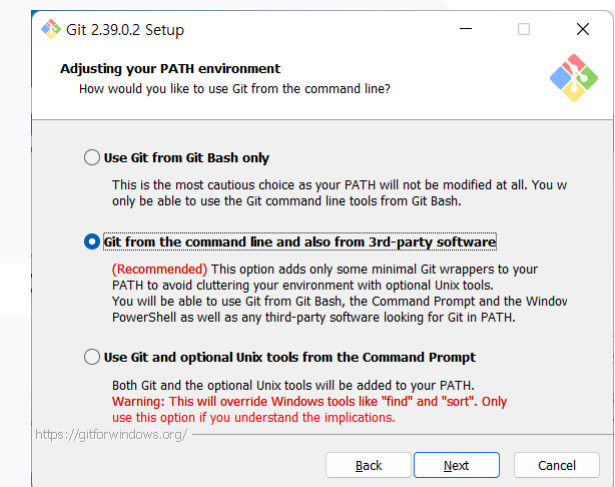
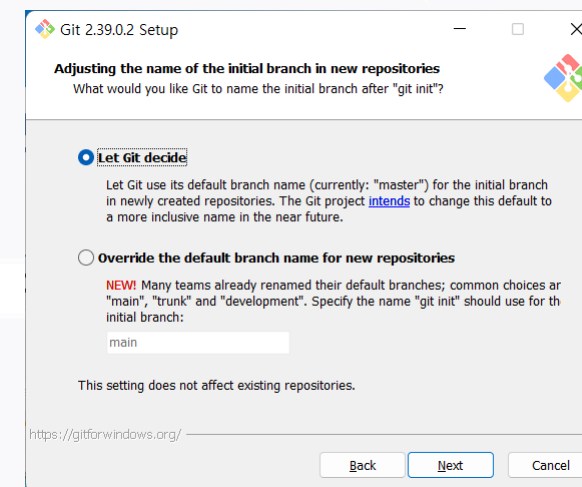
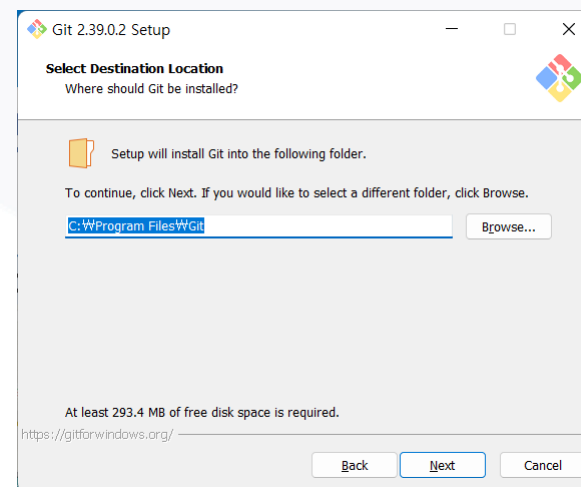


# AI 활용 빅데이터분석 풀스택웹서비스 SW 개발자 양성과정

## GitHub



- 소스코드 및 파일의 변경내역을 저장하는 분산 버전 관리 시스템
- Git 설치 (window)



# Git 초기화 확인

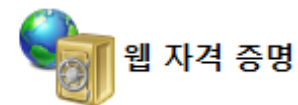
자격 증명 관리자

← → ▼ ↑

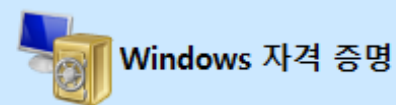
제어판 > 모든 제어판 항목 > 자격 증명 관리자

제어판 홈

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.



웹 자격 증명



Windows 자격 증명

자격 증명 백업(B) 자격 증명 복원(R)

Windows 자격 증명

Windows 자격 증명 추가

SWEC

수정한 날짜: 2023-02-23 ▼

인증서 기반 자격 증명

인증서 기반 자격 증명 추가

인증서가 없습니다.

일반 자격 증명

일반 자격 증명 추가

com.ridi.books/global

수정한 날짜: 2023-01-15 ▼

git:https://github.com

수정한 날짜: 오늘 ^

인터넷 또는 네트워크 주소: git:https://github.com

사용자 이름: cybermin

암호: .....

