



인하공업전문대학
INHA TECHNICAL COLLEGE

TCP/IP

네트워크 프로그래밍

5주차 (프로그래밍실습)

인하공업전문대학 컴퓨터 정보과
김한결 강사



목차

- TCP/IP 개발환경 구축 - cygwin
- 리눅스 사용
- Vim 편집기
- Github 사용하기
- 리눅스 기반 GitHub 사용방법

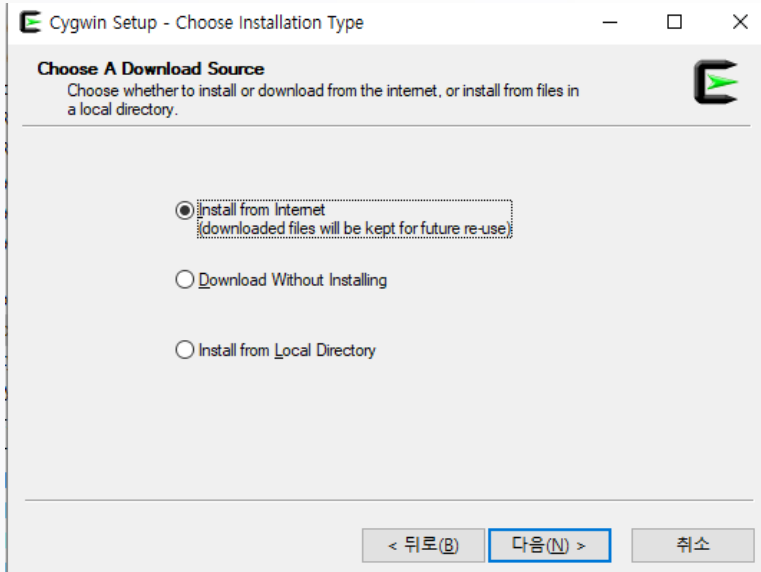
- 구글에서 Cygwin 검색 후 사이트 이동

<https://cygwin.com/install.html>

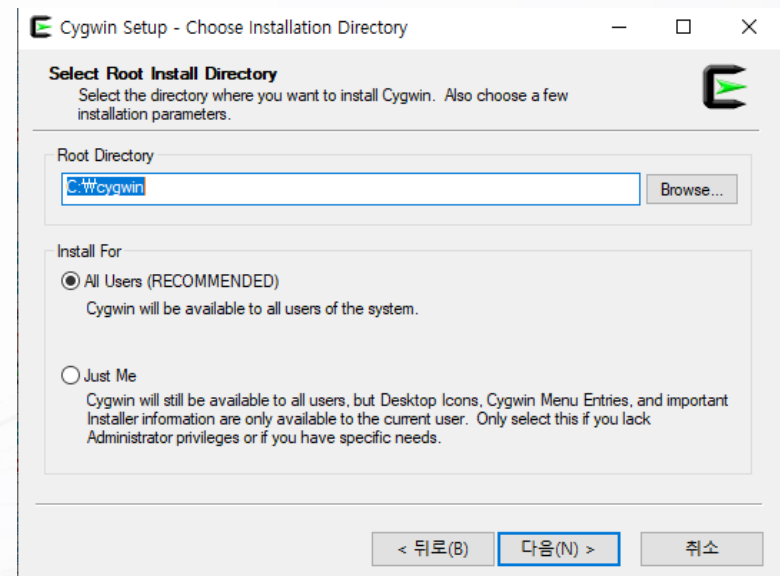


The screenshot shows the Cygwin website's installation page. The browser window has a single tab titled 'Cygwin Installation' and the address bar shows 'cygwin.com/install.html'. On the left is a dark green sidebar with a list of links: 'Cygwin' (with sub-links 'Install Cygwin', 'Update Cygwin', 'Search Packages', 'Licensing Terms'), 'Cygwin/X', 'Community' (with sub-links 'Reporting Problems', 'Mailing Lists', 'Newsgroups', 'IRC channels', 'Gold Stars', 'Mirror Sites', 'Donations'), 'Documentation' (with sub-links 'FAQ', 'User's Guide', 'API Reference', 'Acronyms'), and 'Contributing' (with sub-links 'Snapshots', 'Source in Git'). The main content area has a large 'Cygwin' logo with the tagline 'Get that [Linux](#) feeling - on Windows'. Below the logo is the heading 'Installing and Updating Cygwin Packages'. There are two sub-sections: 'Installing and Updating Cygwin for 64-bit versions of Windows' and 'Installing and Updating Cygwin for 32-bit versions of Windows'. The 32-bit section is highlighted with a red rectangular border. It contains the text: 'Run [setup-x86.exe](#) any time you want to update or install a Cygwin package for 32-bit windows. The [signature](#) for [setup-x86.exe](#) can be used to verify the validity of this binary.' Below this is a section titled 'Signing key transition' with the text: 'The key used to sign setup binaries has been updated. During the transition period, signatures'.

Cygwin install

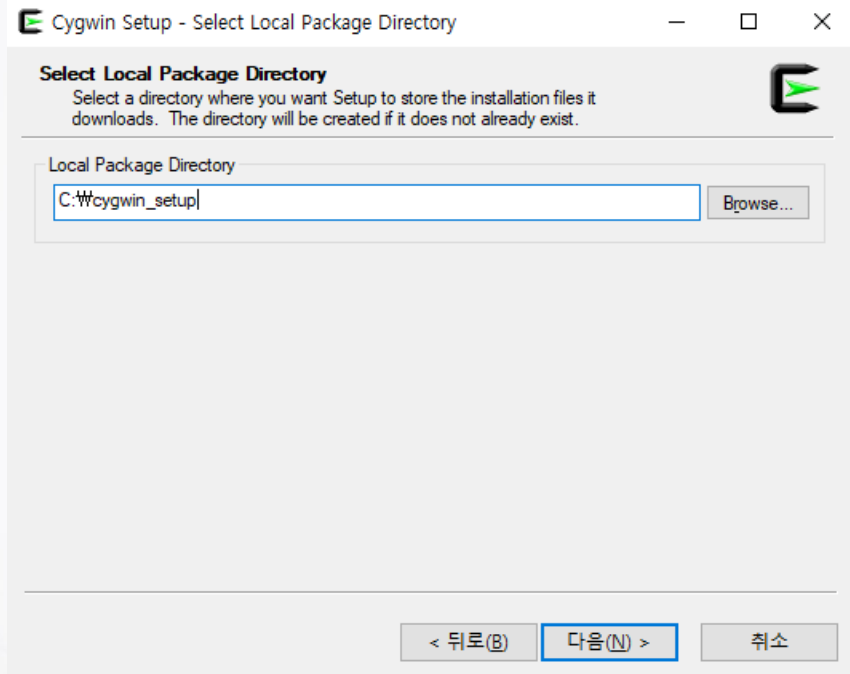


1. Install from internet

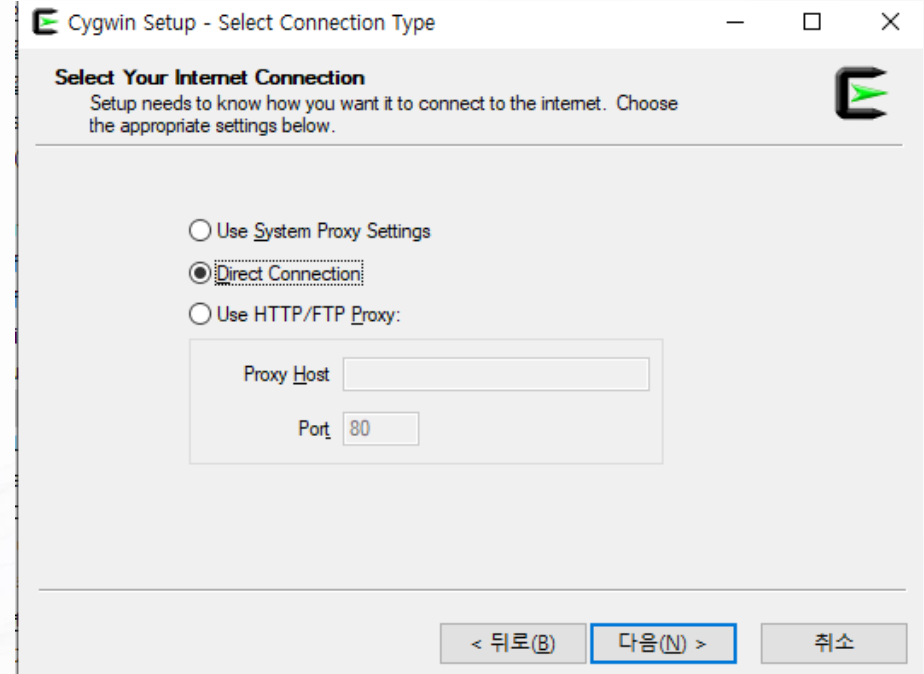


2. C:\cygwin

Cygwin install (Const.)

