Open Source SW Utilization

(524820-2)

송영상(Youngsang Song)

sw.yssong@dankook.ac.kr

Outline

- **Github**
  - **Github 연결**
  - **Github로 소스 관리**
  - **협업(collaboration)**

- **명령어는 녹색으로 표시**

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

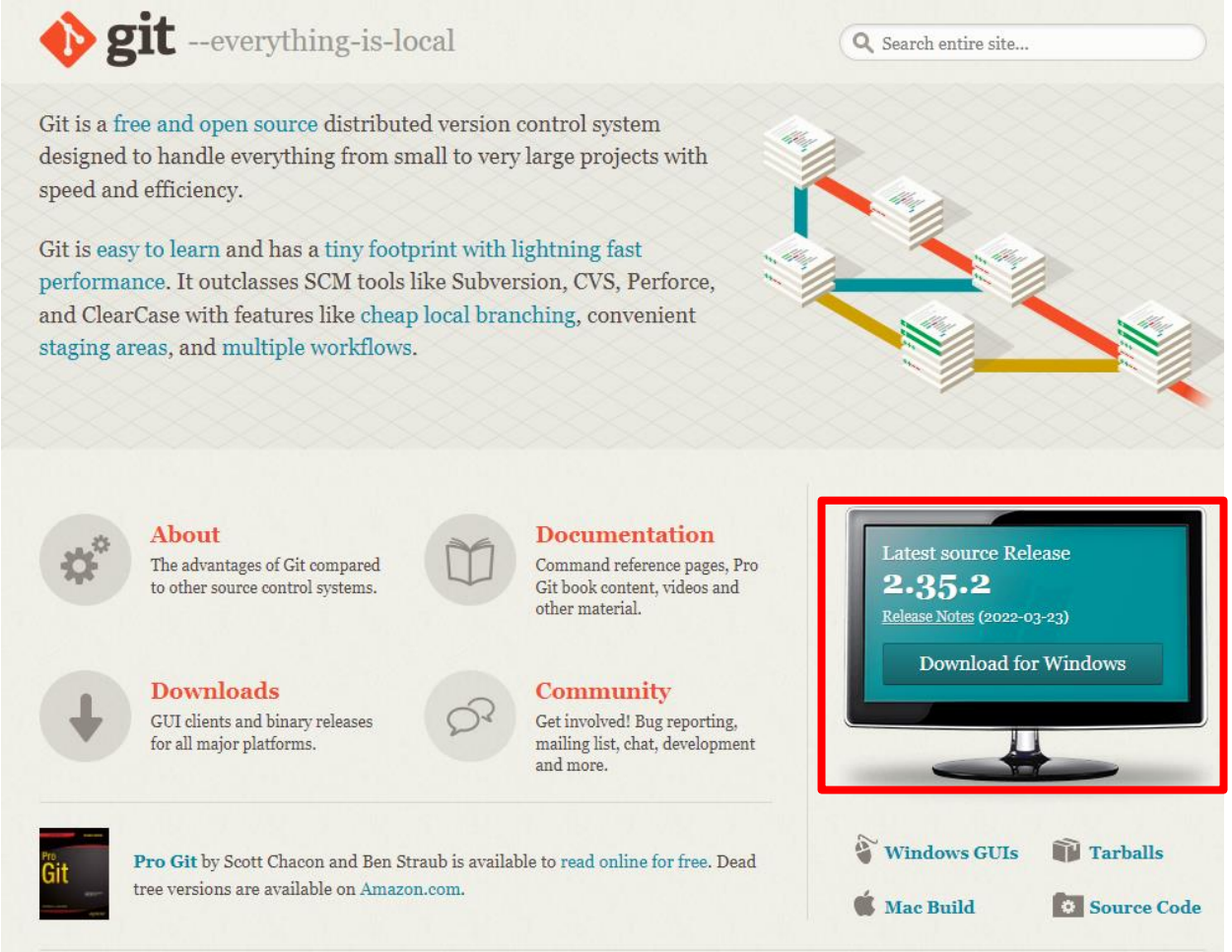
GitHub

Github

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

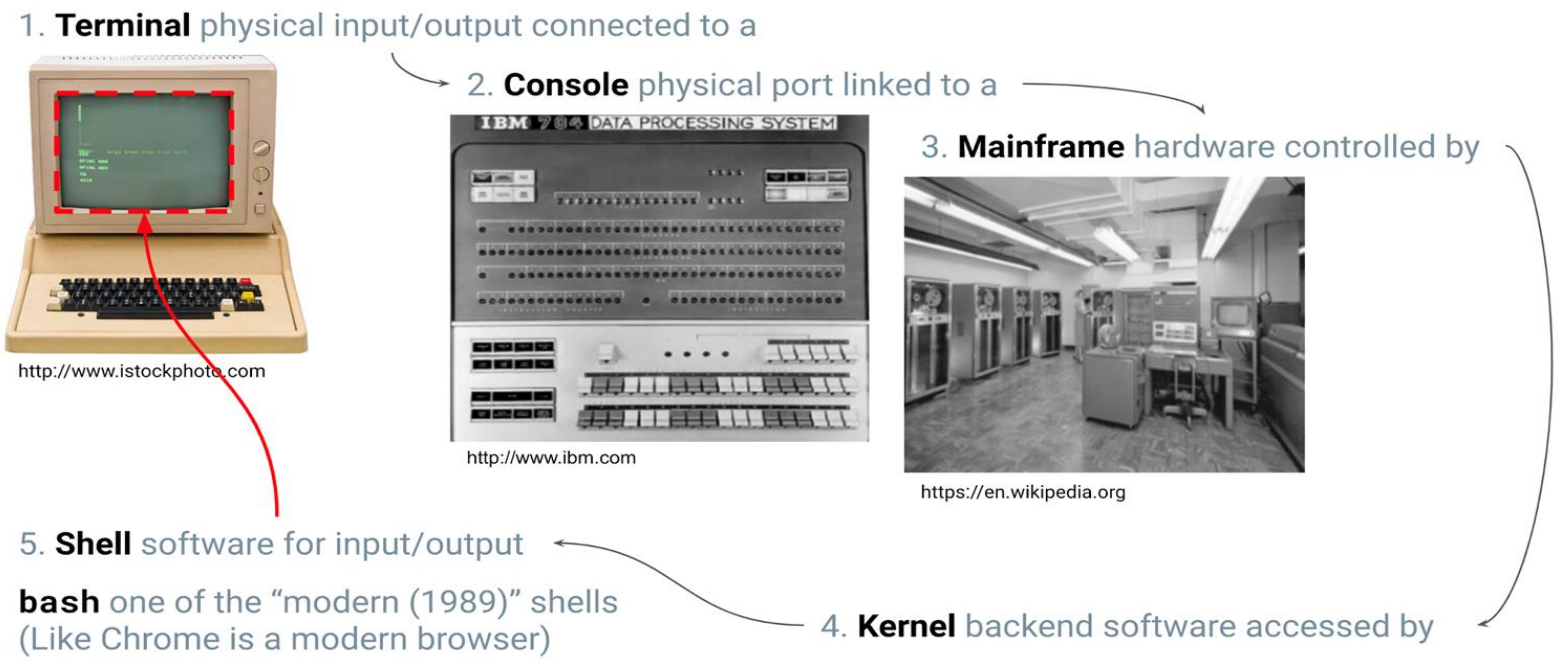
Github

- **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.



Github

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

MINGW64:/c/Users/ysson

```
ysson@DESKTOP-GT35BCL MINGW64 ~
$ git --version
git version 2.35.1.windows.2

ysson@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>
```

Github

- **Git 초기 설정**
  - 사용자명 등록 — $ git config –global user.name “XXXXX”
  - 메일 주소 등록 — $ git config –global user.email “XX@xx.xx”

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>
```

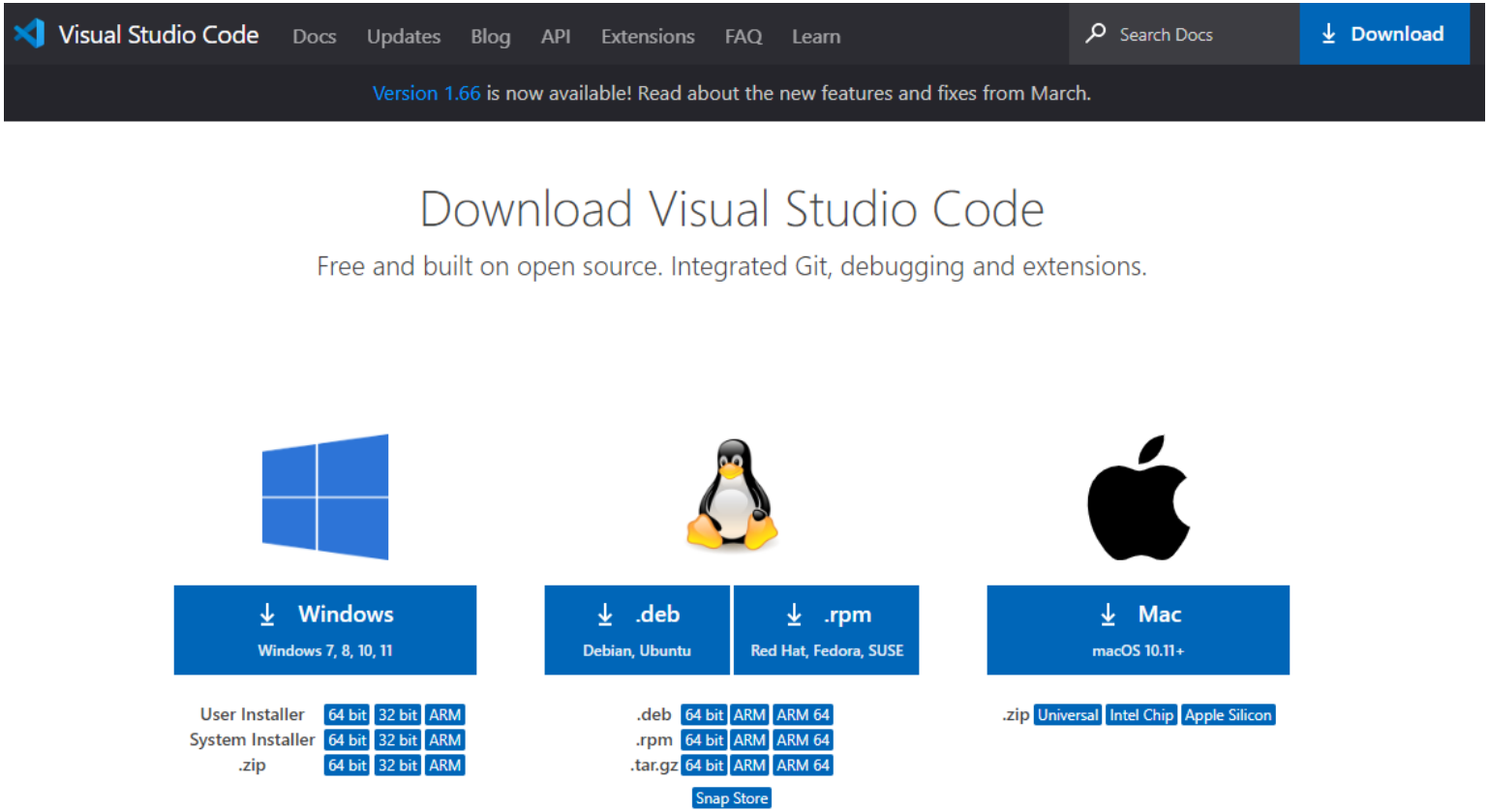
- **설정 확인 : VS code Terminal**
  - git config --list
    - 빠져나오기 :q

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.cr
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 ~
$
```

