin main
```

```
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 Repository 확인

The screenshot shows the GitHub interface for a repository named 'syscrypto / HelloWorld1'. The repository is public and has a 'master' branch with 1 branch and 0 tags. The commit history shows a single commit by 'syscrypto' titled 'first commit' with the hash '2bf270b' and a timestamp of '4 minutes ago'. The file 'index.html' is highlighted, showing its commit history and a preview of its content. The file content is a simple HTML document with a title 'Hello World' and a body containing 'Hello World!'. A green arrow points from the 'index.html' file in the commit history to the file preview below.

Search or jump to... / Pull requests Issues Marketplace Explore

syscrypto / HelloWorld1 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

syscrypto first commit 2bf270b 4 minutes ago 1 commit

index.html first commit 4 minutes ago

master HelloWorld1 / index.html

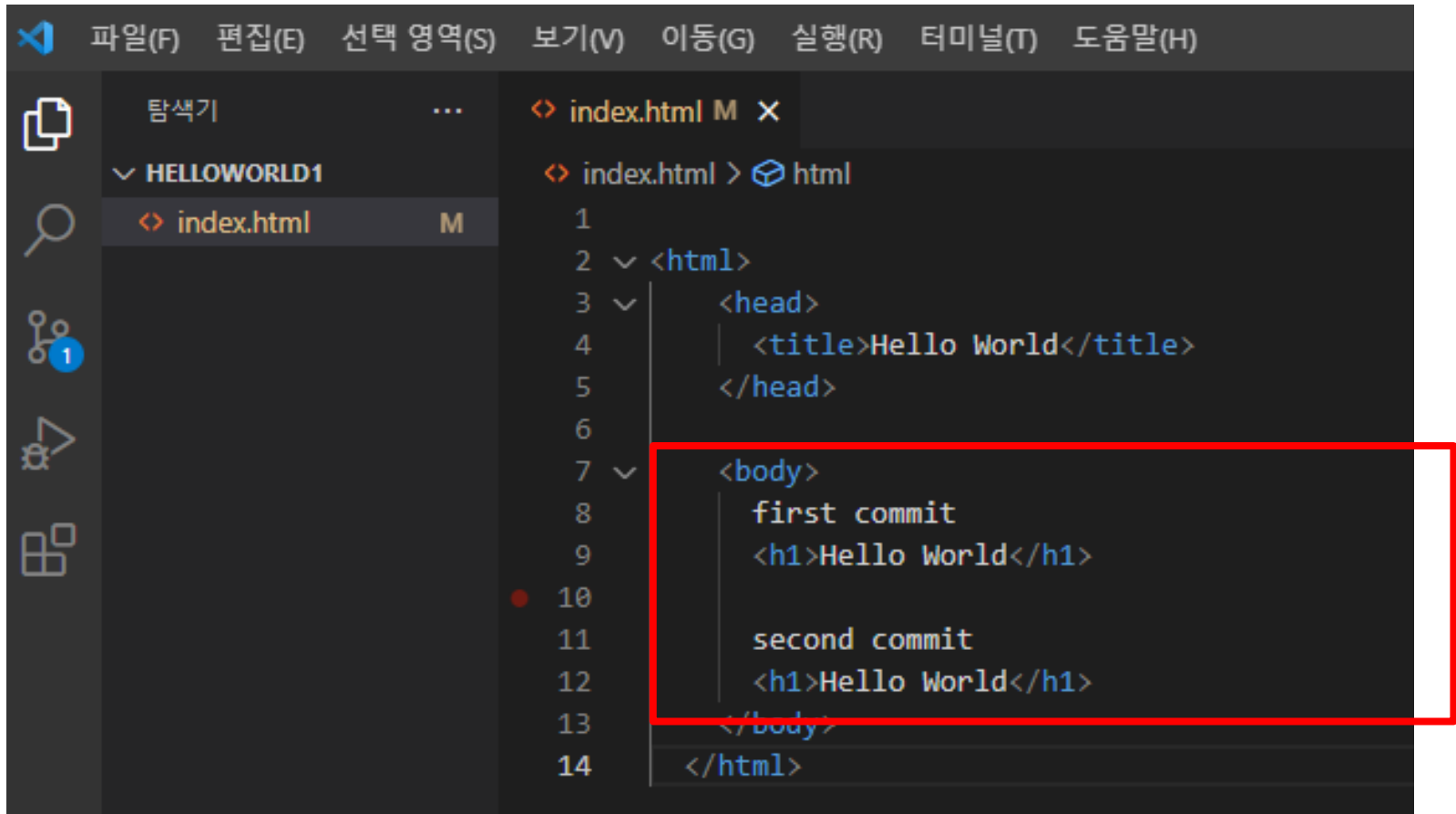
syscrypto first commit

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>
```

● 파일 수정



The screenshot shows the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '실행(R)', '터미널(T)', and '도움말(H)'. The left sidebar shows a file explorer with a folder named 'HELLOWORLD1' containing a file 'index.html'. The main editor area displays the content of 'index.html', which is an HTML document. The code is as follows:

```
<? index.html M X
<? 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>
```

A red rectangular box highlights the content within the `<body>` tags, specifically the two commit entries and their associated `<h1>` elements.

- 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/HelloWorld1 (master)
$ 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
```


● 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>
```

● 변경사항 표시

■ second commit click -> Split/Unified

second commit

master

syscrypto committed 7 minutes ago

1 parent [2bf270b](#) commit [f6bbb0c9e40dab6c3ae0788e7c88776a7285860](#)

[Browse files](#)

Showing 1 changed file with 4 additions and 0 deletions.

4 index.html

...

↑	@@ -5,6 +5,10 @@	
5	</head>	5 </head>
6		6
7	<body>	7 <body>
		8 + first commit
8	<h1>Hello World</h1>	9 <h1>Hello World</h1>
		10 +
		11 + second commit
		12 + <h1>Hello World</h1>
9	</body>	13 </body>
10	</html>	14 </html>

Split Unified

● Github에서 소스 수정

<> Edit file

Preview changes

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>

master 1 branch 0 tags


Go to file

Add file

Code

syscrypto Update index.html 840fdb0 now 3 commits

index.html Update index.html now



Commit changes

Update index.html

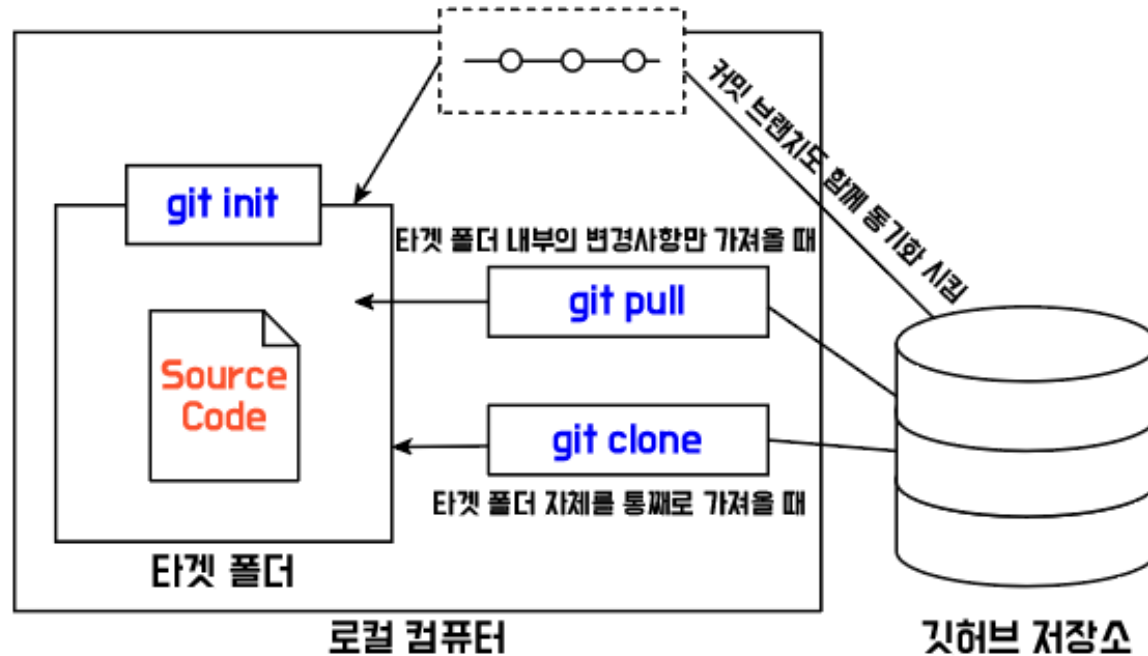
Add an optional extended description...

☒ Commit directly to the master branch.
☐ Create a new branch for this commit and start a pull request.

Commit changes Cancel

● 소스 가져오기

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



- 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
```