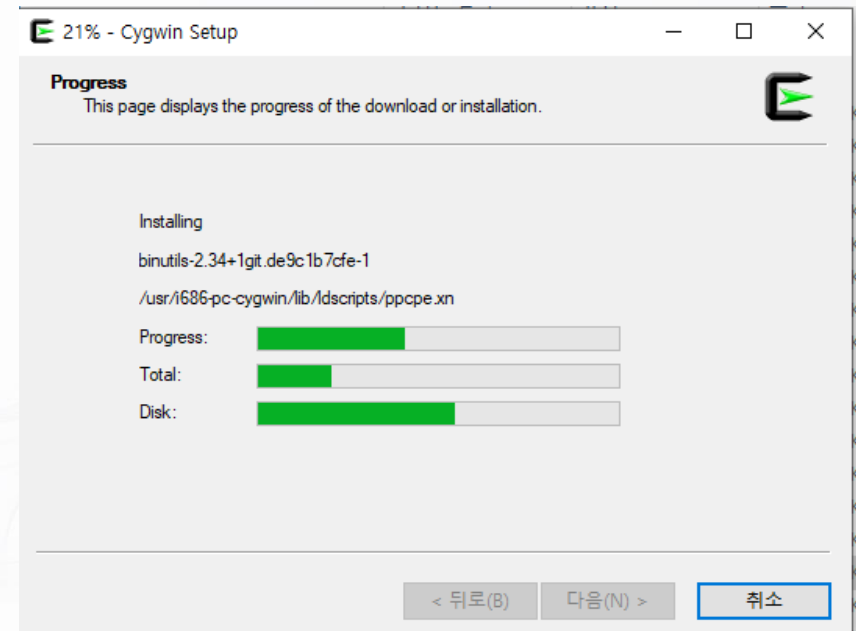
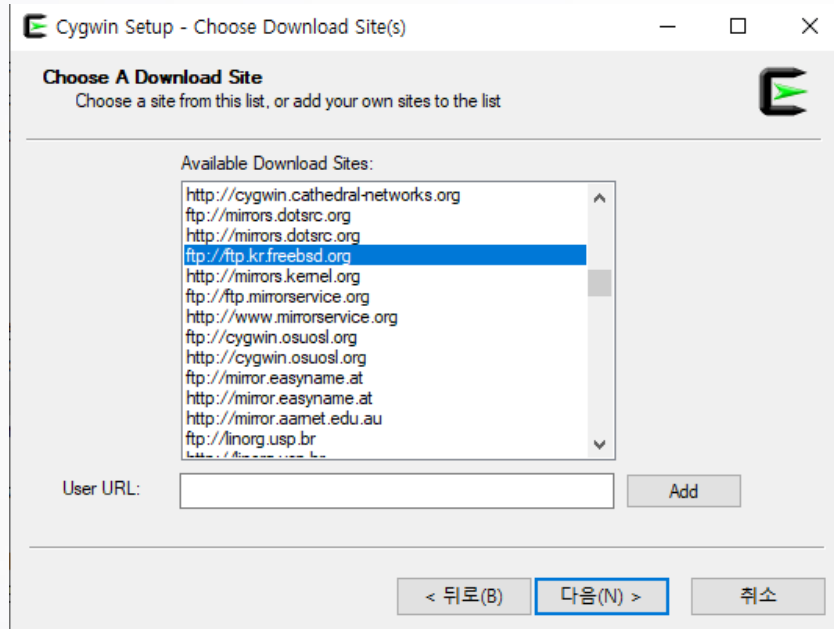


3. Setup package directory



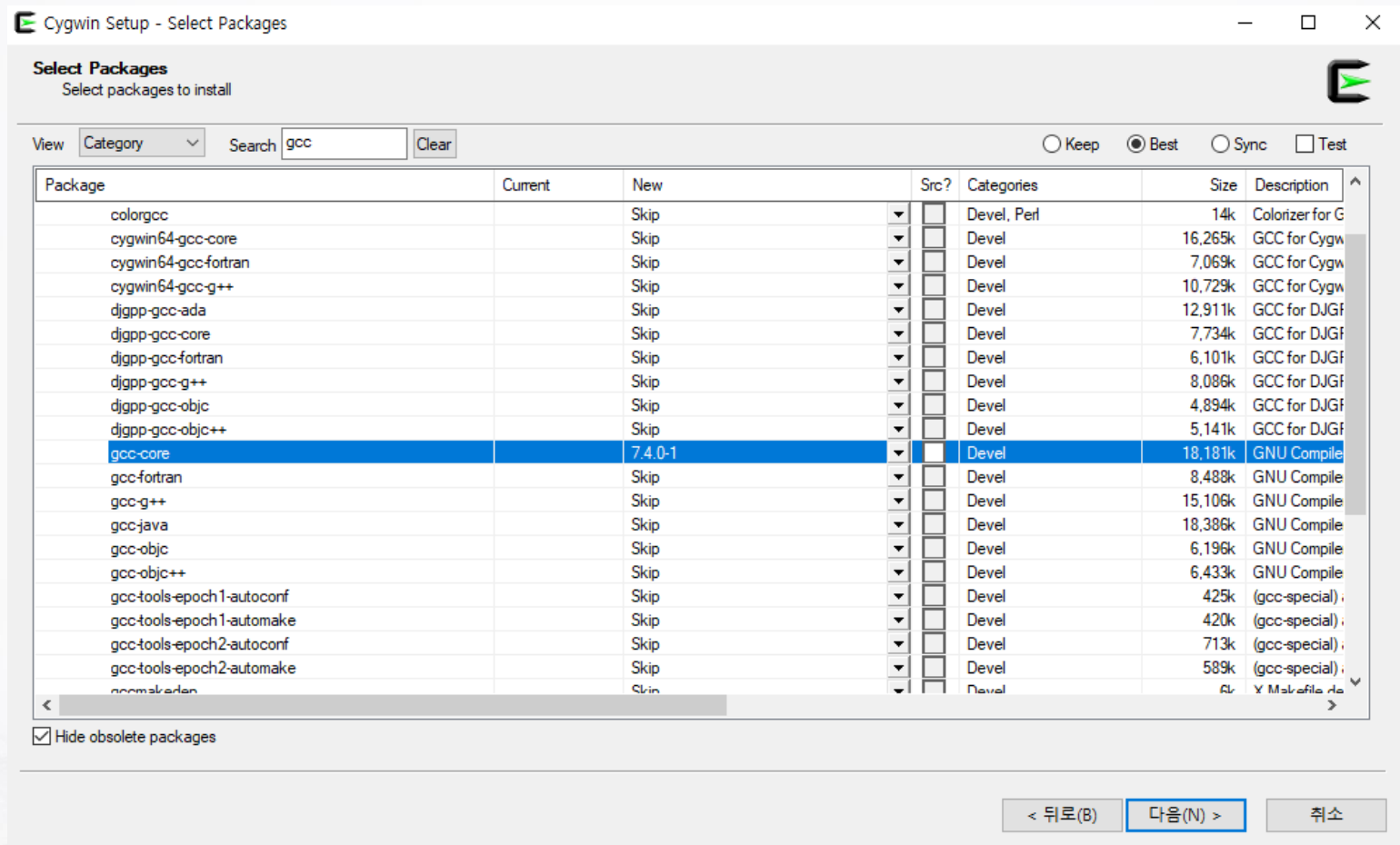
4. Direct Connection

Cygwin install (Const.)