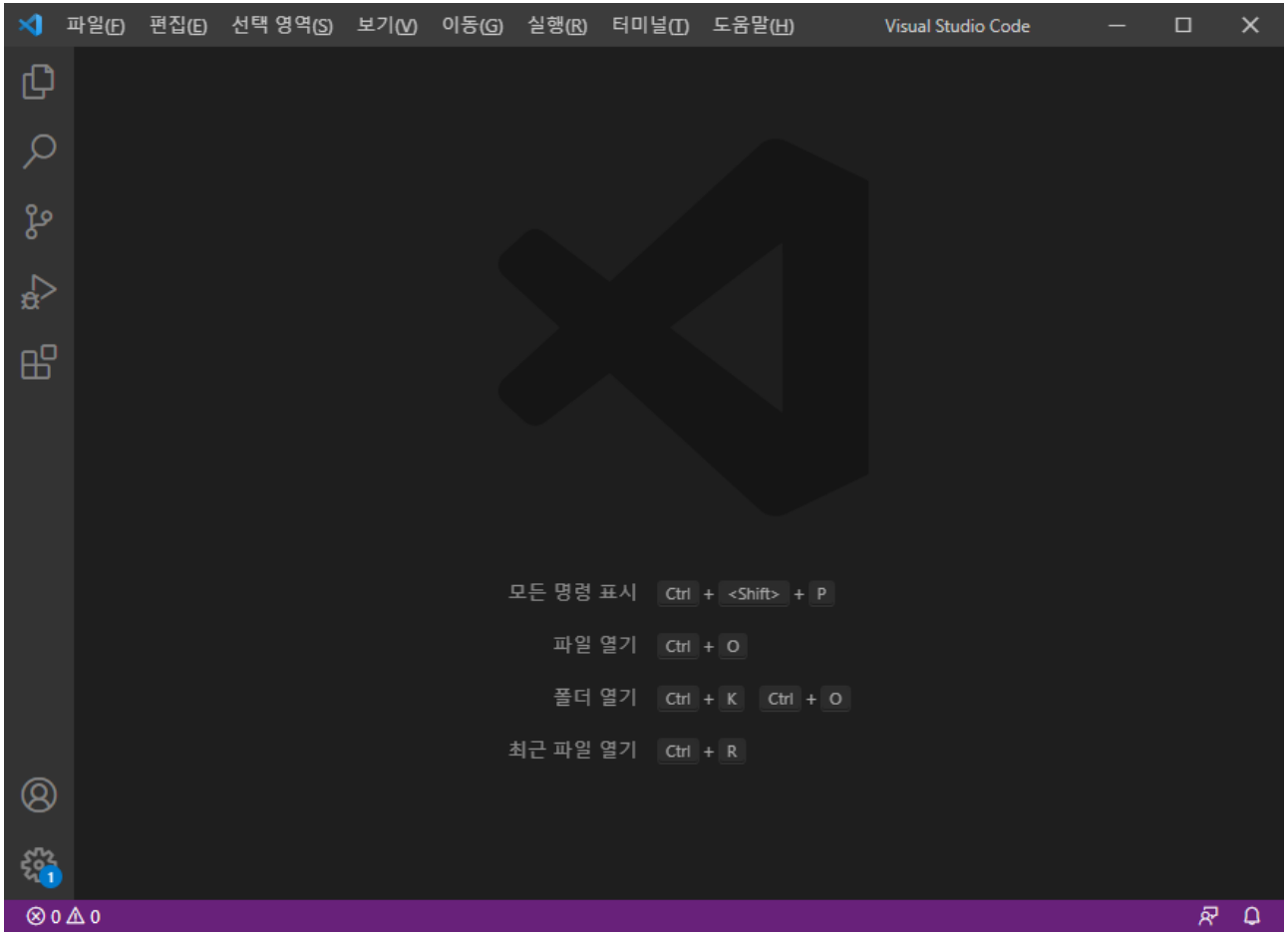
Github

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



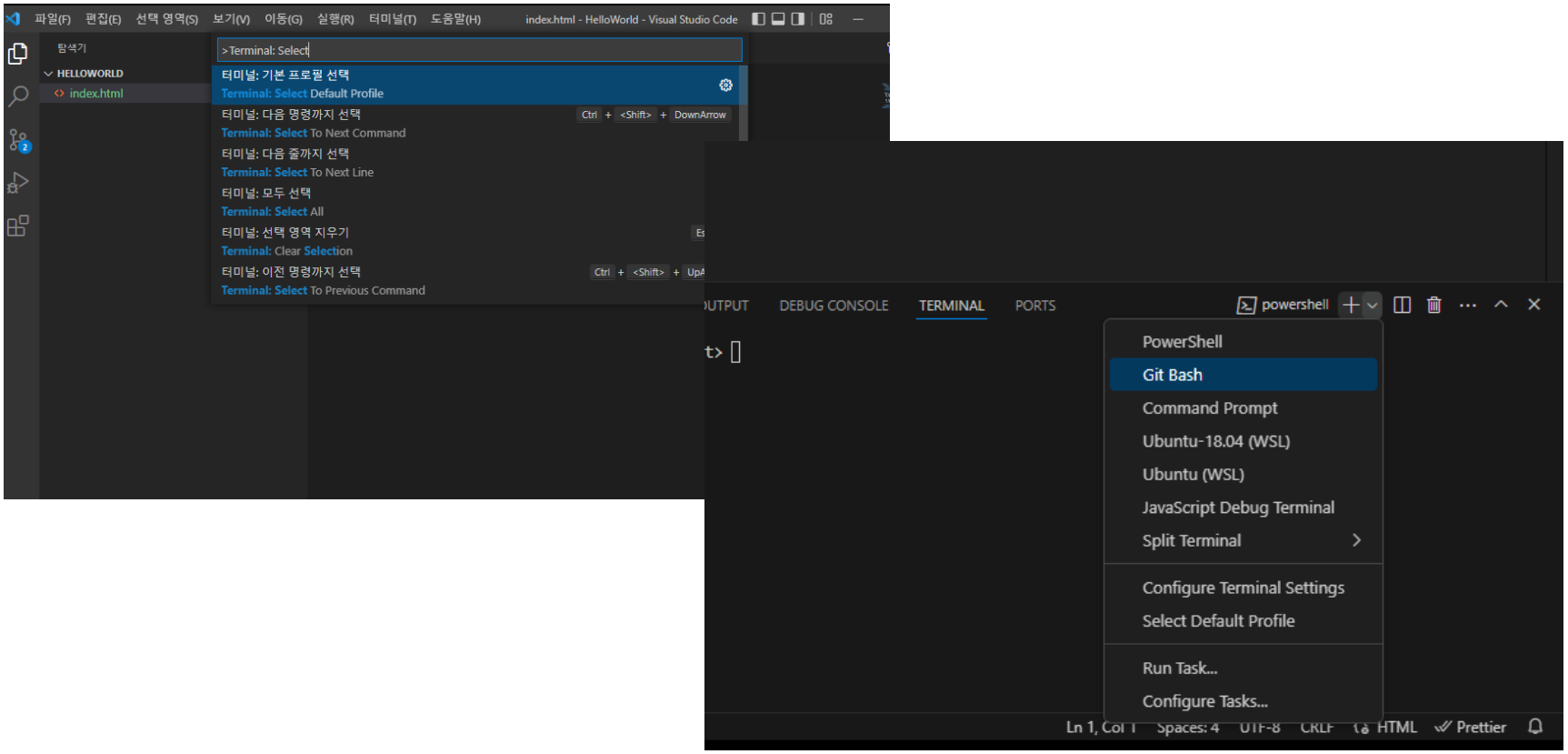
Github

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



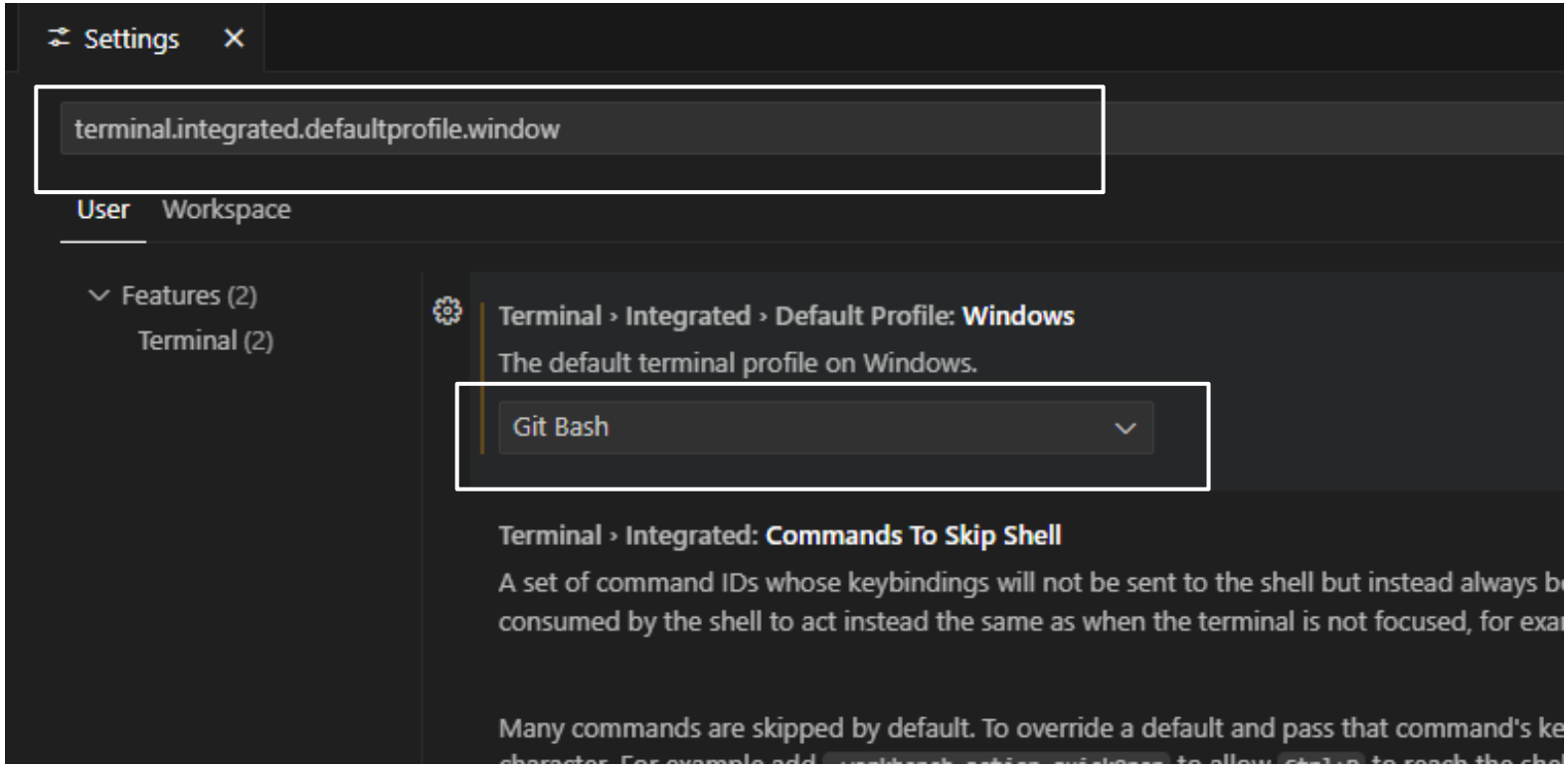
Github

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



Github

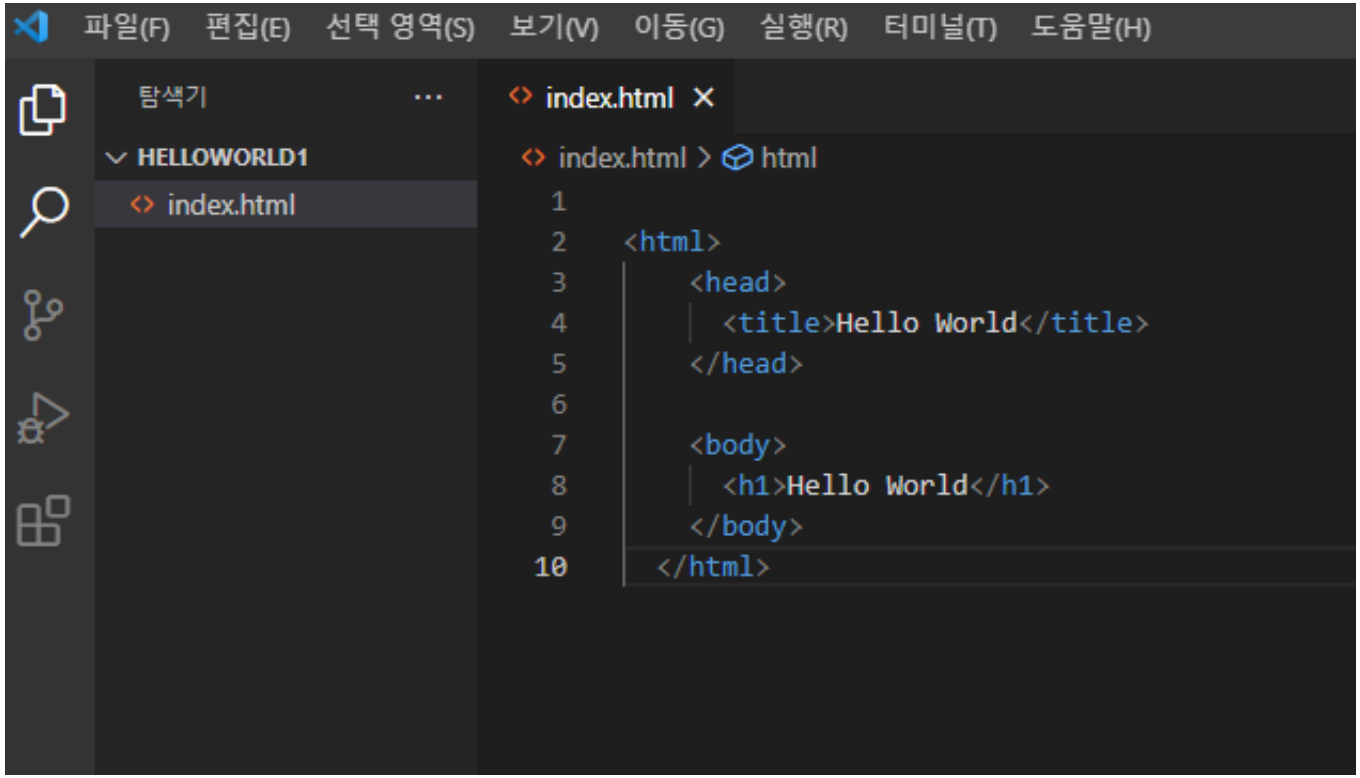
- **Terminal Shell 설정**
  - **Ctrl + ,**
  - **terminal.integrated.defaultprofile.window**
  - **Git Bash 선택**



Github

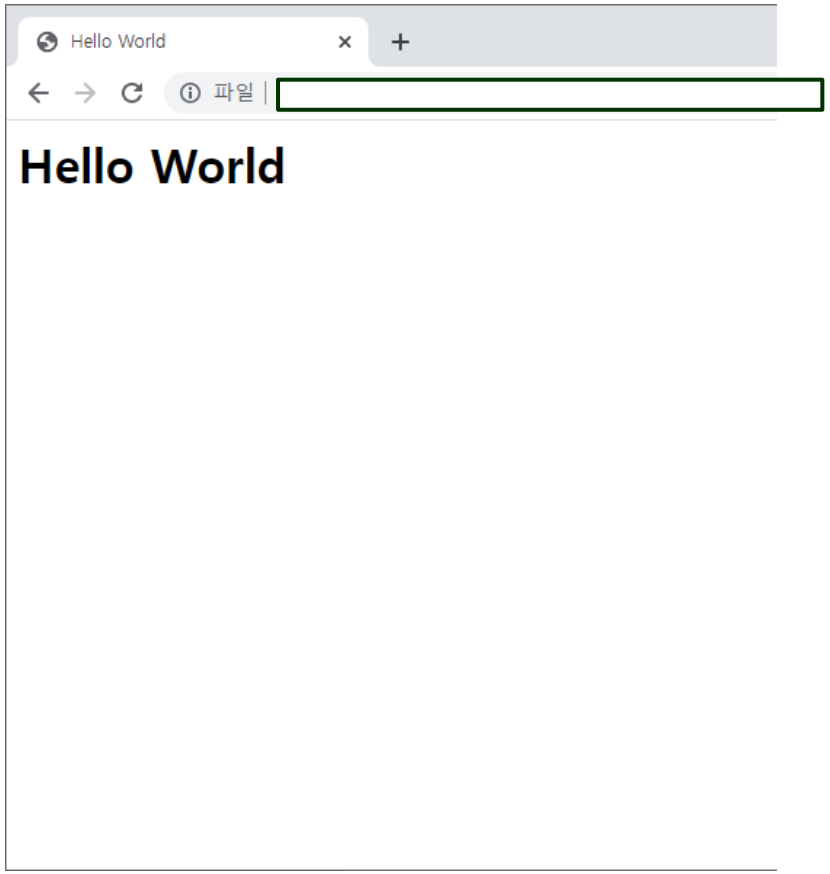
- **HelloWrold Project**
  - **Project실행을 위한 폴더 생성**
  - **VS code 파일 – 폴더열기 : 생성한 폴더 지정**
  - **파일 만들기 : index.html**

