

# Git / Github

source code management system

( 소스관리 도구 , 실습위주 )

focused on training

미래창조과학부 KossLab.  
Taeung Song  
taeung@kosslab.kr

# Speaker

---

**송태웅** (Taeung Song, <https://github.com/taeung>)

- 미래창조과학부 KOSS(Korean Open Source Software) Lab. Software Engineer
- Linux Kernel 프로젝트 Contributor 활동중
- 전) XS inc. 선임연구원 (Linux, NAS, ARM)

## 강의활동

- SK C&C Git/Github 사내교육 동영상 강의
- 삼성 SOSCON 2016 발표 및 Advanced Git 튜토리얼
- 네이버 DEVVIEW 2016 연사
- 서강대, 아주대, 부산대, 제주대, OSS 포럼 등 Git/Github 강의
- 국민대, 이화여대 등 Linux perf, Opensource 참여 관련 시간강사 활동

# Training mode Git 훈련방식

---

우리가 스마트폰을 ( 메뉴얼 없이 ) 사용하면서 이해하듯

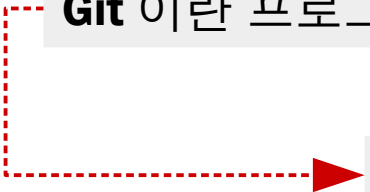
**Git** 이란 프로그램도 일단 써보면서 이해 해보자 ( 실습위주 교육 )