지속성: 로컬 컴퓨터

편집 제거

참고 항목

이유가 없습니다

자격 증명 관리자에서 widows 자격 증명에  
이전 git 로그인 정보가 있는지 확인

# Git 설정

git - version  
• 버전확인

git config - list  
• 설정확인

```
Git CMD
C:\Users\minnote>git --version
git version 2.39.0.windows.2

C:\Users\minnote>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=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\minnote\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait

C:\Users\minnote>
```

초기설정

```
Git CMD
C:\Users\minnote>git config --global user.name "pnumin"
C:\Users\minnote>git config --global user.email pnumin@pusan.ac.kr

C:\Users\minnote>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=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\minnote\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=pnumin
user.email=pnumin@pusan.ac.kr

C:\Users\minnote>
```



# Git을 이용하여 로컬 버전관리

로컬 저장소(내컴퓨터)



1. 워킹디렉토리 생성하고 이동

- mkdir 폴더명
- cd 폴더명

```
Git CMD
C:\wtest>dir /a
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 82EA-121F

C:\wtest 디렉터리

2023-01-15 오전 11:47 <DIR> .
2023-01-15 오전 11:44 <DIR> ..
2023-01-15 오전 11:47 <DIR> .git
2023-01-15 오전 11:46 244 index.html
                  1개 파일          244 바이트
                  3개 디렉터리 794,852,888,576 바이트 남음

C:\wtest>
```

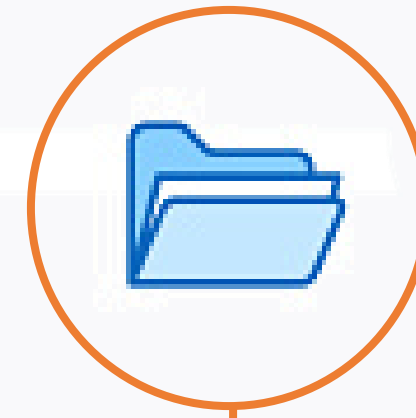
Working디렉토리



2. 워킹디렉토리 초기화

- git init

Staging 영역



3. Git 관리 파일등록

- git add 파일명
- git add .

4. Git 버전 만들기

- git commit -m "버전메시지"