HelloWorld



Github

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

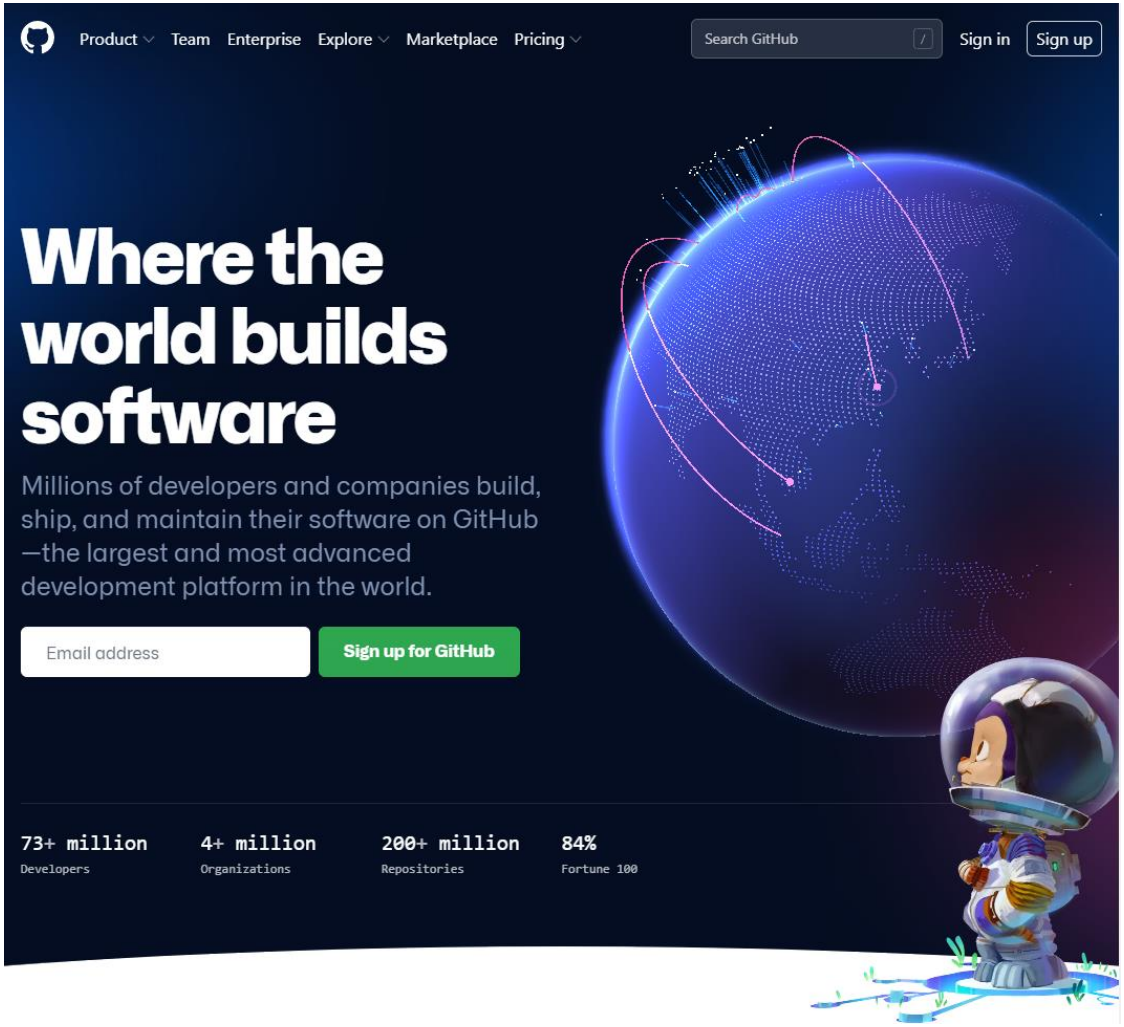


Run 단축키 :

- start debugging (F5)
- 디버깅없이 실행 (Ctrl + F5)
- Stop debugging(shift + F5)
- Restart debugging (ctrl + shift + F5)

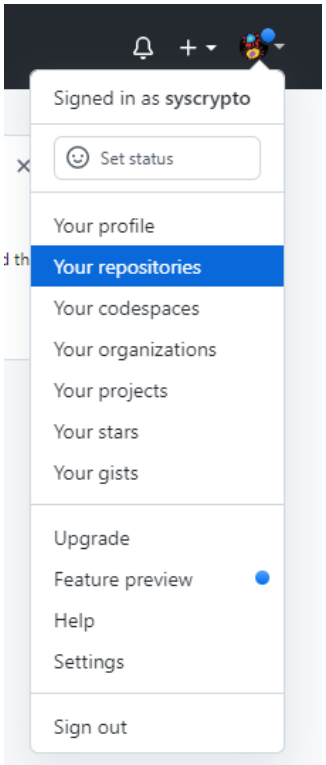
Github

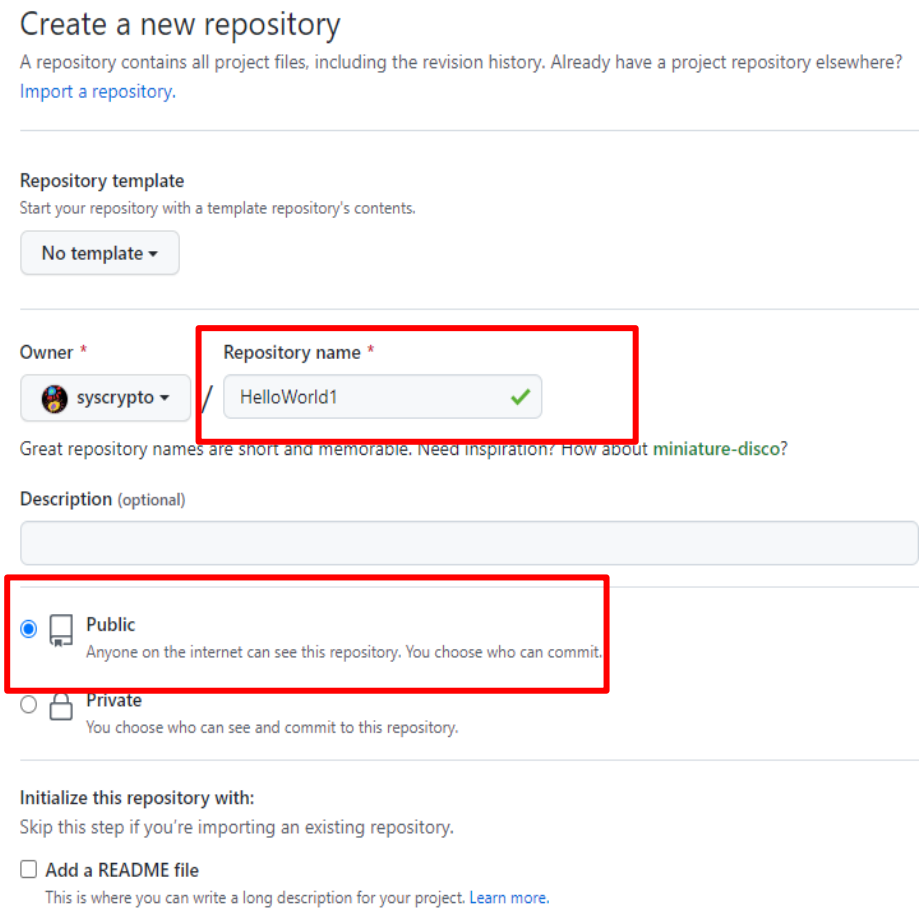
- Github가입



Github

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





Github

- 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
```

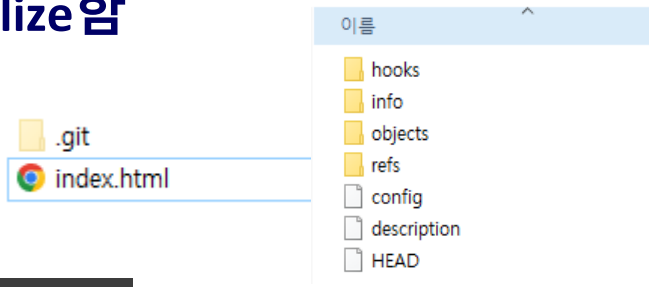
...or import code from another repository

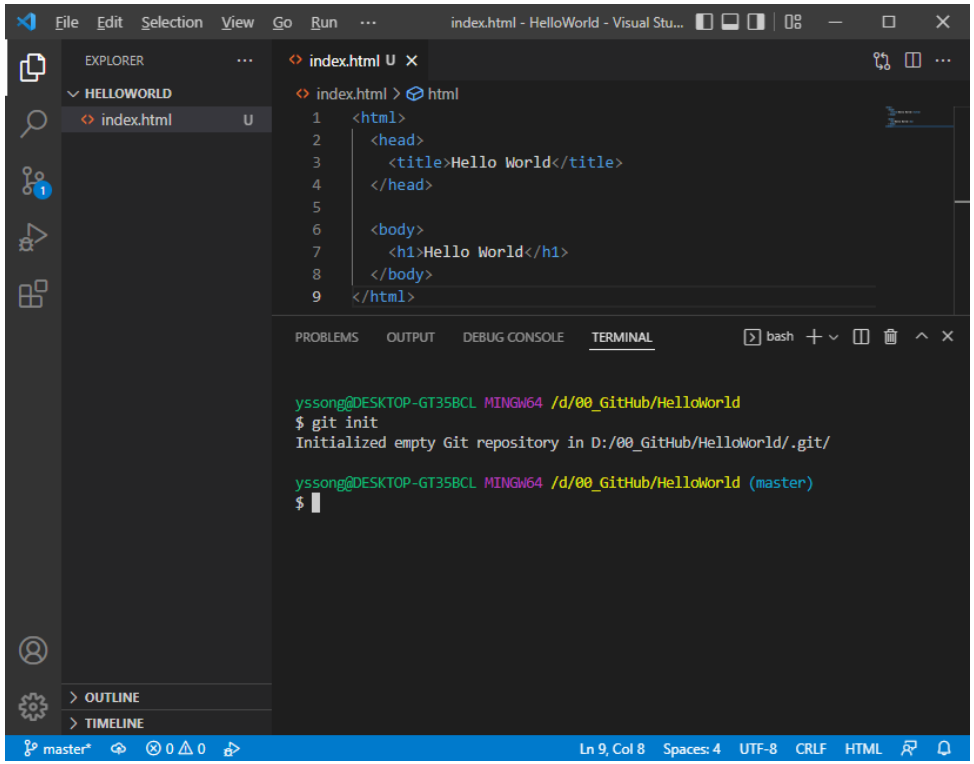
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Github

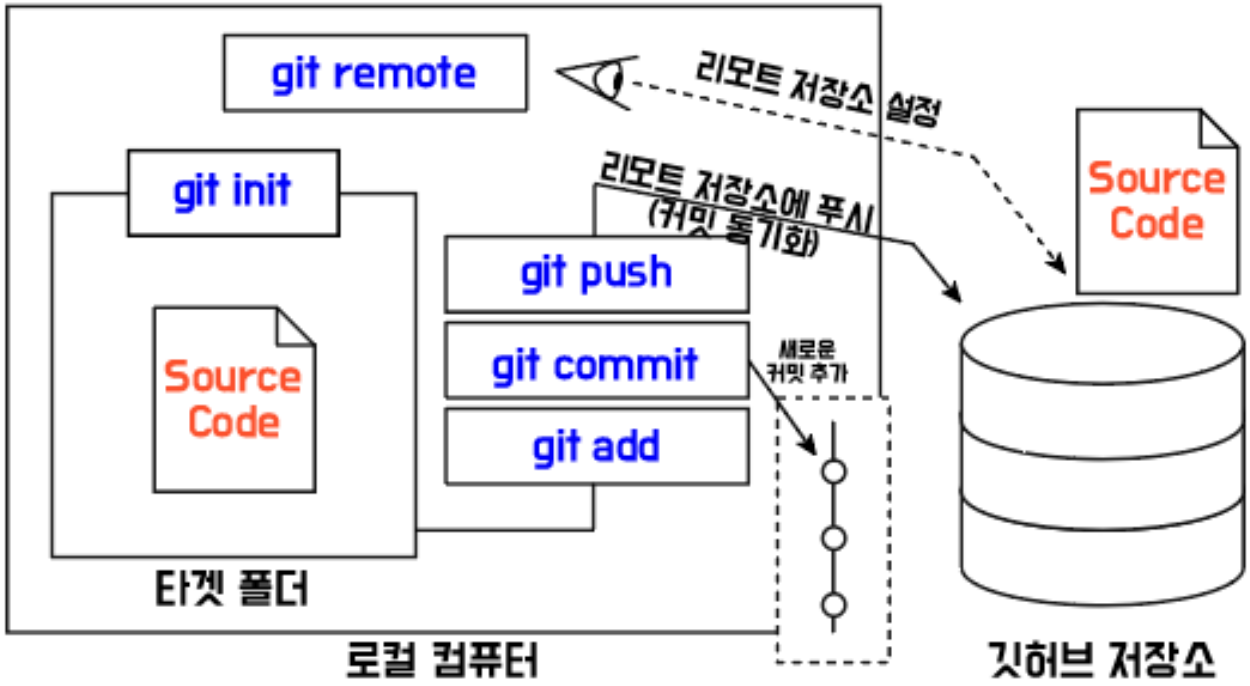
- 초기화
  - 소스코드 폴더를 버전 관리를 위해 initialize함
    - git init
  - Project하위 폴더에 .git폴더 생성 확인
    - 버전관리 정보가 저장





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 main : 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

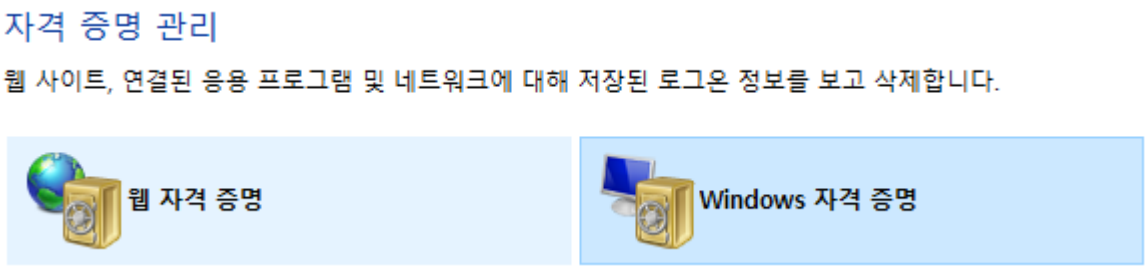
- **Github저장소와 연결**
  - 폴더와 github 저장소의 remote 주소를 연동
    - git remote add *origin* https://github.com/ ...(복사한 주소를 붙여넣기)
    - **git remote –v : 연결 확인**
  - origin : github저장소에 업로드, 다른 주소로도 등록 가능
    - 변경 : git remote set-url origin <git url>

Error Message

오류

