



제품소프트웨어 패키징  
(Git과 GitHub를 활용한 소스코드 관리)

# GitHub에서 Repository 만들기



## 학습내용

- GitHub 시작하기
- GitHub Repository 관리



## 학습목표

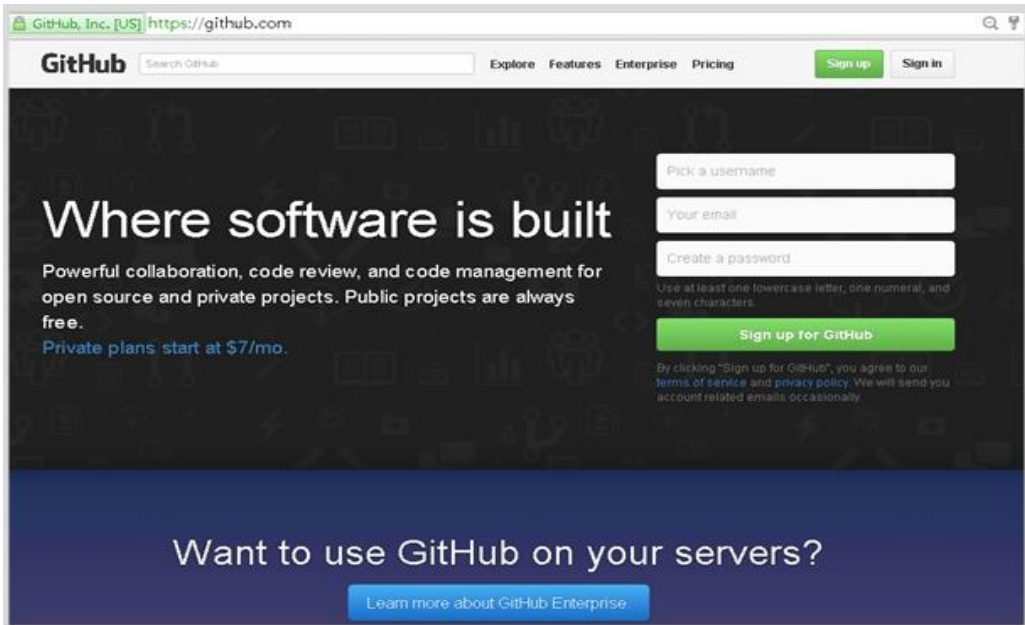
- GitHub에 직접 가입을 해보고 기본적인 기능을 사용할 수 있다.
- GitHub에 Repository를 생성하고 관리할 수 있다.