● 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
```

master

1 branch

0 tags



syscrypto third commit



index.html

third commit

● 협업(Collaboration)

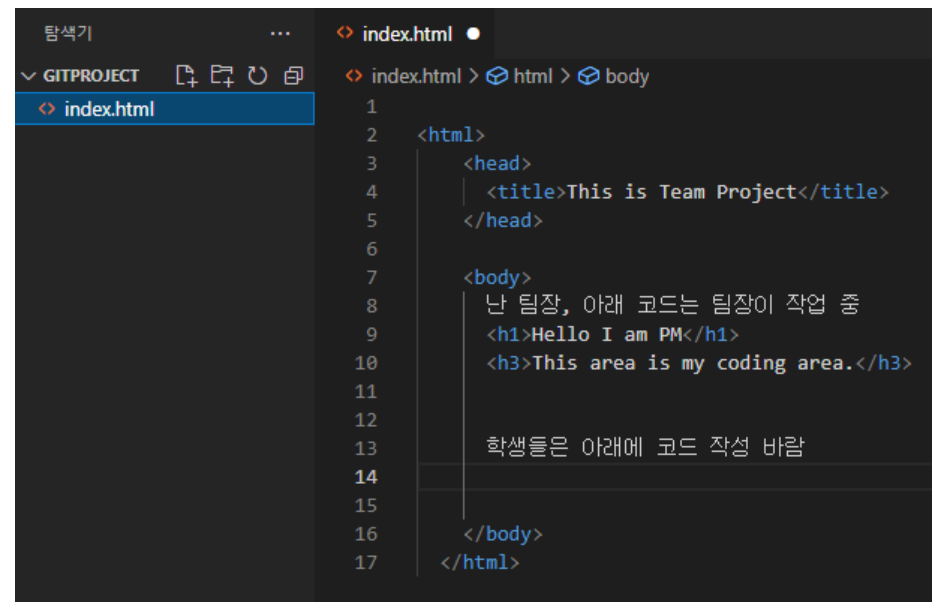
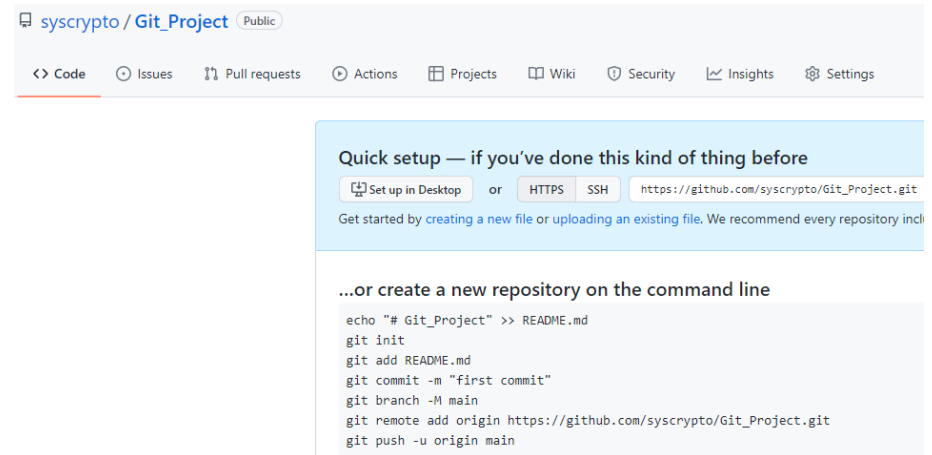
■ Github Repository 생성

- Repository name : Git_Project
- Public

■ VS code Project

● GitProject

» Index.html



● 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
```

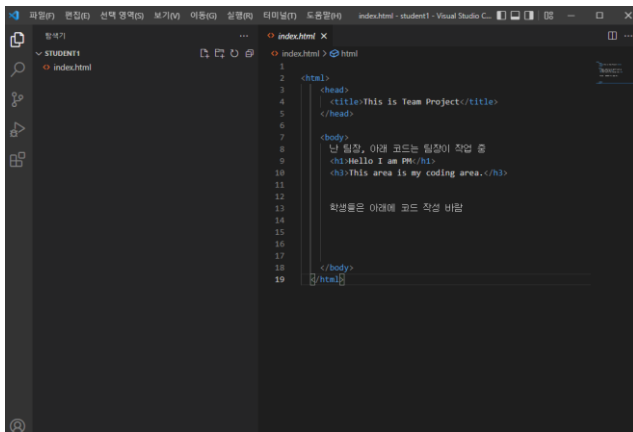
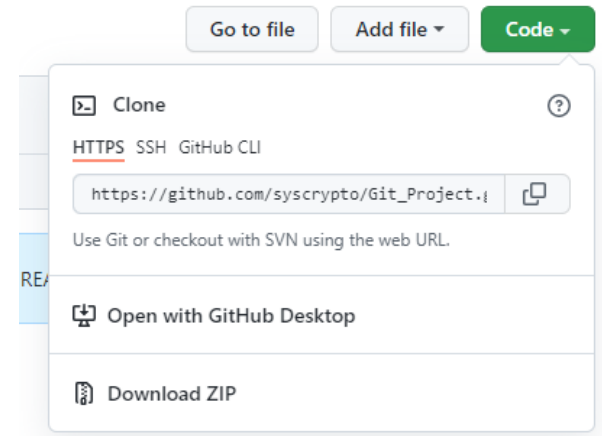