# Training mode Git 훈련방식

---

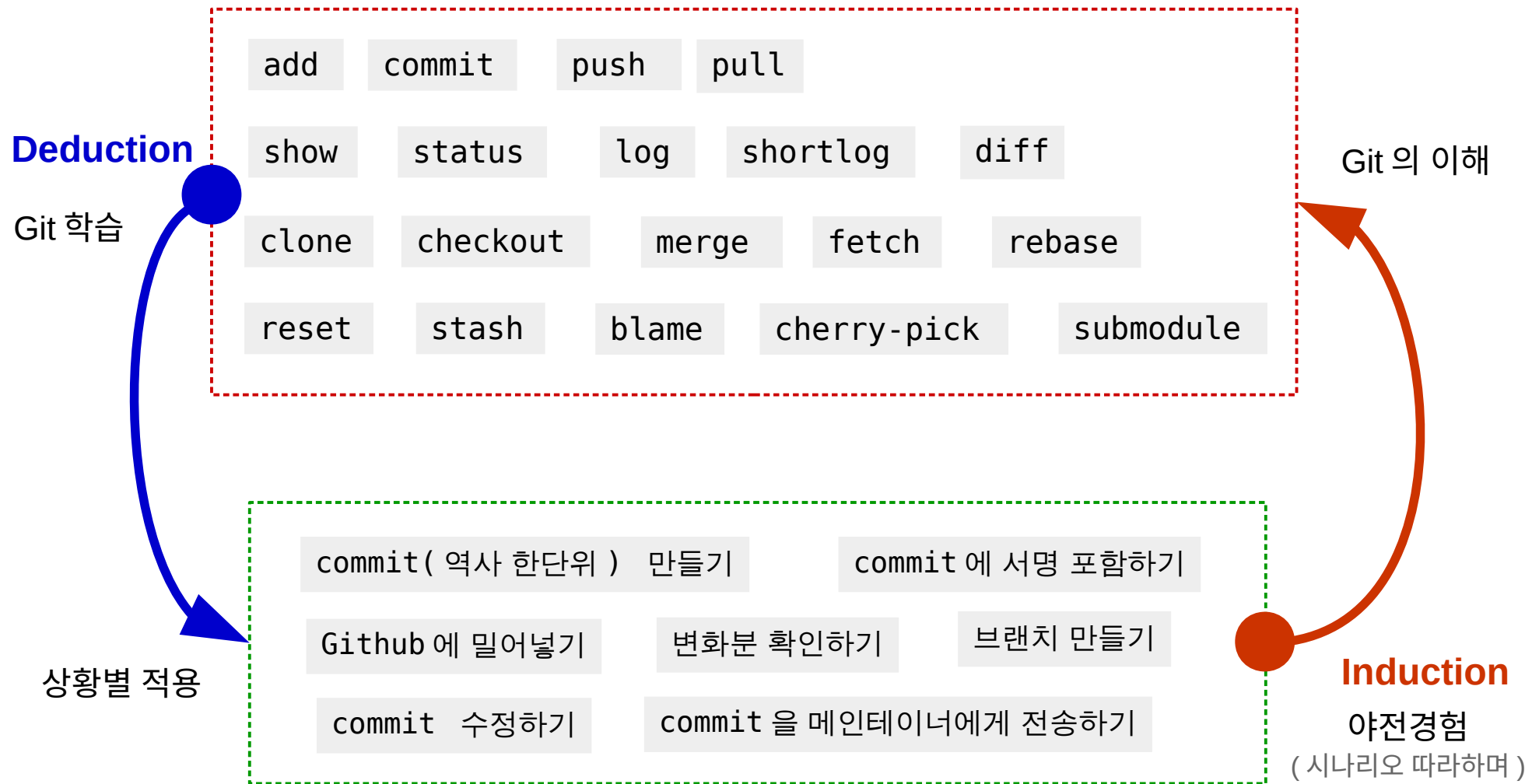
우리가 스마트폰을 ( 메뉴얼 없이 ) 사용하면서 이해하듯

**Git** 이란 프로그램도 일단 써보면서 이해 해보자 ( 실습위주 교육 )



백문이불여일행 ( 百聞不如一行 )

# Training mode Git 훈련방식



# Contents

---



# Git이란 간단한 정의

---

Git == Histroy 관리하는 도구

# Git 이란 간단한 정의

---

Git 개발과정 , 소스파일 등을 관리하는 도구

Histroy 관리가 되어 개발되어온 과정 , 역사를 볼 수 있고 , 특정시점으로 복구가능



# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법

Source code management **tool**

그 사이에 뭐가 바뀌었는지  
**차이 (Diff) 를 알 수 없다 .**



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip

# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법

Source code management **tool**

그 사이에 뭐가 바뀌었는지  
**차이 (Diff) 를 알 수 없다 .**

Ctrl + c, v 를 할수록  
차지하는 **용량 X 2**  
**X 3 ... + diff**



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip

# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법

Source code management **tool**

그 사이에 뭐가 바뀌었는지  
**차이 (Diff)** 를 알 수 없다 .

Ctrl + c, v 를 할수록  
차지하는 **용량 X 2**  
X 3 ... + diff



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip

< History 관리가능 >  
**차이 (Diff)** 가 무엇이고  
**수정 이유**를 Log 를 남길수있다 .

# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법

Source code management **tool**

그 사이에 뭐가 바뀌었는지  
**차이 (Diff)** 를 알 수 없다 .

Ctrl + c, v 를 할수록  
차지하는 **용량 X 2**  
X 3 ... + diff



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip

< History 관리가능 >  
**차이 (Diff)** 가 무엇이고  
**수정 이유**를 Log 를 남길수있다 .

< 타임머신 기능 >  
**현재파일들은 안전한 상태로**  
**과거**상태 그대로 복원가능 (반대도 가능)  
( 각 버전별 **차이만 저장**해서 **size 감소** )

# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법

Source code management **tool**

그 사이에 뭐가 바뀌었는지  
**차이 (Diff)** 를 알 수 없다 .

Ctrl + c, v 를 할수록  
차지하는 **용량 X 2**  
X 3 ... + diff



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip

< History 관리가능 >  
**차이 (Diff)** 가 무엇이고  
**수정 이유**를 Log 를 남길수있다 .

< 특정 버전 관리 >  
tag 나 release 로  
관리가능

< 타임머신 기능 >  
**현재파일들은 안전한 상태로**  
**과거**상태 그대로 복원가능 (반대도 가능)  
( 각 버전별 **차이만 저장**해서 **size 감소** )

# Git 의 기능

Ctrl+c, v 나 Alzip 압축파일 관리법



Git 배우는데 시간소비하느니  
Code 한줄이라도 더 개발 ..

Source code management **tool**



좋은건 알겠는데 ..  
Git 을 쓸 이유가 부족 ..

Wants 는 맞지만 Needs 는 아니야



과제 1\_ 최종 \_2016\_02\_28.zip

과제 1\_ 진짜최종 \_2016\_02\_29.zip

과제 1\_ 진짜진짜최종 \_2016\_03\_01.zip



# Git 실습 각 단계별

---



# Git 실습 각 단계별

---

Don't think about git,  
just do git

똑같이 따라 해보자 Git (<https://github.com/Taeung/git-training>)



# Git 실습 준비하기

---

## 1. 예제소스 다운받기

[https://www.dropbox.com/sh/3ywkargf9xzcfoi/AAC63uvN4eQBQhT\\_1m4GmCMLa?dl=1&pv=1](https://www.dropbox.com/sh/3ywkargf9xzcfoi/AAC63uvN4eQBQhT_1m4GmCMLa?dl=1&pv=1)

## 2. Git 설치하기

▶ <https://git-scm.com/downloads>

## 3. Editor 다운받기


▶ <https://atom.io/>

## 4. Github 회원가입

▶ <https://github.com/join>


# Git 실습 Github 회원가입


가입정보 기입


 [Personal](#) [Open source](#) [Business](#) [Explore](#) [Pricing](#) [Blog](#) [Support](#)  [Sign in](#) [Sign up](#)

## Join GitHub

The best way to design, build, and ship software.