```
$ git push origin main
remote: Permission to <git주소> denied to <git hub 계정1>
fatal: unable to access '<git주소>/': The requested URL returned error: 403
```

- **Window – 자격증명관리자**
- **Github 사용자 계정 확인 및 변경**

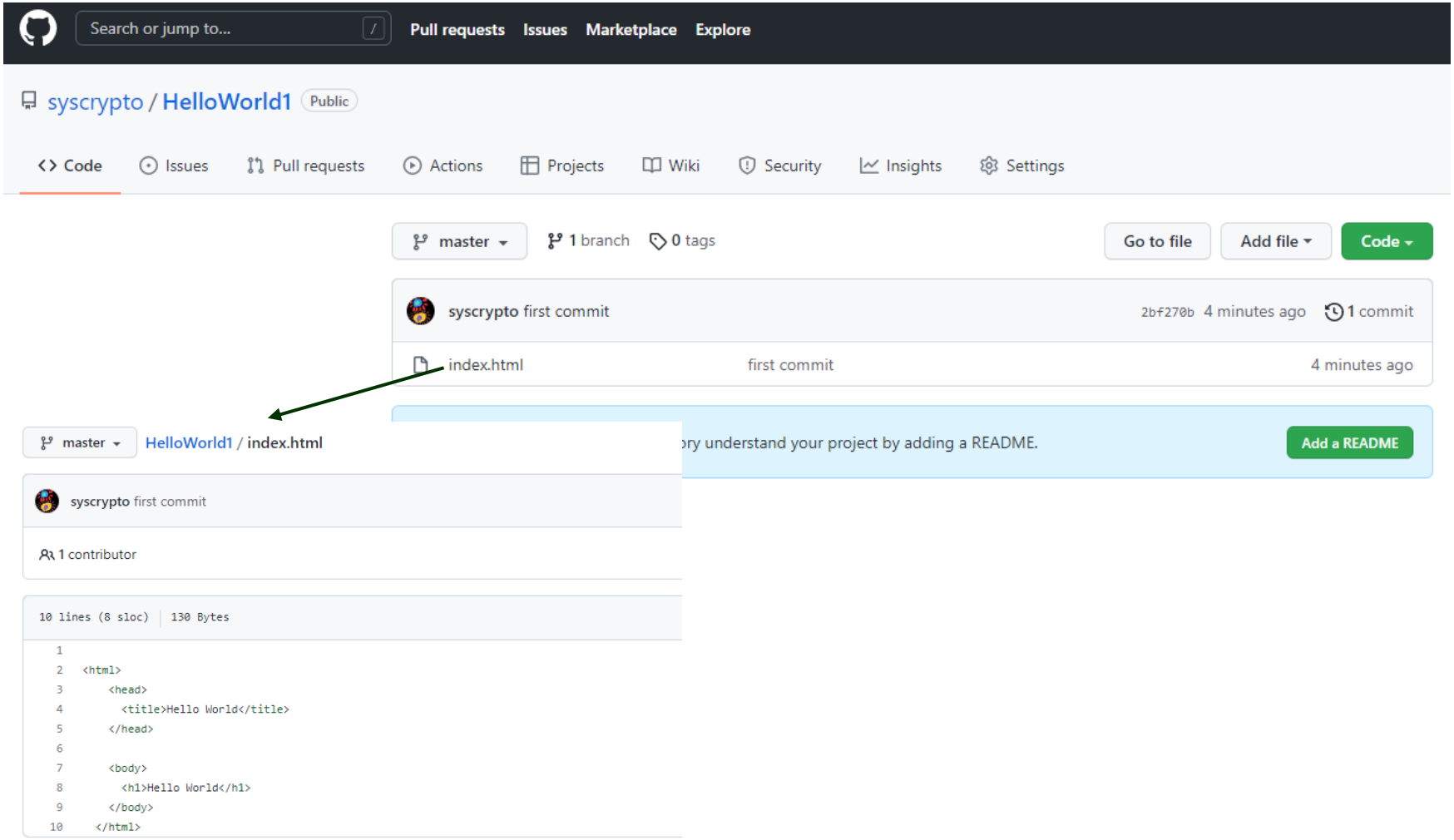



```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git status
On branch main
nothing to commit, working tree clean
```

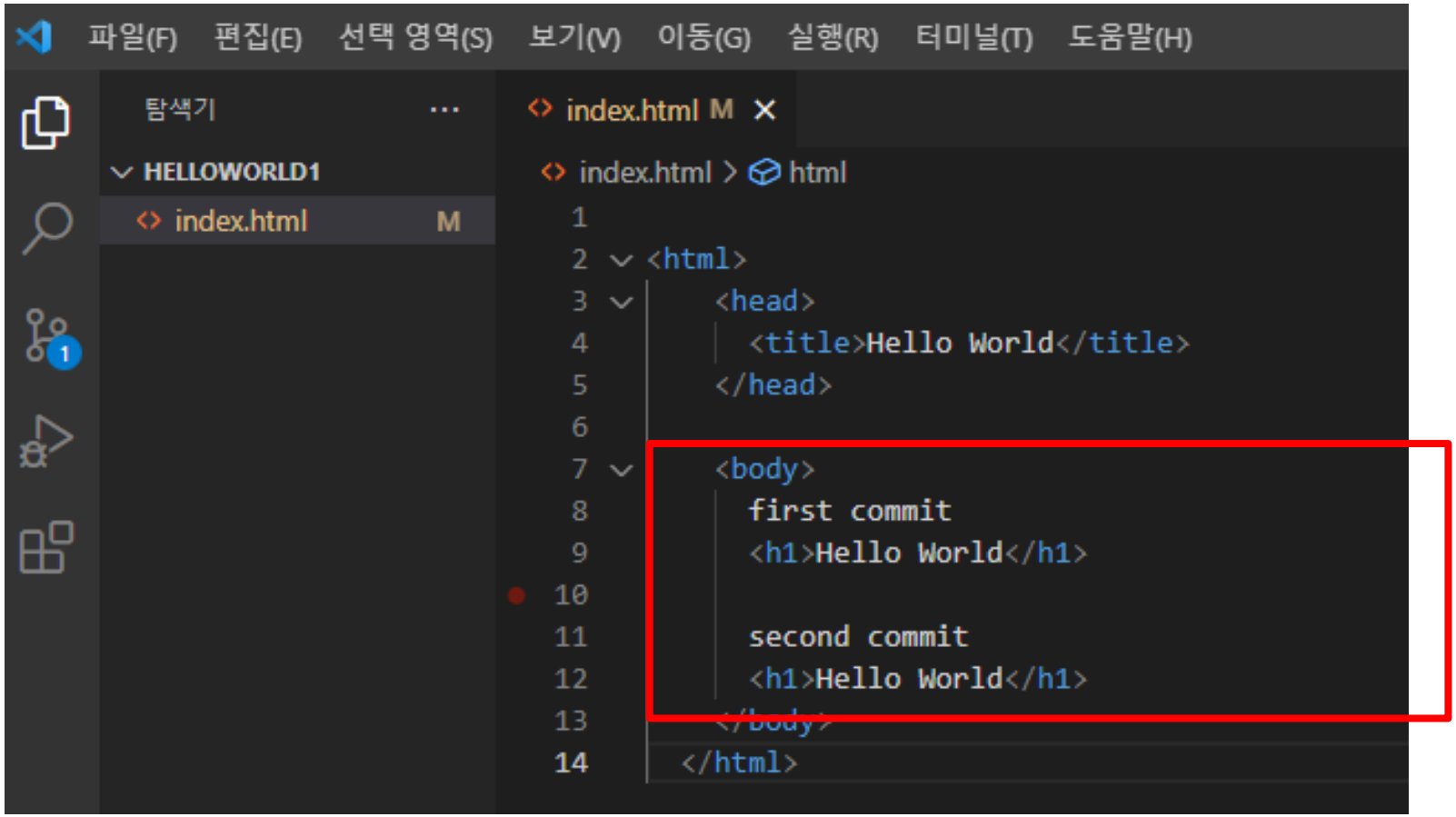
Github

- **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/Hel
$ 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

second commit

Browse files

master

syscrypto committed 7 minutes ago 1 parent 2bf270b commit f6bbbf0c9e40dab6c3ae0788e7c88776a7285860

Showing 1 changed file with 4 additions and 0 deletions.

Split | Unified

4 index.html

```
@@ -5,6 +5,10 @@
 5        </head>
 6
 7        <body>
 8          <h1>Hello World</h1>
 9        </body>
10      </html>
```

```
 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

- **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>
```

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

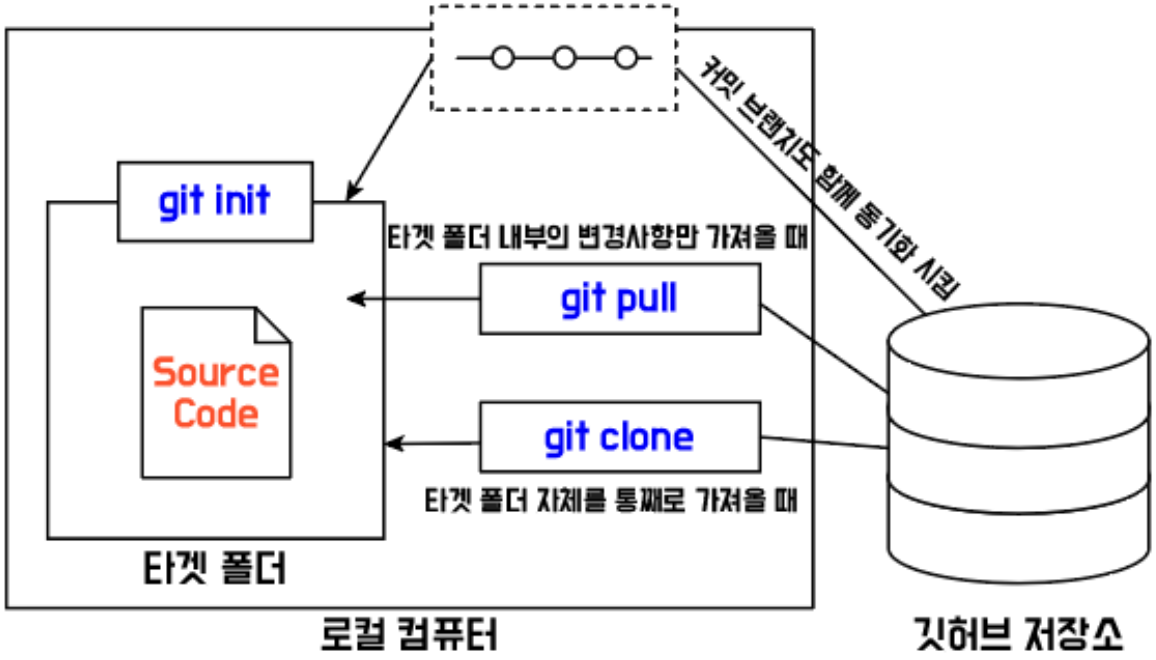
master | 1 branch | 0 tags | Go to file | Add file | Code

syscrypto Update index.html 840fdb0 now 3 commits

index.html Update index.html now

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 > ...