● 학생과 협업

■ 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

```
C:\> 선택 C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lysson> d:
D:\> cd 00_Student
D:\00_Student> _
```



```
D:\00_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:\00_Student>
```

● 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'.

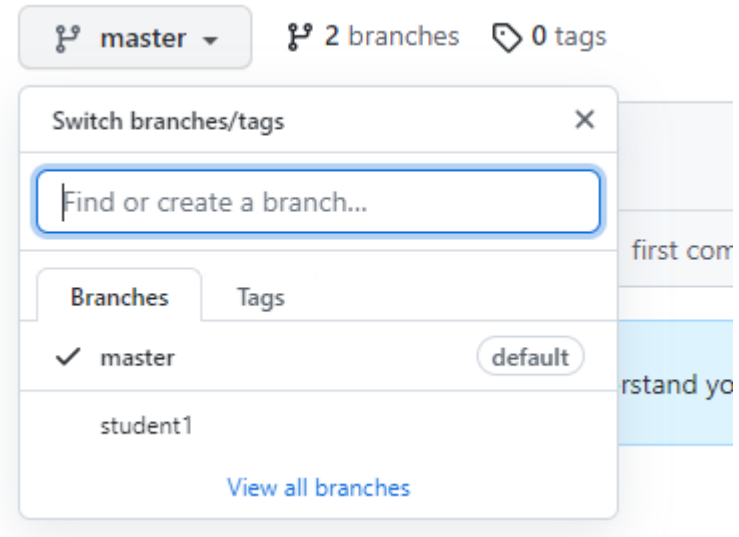
nothing to commit, working tree clean

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
```

● Branch

- master
- student1 (student1이 생성)



student1 2 branches 0 tags

This branch is up to date with master.


 syscrypto first commit

 index.html first commit

Help people interested in this repository understand your project by adding a README.

● 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>
```

- Pull requests 요청
 - 기존 소스에 merge 요청

The screenshot shows the GitHub interface for the repository 'syscrypto / Git_Project'. The 'Pull requests' tab is selected in the top navigation bar. A notification banner at the top states: 'Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with good first issue.' Below this, a yellow banner indicates 'student1 had recent pushes 2 minutes ago'. On the right side of this banner, a green button labeled 'Compare & pull request' is highlighted with a red rectangle. The main section is titled 'Open a pull request' with the subtitle 'Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.' Below the subtitle, there are filters for 'Labels' (9) and 'Milestones' (0), and a green 'New pull request' button. The comparison section shows 'base: master' and 'compare: student1' with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' Below this, the 'student1 second commit' is displayed in a preview window. The preview window has a 'Write' tab and a 'Preview' tab. The content of the preview is '수정된 파일 적용 바랍니다.' (Apply modified files). At the bottom of the preview window, there is a dashed line and the text 'Attach files by dragging & dropping, selecting or pasting them.' Below the preview window, a green button labeled 'Create pull request' is highlighted with a red rectangle.

● Merge

■ Main source에 합쳐지기 때문에 신중하게 선택

The screenshot shows a GitHub Pull Request interface for the repository 'syscrypto / Git_Project'. The 'Pull requests' tab is selected and highlighted with a red box. The pull request is titled 'student1 second commit #1' and is from the 'student1' branch to the 'master' branch. A green 'Open' button is visible. Below the title, a comment from 'syscrypto' says '수정된 파일 적용 바랍니다.' (Please apply the modified files). The commit 'student1 second commit' is listed and highlighted with a red box. A green arrow points from this commit to the code diff view on the left. The diff shows changes to 'index.html'. At the bottom, a green 'Merge pull request' button is highlighted with a red box. The interface also shows a status bar indicating 'This branch has no conflicts with the base branch' and a note about continuous integration.

syscrypto / Git_Project Public

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

student1 second commit #1

Open syscrypto wants to merge 1 commit into master from student1

Conversation 0 Commits 1 Checks 0 Files changed 1

syscrypto commented now

수정된 파일 적용 바랍니다.