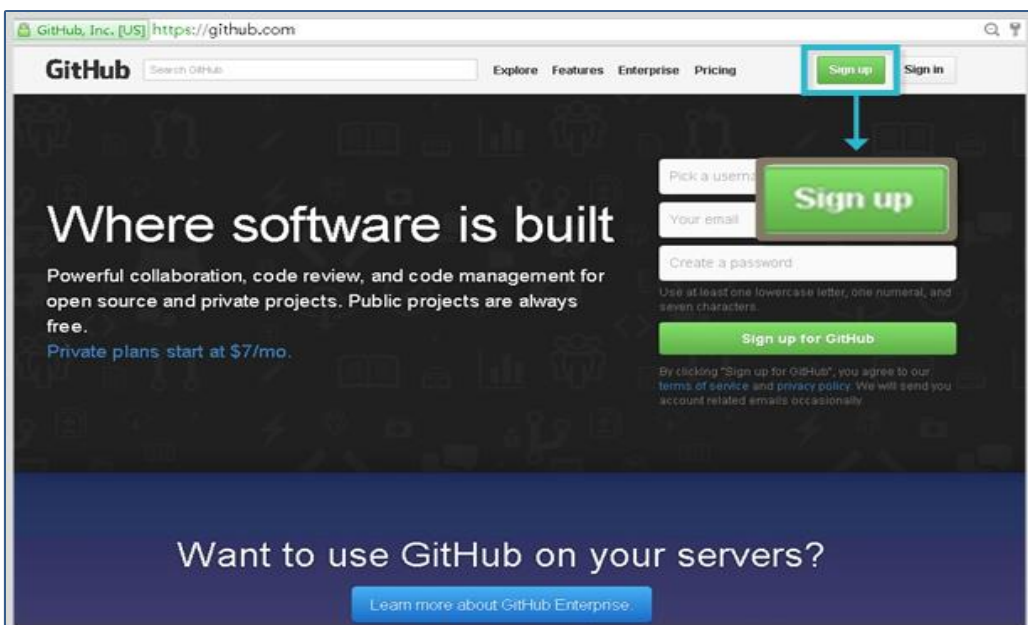
## GitHub 시작하기

### 1. GitHub 가입하기

#### 1) 홈페이지 접속



#### 2) 가입하기





## GitHub 시작하기

### 1. GitHub 가입하기

#### 3) 로그인

The screenshot shows the GitHub 'Join GitHub' page. At the top, there's a navigation bar with links for 'Explore', 'Features', 'Enterprise', and 'Pricing', along with 'Sign up' and 'Sign in' buttons. The main heading is 'Join GitHub' with the tagline 'The best way to design, build, and ship software.' Below this, a progress bar indicates three steps: 'Step 1: Set up a personal account' (active), 'Step 2: Choose your plan', and 'Step 3: Go to your dashboard'. The 'Create your personal account' section includes input fields for 'Username', 'Email Address', and 'Password'. A note states: 'This will be your username — you can enter your organization's username next.' Another note says: 'You will occasionally receive account related emails. We promise not to share your email with anyone.' A password requirement note says: 'Use at least one lowercase letter, one numeral, and seven characters.' At the bottom of this section is a green 'Create an account' button. To the right, a box titled 'You'll love GitHub' lists benefits: 'Unlimited collaborators', 'Unlimited public repositories', 'Great communication', 'Friction-less development', and 'Open source community'.

#### 4) 이용방법 설정 - 사용형태에 따라

- 무료 서비스
  - 소스코드와 프로젝트 페이지가 모두 공개로 되어야 함
  - 기능상 제약 없이 거의 모두 사용할 수 있음
- 유료 서비스
  - 소스코드 공개여부를 결정할 수 있음



## GitHub 시작하기

### 1. GitHub 가입하기

#### 4) 이용방법 설정 - 사용형태에 따라

**Welcome to GitHub**  
You've taken your first step into a larger world, @GirrrMasters.

Completed Set up a personal account    Step 2: Choose your plan    Step 3: Go to your dashboard

**Choose your personal plan**

| Plan   | Cost (view in KRW) | Private repositories |                        |
|--------|--------------------|----------------------|------------------------|
| Large  | \$50/month         | 50                   | <a href="#">Choose</a> |
| Medium | \$22/month         | 20                   | <a href="#">Choose</a> |
| Small  | \$12/month         | 10                   | <a href="#">Choose</a> |
| Micro  | \$7/month          | 5                    | <a href="#">Choose</a> |
| Free   | \$0/month          | 0                    | <a href="#">Chosen</a> |

Each plan includes:  
Unlimited collaborators  
Unlimited public repositories


- ✓ Free setup
- ✓ HTTPS Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages,...

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an estimate based on current exchange rates. Local prices will change as the exchange rate fluctuates. Don't worry, you can cancel or upgrade at any time.

☐ **Help me set up an organization next**  
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees. [Learn more about organizations.](#)

[Finish sign up](#)

#### 5) 사용자 인증



Hi @GirrrMasters!

Help us secure your GitHub account by verifying your email address (ID@도메인주소). This lets you access all of GitHub's features.