```html

<html>
    <head>
      <title>Hello World</title>
    </head>

    <body>
      first commit
      <h1>Hello World</h1>

      second commit
      <h1>Hello World</h1>

      Modify line
      <h1>Hello World</h1>

    </body>
  </html>

```

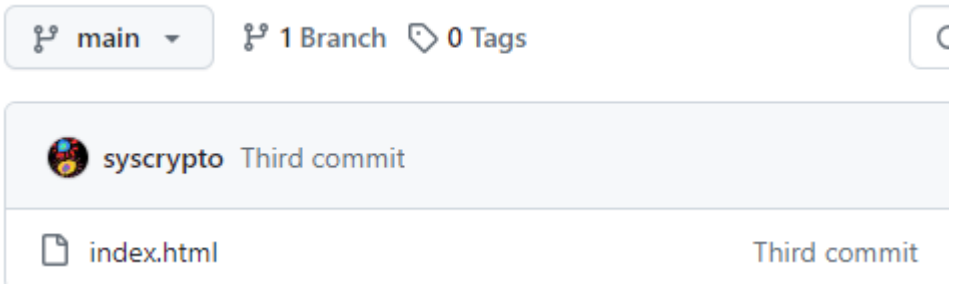
Github

- **Third commit**

index.html > html > body

```html

<html>
    <head>
      <title>Hello World</title>
    </head>

    <body>
      first commit
      <h1>Hello World</h1>

      second commit
      <h1>Hello World</h1>

      Modify line
      <h1>Hello World</h1>

      third commit
      <h1>Hello World</h1>

    </body>
  </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
```



Checking

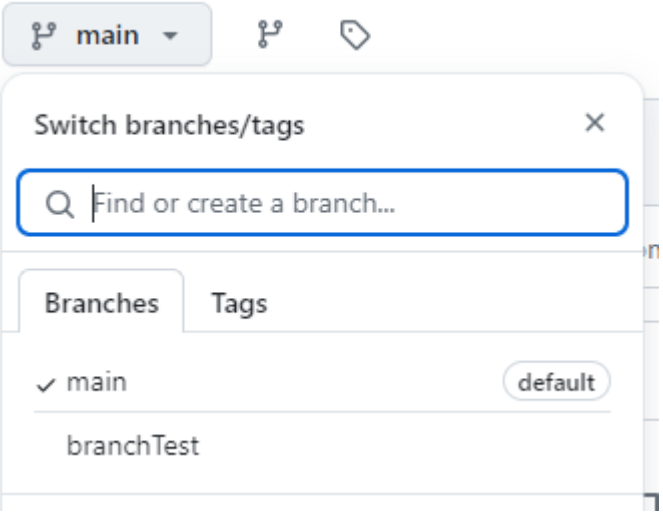
- 변경된 내용 없이 commit 하는 경우

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git add .

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git commit -m "third commit"
On branch main
nothing to commit, working tree clean

swcu_sys@DESKTOP-3S6F5MB MIN
$ git push origin main
Everything up-to-date
```

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git add .

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git commit -m "fourth commit"
[main c7787eb] fourth commit
 1 file changed, 2 insertions(+)

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/syscrypto/HelloWorld.git
   dad28af..c7787eb  main -> main
```

Branch 생성 및 활용

- Branch 생성
  - git branch 브랜치이름
- Branch로 전환
  - git checkout 브랜치이름
- Github에 반영
  - git push origin 브랜치이름



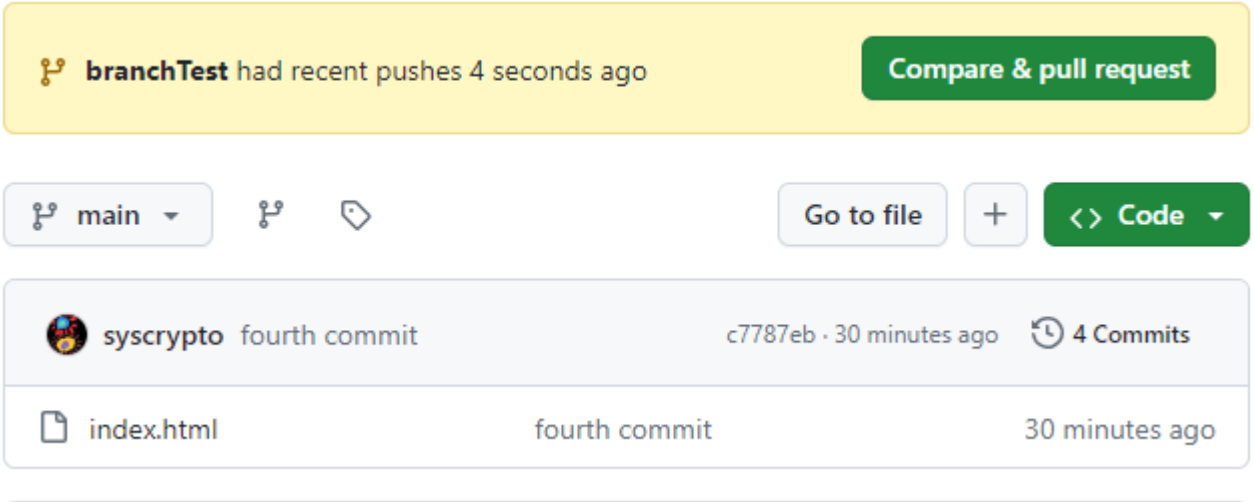
- git checkout –b 브랜치이름

- 수정 파일 브랜치에 추가

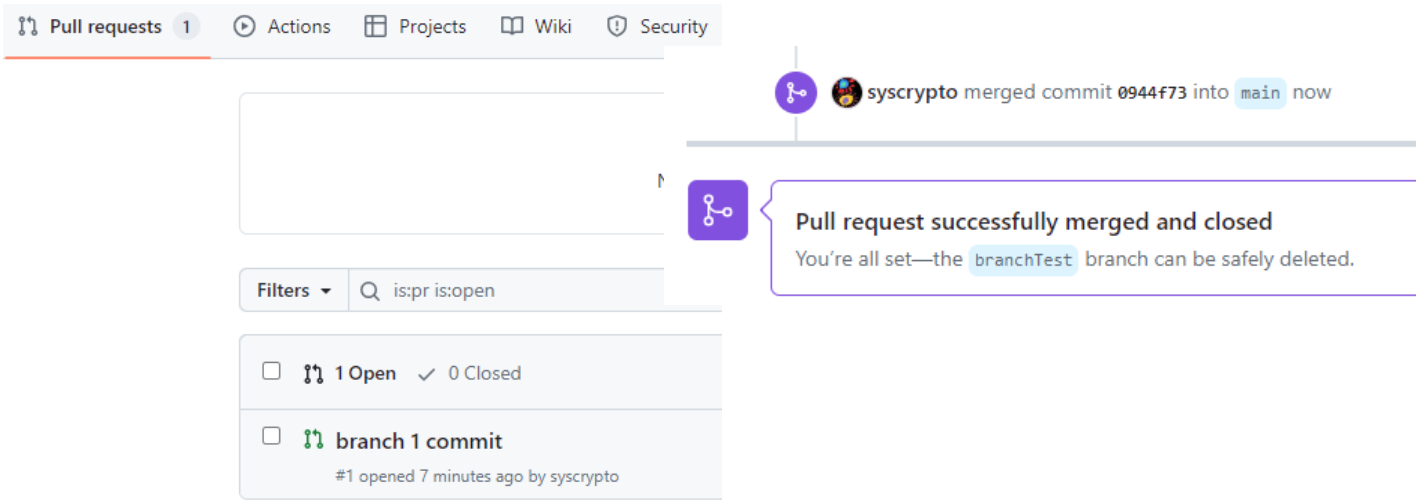
```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (branchTest)
$ git add .

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (branchTest)
$ git commit -m "branch 1 commit"
[branchTest 67d202b] branch 1 commit
 1 file changed, 4 insertions(+)

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest (branchTest)
$ git push origin branchTest
```

Branch 생성 및 활용

- **Github branch compare & pull request**

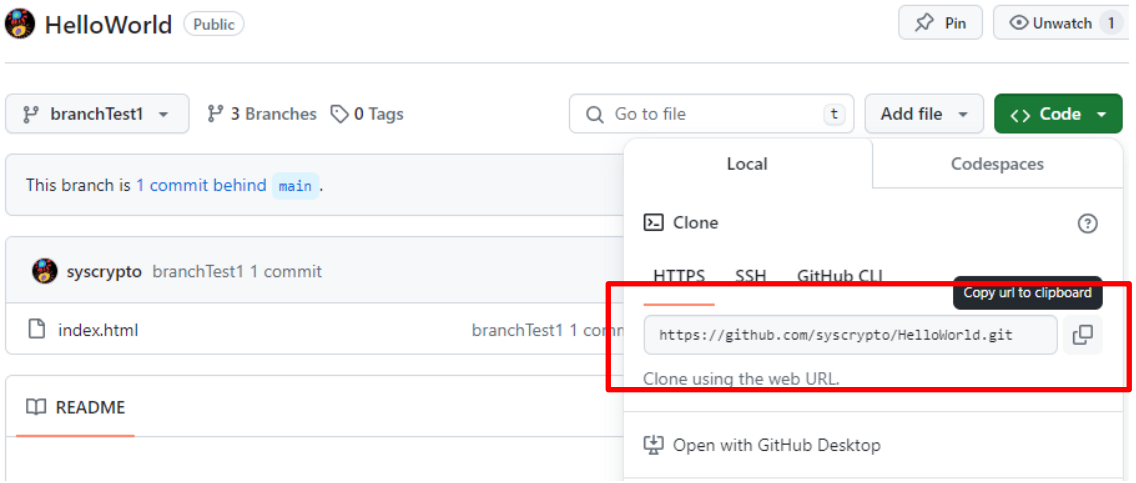


- **Merge**



Clone

- Vscode 새로운 창열기
- Terminal 열기
- 위치 이동
  - cd / -> cd 폴더명



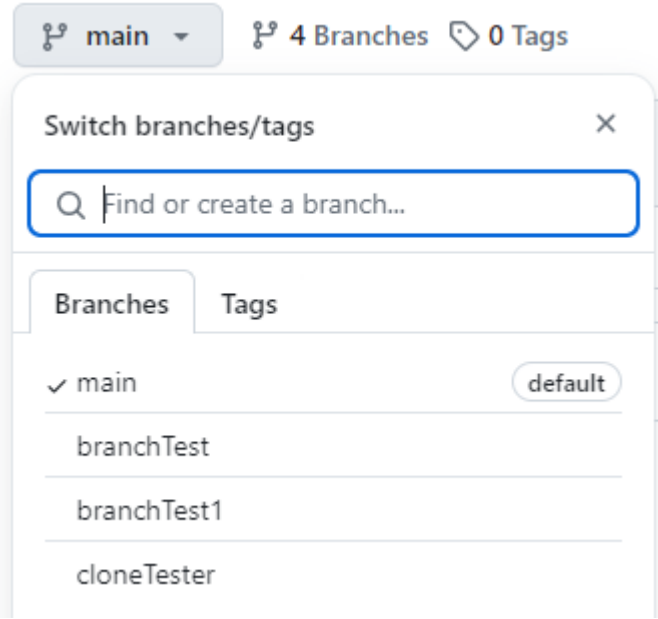
- Clone
  - git clone [github clone url] [받아올 디렉토리]
  - Ex. : git clone https://github.com/XXXXX/HelloWorld.git cloneTest
  - 받은 파일을 수정 후 main push

- Clone branch 생성
  - git checkout –b cloneTester
  - 현재 branch 확인 : git branch

Clone

- **Branch 확인 : git branch**

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest/cloneTest (cloneTester)
$ git branch
* cloneTester
  main
```



- **파일 수정 후 commit**
  - Git add .
  - Git commit –m “cloneTester commit”
  - Git push origin cloneTester

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest/cloneTest (cloneTester)
$ git add .

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest/cloneTest (cloneTester)
$ git commit -m "cloneTester branch"
[cloneTester 2e851ba] cloneTester branch
 1 file changed, 4 insertions(+)

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/gitTest/cloneTest (cloneTester)
$ git push origin cloneTester
```

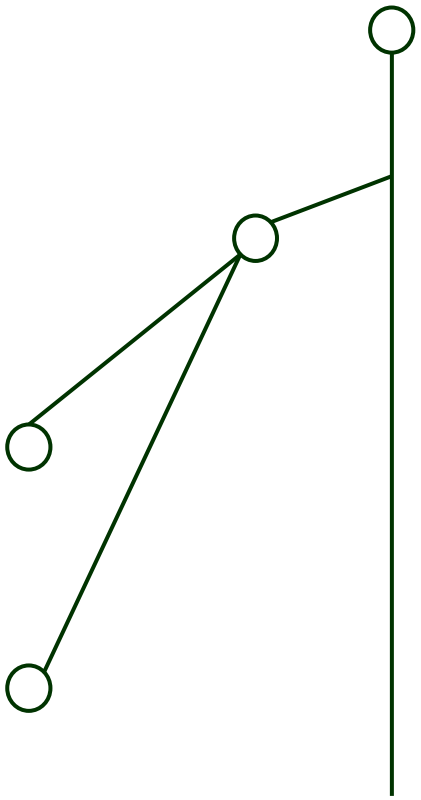
Github Collaboration

Project Manager
(branch : main)

Main copy branch
(branch : develop)

Programmer1
(branch : function-a)

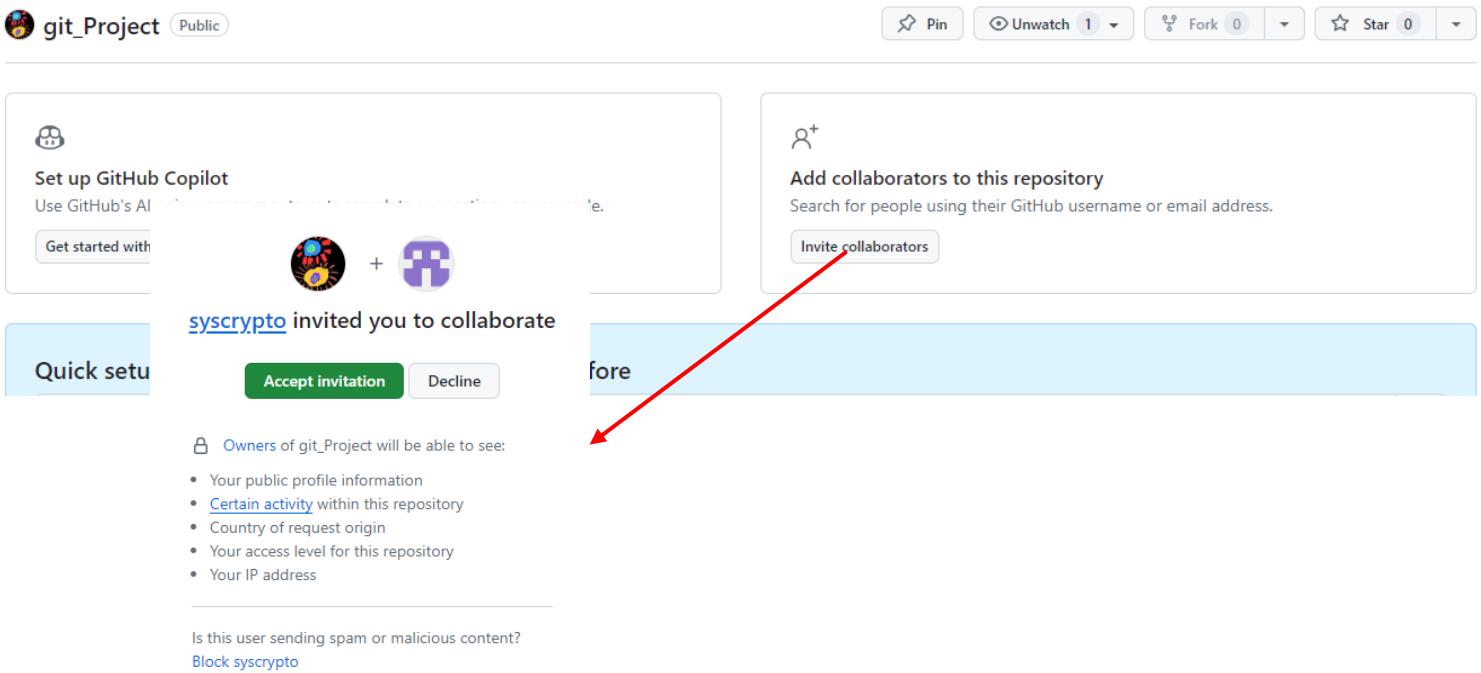
Programmer2
(branch : function-a)



Github project menu활용

Repository 생성

- **Repository name : git_Project**
- **Invite collaborators**
  - **Email or username으로 초대**
  - **초대 수락 : Accept invitation**



Repository 생성

- **Repository name : git_Project**
- **초기 셋팅**

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

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

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitign

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