local 레포지토리



Git 현재 시점 변경

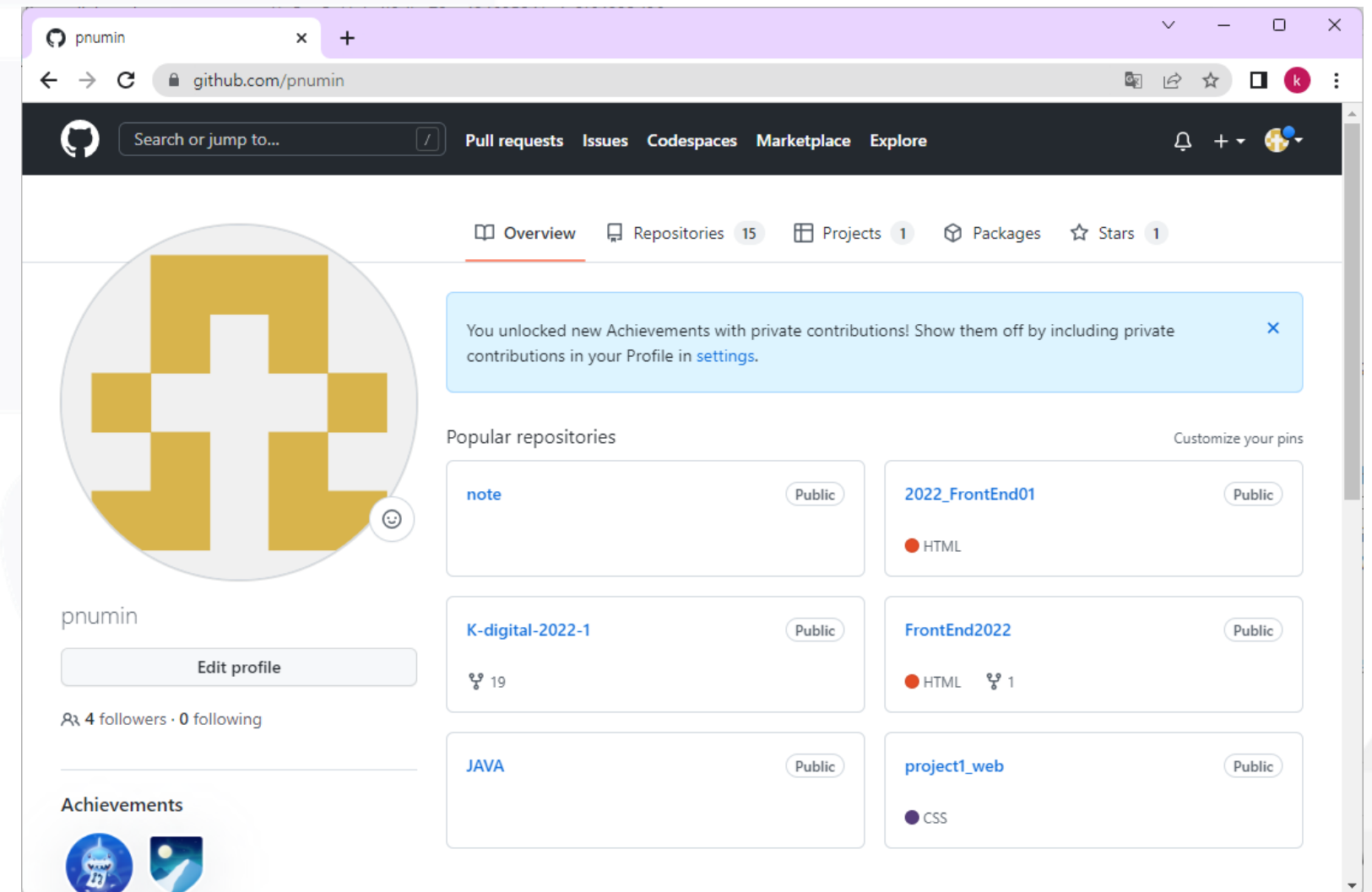
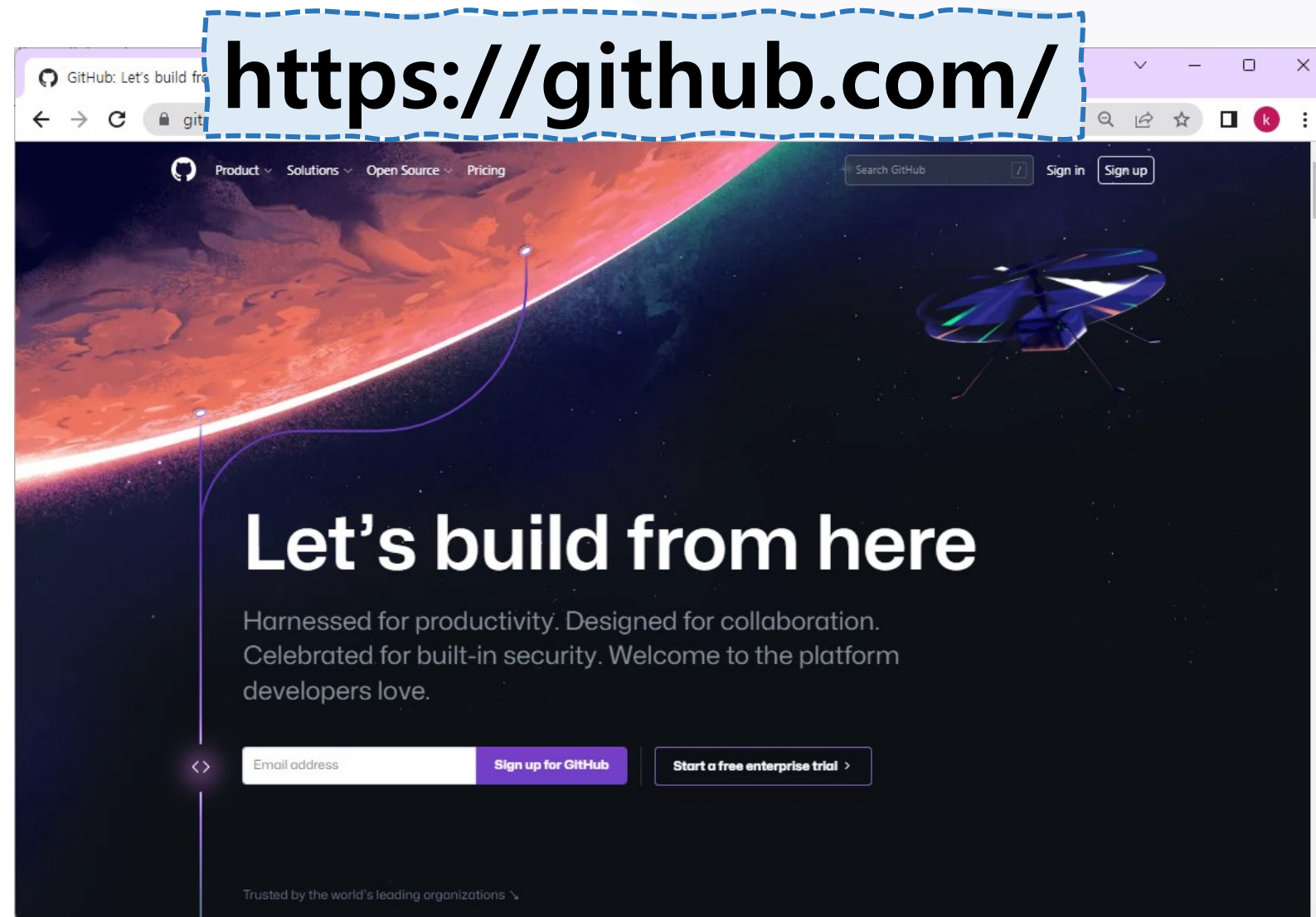
- git checkout (커밋아이디) // 커밋아이디는 git log확인
- git checkout master //최근 커밋 상태로 돌아오기