[Verify email address](#)

Button not working? Paste the following link into your browser:  
[https://github.com/users/GirrrMasters/emails/17849880/confirm\\_verification/64937996a742224c4981d0049e6dac4c3c0c0dc3](https://github.com/users/GirrrMasters/emails/17849880/confirm_verification/64937996a742224c4981d0049e6dac4c3c0c0dc3)

You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.

**GitHub**

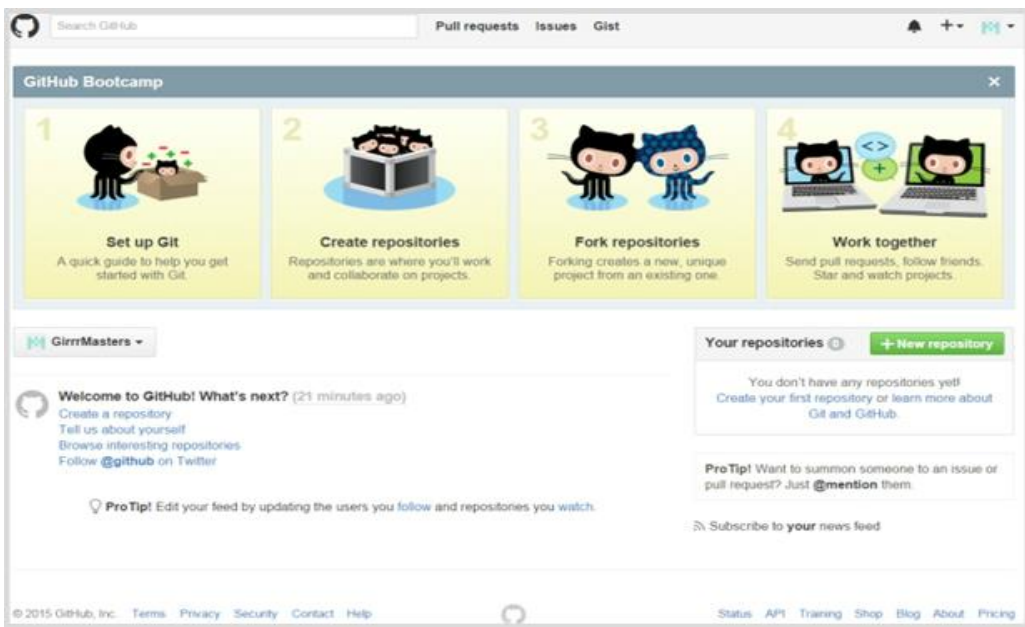
Sent with <3 by GitHub.  
GitHub, Inc. 88 Colin P Kelly Jr Street  
San Francisco, CA 94107