student1 second commit

Add more commits by pushing to the student1 branch on syscrypto/Git_Project.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

student1 second commit

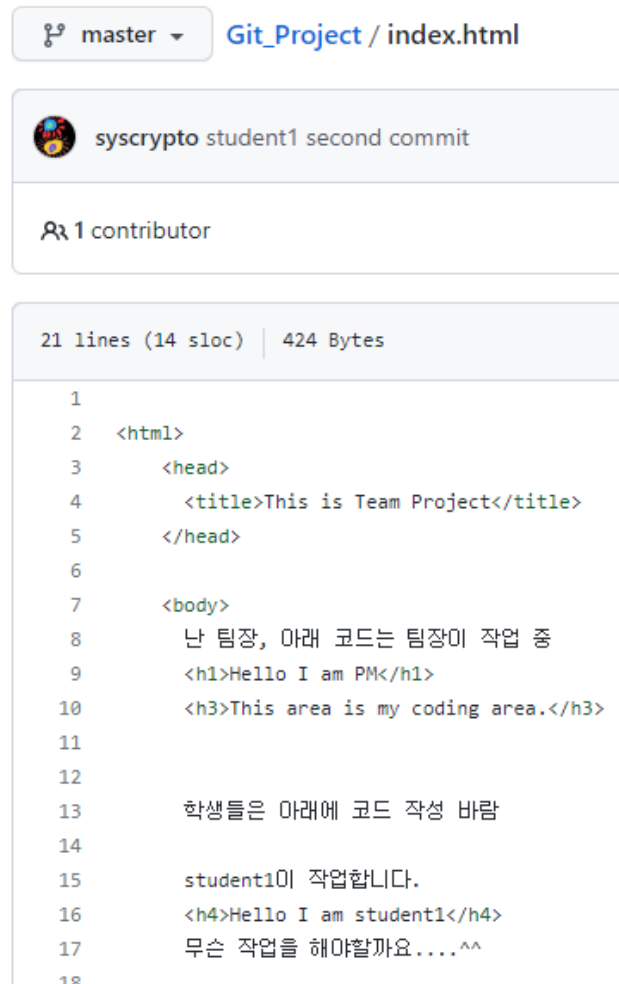
student1 (#1)

syscrypto committed 7 minutes ago

index.html

```
@@ -11,8 +11,10 @@ <h3>This area is my coding area.</h3>
11 11
12 12
13 13 학생들은 아래에 코드 작성 바람
14 -
15 14
16 15 + student1이 작업합니다.
17 16 + <h4>Hello I am student1</h4>
18 17 + 무슨 작업을 해야할까요....^^
```