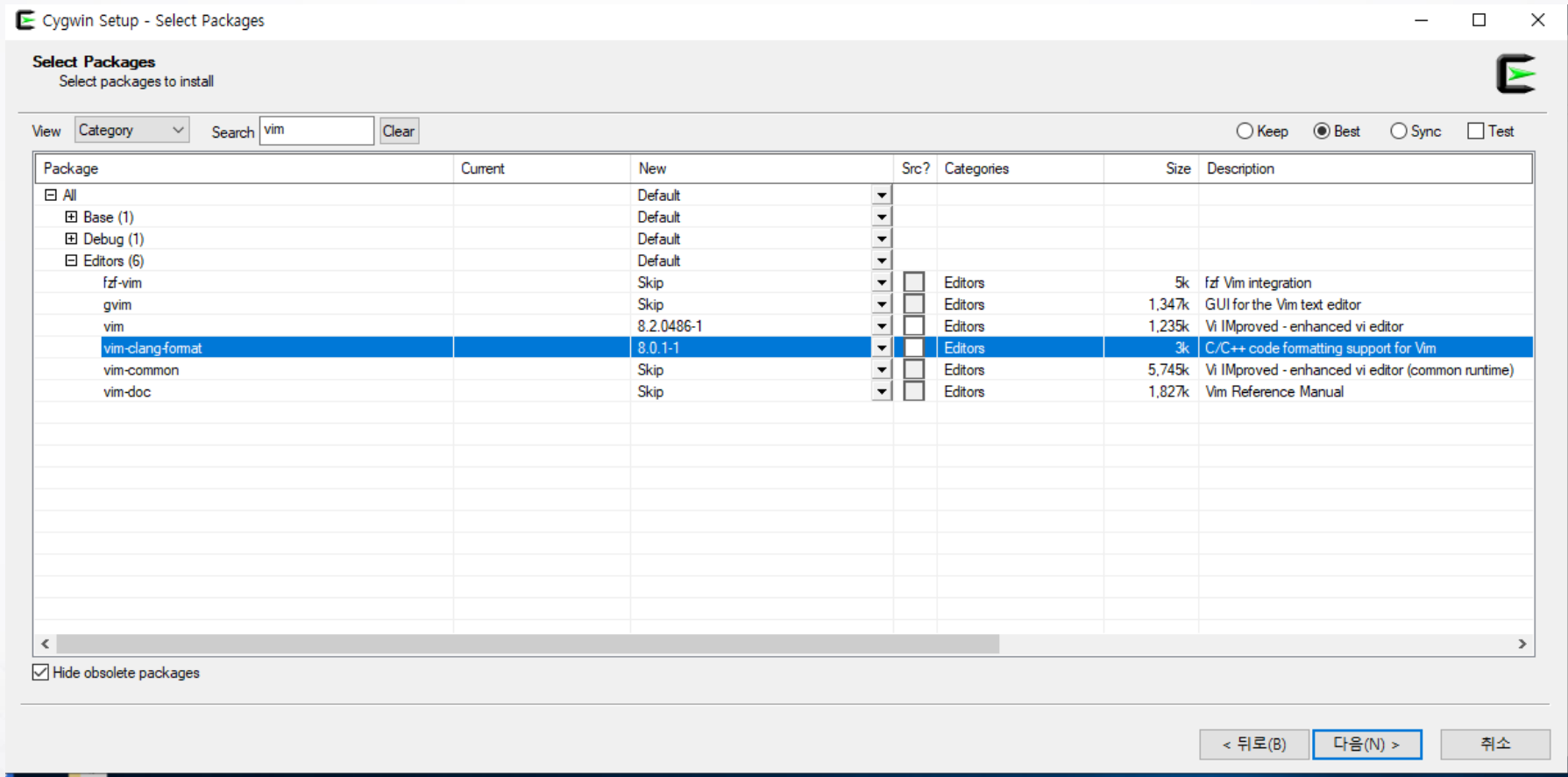
5. <ftp://ftp.kr.freebsd.org>

Cywin install (Const.)



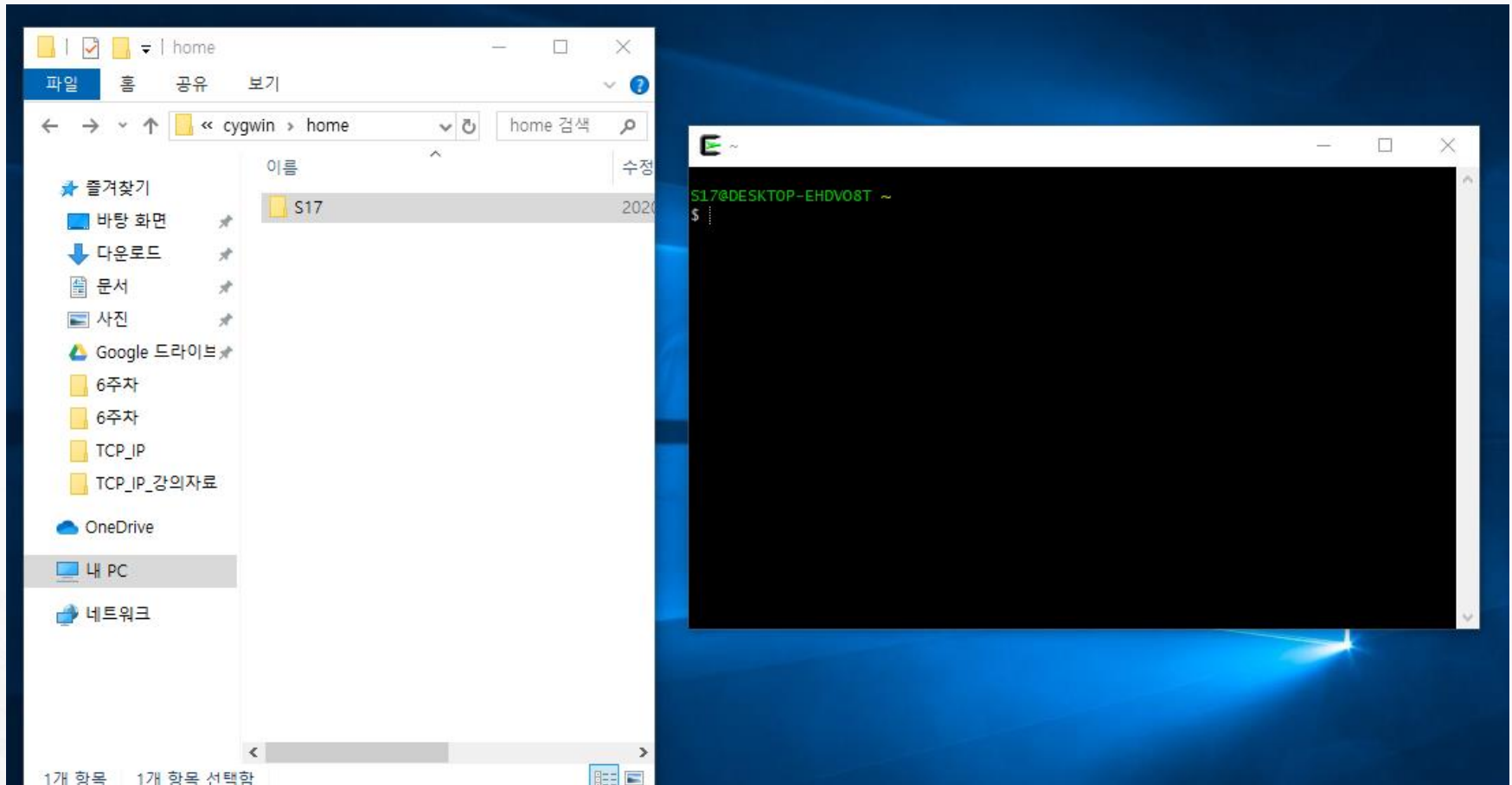
6. Category -> gcc -> gcc-core -> 7.4.0-1

Cygwin install (Const.)