Git 이전 commit으로 돌아가고 이후 commit 삭제

- git reset --hard (커밋아이디)

# GitHub 원격 버전관리

- 분산 버전 관리 툴인 Git을 관리해주는 웹 호스팅 서비스가 바로 GitHub
- Git을 클라우드 환경에서 사용할 수 있게 제공하는 공간이 GitHub



# GitHub 저장소(repository) 만들기

The image displays three overlapping screenshots of the GitHub website, illustrating the steps to create a new repository.

**Left Screenshot (github.com/pnumin):** Shows the user's profile page for 'pnumin'. It includes a profile picture, a bio, and a list of popular repositories. The 'note' repository is highlighted.

**Middle Screenshot (github.com/new):** Shows the 'Create a new repository' page. The 'Owner' is 'pnumin' and the 'Repository name' is 'K-digital-2023-2'. The 'Description' field is empty. The 'Public' option is selected under 'Initialize this repository with:'. The 'Add a README file' checkbox is checked. The 'Add .gitignore' section shows a dropdown menu with 'None' selected. The 'Choose a license' section shows a dropdown menu with 'None' selected. A green 'Create repository' button is at the bottom.

**Right Screenshot (github.com/pnumin/K-digital-2023-2):** Shows the newly created repository page for 'pnumin/K-digital-2023-2'. It includes a 'Quick setup' section with a 'Set up in Desktop' button and a 'HTTPS' button. Below this, there are sections for '...or create a new repository on the command line' and '...or push an existing repository from the command line', each with a code block and a copy icon. The '...or import code from another repository' section is also visible.

# 로컬저장소와 원격저장소 연결

## • 로컬저장소에 자료가 있는 경우

The image illustrates the process of connecting a local Git repository to a remote GitHub repository. It consists of three main components:

- GitHub Repository Page (Left):** Shows the repository `pnumin/K-digital-2023-2` with a public status. It includes a "Quick setup" section with instructions for creating a new repository on the command line. A red dashed box highlights the commands for pushing an existing repository from the command line.
- Git CMD Terminal (Center):** Shows the execution of the following commands:

```
C:\Wgtest>git remote add origin https://github.com/pnumin/K-digital-2023-2.git
C:\Wgtest>git branch -M main
C:\Wgtest>git push -u origin main
```

The output shows the successful push of the main branch to the remote repository, including object enumeration, compression, and writing details.
- GitHub Repository Page (Right):** Shows the repository `pnumin/K-digital-2023-2` with a public status. It includes a "Code" button and a "README" section. A red dashed box highlights the "Code" button, which is linked to the "Add a README" button in the "Releases" section.

Arrows indicate the flow of information: from the "Code" button on the left GitHub page to the Git CMD terminal, and from the "Add a README" button on the right GitHub page to the "Code" button on the left GitHub page.