```
echo "# git_Project" >> 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/git_Project.git
git push -u origin main
```

- **Main code commit**
  - **git add .**
  - **git commit –m "main second commit"**
  - **git push origin main**

Main Branch 복사본 만들기

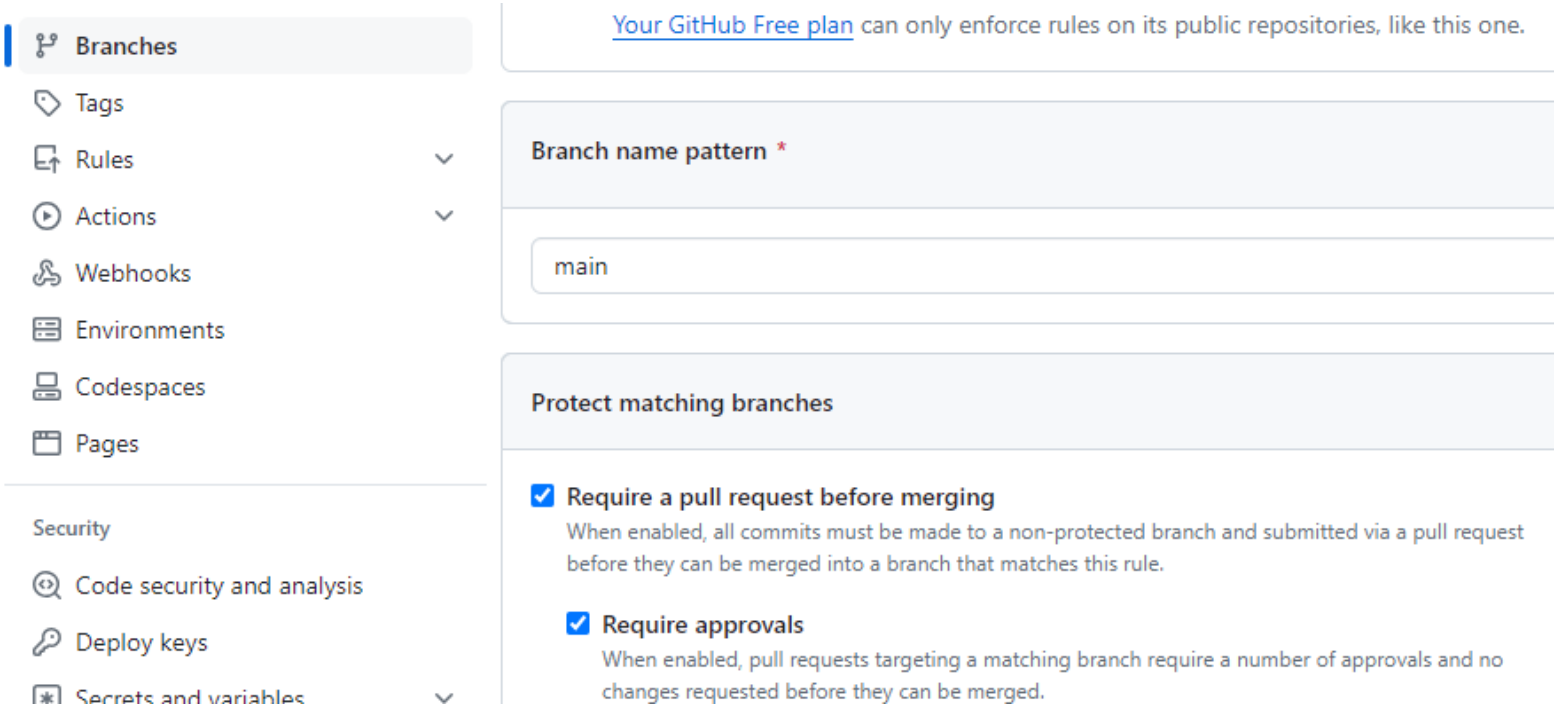
- **Develop branch**
  - Main branch는 서비스용 파일
  - 복사본을 만들어 main에 merge하기 전에 테스트 및 검증용으로 사용

- **develop branch 생성**
  - git checkout –b develop
  - git push
  - git push --set-upstream origin develop
  - Github에서 branch 확인

- **참고 사항**
  - Branch 변경 : git switch branch명 또는 git checkout branch명
  - Branch 확인 : git branch

Main branch Protection

- **Settings – Branches – Add branch protection rule**
  - **Branch name pattern : main**
  - **Protect matching branches**
    - **Require a pull request before merging**
    - **Lock branch**



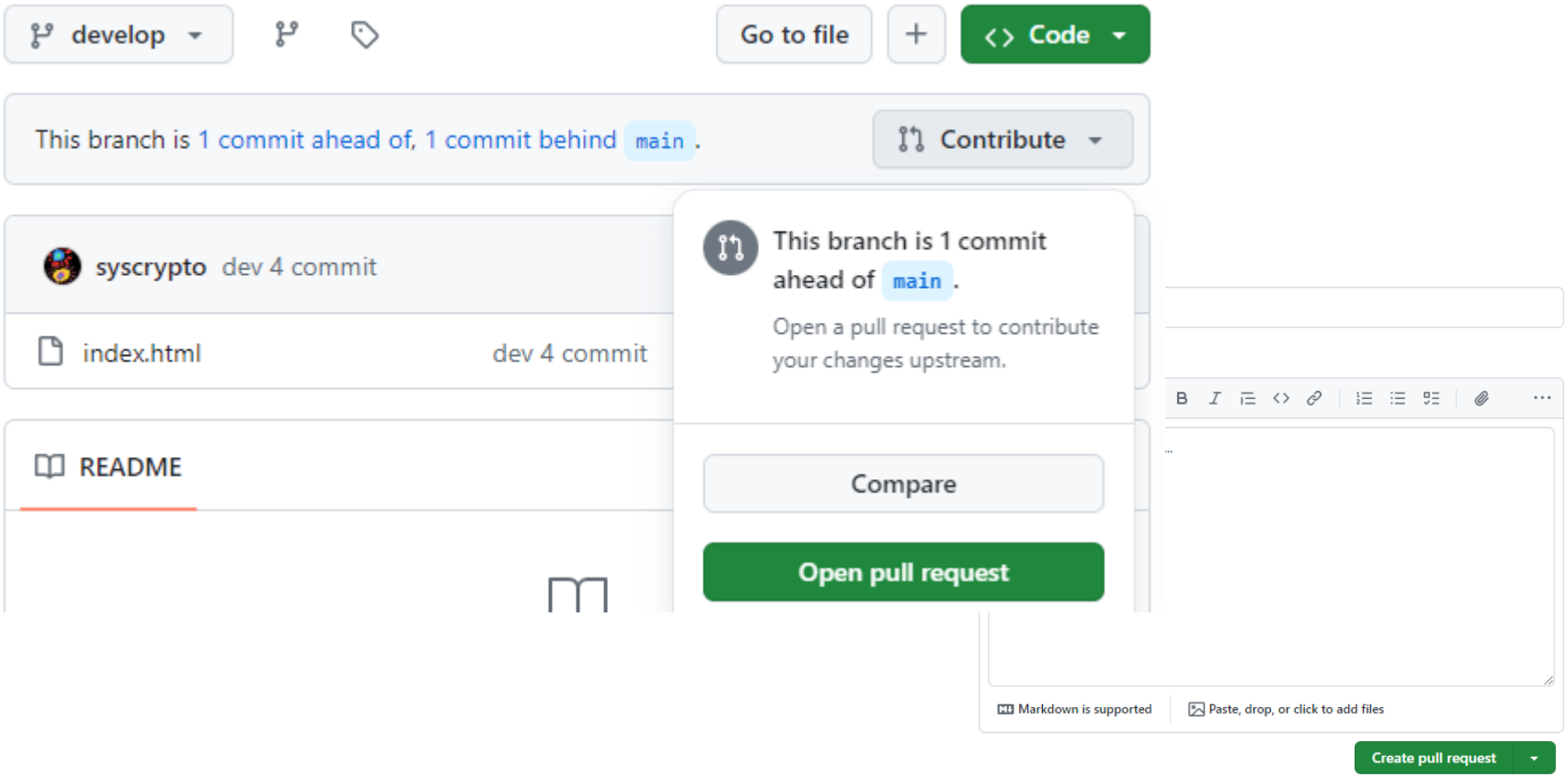
Protect Test

- Developer가 수정한 코드를 직접 main에 올릴 경우 error

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git add .

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git commit -m "dev 3 commit"
[develop 981d7ad] dev 3 commit
 1 file changed, 2 insertions(+)

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git push origin main
To https://github.com/syscrypto/git_Project.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/syscrypto/git_Project.git'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. If you want to integrate the remote changes, use 'git pull'
hint: before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git push origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 159.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/syscrypto/git_Project.git
   36ed580..981d7ad  develop -> develop
```