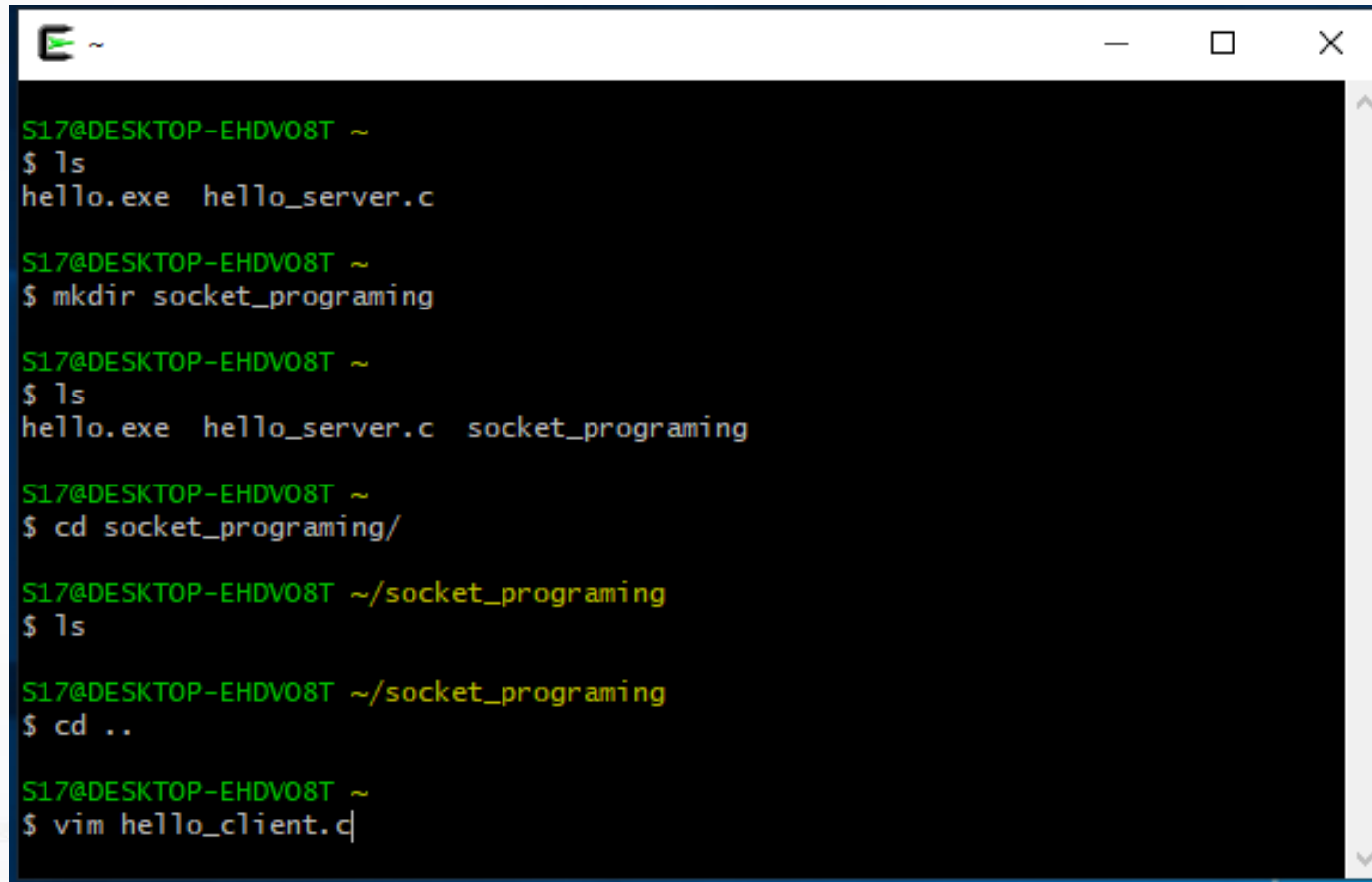


7. Category -> vim -> vim , vim-clang-format

Cygwin view



Cygwin view

A screenshot of a Cygwin terminal window. The window has a title bar with a green icon and a tilde (~) on the left, and standard Windows window controls (minimize, maximize, close) on the right. The terminal background is black with green text. The prompt is 'S17@DESKTOP-EHDV08T ~'. The user enters '\$ ls' and the output is 'hello.exe hello_server.c'. Then the user enters '\$ mkdir socket_programing'. The prompt changes to 'S17@DESKTOP-EHDV08T ~/socket_programing'. The user enters '\$ ls' and the output is 'hello.exe hello_server.c socket_programing'. Then the user enters '\$ cd socket_programing/'. The prompt changes to 'S17@DESKTOP-EHDV08T ~/socket_programing'. The user enters '\$ ls' and the output is 'hello.exe hello_server.c socket_programing'. Then the user enters '\$ cd ..'. The prompt changes to 'S17@DESKTOP-EHDV08T ~'. Finally, the user enters '\$ vim hello_client.c|'.

```
S17@DESKTOP-EHDV08T ~  
$ ls  
hello.exe hello_server.c  
  
S17@DESKTOP-EHDV08T ~  
$ mkdir socket_programing  
  
S17@DESKTOP-EHDV08T ~  
$ ls  
hello.exe hello_server.c socket_programing  
  
S17@DESKTOP-EHDV08T ~  
$ cd socket_programing/  
  
S17@DESKTOP-EHDV08T ~/socket_programing  
$ ls  
hello.exe hello_server.c socket_programing  
  
S17@DESKTOP-EHDV08T ~/socket_programing  
$ cd ..  
  
S17@DESKTOP-EHDV08T ~  
$ vim hello_client.c|
```

gcc c파일이름 -o 실행파일이름

Ex) gcc hello_server.c -o hserver

hello_server 실행 ./hserver 9190(포트번호)

hello_client 실행 ./client 127.0.0.1 9190

리눅스 사용방법 기초

기본 디렉토리 명령어

ls

현재 경로 파일 리스트

cd

디렉토리 변경

mkdir

디렉토리 생성

pwd

현재 디렉토리

운영 관련 명령어

ps -aux

현재 실행중인 프로세스

netstat -tnl

현재 사용중인 TCP 포트

ufw

방화벽 설정

history

명령어 history

주요 명령어

apt-get

소프트웨어 설치/제거

sudo

슈퍼유저권한 대행

su

User 변경

vim

편집기

기본 디렉토리 명령어

ls

현재 경로 파일 리스트

ls -al

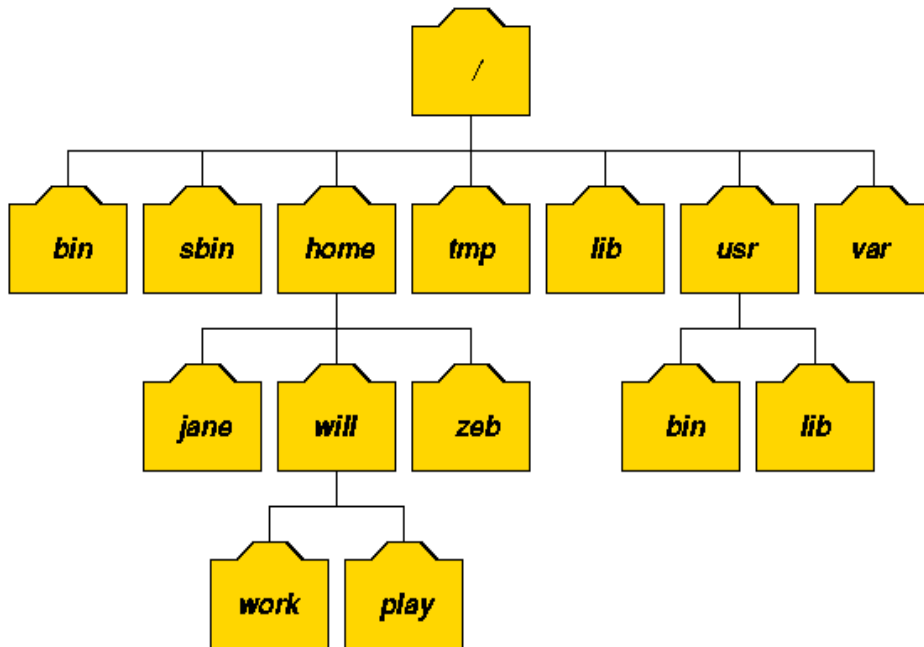
현재 경로 파일 리스트
및
권한, 크기, 날짜

pi@raspberrypi: ~

```
pi@raspberrypi:~ $ ls
Bookshelf Documents Music Public Videos
Desktop Downloads Pictures Templates work
pi@raspberrypi:~ $ ls -al
total 104
drwxr-xr-x 16 pi pi 4096 Jul 9 23:32 .
drwxr-xr-x 3 root root 4096 May 27 16:10 ..
-rw----- 1 pi pi 21 Jul 9 22:55 .bash_history
-rw-r--r-- 1 pi pi 220 May 27 16:10 .bash_logout
-rw-r--r-- 1 pi pi 3523 May 27 16:10 .bashrc
drwxr-xr-x 2 pi pi 4096 May 27 16:18 Bookshelf
drwxr-xr-x 5 pi pi 4096 May 27 16:46 .cache
drwx----- 3 pi pi 4096 May 27 16:46 .config
drwxr-xr-x 2 pi pi 4096 May 27 16:46 Desktop
drwxr-xr-x 2 pi pi 4096 May 27 16:46 Documents
drwxr-xr-x 2 pi pi 4096 May 27 16:46 Downloads
drwx----- 3 pi pi 4096 May 27 16:46 .gnupg
drwxr-xr-x 3 pi pi 4096 May 27 16:18 .local
drwxr-xr-x 2 pi pi 4096 May 27 16:46 Music
drwxr-xr-x 2 pi pi 4096 May 27 16:46 Pictures
-rw-r--r-- 1 pi pi 807 May 27 16:10 .profile
```

리눅스 사용방법 기초

리눅스 디렉토리 구조



/	/
bin	tmp
실행 파일 모음	임시파일
home	root
user 폴더	root home 디렉토리
sbin	var
시스템 관리용 실행 파일	로그, 시스템 운영중 갱신데이터 저장
usr	lib
각종 프로그램 설치되는 폴더	각종 라이브러리 설치되는 폴더

리눅스 사용방법 기초

기본 디렉토리 명령어

cd

디렉토리 변경

cd ..

한단계 상위수준
디렉토리로 변경

cd /home/pi/Document

절대경로 기준으로
디렉토리로 변경

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ ls  
Bookshelf Documents Music Public Videos  
Desktop Downloads Pictures Templates work  
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $ cd Documents/  
pi@raspberrypi:~/Documents $ pwd  
/home/pi/Documents  
pi@raspberrypi:~/Documents $ cd ..  
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $ cd /home/pi/Documents/  
pi@raspberrypi:~/Documents $ pwd  
/home/pi/Documents  
pi@raspberrypi:~/Documents $ cd ..  
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $
```


Vim 편집기

Vim Editor Setting

파일명: .vimrc

경로: /home/pi/.vimrc

pi@raspberrypi: ~