# 로컬저장소와 원격저장소 연결

## • 원격저장소에 자료가 있는 경우

```
Git CMD
C:\Wgtest>
C:\Wgtest>mkdir p1

C:\Wgtest>cd p1

C:\Wgtest\p1>git clone https://github.com/pnumin/project1_web
.git
Cloning into 'project1_web'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 47% (11/23)
remote: Compressing objects: 52% (12/23)Receiving objects:
remote: Compressing objects: 100% (23/23), done.
remote: Total 25 (delta 2), reused 15 (delta 0), pack-reused
Receiving objects: 68% (17/25)
Receiving objects: 100% (25/25), 47.62 KiB | 4.33 MiB/s, don
e.
Resolving deltas: 100% (2/2), done.
C:\Wgtest\p1>_
```

EXPLORER

- ✓ GTEST
  - ✓ p1\project1\_web
    - > images
    - > styles
    - <> index.html
    - ① README.md
    - ① README.md

<> index.html X

```
p1 > project1_web > <> index.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>K디지털 웹</title>
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <link rel="stylesheet" href="/styles/index.css">
7 <link rel="stylesheet" href="/styles/nav.css">
8 <link rel="stylesheet" href="/styles/header.css">
9 <!--웹아이콘 가져오기-->
10 <script src="https://kit.fontawesome.com/74df50e688.js" crossorigin="anonymou
11 <!--웹폰트-->
12 <link rel="preconnect" href="https://fonts.googleapis.com">
13 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
14 <link href="https://fonts.googleapis.com/css2?family=Dongle&family=Nanum+Goth
15 </head>
16 <body>
17 <nav>
18 <div class="nav_logo">
19 <p>mylogo</p>
20 </div>
21 <ul class="nav_menu">
22 <li><a href="#">HTML</a></li>
23 <li><a href="#">CSS</a></li>
24 <li><a href="#">JavaScript</a></li>
25 <li><a href="#">React</a></li>
26 </ul>
27 <ul class="nav_icon">
28 <li><i class="fa-brands fa-facebook-f"></i></li>
29 <li><i class="fa-brands fa-square-instagram"></i></li>
30 </ul>
31 </nav>
32 </body>
```

# 로컬저장소와 원격저장소

로컬 저장소(내컴퓨터)



원격저장소 자료 가져오기

- git pull origin main

원격저장소에 자료 올리기

- git push origin main

```
Git CMD
C:\gtest>git add .
C:\gtest>git commit -m "local 자료 수정1"
[main c0e5a2a] local 자료 수정1
1 file changed, 1 insertion(+), 1 deletion(-)
C:\gtest>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 355 bytes | 355.00 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/pnmin/K-digital-2023-2.git
4558dfb..c0e5a2a main -> main
C:\gtest>
```

원격 저장소



# GitHub로 웹 호스팅

The screenshot shows the GitHub Pages settings page for the repository 'pnumin/testp2'. The 'Settings' tab is highlighted with a red dashed box. The 'Build and deployment' section is expanded, showing the 'Source' as 'Deploy from a branch'. The 'Branch' section is also expanded, showing 'main' as the selected branch and '/ (root)' as the directory. A red dashed box highlights the 'main' branch and the '/ (root)' directory. The 'Custom domain' section is visible below, with a text input field and 'Save' and 'Remove' buttons.

Pages

github.com/pnumin/testp2/settings/pages

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

pnumin / testp2 Public

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Codespaces

**Pages**

Security

Code security and analysis

Deploy keys

Secrets and variables

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

### Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more.](#)

main / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

### Custom domain

Custom domain

Custom domains allow you to serve your site from a domain other than pnumin.github.io. [Learn more.](#)

Save Remove

DNS records should point to the [internationalized domain name](#).

The screenshot shows the GitHub Actions workflow run for 'pages build and deployment'. The 'Actions' tab is highlighted with a red dashed box. The workflow is titled 'pages build and deployment #1'. The 'Summary' section shows the workflow was triggered via dynamic 1 minute ago, with a status of 'Success', a total duration of 52s, and 1 artifact. The 'Run details' section shows the workflow steps: 'build' (22s), 'report-build-status' (2s), and 'deploy' (7s). The 'deploy' step is highlighted with a red dashed box, showing the URL 'https://pnumin.github.io/testp2/'.