 **Step 1:**  
Set up a personal account

 **Step 2:**  
Choose your plan

 **Step 3:**  
Go to your dashboard

### Create your personal account

**Username**

This will be your username — you can enter your organization's username next.

**Email Address**

You will occasionally receive account related emails. We promise not to share your email with anyone.

**Password**




Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

[Create an account](#)

#### You'll love GitHub

**Unlimited** collaborators  
**Unlimited** public repositories

 Great communication  
 Friction-less development  
 Open source community

<https://github.com/join>

# Git 실습 Github 메일인증하기

본인 메일확인해서 버튼 누르기



Hi @yukult400!

Help us secure your GitHub account by verifying your email address (yukult400@naver.com). 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/yukult400/emails/21042026/confirm\\_verification/388b1d3024fc012d64848df72d2d8fc4d7dc3b5a](https://github.com/users/yukult400/emails/21042026/confirm_verification/388b1d3024fc012d64848df72d2d8fc4d7dc3b5a)

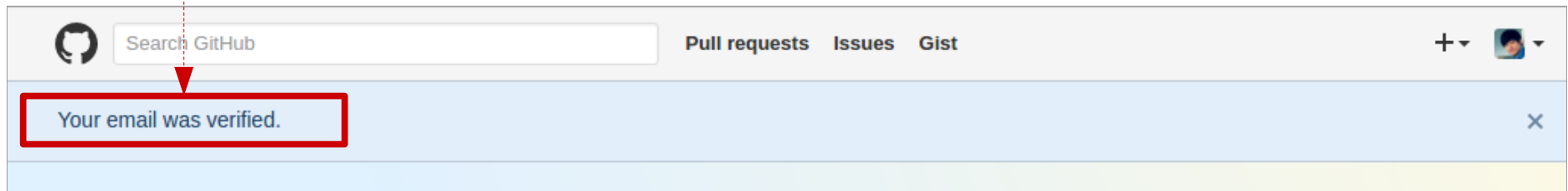
You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.