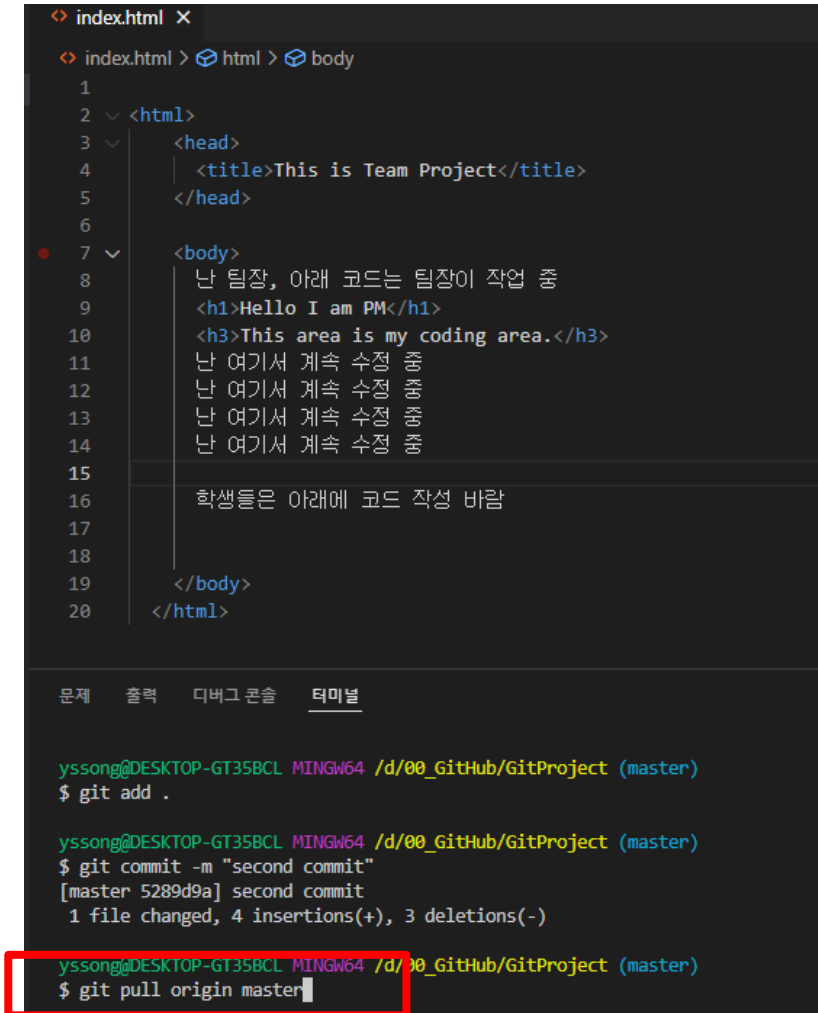
- Merge후 master code에 적용 화면



The screenshot shows a GitHub commit page for the file 'index.html' in the 'Git_Project' repository. The commit is on the 'master' branch and was made by 'syscrypto' with the message 'student1 second commit'. It shows 1 contributor. The commit details indicate 21 lines (14 sloc) and 424 Bytes. The code content is as follows:

```
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
```

PM의 Vs_code와 github 수정된 파일 동기화 : pull



The screenshot shows the VS Code editor with the file `index.html` open. The file content is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<html>
  <head>
    <title>This is Team Project</title>
  </head>
  <body>
    난 팀장, 아래 코드는 팀장이 작업 중
    <h1>Hello I am PM</h1>
    <h3>This area is my coding area.</h3>
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    학생들은 아래에 코드 작성 바람
  </body>
</html>
```

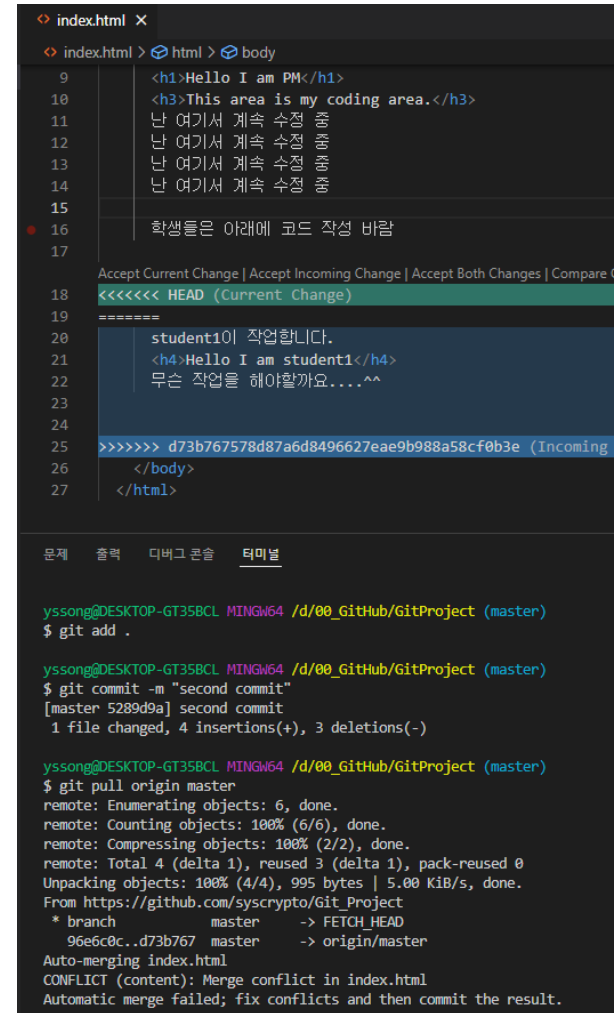
Below the editor, the terminal window shows the following commands and output:

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

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git commit -m "second commit"
[master 5289d9a] second commit
1 file changed, 4 insertions(+), 3 deletions(-)

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git pull origin master
```