```
1 set nu
2 set cindent
3 set bg=dark
4 set autoindent
5 set expandtab
6 set softtabstop=4
7 set tabstop=4
8 let python_version_2 = 1
9 let python_highlight_all = 1
10 filetype indent plugin on
11 syntax on
```

syntax on

filetype indent plugin on

let python_version_2 = 1

let python_highlight_all = 1

set tabstop=4

set expandtab

set softtabstop=4

set autoindent

set bg=dark

set nu

-----> 문법 기능 ON

-----> 확장자로 문서 형식 파악

-----> python 2 문법을 따름(플러그 인)

-----> 모든 강조(색상) ON(플러그 인)

-----> 탭의 길이 설정

-----> 탭을 누를 때 실제로 벌어지는 간격

-----> 자동 들여쓰기

-----> 배경을 어둡게(필수!) 가독성이 좋아짐

-----> 라인 숫자 ON

GitHub - 분산 버전 관리 툴

Github

<https://github.com/>

- 깃허브는 버전관리와 협업을 위한 코드 호스팅 플랫폼
- 소프트웨어 개발 프로젝트를 위한 소스 코드 관리 서비스

The screenshot shows the GitHub repository page for 'adafruit / DHT-sensor-library'. The repository is an Arduino library for DHT11, DHT22, etc. Temperature & Humidity Sensors. It has 125 commits, 3 branches, 0 packages, 18 releases, and 27 contributors. The latest commit is 13d25e6, 9 days ago. The repository includes files such as .github, examples, .gitignore, CONTRIBUTING.md, DHT.cpp, DHT.h, DHT_U.cpp, DHT_U.h, README.md, and code-of-conduct.md.

adafruit / DHT-sensor-library

Watch 152 Star 1.3k Fork 1.1k

<> Code Issues 7 Pull requests 12 Actions Projects 0 Security 0 Insights

Arduino library for DHT11, DHT22, etc Temperature & Humidity Sensors <https://learn.adafruit.com/dht>

125 commits 3 branches 0 packages 18 releases 27 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

dherrada Bump to 1.3.10 Latest commit 13d25e6 9 days ago

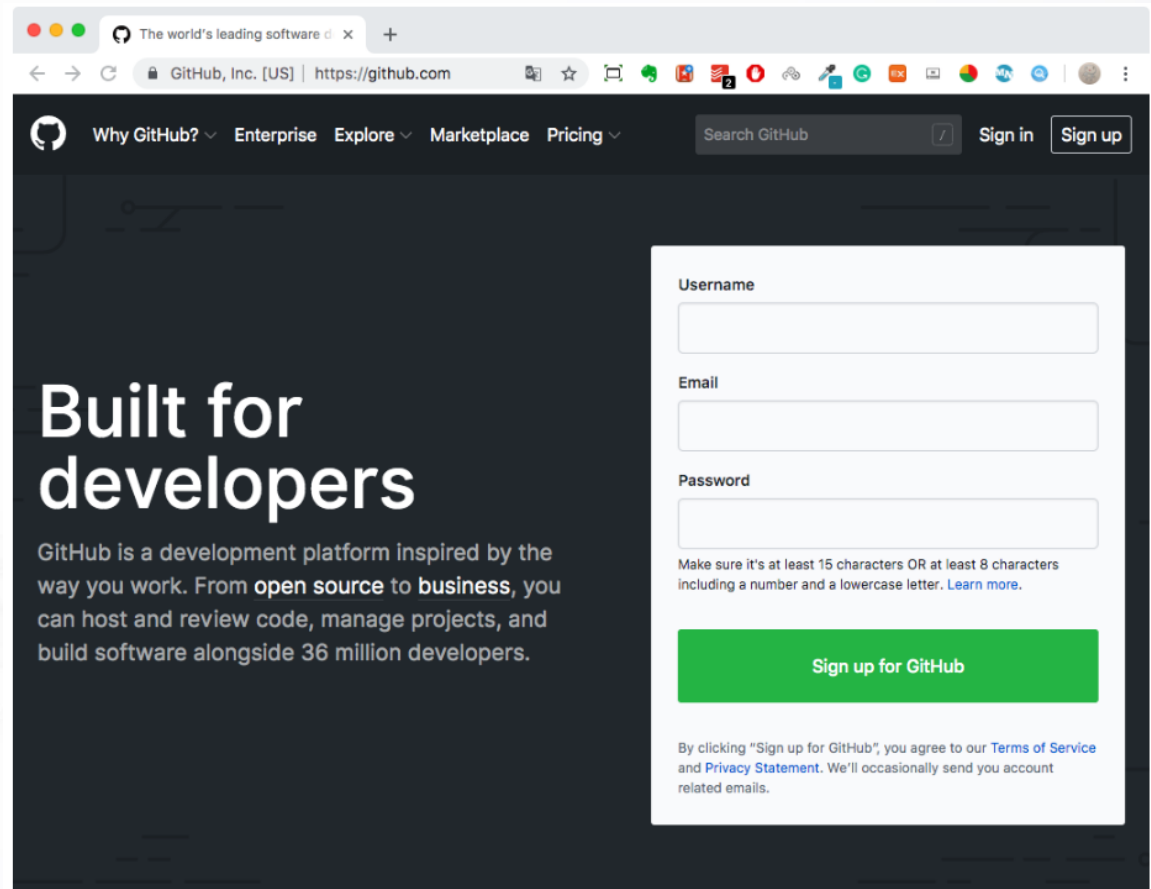
.github	actionified, formatted and doxy'd	2 months ago
examples	Change initial HIGH delay, fix comments in examples	16 months ago
.gitignore	Add .gitignore	11 months ago
CONTRIBUTING.md	[Update URL]	7 months ago
DHT.cpp	Merge pull request #159 from Rotzbua/patch-1	14 days ago
DHT.h	actionified, formatted and doxy'd	2 months ago
DHT_U.cpp	actionified, formatted and doxy'd	2 months ago
DHT_U.h	actionified, formatted and doxy'd	2 months ago
README.md	actionified, formatted and doxy'd	2 months ago
code-of-conduct.md	actionified, formatted and doxy'd	2 months ago

GitHub - 분산 버전 관리 툴

Github

<https://github.com/>

- 회원 가입(Sign in)



The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays "GitHub, Inc. [US] | https://github.com". The page features a dark theme with a navigation bar at the top containing links like "Why GitHub?", "Enterprise", "Explore", "Marketplace", and "Pricing". A search bar and "Sign in" / "Sign up" buttons are also present. The main content area has the heading "Built for developers" and a description of GitHub as a development platform. On the right side, there is a white sign-up form with fields for "Username", "Email", and "Password". Below the password field, there is a note about password requirements and a link to "Learn more". A green "Sign up for GitHub" button is at the bottom of the form. At the very bottom of the form, there is a disclaimer about agreeing to the Terms of Service and Privacy Statement.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

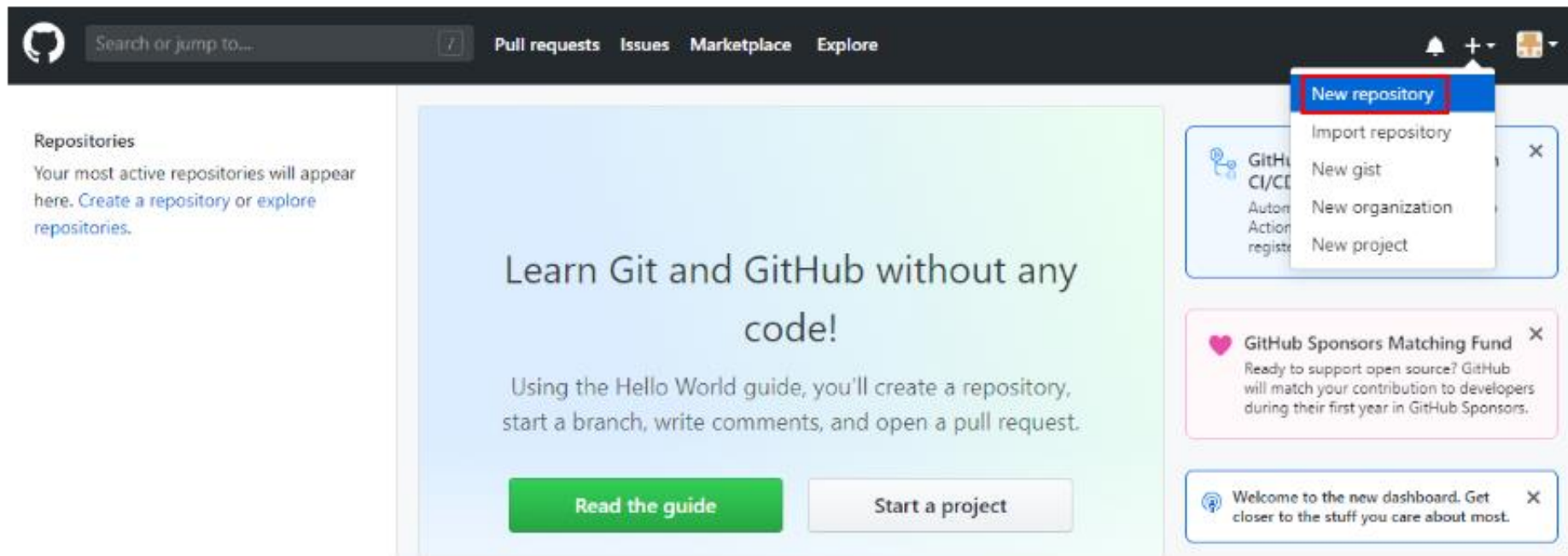
[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

GitHub - 분산 버전 관리 툴

Step 1. 저장소(Repository) 생성하기

- 우측 상단의 '+' 버튼을 클릭 -> New repository




Step 1. 저장소(Repository) 생성하기

- Repository Name : MyFirstRepository
- Description : Tutorial Repository
- Check Public
- Check Initialize this repository with a README
- Create Repository 클릭


Create a new repository


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

Owner *  sonnonet / Repository name * MyFirstRepository ✓

Great repository names are short, lowercase, and unique. **MyFirstRepository** is available. Get inspiration? How about **supreme-octo-barnacle**?

Description (optional)
Tutorial Repository

☒  **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.

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

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ Add a license: None ▾ ⓘ

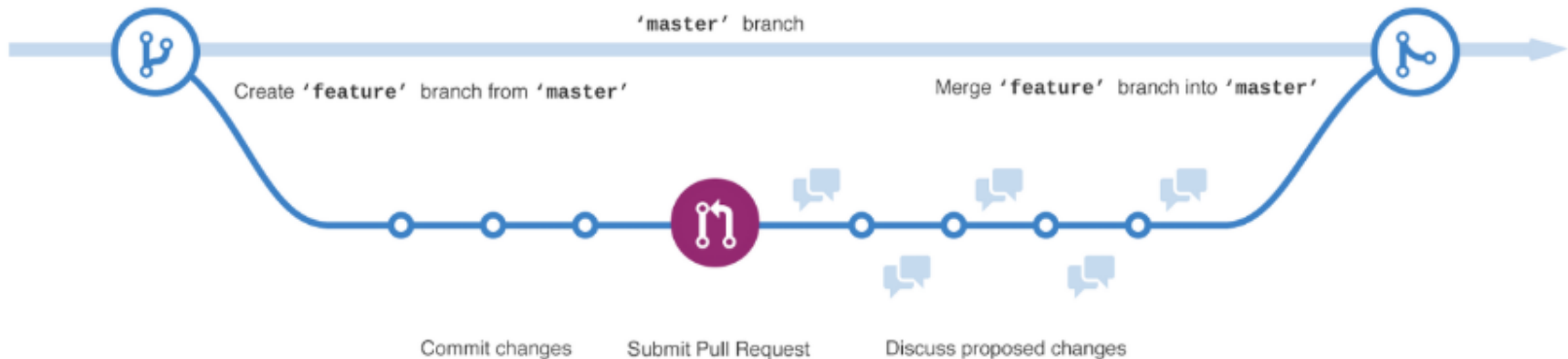
Create repository

Step 2. 브랜치(Branch) 생성하기

- 브랜칭(Branching)은 하나의 저장소에서 서로 다른 버전을 동시에 작업하는 방법
- 기본적으로 저장소는 "Master"라는 이름의 메인 브랜치를 가짐.
- "Master" 외의 다른 브랜치를 만들어서 "Master"에 커밋하기 전에 수정, 테스트
 - 커밋 (commit) : 파일을 추가하거나 변경 내용을 저장소에 저장하는 작업
- Master 외의 브랜치가 만들어지면 그순간에 Master의 Copy 혹은 스냅샷이 생성
- 새로운 브랜치에서 작업하는 동안 다른 사용자가 Master 브랜치를 수정한다면, 그 업데이트된 저장소를 풀로 받을수 있음.
 - 풀(pull) 다른 사용자의 업데이트 혹은 커밋(commit) 내용을 클라이언트로 내려받는 명령어.

Step 2. 브랜치(Branch) 생성하기

- master 브랜치
- feature라는 새로운 브랜치
- feature 브랜치가 master로 병합(merge)되기 전까지의 과정

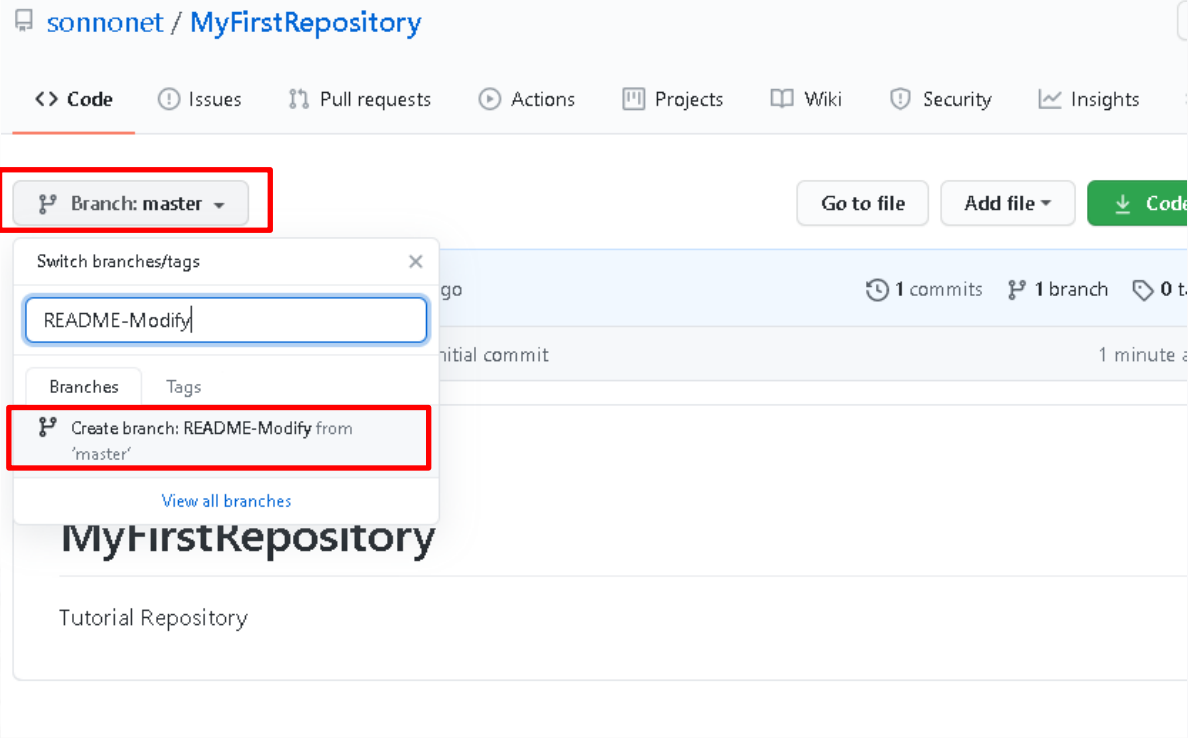


- 깃허브에서 개발자, 관리자, 디자이너들은 브랜치를 사용하여 버그 수정 및 특정 작업을 Master 로부터 분리, 수정이 완료되면 작업된 브랜치를 Master로 병합함.

GitHub - 분산 버전 관리 툴

Step 2. 브랜치(Branch) 생성하기

- Branch:master 클릭.
- "README-Modify"라는 master copy 브랜치 생성.
- Create branch:README-Modify from 'master' 클릭.
- 브랜치가 두개 생성
 - master(기본)
 - README-Modify



GitHub - 분산 버전 관리 툴

Step 2. 브랜치(Branch) 생성하기

- README-Modify 브랜치 작업

The screenshot shows the GitHub interface for the repository 'sonnonet / MyFirstRepository'. The 'Code' tab is selected. A red box highlights the 'Branch: README-Modify' dropdown menu. Below this, a message states 'This branch is even with master.' with links for 'Pull request' and 'Compare'. A commit history section shows a commit by 'sonnonet' with the message 'committed 98FF493 6 minutes ago', including details for '1 commits', '2 branches', and '0 tags'. The file 'README.md' is listed as an 'Initial commit' from '6 minutes ago'. The content of the README.md file is displayed, showing the title 'MyFirstRepository' and the subtitle 'Tutorial Repository'.

Step 3. 수정하고 커밋(commit)하기

- README.md 클릭

The screenshot shows a GitHub repository interface. At the top, there's a branch selector set to 'Branch: README-Modify'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below this, a status bar indicates 'This branch is even with master.' with links for 'Pull request' and 'Compare'. A commit history section shows a commit by 'sonnonet' 9 minutes ago. Below the commit list, the 'README.md' file is highlighted with a red rectangle. The file content area shows the title 'MyFirstRepository' and the subtitle 'Tutorial Repository'.

Branch: README-Modify

Go to file Add file Code

This branch is even with master. Pull request Compare

sonnonet committed 98ff493 9 minutes ago 1 commits 2 branches 0 tags

README.md Initial commit 9 minutes ago

README.md

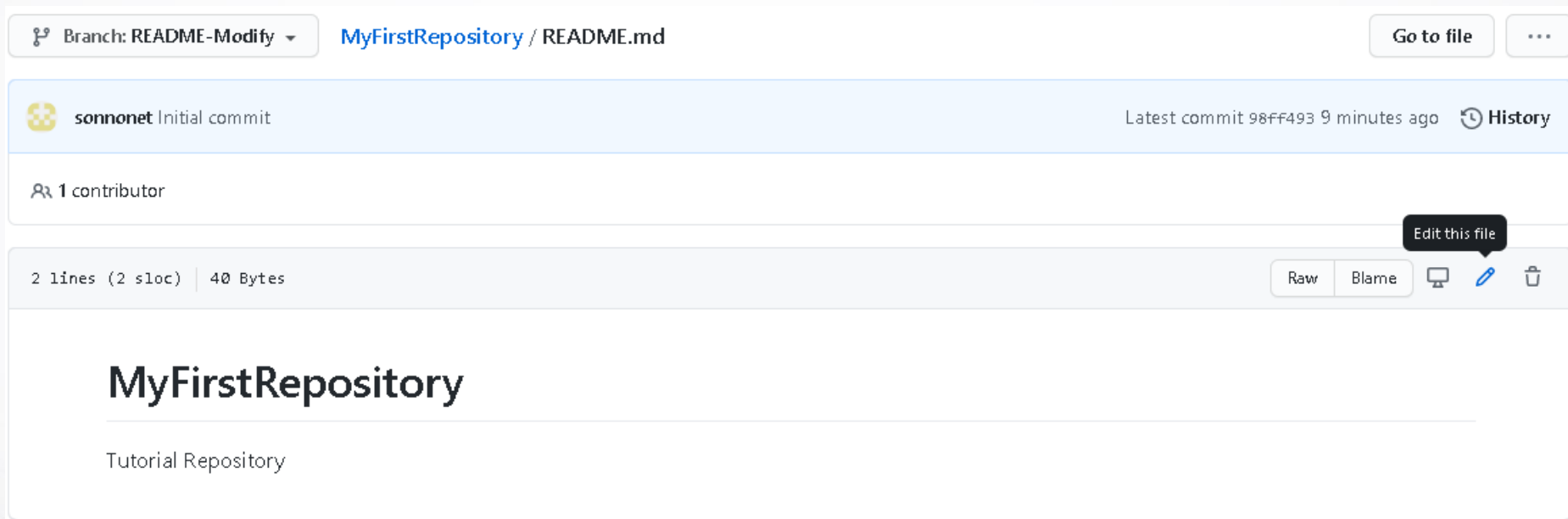
MyFirstRepository

Tutorial Repository

GitHub - 분산 버전 관리 툴

Step 3. 수정하고 커밋(commit)하기

- Edit this file 버튼 클릭



Branch: README-Modify ▾ MyFirstRepository / README.md Go to file ...

sonnonet Initial commit Latest commit 98ff493 9 minutes ago History

1 contributor

2 lines (2 sloc) | 40 Bytes Raw Blame Edit this file

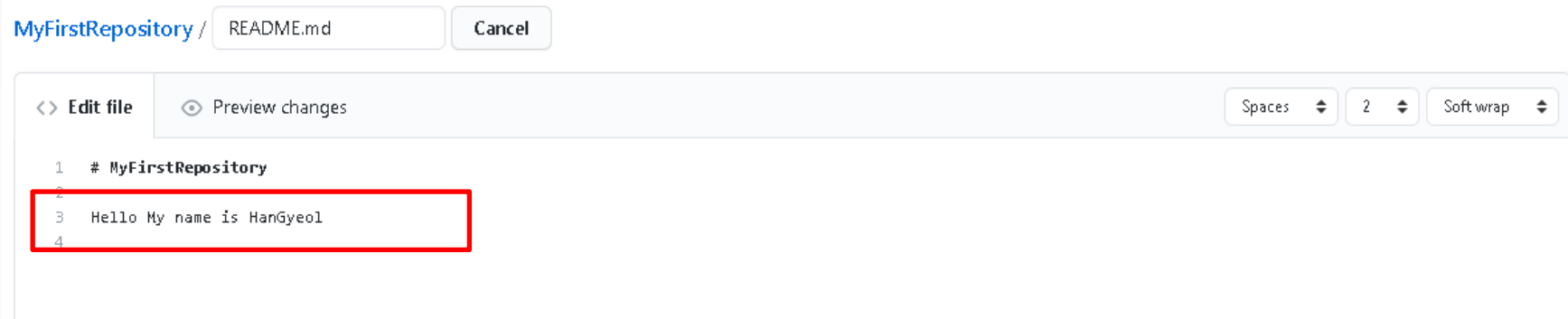
MyFirstRepository

Tutorial Repository

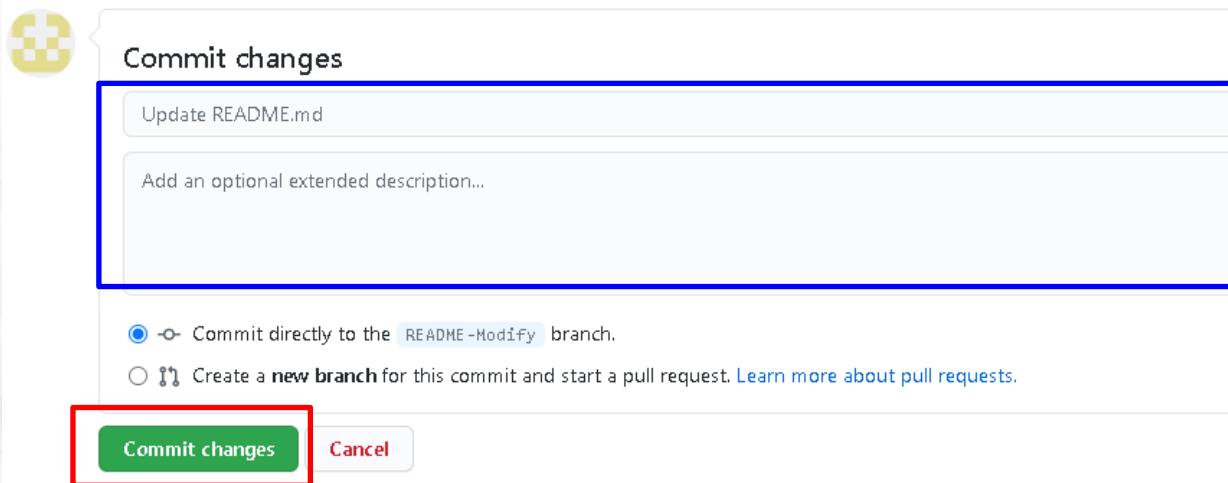
GitHub - 분산 버전 관리 툴

Step 3. 수정하고 커밋(commit)하기

- 원하는 내용으로 수정

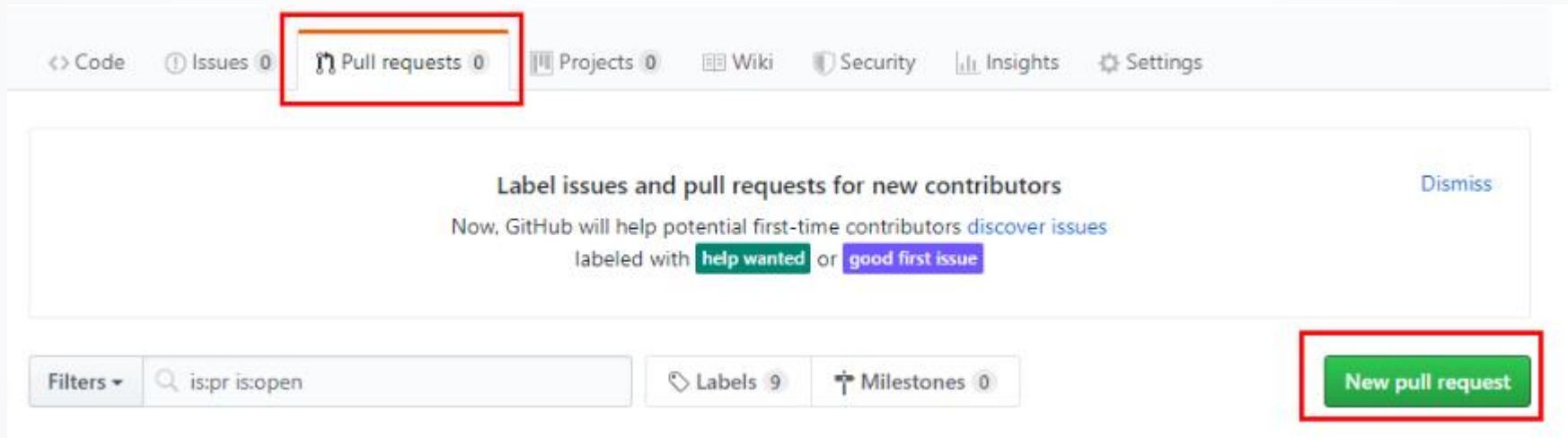


- 커밋 메시지를 작성후, Commit changes 클릭



Step 4. 풀(Pull)요청하기

- “Master” 외의 브랜치에서 수정발생, 풀 요청(Pull Request) 가능



- 풀요청(Pull Request)은 깃허브에서 협업의 핵심
- 사용자는 풀 요청을 하여 수정한 내용을 다른 사용자에게 제안하고, 다른 사람의 리뷰를 요청하여, 수정된 내용을 그들의 브랜치에 병합
- 커밋을 하면 바로 풀 요청을 열어서 토의를 시작할 수 있음.
- 깃허브의 멘션시스템을 사용하여, 당신의 특정 사람이나 팀에게 피드백을 요청 할 수 있음.

GitHub - 분산 버전 관리 툴

Step 4. 풀(Pull)요청하기

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: master ← compare: README-Modify ✓ **Able to merge.** These branches can be automatically merged.

② **Create pull request** Discuss and review the changes in this comparison with others.

1 commit

± 1 file changed

0 commit comments

1 contributor

Commits on Jul 12, 2020

sonnonet Update README.md

Verified 1f54f8c

Showing 1 changed file with 2 additions and 1 deletion.

Unified Split

3 README.md

<> [icon] ...