GitHub

# Git 실습 Github 메일인증하기

---

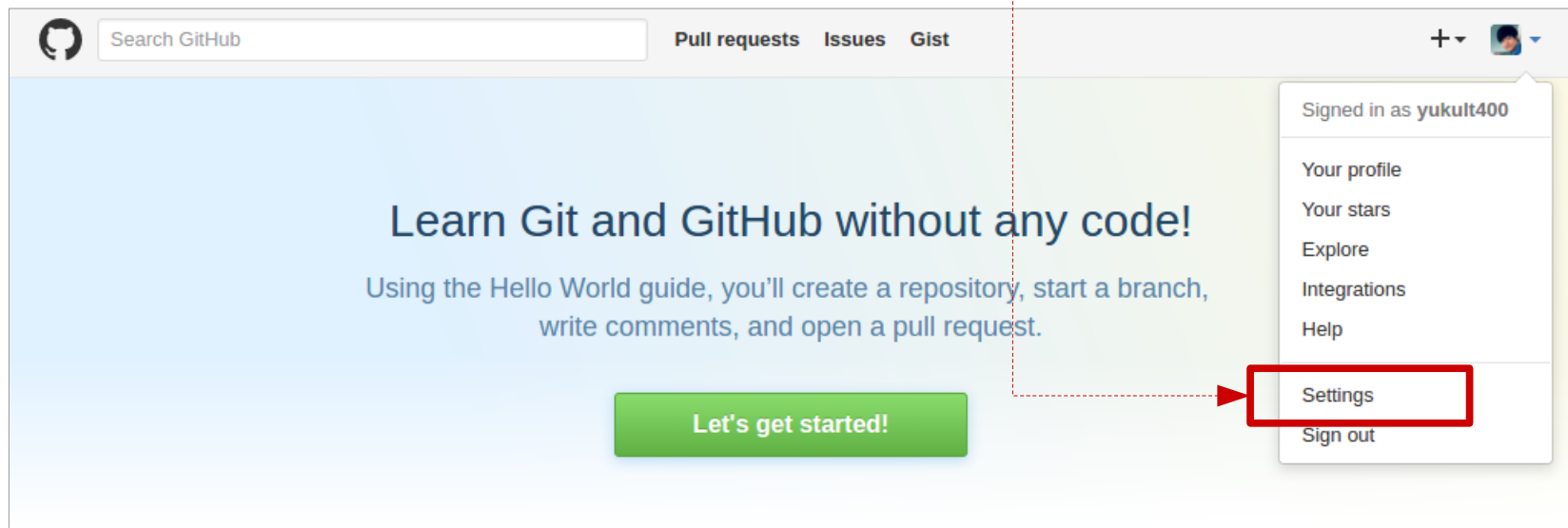
다시 Github 접속해서 메일인증 확인하기



<http://github.com>

# Git 실습 Github 메일인증하기

기다려도 메일이 혹시 안왔으면 설정버튼



<http://github.com>

# Git 실습

## Github 메일인증하기

메일이 혹시 안왔으면 인증메일 다시 보내기

**Personal settings**

- Profile
- Account
- Emails** ⚠️
- Notifications
- Billing
- SSH keys

**Email**

Your **primary GitHub email address** will be used for account-related notifications (e.g. account changes and billing receipts) as well as any web-based GitHub operations (e.g. edits and merges).

Please **verify an email address** to collaborate on GitHub, receive notifications, and help us protect your account.

yukult400@naver.com **Primary** Public Unverified

**Add email address**

**Add**

Verification email sent. [Resend](#) 🗑️

<https://github.com/settings/emails>

# Git 실습 준비하기

---

## 5. git-training-ex-v2.zip 압축풀고 폴더 열어두기

→ Ctrl-c, v 복사 , 붙여넣기 할 소스들

## 6. 편집기 (atom, sublime text) 열어두기

→ Ctrl-c, v 복사 , 붙여넣기 용도 편집기

# Git 실습 방법 , 시나리오

---

Git 실습 방법은 ?

직접 C 프로그래밍을 짜면서 Git 을 사용한다는 시나리오

report card ( 성적 출력하기 문제 )

- 1) 코딩은 Ctrl-c, v 로 하고
- 2) Git 은 직접 명령어 쳐서 (git-bash)

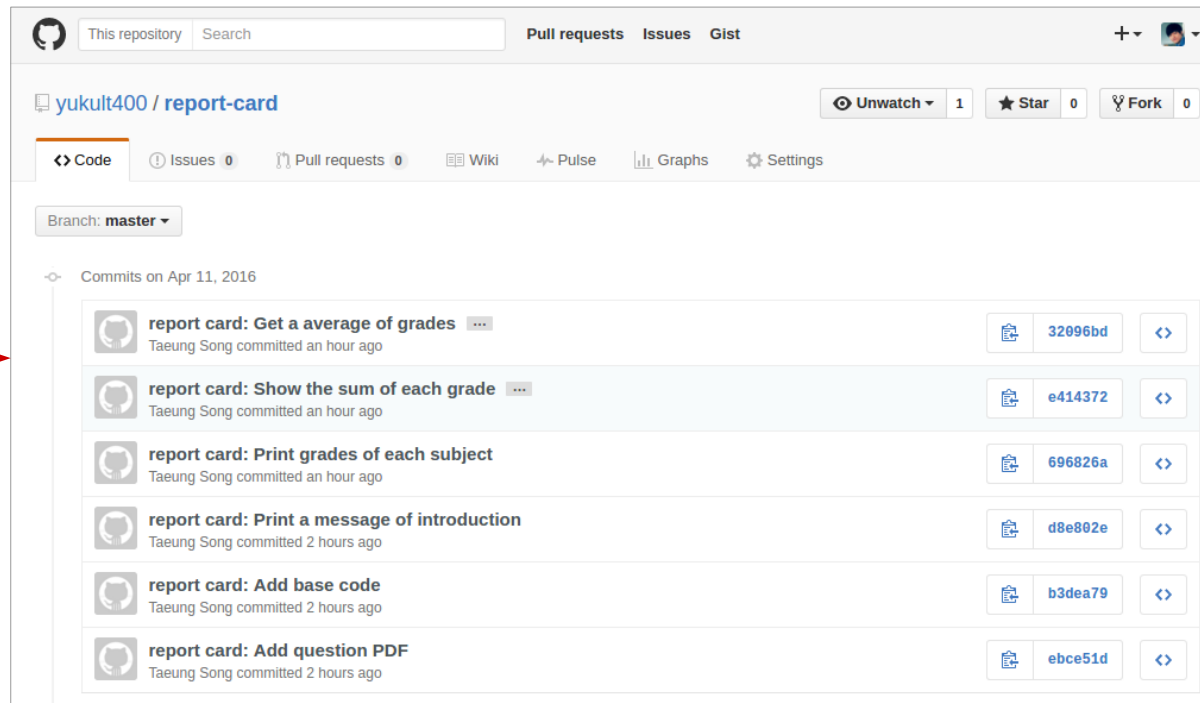


# Git 실습 준비하기

## 8. Stage 해결하기 위해서 수단과 방법을 가리지말자 !

본인의 Github 계정에 본인의 프로젝트에 commit 기록이 다음과 똑같아 지면 된다 .