## GitHub 시작하기

### 1. GitHub 가입하기

#### 6) 계정생성



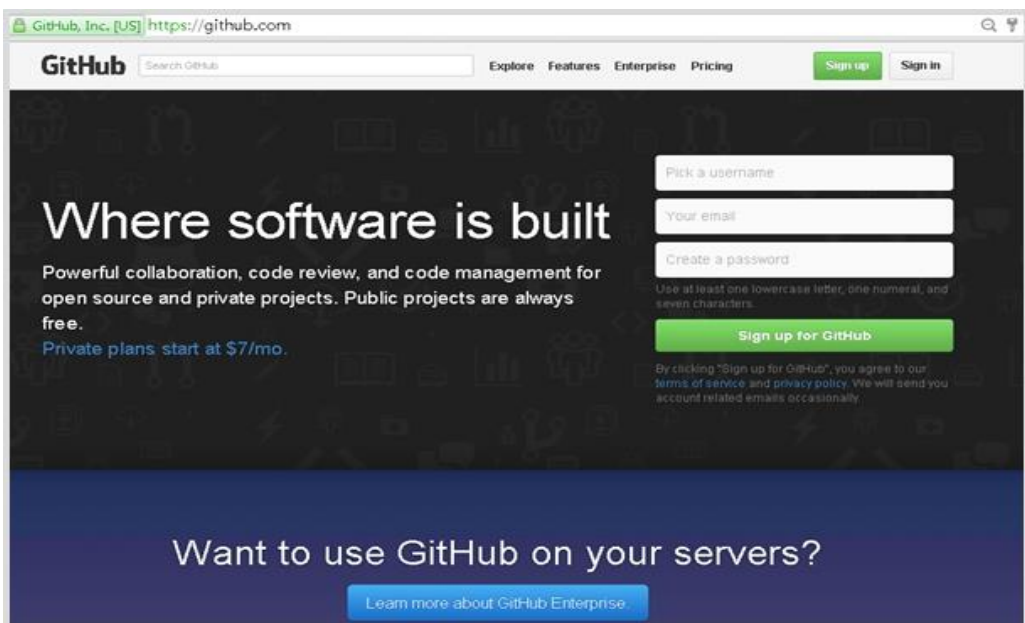


## GitHub 시작하기

### 2. GitHub 서비스 정책

#### 1) GitHub의 특징

- Git을 사용하여 소스코드 관리를 하는 사람들의 편의를 위해 개발됨
- 오픈소스를 개발하는 개발자들에 의해 창업한 회사임
- 전체적으로 무료 형태의 서비스 위주로 진행됨
  - 오픈소스 프로젝트나 공개적으로 개발해도 문제가 없는 경우 무료 서비스로 충분히 사용 가능함





## GitHub 시작하기

### 2. GitHub 서비스 정책

#### 2) Personal plans

- 다른 사람들과 함께 공유하면서 진행할 프로젝트를 위한 요금제
- 기본적으로 프로젝트를 다른 사람들과 공유할 목적이라면 아무 제약 없이 무료로 사용가능
  - GitHub의 최대의 홍보 방법이자 전략
- 비공개로 진행할 프로젝트의 경우 Repository 수에 따라 비용 책정
- 개발자 참여 수 등에는 별도의 제한이 없음

#### 3) Organization plans

- 특정 기업이나 단체 등을 위해 제공되는 요금제
- Organization plans의 특징
  - 기업에서 내부적으로 프로젝트 진행에 사용하기에 적합한 요금 정책임
  - 공개 프로젝트 진행 시 무료로 이용 가능함
  - 비공개 프로젝트 진행 시 Private 저장소 개수에 따라 요금이 차등 부과
- GitHub는 별도의 설치나 관리에 필요한 인력이 없으므로, 요금만을 비교하기보다 각각의 상황에 따라 선택을 해야 함

#### 4) Add-ons

- 별도의 파일을 저장하기 위한 공간이 필요할 때 사용
  - 소스코드를 올리는 공간에 대한 요금은 이전의 요금제에 모두 포함
  - 오픈소스를 사용 시 추가 요금없이 사용
  - 큰 용량의 파일을 별도로 올려놓는 경우 : GitHub를 이용한다면 별도의 저장공간만을 유료로 활용
- 저장소의 용량과 트래픽에 따라 요금을 부과함