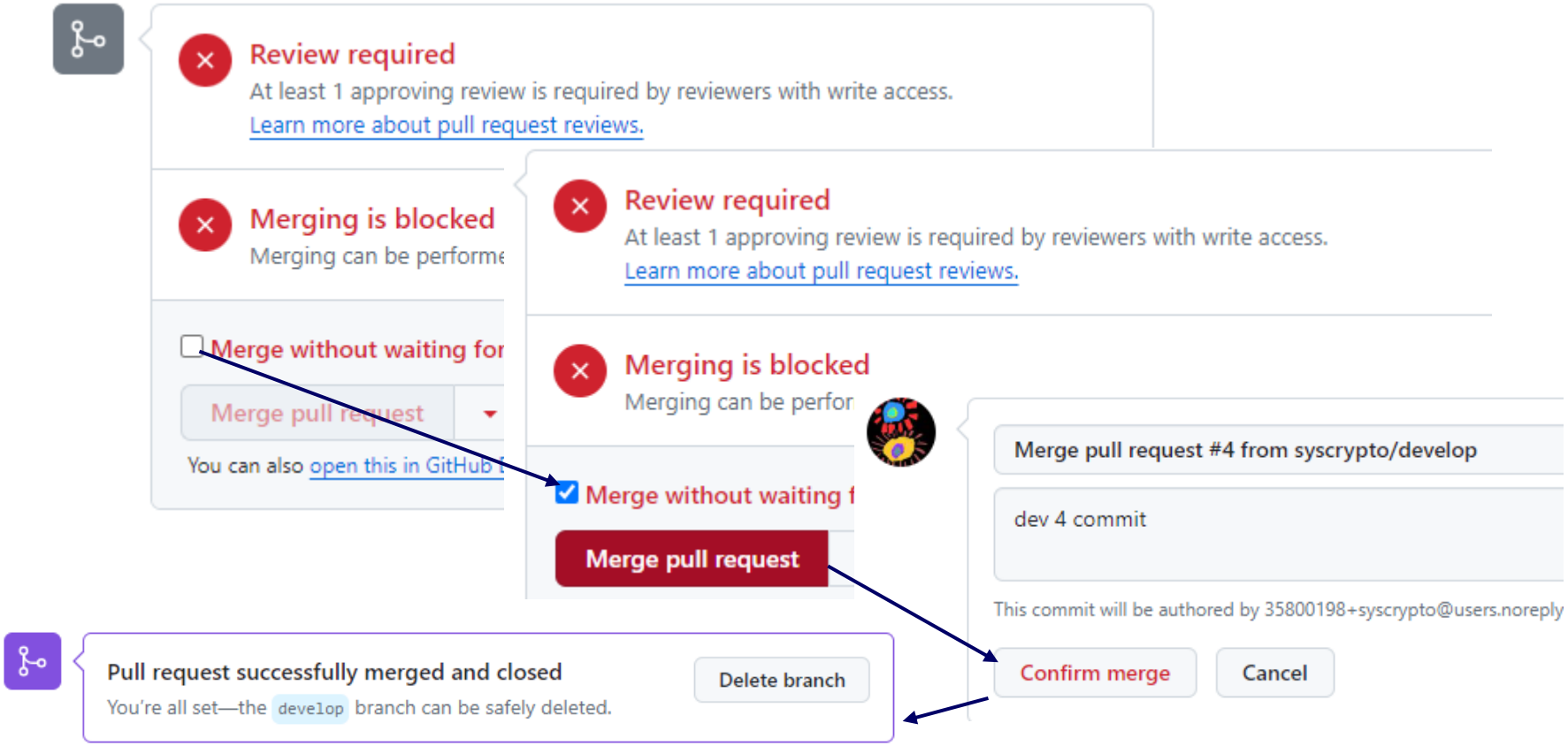
Main branch 에 merge

- Commit된 파일 확인
- Contribute에서 open pull request
- Create pull request



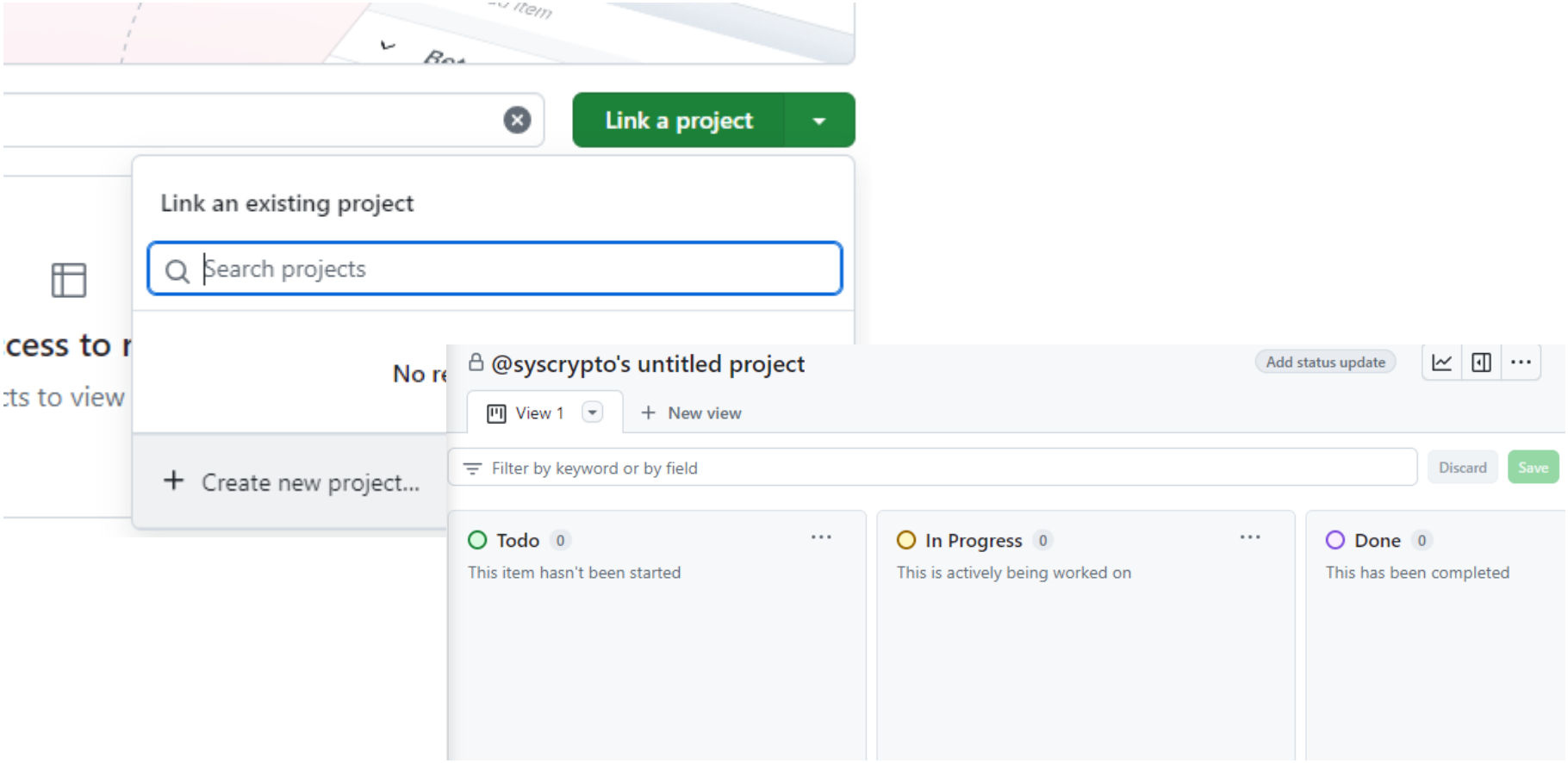
Main branch 에 merge

- 현재 main branch는 protect되어 있음
- Checkbox 선택 후 Merge pull request 선택
- Confirm merge



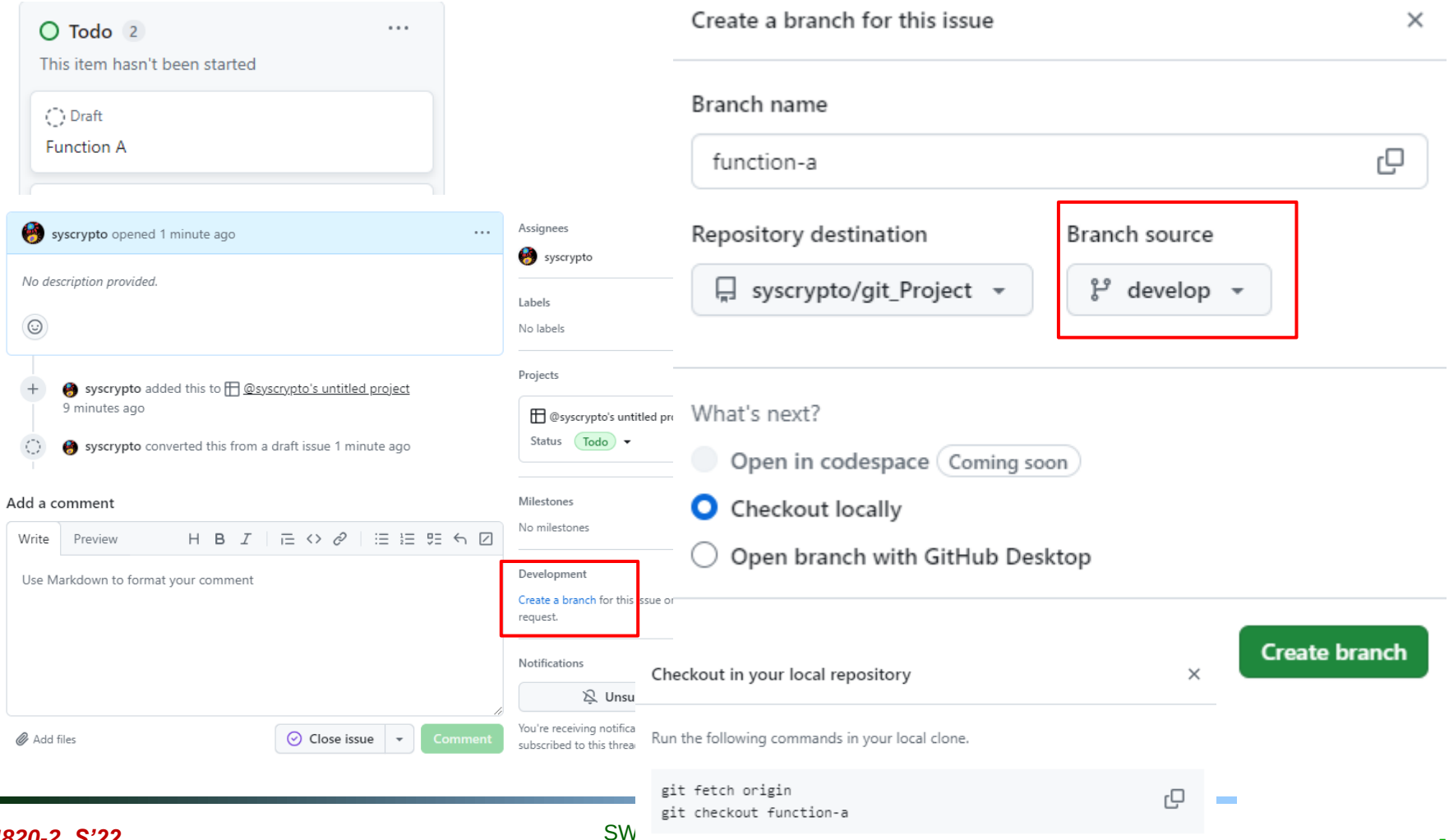
Project Management

- **Projects – Link a project – Create new project – Board – Create project**



Project Management

- **Add item – Function A, Function B – A, B중에 선택 – convert to issue – Create a branch – Branch Source(develop)**



Project Management

- Branch 정보 copy
- Vscode terminal에 paste -> branch 생성

Checkout in your local repository ×

Run the following commands in your local clone.

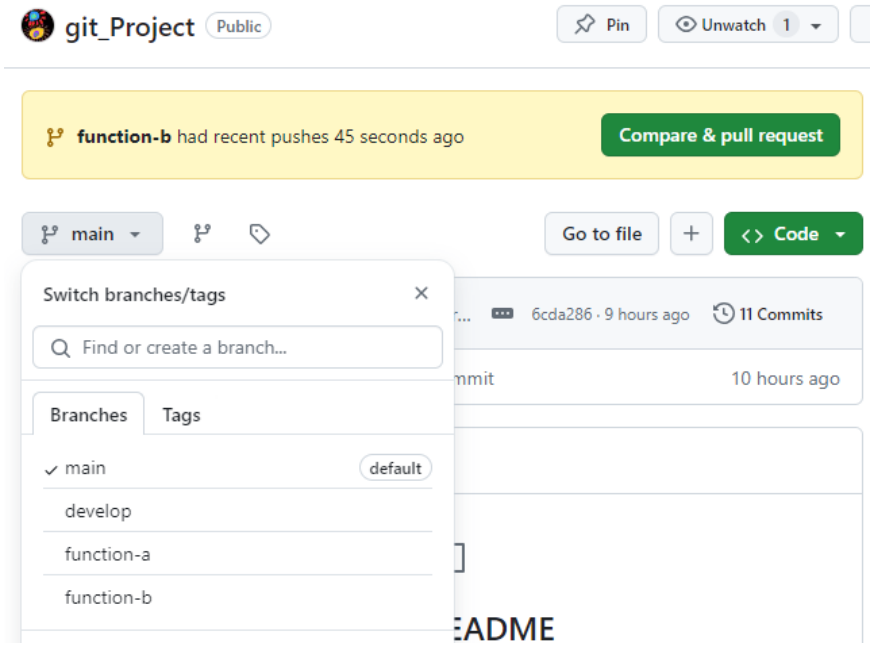
```
git fetch origin
git checkout function-a
```

```
swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git fetch origin
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 1.75 KiB | 162.00 KiB/s, done.
From https://github.com/syscrypto/git_Project
 * [new branch]      function-a -> origin/function-a
   3560444..6cda286  main       -> origin/main

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git checkout function-a
branch 'function-a' set up to track 'origin/function-a'.
Switched to a new branch 'function-a'
```

Project clone

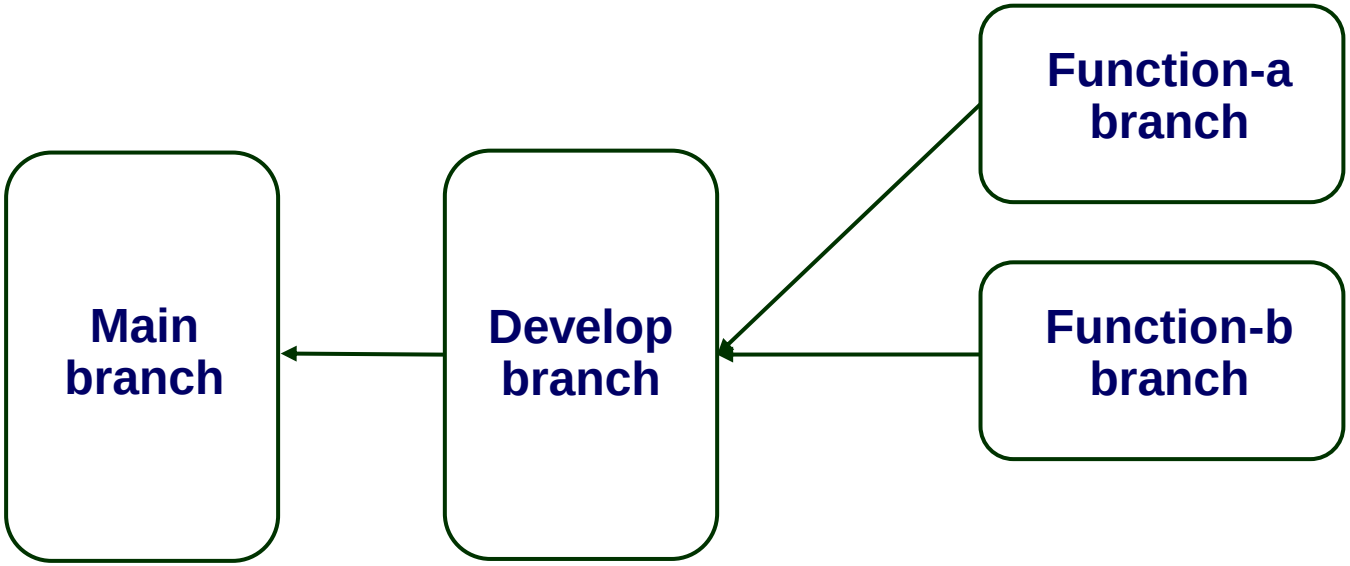
- Programmer1이 git_Project 에 참여
  - 프로젝트 시작하려는 위치에서 clone
  - git clone project url directory name
    - 현재 작업하는 디렉토리 내에 만들면 안됨

- Github project에서 Function-b branch생성 및 assignment
- 코드 수정 후 commit
  - git add .
  - git commit –m “pro1 first commit”
  - git push origin function-b