<동기화 전 화면>



The screenshot shows the VS Code editor with the file `index.html` open. The file content is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<html>
  <head>
    <title>This is Team Project</title>
  </head>
  <body>
    난 팀장, 아래 코드는 팀장이 작업 중
    <h1>Hello I am PM</h1>
    <h3>This area is my coding area.</h3>
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    난 여기서 계속 수정 중
    학생들은 아래에 코드 작성 바람
  </body>
</html>
```

Below the editor, the terminal window shows the following commands and output:

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

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git commit -m "second commit"
[master 5289d9a] second commit
1 file changed, 4 insertions(+), 3 deletions(-)

yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master)
$ git pull origin master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (4/4), 995 bytes | 5.00 KiB/s, done.
From https://github.com/syscrypto/Git_Project
* branch      master       -> FETCH_HEAD
   96e6c0c..d73b767 master   -> origin/master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

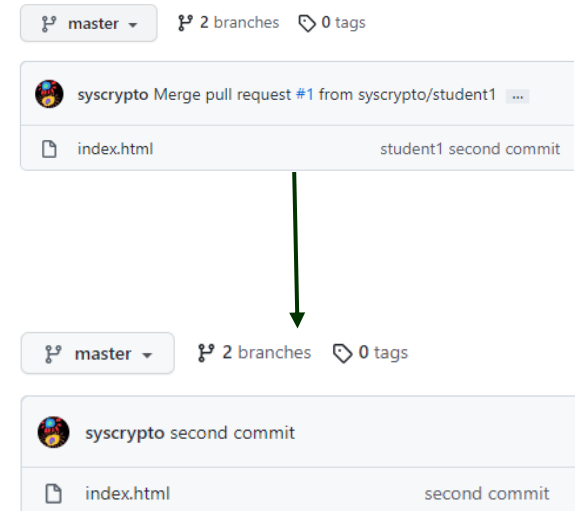
<동기화 후 화면>

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

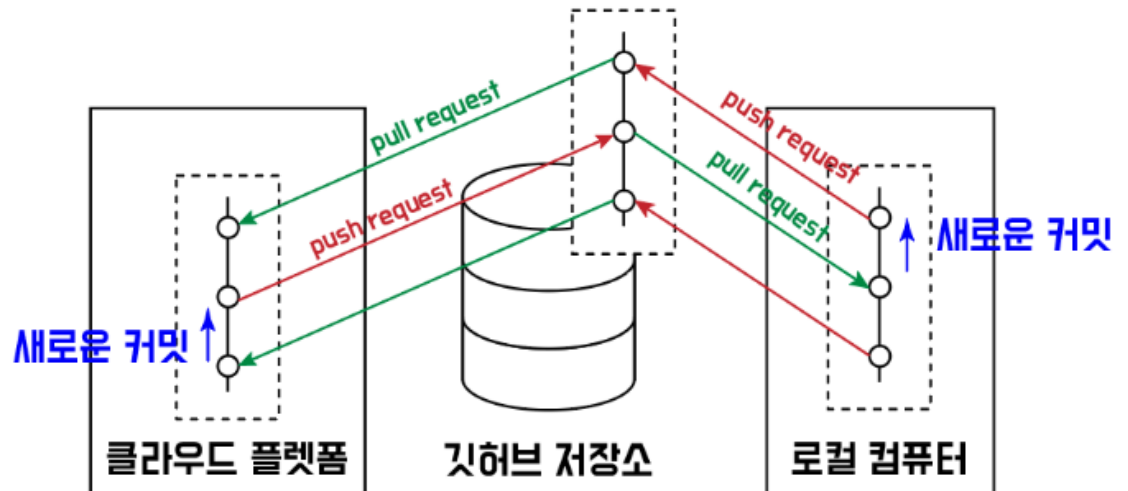
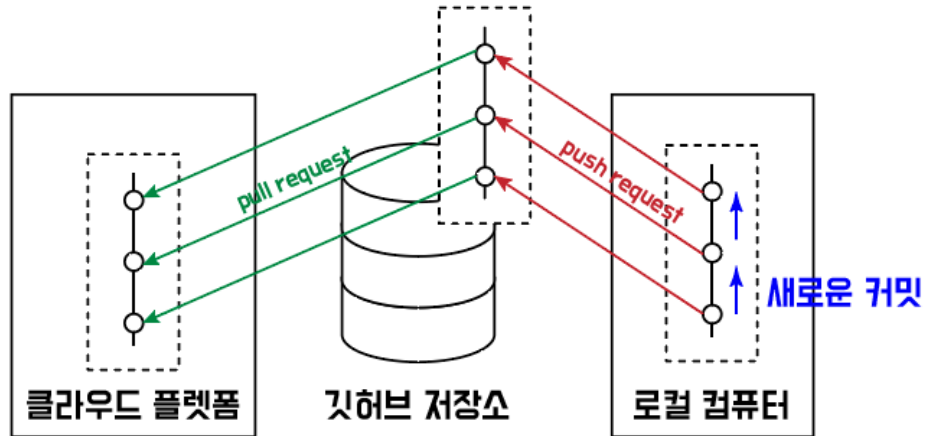
```
yssong@DESKTOP-GT35BCL MINGW64 /d/00_GitHub/GitProject (master|MERGING)
$ git add .