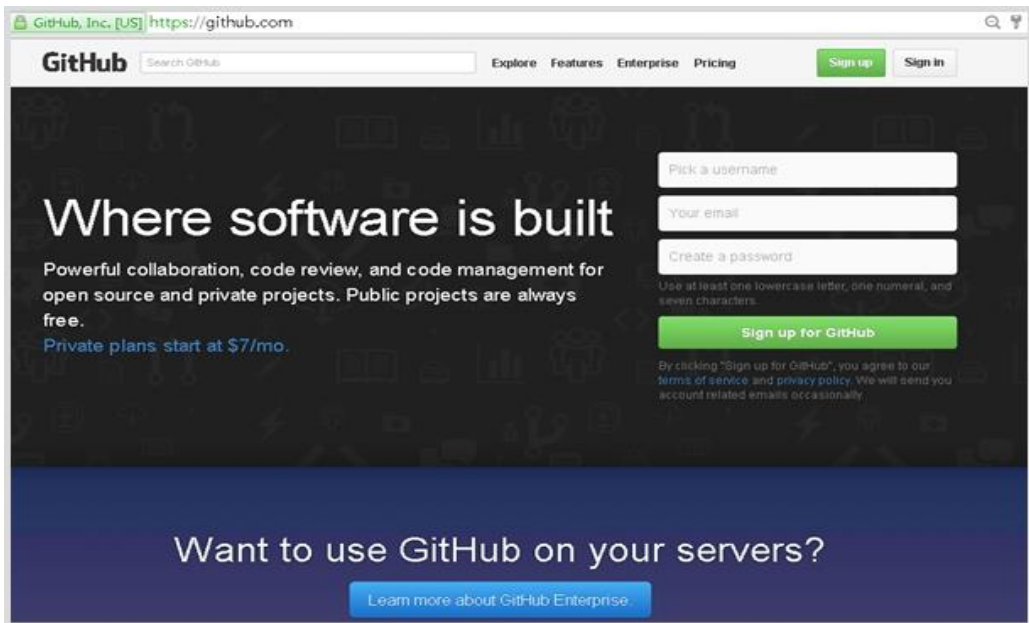




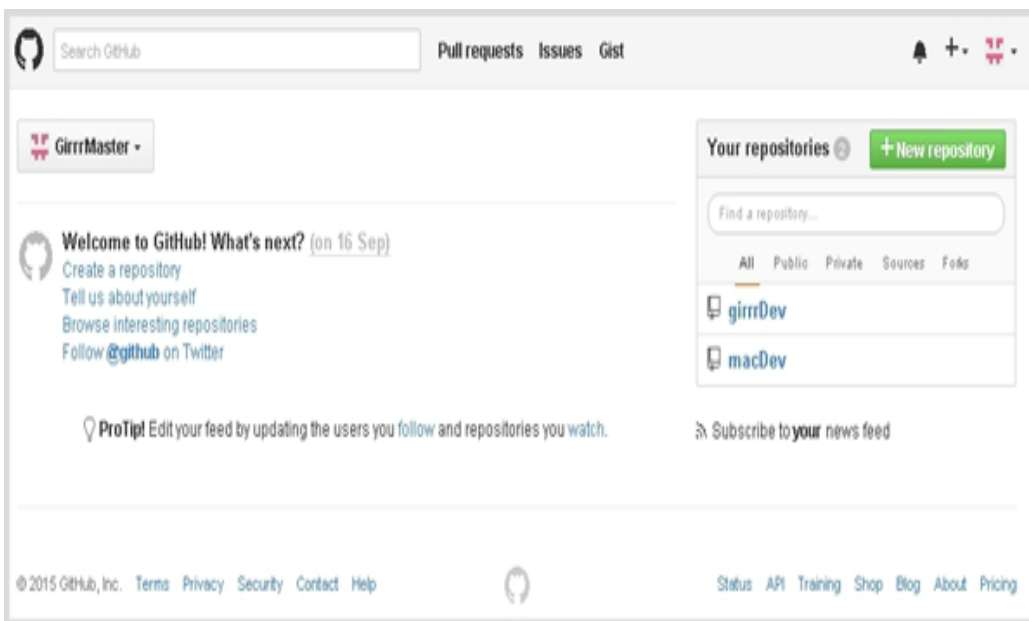
## GitHub Repository 관리

### 1. Repository 생성

#### 1) 로그인



#### 2) New repository 클릭





## GitHub Repository 관리

### 1. Repository 생성

#### 3) New repository 생성에 필요한 항목 입력

Search GitHub Pull requests Issues Gist

### Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: GirrrMaster Repository name:

Great repository names are short and memorable. Need inspiration? How about `feckless-octo-broccoli`.