```
... @@ -1,2 +1,3 @@
1 1 # MyFirstRepository
2 - Tutorial Repository
2 +
3 + Hello My name is HanGyeol
```

※ ① base:master <- compare:README-Modify 확인

- 풀요청은 두 브랜치로부터 차이점을 보여줌, 수정,추가 (녹색) , 제외(빨간색)
- ② 이상이 없다면 Create pull request 클릭

Step 4. 풀(Pull)요청하기

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master

←

compare: README-Modify

✓ Able to merge. These branches can be automatically merged.

Update README.md

Add header text

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻

Hello and myname is Added

Attach files by dragging & dropping, selecting or pasting them.

MS

Create pull request

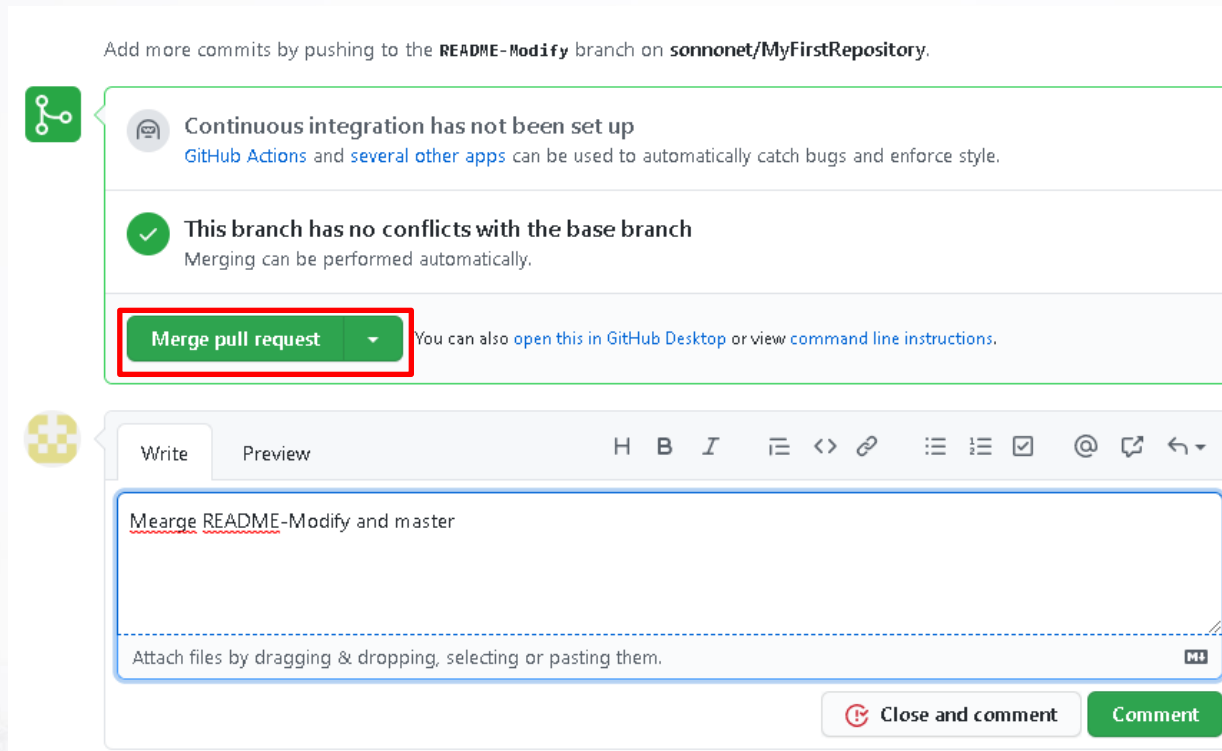
①

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

- 간단한 설명을 작성하고, ① Create pull request 클릭

GitHub - 분산 버전 관리 툴

Step 5. 병합(Merge) 하기



- Merge pull request 클릭

GitHub - 분산 버전 관리 툴

Step 5. 병합(Merge) 하기

Update README.md #1

 Open sonnonet wants to merge 1 commit into `master` from `README-Modify` 

 Conversation 0  Commits 1  Checks 0  Files changed 1



sonnonet commented 3 minutes ago

Owner  ...

Hello and myname is Added



 Update README.md

Verified 1f54f8c

Add more commits by pushing to the `README-Modify` branch on `sonnonet/MyFirstRepository`.



Merge pull request #1 from sonnonet/README-Modify

Mearge README-Modify and master

Confirm merge

Cancel

- Confirm merge 클릭

Step 5. 병합(Merge) 하기

Update README.md #1

 **Merged** sonnonet merged 1 commit into `master` from `README-Modify`  now

 Conversation 0  Commits 1  Checks 0  Files changed 1



sonnonet commented 4 minutes ago

Owner



Hello and myname is Added

  Update README.md

Verified

1f54f8c



 sonnonet merged commit **3429025** into `master` now

Revert



Pull request successfully merged and closed

You're all set—the `README-Modify` branch can be safely deleted.

Delete branch

- README-modify 는 더 이상 필요 없으므로 Delete branch 클릭

GitHub - 분산 버전 관리 툴

최종 확인

sonnonet / MyFirstRepository

Unwatch

1

Star

0

Fork

0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters

is:pr is:closed

Labels 9

Milestones 0

New pull request

Clear current search query, filters, and sorts

0 Open ✓ 1 Closed

Update README.md

#1 by sonnonet was merged 2 minutes ago

Author

Label

Projects

Milestones

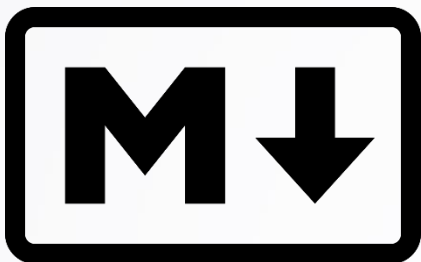
Reviews

Assignee

Sort

README 파일수정 (md)

마크 다운 문법



<https://ko.wikipedia.org/wiki/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4>

DHT sensor library Arduino Library CI passing

Description

An Arduino library for the DHT series of low-cost temperature/humidity sensors.

You can find DHT tutorials [here](#).

Dependencies

- [Adafruit Unified Sensor Driver](#)

Contributing

Contributions are welcome! Not only you'll encourage the development of the library, but you'll also learn how to best use the library and probably some C++ too

Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

<> Edit file

Preview changes

Spaces 2 Softwrap