<https://github.com/yukult400/report-card/commits/master>



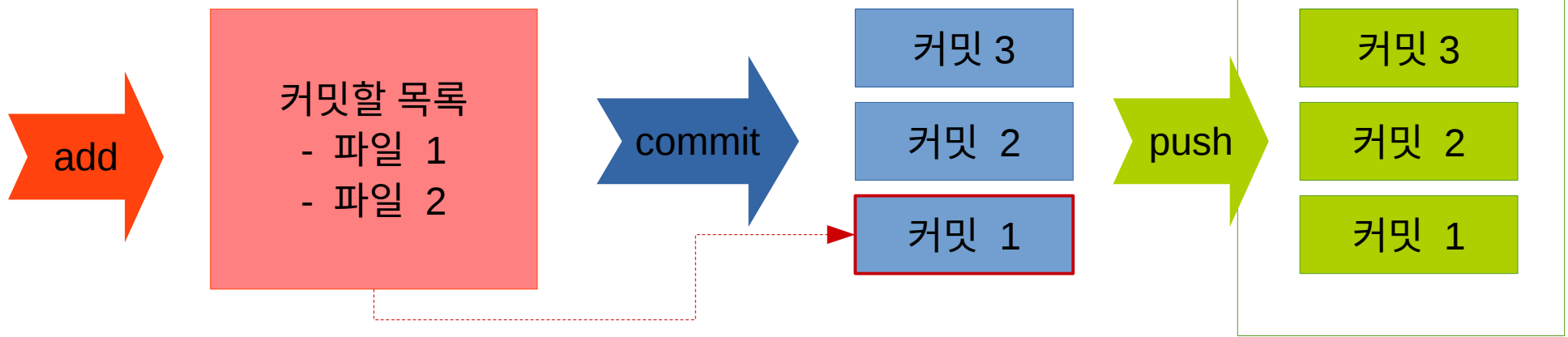
# Git 실습 준비하기

## < Git 필수 명령 >

**add** : 커밋할 **목록에 추가**

**commit** : 커밋 ( 히스토리의 **한단위** ) **만들기**

**push** : 현재까지 역사 (commits) Github 에 **밀어넣기**



# Git 실습 본격적인 실습에 앞서서 ..

---

- 본 강의는 실습 90% 이론 10% (스파르타)
- 나중에 하는건 없다 무조건 오늘목표는 이루고 가자
- 질문은 모두를 이롭게 한다
- 내 손으로 명령어를 직접 입력해보자
- 터미널 환경 경험하자
- GUI(Graphic User Interface) 대신 CLI(Command Line Interface) 경험하자
- vi 에디터를 사용해보자

# Git 실습 본격적인 실습에 앞서서 ..

Prompt, 프롬프트

Arguments, 인자 (1 개)

```
# git add report_card.pdf
```

Git 의 Sub-command 중 하나

Arguments, 인자 (2 개)

```
# git config --global user.name "Taeung Song"
```

-- 로 시작하면 보통은 Long Name 옵션

```
# git commit -s
```

- 로 시작하면 보통은 Short Name 옵션

# Git 실습 기본설정

---

1) Git-bash 혹은 터미널 실행 후에

2) 미리 캐치저장되어 있을지 모를 계정정보 삭제 (처음설치시 생략가능)

```
# git config --global --unset credential.helper
```

```
# git config --system --unset credential.helper
```

3) 나의 Github 계정 이메일 (Github 계정이메일) 과 이름 (본인 영문이름, Github 아이디 x) 을 적자

```
# git config user.email "본인메일적으세요@gmail.com"
```

```
# git config user.name "본인이름적으세요Taeung Song"
```

4) git-training-ex-v2.zip 압축파일 폴더 열기 (window 폴더탐색기로)

# Git 실습 Stage 1 초기화 및 첫 commit 하기 (Basic)

1) git-bash 또는 터미널 실행하고 HOME 경로 (~) 로 이동

```
# cd ~
```

2) report-card 폴더 만들고 window 폴더탐색기로도 열어두기

```
# mkdir report-card
```

3) 경로 이동 ( **pwd** 명령어로 현재경로 확인하기 )

```
# cd report-card
```

4) 해당 폴더를 git 초기화 ( **ls -A** 명령어로 생성된 .git 폴더 확인하기 )

```
# git init
```

5) 프로그램 문제 PDF 파일 추가 ( **커밋할 목록에 추가 add** ) (commit1 폴더내 파일 활용 )

```
# git add report_card.pdf
```

6) 첫 commit 하기 (역사 한단위 만들기 )

```
# git commit -m "report card: Add question PDF"
```