Description (optional):

☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

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

Create repository

© 2015 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

- Repository name : 생성한 Repository의 ID
- Description : 생성할 저장소에 대한 간략한 설명
- Public / Private : 저장소의 공개 여부
- Initialize this repository with a README : 저장소에 README.md 파일 생성. 다른 유저들이 Repository를 이용할 때 가이드를 줄 수 있음
- Add .gitignore : 파일 업로드 하기 전에 업로드를 무시할 파일 형태 지정
- Add license : 해당 라이선스 지정



## GitHub Repository 관리

### 1. Repository 생성

#### 4) Create repository 클릭

The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. The main heading is 'Create a new repository' with a subtext: 'A repository contains all the files for your project, including the revision history.' Below this, there are two input fields: 'Owner' (set to 'GirrrMaster') and 'Repository name' (set to 'myRepo' with a green checkmark). A note says: 'Great repository names are short and memorable. Need inspiration? How about `jocular-octo-tatertot`.' There's a 'Description (optional)' field with the text 'Github repository'. Below that, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. A checkbox 'Initialize this repository with a README' is checked. At the bottom, there are two dropdown menus: 'Add .gitignore' (set to '.C') and 'Add a license' (set to 'GNU General Public License v2.0'). A green 'Create repository' button is at the bottom. The footer contains copyright information and various links like 'Status', 'API', 'Training', etc.

#### 5) New repository 생성 완료

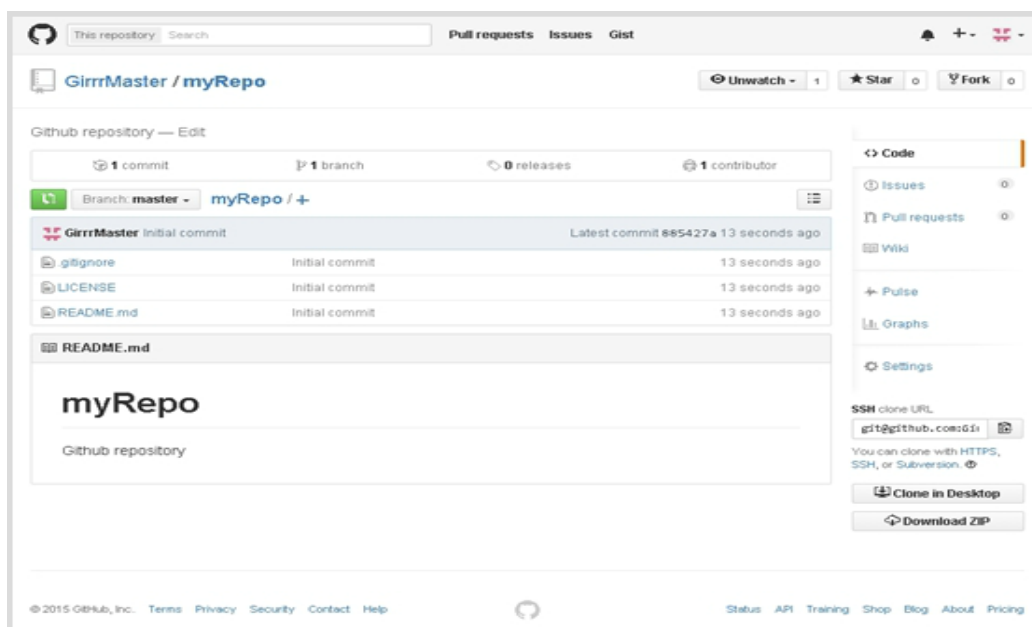
The screenshot shows the GitHub repository page for 'GirrrMaster / myRepo'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. The repository name 'GirrrMaster / myRepo' is prominently displayed. Below it, there are buttons for 'Unwatch', 'Star', and 'Fork'. The main content area shows the repository details: '1 commit', '1 branch', '0 releases', and '1 contributor'. A table lists the initial commit files: '.gitignore', 'LICENSE', and 'README.md', all committed 13 seconds ago. Below the table is a 'README.md' section with the text 'myRepo' and 'Github repository'. On the right side, there's a sidebar with links for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. At the bottom of the sidebar, there's an 'SSH clone URL' section with the URL 'git@github.com:GirrrMaster/myRepo.git' and buttons for 'Clone in Desktop' and 'Download ZIP'. The footer contains copyright information and various links like 'Status', 'API', 'Training', etc.