```
1 # DHT sensor library \[\[Build Status\]\](https://github.com/adafruit/DHT-sensor-library/workflows/Arduino%20Library%20CI/badge.svg)(https://github.com/adafruit/DHT-sensor-library/actions)
2
3 ## Description
4
5 An Arduino library For the DHT series of low-cost temperature/humidity sensors.
6
7 You can find DHT tutorials \[here\](https://learn.adafruit.com/dht).
8
9 # Dependencies
10 * \[Adafruit Unified Sensor Driver\](https://github.com/adafruit/Adafruit_Sensor)
11
12 # Contributing
13
14 Contributions are welcome! Not only you'll encourage the development of the library, but you'll also learn how to best use the library and probably some C++ too
15
16 Please read our \[Code of Conduct\](https://github.com/adafruit/DHT-sensor-library/blob/master/CODE_OF_CONDUCT.md) before contributing to help this project stay welcoming.
17
18 ## Documentation and doxygen
19
20 Documentation is produced by doxygen. Contributions should include documentation for any new code added.
21
22 Some examples of how to use doxygen can be found in these guide pages:
23
```

GitHub – 리눅스 기반 사용방법

Git 저장소 다운로드

pi@raspberrypi: ~ \$

git clone https://github.com/<user_name>/<repository_name>



pi@raspberrypi: ~/work/jjvision

pi@raspberrypi:~/work \$ git clone https://github.com/sonnonet/jjvision

Cloning into 'jjvision'...

remote: Enumerating objects: 44, done.

remote: Counting objects: 100% (44/44), done.

remote: Compressing objects: 100% (35/35), done.

remote: Total 44 (delta 12), reused 17 (delta 5), pack-reused 0

Unpacking objects: 100% (44/44), done.

pi@raspberrypi:~/work \$ cd jjvision/

pi@raspberrypi:~/work/jjvision \$ ls

datasheet libjffi-1.2.so pir.py README.md simpletest.py

pi@raspberrypi:~/work/jjvision \$

GitHub – 리눅스 기반 사용방법

Git 저장소 업로드