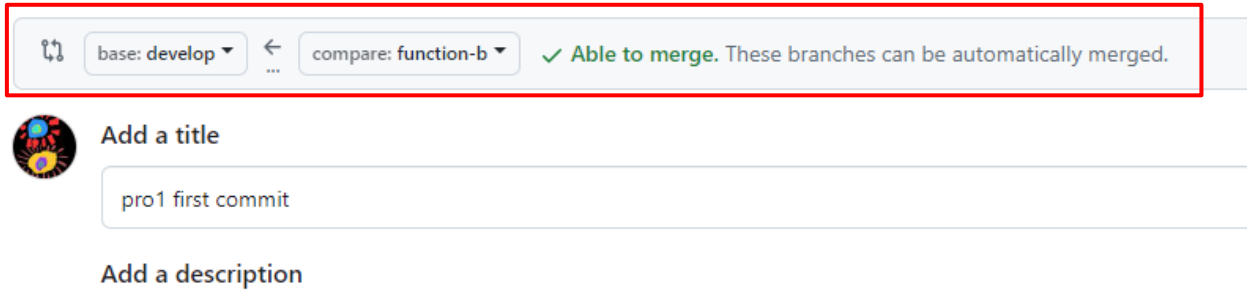


Project management

- Programmer1이 작업한 파일을 function-b로 commit하고
- 이를 develop branch로 이동

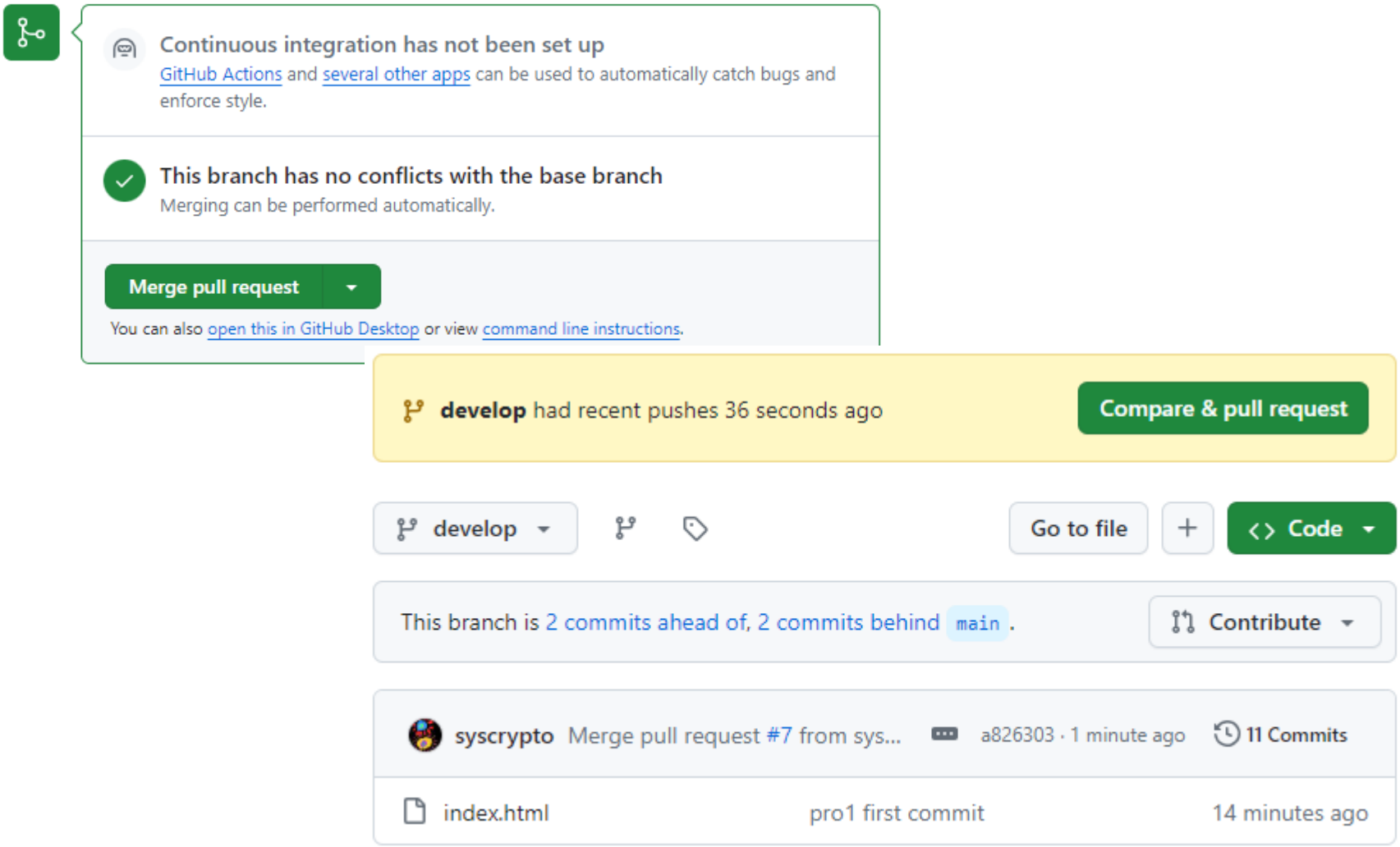


- Pull request
  - Function-b branch에서 develop branch로



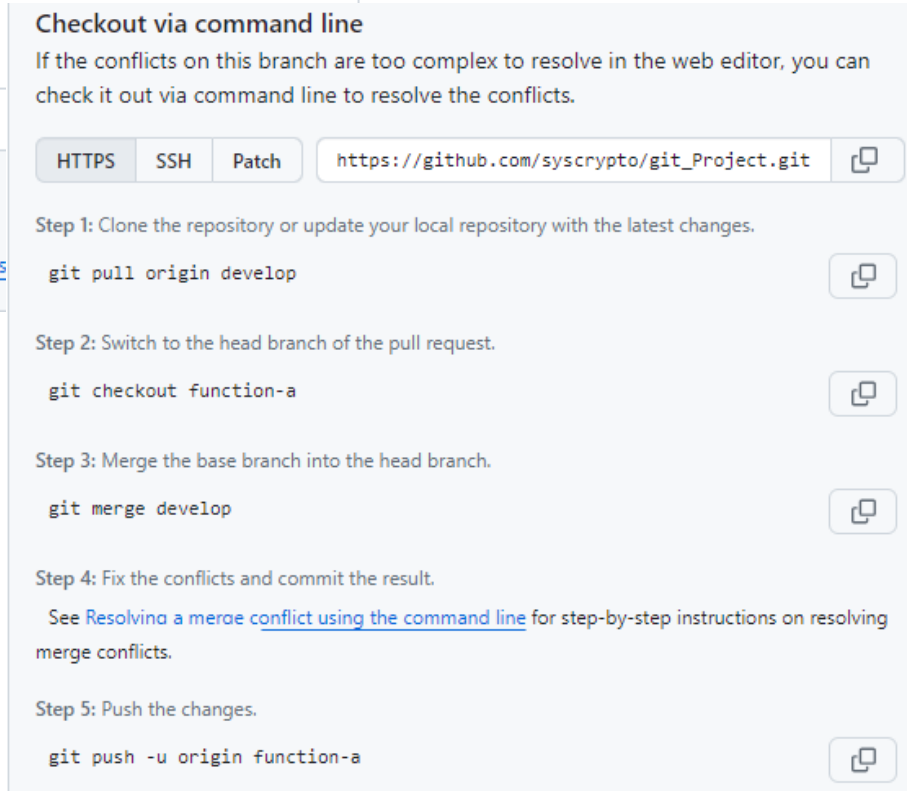
Pull request

- **Checking & Merge**



Conflict 해결

- PM이 작성한 수정 코드를 develop branch에 commit하고 pull request한 후 Merge 전에 conflict 발생



- Command line click
  - git switch develop 변경 후
  - 순차적으로 수행

Merge – conflict해결

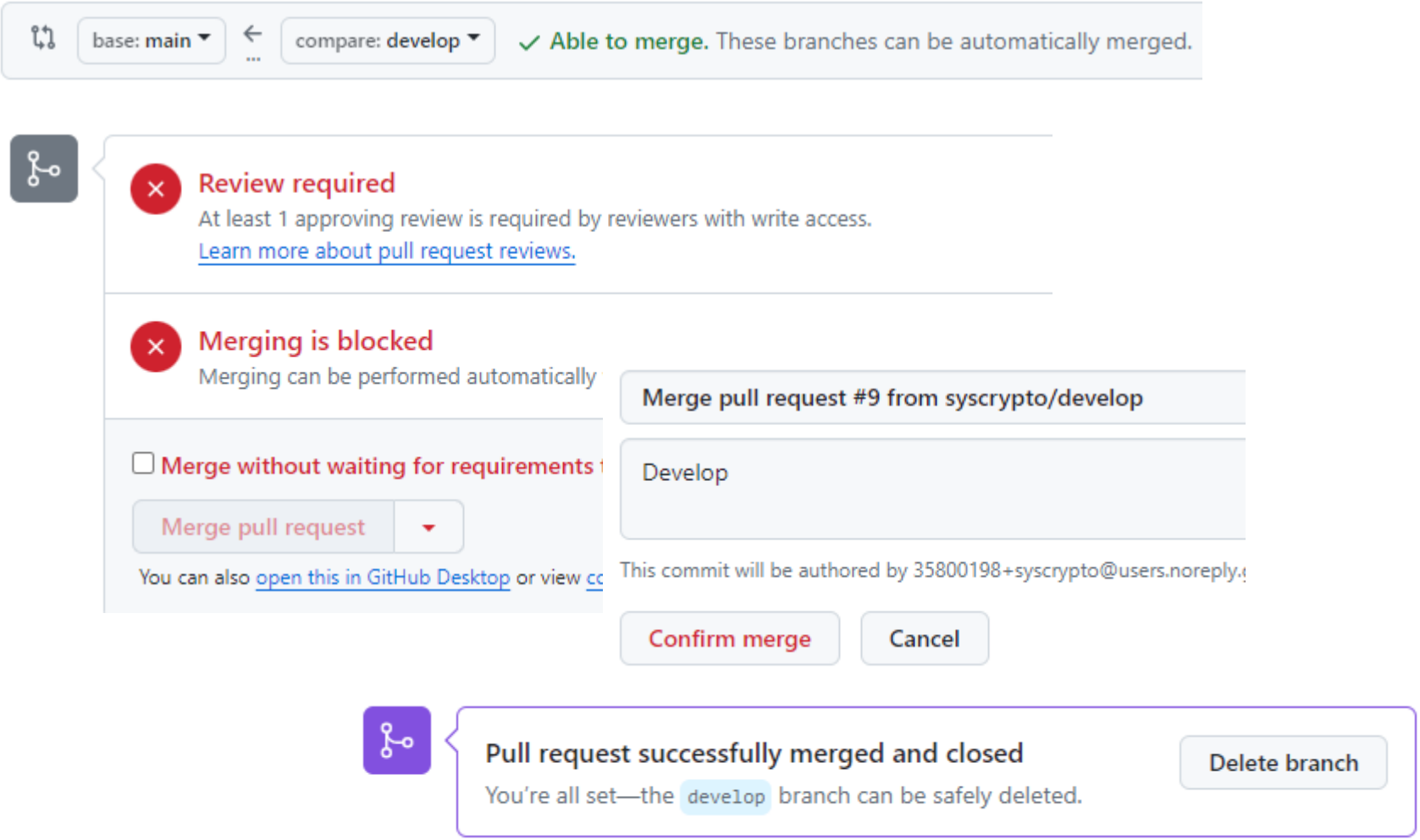
```
17          <h1> developer가 추가 코드를 main에 merge 예제 </h1>
18
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
19  <<<<<<< HEAD (Current Change)
20          <h1> PM이 Function-a 작업한 코드를 develop에 반영 </h1>
21  =======
22          <h1> programmer1은 functhion-b를 개발  </h1>
23  >>>>>>> develop (Incoming Change)
24      </body>
25  </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

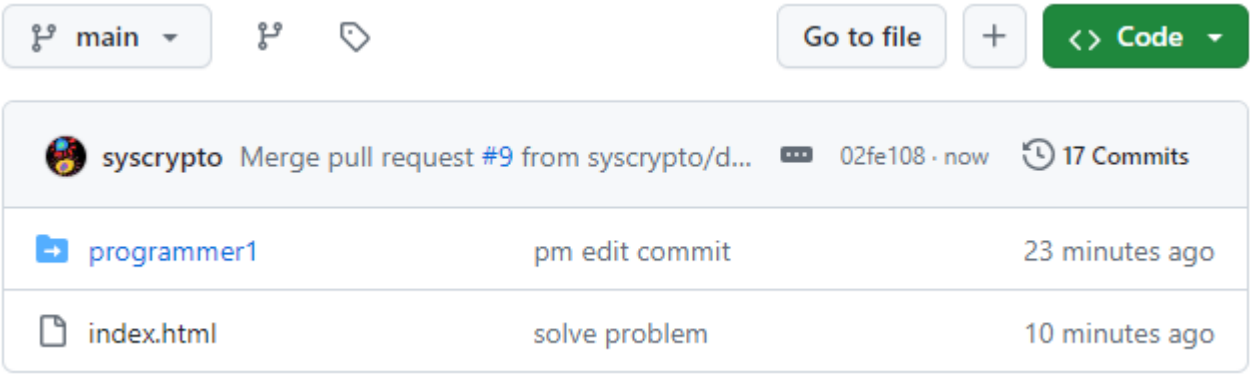
```
Updating 93a0b20..a826303
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (develop)
$ git checkout function-a
Switched to branch 'function-a'
Your branch is up to date with 'origin/function-a'.

swcu_sys@DESKTOP-3S6F5MB MINGW64 /c/git_Project (function-a)
$ git merge develop
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

최종 파일 올리기

- **Develop에서 main으로 최종코드 반영**



최종 배포 파일


main
Go to file
Code
syscrypto Merge pull request #9 from syscrypto/d...
02fe108 · now
17 Commits
programmer1
pm edit commit
23 minutes ago
index.html
solve problem
10 minutes ago

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
- https://inpa.tistory.com/entry/GIT-%E2%9A%A1%EF%B8%8F-%EA%B0%9C%EB%85%90-%EC%9B%90%EB%A6%AC-%EC%89%BD%EA%B2%8C%EC%9D%B4%ED%95%B4
- https://www.youtube.com/watch?v=tkkbYCajCjM

중간고사 대체 과제

- <Github 활용>
  - ‘Mid-Term-Project’ Repository를 생성해서 작업
  - 최소 15회이상 Commit
  - Github Page 또는 서버구축하여 webpage 운영
    - URL 제출

- <내용>
  - 자기 소개를 위한 개인 홈페이지 작성
    - Template 사용가능
  - 최소 이미지 2개 이상
  - 옵션사항
    - 동영상 또는 동영상 링크 추가