Git 상태확인 명령어

( 중간중간 치면서 수시로 확인하자 )

```
# git show  
# git log  
# git shortlog  
# git diff  
# git status
```

# Git 실습 Stage 1 초기화 및 첫 commit 하기 (Basic)

**몰라도 좋으니, 일단 따라해보자.**

7) 소스코드 추가하기 ( 커밋할 목록에 추가 add) (commit2 폴더내 파일 활용 )

```
# git add report_card.c
```

8) commit 하기 ( 역사 한단위 만들기 )

```
# git commit -m "report card: Add base code"
```

Git 상태확인 명령어

( 중간중간 치면서 수시로 확인하자 )

```
# git show  
# git log  
# git shortlog  
# git diff  
# git status
```

# Git 실습 Stage 2 diff 사용과 추가 commit 하기 (Basic)

**몰라도 좋으니, 일단 따라해보자.**

1) 상태를 확인한다

```
# git status
```

2) **commit3 폴더**에 있는 report\_card.c 소스 파일로 수정, 덮어쓰기 후 확인

```
# git diff
```

3) diff 를 통해서 변화분을 확인했다면 add 진행

```
# git add report_card.c
```

4) 준비된 소스파일을 commit 한다.

```
# git commit -m "report card: Print a message of introduction"
```

5) 지금까지한 3 개의 commit 들을 확인하자

```
# git log
```



# Git 실습 Stage 2 diff 사용과 추가 commit 하기 (Basic)

---

**몰라도 좋으니, 일단 따라해보자.**

6) **commit4** 폴더에 있는 report\_card.c 소스 파일로 수정, 덮어쓰기 후 확인

```
# git diff
```

7) diff 를 통해서 변화분을 확인했다면 add 진행

```
# git add report_card.c
```

8) 준비된 소스파일을 commit 한다.

```
# git commit -m "report card: Print grades of each subject"
```

9) 지금까지한 **4 개의 commit** 들을 **확인**하자

```
# git log
```

# Git 실습

## Stage 3 commit 에 본인서명 포함하기 (Basic)

---

1) **commit5** 폴더에 있는 report\_card.c 소스 파일로 수정 , 덮어쓰기 후 확인

```
# git diff
```

2) diff 를 통해서 변화분을 확인했다면 add 진행

```
# git add report_card.c
```

3) 서명과함께 commit 한다 . (-s 옵션으로 서명을 포함한다 .)

```
# git commit -sm "report card: Show the sum of each grade"
```

\* 서명의 의미는 본 오픈소스의 라이선스를 제대로 이해하고 작성한 commit 이라는 확인서명 ( 주로 리눅스커널에서 commit 을 공식적으로 만들때 많이 이용된다 .)

# Git 실습

## Stage 3 commit 에 본인서명 포함하기 (Basic)

---

\* 반복해서 추가 commit 만든다

4) **commit6** 폴더에 있는 report\_card.c 소스 파일로 수정 , 덮어쓰기 후 확인

```
# git diff
```

5) diff 를 통해서 변화분을 확인했다면 add 진행

```
# git add report_card.c
```

6) 서명과함께 commit 한다 .

```
# git commit -sm "report card: Get a average of grades"
```

# Git 실습

## Stage 4 지금까지의 commit 을 push 하자 (Basic)

---

**몰라도 좋으니, 일단 따라해보자.**

1) 상태를 확인하고 현재 브랜치명 master 를 확인하자

```
# git status
```

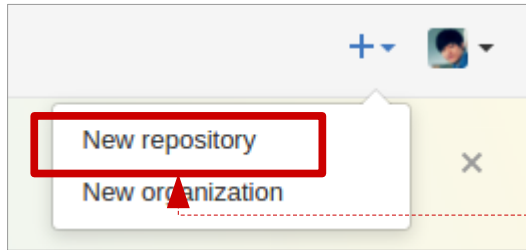
2) 지금까지한 commit 들을 확인하자 (6 개가 아니면 다시 확인하자)

```
# git shortlog
```