```
pi@raspberrypi: ~ $ git add .
```

```
pi@raspberrypi: ~ $ git commit -m "<주석 내용>"
```

```
pi@raspberrypi: ~ $ git push
```

```
pi@raspberrypi: ~/work/jjvision
```

```
pi@raspberrypi:~/work/jjvision $ git add .
```

```
pi@raspberrypi:~/work/jjvision $ git commit -m "upload pir ver1.py"
```

```
[master c165449] upload pir_ver1.py  
1 file changed, 20 insertions(+)  
create mode 100644 pir_ver1.py
```

```
pi@raspberrypi:~/work/jjvision $ git push
```

```
Username for 'https://github.com': sonnonet
```

```
Password for 'https://sonnonet@github.com':
```

```
Enumerating objects: 4, done.
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 4 threads
```

```
Compressing objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 479 bytes | 239.00 KiB/s, done.
```

```
Total 3 (delta 1), reused 0 (delta 0)
```

```
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
```

```
To https://github.com/sonnonet/jjvision
```

```
0a23124..c165449 master -> master
```

```
pi@raspberrypi:~/work/jjvision $
```

GitHub – 리눅스 기반 사용방법 추가

GitHub 세부 명령어

git 기본 명령어

```
저장소 생성 : git init
저장소에 파일 추가 : git add 파일이름
저장소에 수정 내역 제출 : git commit -a(변경된 모든파일), git commit -m "커밋메세지", git commit
저장소 상태 확인 : git status
```

git branch 명령어

```
저장소에 브랜치 추가 : git branch 이름
작업 중인 브랜치 변경 : git checkout 브랜치 이름
브랜치 병합하기 : git merge 브랜치 이름
```

.gitignore 불필요한 파일 및 폴더 무시

```
touch .gitignore
```

<https://www.gitignore.io/> (자동생성 웹사이트)

git log 기록보기

```
git log -p 각 커밋에 적용된 실제 변경 내용을 보여줌
git log --stat 각 커밋에서 수정된 파일에 통계 정보를 보여줌
git log --graph 브랜치 분기와 병합 내역을 아스키 그래프로 보여줌
```

5주차 수업이 끝났습니다

고생하셨습니다.