## GitHub Repository 관리

### 2. Repository 화면 구성

#### 1) Repository 기능



- Master Branch : Master Branch 표시. Branch가 있음
- 1branch : 해당 Repository에서 사용되고 있는 Branch 수. 브랜치가 여러 개 생성되면 값이 변경됨
- .gitignore / License / README.md : 생성시 사용자가 추가한 파일들이 보임. 실제 개발자가 사용하며 생성되는 소스코드들이 나타남
- myRepo : 생성된 파일 중 README.md 파일의 내용
- issue / pull request : 해당 Repository에 현재 issue와 pull request 수
- 0 release : Release하면 이름 붙은 Tar Ball을 다운로드 할 수 있음
- 1contributor : 해당 Repository에 참여하는 참여자. Contributor가 많다면 많이 활성화된 오픈소스 프로젝트라고 판단하는 척도가 됨
- Settings : Repository의 기능 수정
- GirrMaster Initial commit : 각각의 항목을 클릭하여 파일들을 직접 수정하고 Commit 하거나 내용을 조회할 수 있음
- SSH clone URL : 해당 Repository를 사용자의 로컬 환경에서 URL을 통하여 내용을 가져올 수 있게 하는 주소 생성



## GitHub Repository 관리

### 2. Repository 화면 구성

#### 2) Repository 기능 - .gitignore 파일

```
Branch: master - myRepo / .gitignore
GerrMaster Initial commit 885427a 21 hours ago
1 contributor

33 lines (27 sloc) 246 Bytes
Raw Blame History

1 # Object files
2 *.o
3 *.ko
4 *.obj
5 *.elf
6
7 # Precompiled Headers
8 *.gch
9 *.pch
10
11 # Libraries
12 *.lib
13 *.a
14 *.la
15 *.lo
16
17 # Shared objects (inc. windows DLLs)
18 *.dll
19 *.so
20 *.so.*
21 *.dylib
22
23 # Executables
24 *.exe
25 *.out
26 *.app
27 *.S*86
28 *.x86_64
29 *.hex
30
31 # Debug files
32 *.dSYM/
```

- Repository 생성시 C 언어의 타입 선택
- Object files, Precompiled Header, Shared library 등 제외시킬 항목의 리스트 설정
- 이 내용들을 제외하여 순수 해당 프로젝트에서 관리할 소스코드들만 들어갈 수 있도록 설정
- 개발자 각 환경에서 자동으로 생성되는 빌드 파일 등을 제외시키는데 사용
- 필요하다 하는 파일은 .gitignore 파일을 수정하여 제외 시킬 수 있음
- Gitignore를 제대로 활용하지 않고 모든 소스코드를 포함 시킬 경우, 빌드 파일 등 특정 환경에서만 변경되는 파일들이 자꾸 수정사항으로 반영되어 관리가 어려울 수 있음