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
```

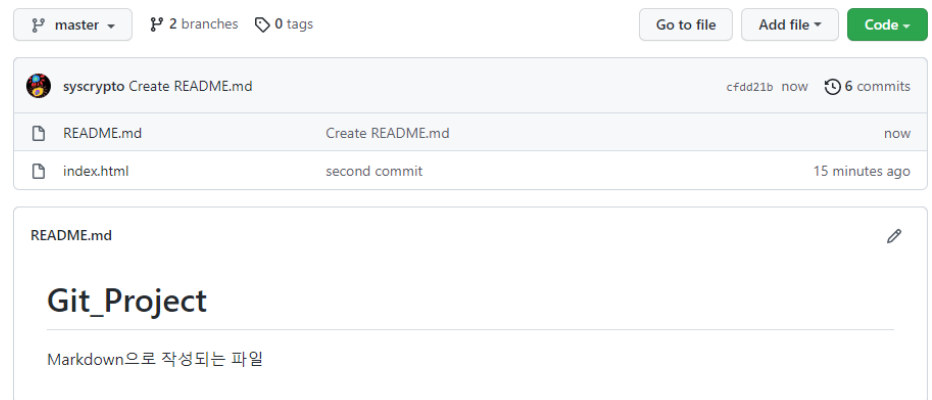


- 여러 장소에서 개발, Github 활용



● 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 콘텐츠를 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.

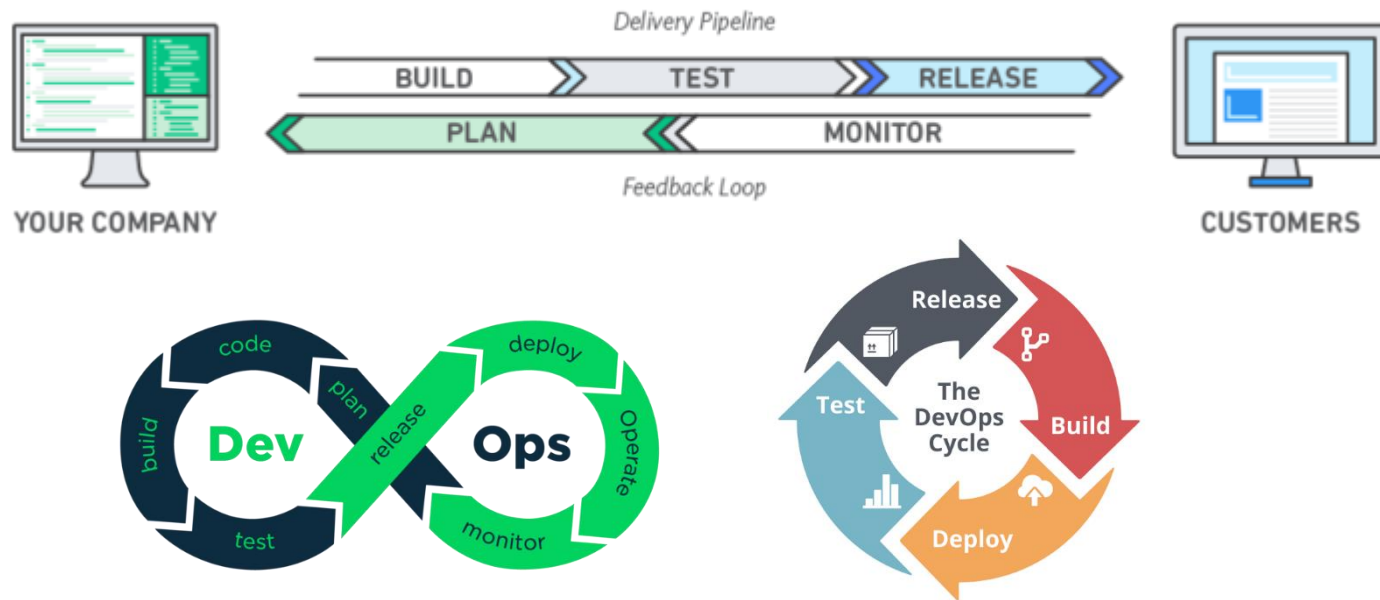


● GitLab

- 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.

● 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



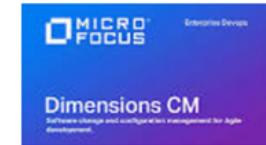
SVN



Mercurial



Dimensions CM



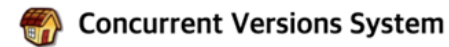
Perforce

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)