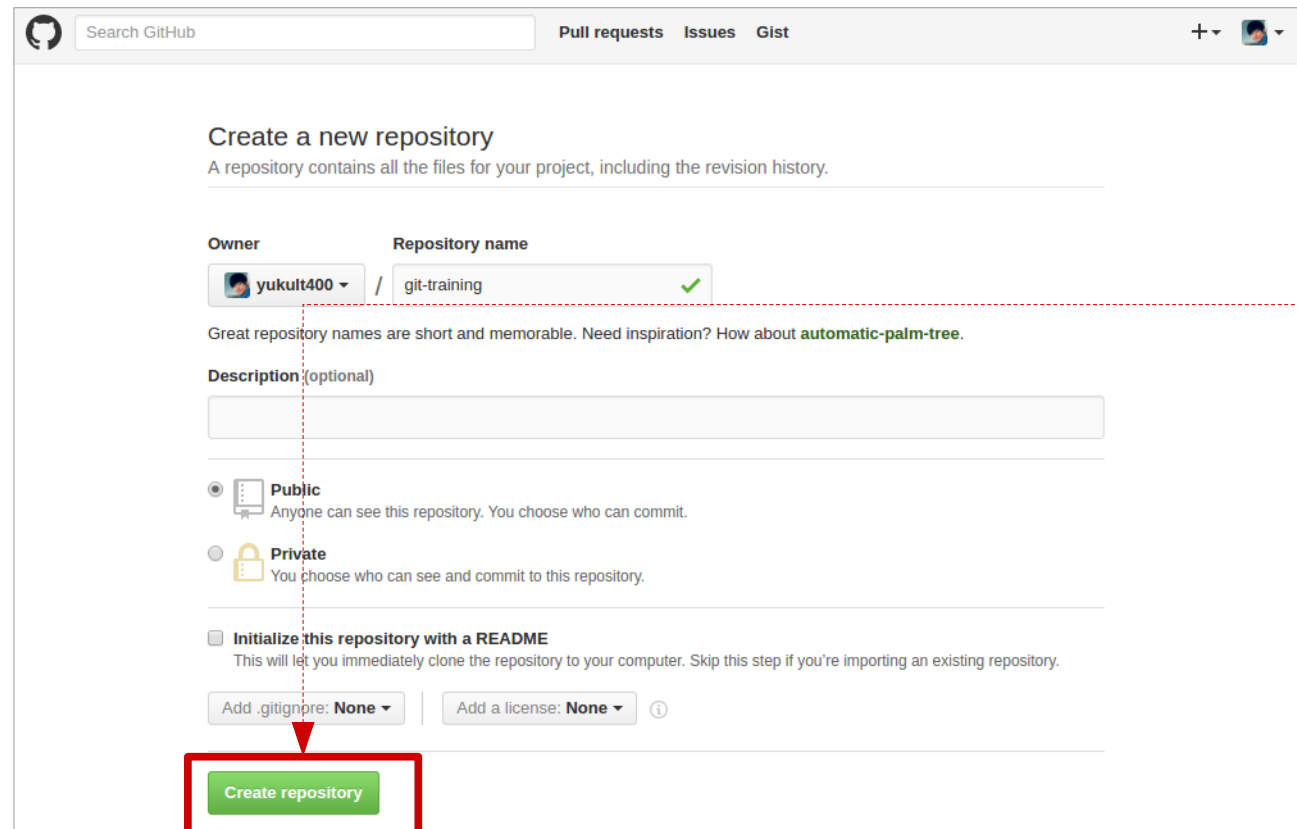
3) Github 원격저장소 URL 를 등록하자

**( 잠깐 멈추고 <http://github.com> 를 켜고 repository 새로 생성하자 )**

# Git 실습 Github 에서 원격저장소 만들기



새로운 원격 저장소를 생성하자 (클릭, 프로젝트명 자유)

A screenshot of the 'Create a new repository' form on GitHub. The form is titled 'Create a new repository' and includes a subtitle 'A repository contains all the files for your project, including the revision history.' The form has several fields and options: 'Owner' (yukult400), 'Repository name' (git-training), 'Description (optional)' (empty text box), 'Public' (selected radio button), 'Private' (unselected radio button), 'Initialize this repository with a README' (checkbox), 'Add .gitignore: None', and 'Add a license: None'. The 'Create repository' button at the bottom is highlighted with a red rectangular box. A red arrow points from the 'New repository' option in the previous screenshot to this button.

# Git 실습 Github 에서 원격저장소 만들기

해당 URL 복사해서 Stage5 의 4) 으로 이어서 진행

The screenshot shows the GitHub interface for the repository 'yukult400 / git-training'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch', 'Star', and 'Fork'. The 'Code' tab is selected, showing options for 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Quick setup' section is highlighted, showing the HTTPS URL 'https://github.com/yukult400/git-training.git' which is enclosed in a red box. Below this, there's a section for creating a new repository on the command line with a code block containing the following commands:

```
echo "# git-training" >> README.md
git init
git add README.md
```

# Git 실습

## Stage 4 지금까지의 commit 을 push 하자 (Basic)

---

4) 아까복사한 URL 로 Github 원격저장소 등록하자 ('<' 와 '>' 기호는 제외하고 입력한다)

```
# git remote add origin <아까복사한 URL>
```