## GitHub Repository 관리

### 2. Repository 화면 구성

#### 3) Repository 기능 - LICENSE

```
Branch: master - myRepo / LICENSE
GirrrMaster Initial commit 885427a 21 hours ago
1 contributor

341 lines (281 sloc) 17.6 KB
Raw Blame History

1 GNU GENERAL PUBLIC LICENSE
2 Version 2, June 1991
3
4 Copyright (C) 1989, 1991 Free Software Foundation, Inc., <http://fsf.org/>
5 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
6 Everyone is permitted to copy and distribute verbatim copies
7 of this license document, but changing it is not allowed.
8
9 Preamble
10
11 The licenses for most software are designed to take away your
12 freedom to share and change it. By contrast, the GNU General Public
13 License is intended to guarantee your freedom to share and change free
14 software--to make sure the software is free for all its users. This
15 General Public License applies to most of the Free Software
16 Foundation's software and to any other program whose authors commit to
17 using it. (Some other Free Software Foundation software is covered by
18 the GNU Lesser General Public License instead.) You can apply it to
19 your programs, too.
20
21 When we speak of free software, we are referring to freedom, not
22 price. Our General Public Licenses are designed to make sure that you
23 have the freedom to distribute copies of free software (and charge for
24 this service if you wish), that you receive source code or can get it
25 if you want it, that you can change the software or use pieces of it
26 in new free programs; and that you know you can do these things.
```

- Revision History와 내용 편집 아이콘
- History 파일 : 해당 파일의 수정 이력 등을 확인
- 편집 아이콘 : LICENSE 파일 수정



## GitHub Repository 관리

### 2. Repository 화면 구성

#### 4) Repository 기능 - LICENSE 수정



- 첫 줄에 '[GirrrMaster GitHub]' 입력
- 화면 하단 부분에 Commit changes 안에서 commit message와 간단한 description을 추가
- 하단 두 가지 선택 부분에서는 새롭게 branch를 만들어서 commit 을 할 것인지 아니면 원본을 수정할 것인지를 체크하고 Commit changes 버튼을 클릭하여 수정완료



## GitHub Repository 관리

### 2. Repository 화면 구성

#### 4) Repository 기능 - LICENSE 수정, commit

The screenshot shows the GitHub interface for a repository named 'myRepo' with a file named 'LICENSE'. The 'History' tab is active, displaying a list of commits. The first commit, 'Initial commit' (885427a), is selected, and its details are shown below. The commit message is 'Initial commit' and it was authored by 'GirrrMaster' 22 hours ago. The commit content is the GNU General Public License (Version 2, June 1991). The license text is displayed in a monospaced font, with line numbers 1 through 15 visible. The license text includes the title 'GNU GENERAL PUBLIC LICENSE', the version 'Version 2, June 1991', the copyright notice 'Copyright (C) 1989, 1991 Free Software Foundation, Inc.', and the full text of the license, including the preamble and the main body of the license.

- 수정이 완료된 화면을 보면 새롭게 입력한 부분이 추가
- 수정된 파일의 History와 Blame 기능을 이용하여 변경된 내용 확인
- GitHub에서는 모든 변화에 대한 ID 값이 부여

#### 5) GitHub의 특징

- 개발자의 PC에서 Git을 통해 소스코드를 받아 내용을 수정하고 다시 Commit하여 반영하는 절차가 없음
- 바로 웹 상에서도 내용을 수정할 수 있고, 그 이력을 바로 확인할 수 있는 기능을 제공하고 있음
- 단순히 이력에 대한 확인 및 단순 수정의 경우, 바로 수정이 가능하므로 매우 편리한 기능임
- GitHub는 단순히 Git의 저장소를 관리하는 기능 뿐 아니라 프로젝트 진행을 위한 다양한 기능들을 제공하고 있음





## ! 핵심정리



### GitHub 시작하기

#### 1. GitHub 가입하기

- 홈페이지 접속
- 가입하기
- 로그인
- 이용방법 설정
- 사용자 인증

#### 2. GitHub 서비스 정책

- Personal plans : 다른 사람들과 함께 공유하면서 진행할 프로젝트를 위한 요금제
- Organization plans : 특정 기업이나 단체 등을 위해 제공되는 요금제
- Add-ons : 추가 기능을 이용할 수 있는 요금

## ! 핵심정리



### GitHub Repository 관리

#### 1. Repository 생성

- 로그인
- New repository 클릭
- New repository 생성에 필요한 항목 입력
- Create repository 클릭
- New repository 생성 완료