pages build and deployment · p x

github.com/pnumin/testp2/actions/runs/3923232550

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

pnumin / testp2 Public

Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

### pages build and deployment #1

Re-run all jobs

#### Summary

Triggered via dynamic 1 minute ago

Status Success

Total duration 52s

Artifacts 1

Jobs

- build
- report-build-status
- deploy

#### Run details

Usage

pages-build-deployment on: dynamic

build 22s

report-build-status 2s

deploy 7s

<https://pnumin.github.io/testp2/>

Artifacts

# 원본 원격저장소 복사

복사할 원격 저장소

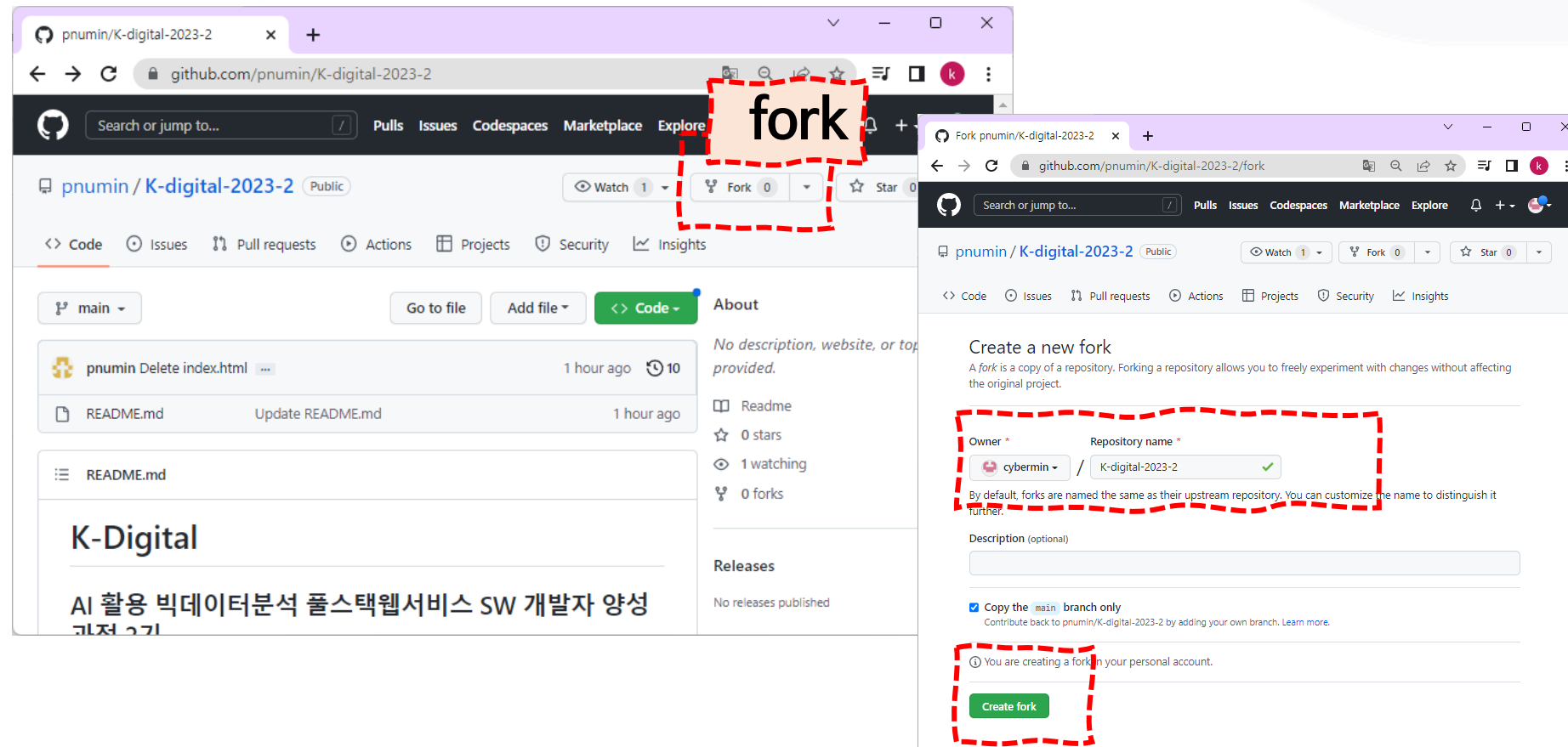


1. 본인의 원격저장소에서  
복사할 원격저장소로 이동

본인의 원격 저장소



2. fork





# 원본 원격저장소 복사

복사할 원격 저장소



4. 수정내용 반영

본인의 원격 저장소



3. 본인의 원격저장소에서  
수정후 pull requests보내기