5) Github 원격저장소 (origin) 에다가 밀어 넣자 .

```
# git push origin master
```

6) Github 웹페이지 열고 확인하자

# Git 실습 Stage 4 지금까지의 commit 을 push 하자 (Basic)

Github 들어가서 **본인** 프로젝트의 commit 기록 눌러보자

The screenshot shows the GitHub interface for the repository 'yukult400 / report-card'. A red dashed line with an arrow points from the text 'Github 들어가서 본인 프로젝트의 commit 기록 눌러보자' to the repository name 'yukult400 / report-card', which is highlighted with a red box. The repository page includes a header with the GitHub logo, a search bar, and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch', 'Star', and 'Fork'. The main content area shows the repository description 'No description or website provided. — Edit' and statistics: '6 commits', '1 branch', '0 releases', and '0 contributors'. There are also buttons for 'New pull request', 'New file', 'Upload files', 'Find file', 'HTTPS', and 'Download ZIP'. The commit history is displayed as a table with columns for the commit message, the file changed, and the time since the commit. The latest commit is 'Taeung Song report card: Get a average of grades' by 'yukult400', committed 2 hours ago. The commit history table is as follows:

Commit Message	File Changed	Time
Taeung Song report card: Get a average of grades	report_card.c	2 hours ago
report card: Get a average of grades	report_card.c	2 hours ago
report card: Add question PDF	report_card.pdf	2 hours ago

At the bottom, there is a prompt to 'Add a README' to help people understand the project.



# Git 실습 Stage 4 지금까지의 commit 을 push 하자 (Basic)

본인이 추가한 commit 들이 나오는걸 확인하자 ( 본인 Github)

The screenshot shows the GitHub interface for the repository 'yukult400 / report-card'. The 'Code' tab is selected, and the 'master' branch is chosen. A list of commits is displayed, with a red box highlighting the entire list and a red arrow pointing to the third commit.

Commit Message	Author	Time	SHA-1	Code Link
report card: Get a average of grades ...	Taeung Song	committed an hour ago	32096bd	<>
report card: Show the sum of each grade ...	Taeung Song	committed an hour ago	e414372	<>
report card: Print grades of each subject	Taeung Song	committed an hour ago	696826a	<>
report card: Print a message of introduction	Taeung Song	committed 2 hours ago	d8e802e	<>
report card: Add base code	Taeung Song	committed 2 hours ago	b3dea79	<>
report card: Add question PDF	Taeung Song	committed 2 hours ago	ebce51d	<>

# Git 실습 Stage 5 커밋 수정하기 (Basic)

- 1) report\_card.c 소스내의 'Mean' 변수명을 'Average' 로 바꾼다고 가정하자  
(commit6-1 폴더내 소스 활용 )

```
# git diff
```

- 2) diff 를 통해서 변화분을 확인했다면 add 진행

```
# git add report_card.c
```

- 3) 가장 위에 있는 commit 을 수정하자 (vi 에디터등 등록된 에디터 프로그램 켜지면 저장후 닫기 )

```
# git commit --amend
```

\* vi 에디터 또는 지정된 에디터 ( 메모장 등 ) 이 열릴수있다 .  
Commit 메시지를 수정하거나 수정없이 에디터를 닫으면 완료

\* vi 에디터는 i 또는 a 키를 눌러 수정모드로 변경하여 수정 후  
ESC 키 누르고 :wq 명령어를 입력하여 Enter 눌러 나올 수 있다 .

# Git 실습 Stage 5 커밋 수정하기 (remote 도 ) (Basic)

---

4) 바로 push 해보자 ( 충돌 오류발생 )

```
# git push origin master
```

5) 강제로 push 해서 수정하자 ( - -force 또는 -f 옵션 사용가능 )

```
# git push origin master - -force
```

6) 다시 Github 가서 제대로 변경되었는지 확인해보자

\* 4) 의 충돌이유는 Local( 본인 노트북 ,PC) 에 기록된 commit 들과 Github 에 먼저 push 하여 저장된 commit 들의 commit ID 가 일치하지 않는부분이 있기 때문이다 .

\* 물론 모든 commit ID 가 일치한상태에서 Local 에만 새로운 추가 commit 있을때는 push 가능

# Git 실습

## Stage 6 add 한거 취소하기 (Basic)

---

1) touch 로 빈파일 생성하고 add 하자 (; 로 명령어들을 연속적 실행가능 )

```
# touch test; git add test
```

2) 현재상태 확인해보고

```
# git status
```

3) reset 으로 add 한거 취소해보자

```
# git reset
```

4) 현재 상태 다시한번 확인해본다 .

```
# git status
```

# Git 실습 Stage 7 commit 한거 없애기 (Basic)

1) 아까 test 파일 여전히 존재하는지 확인 (지웠으면 다시만들기)

```
# git status
```

2) 임의로 실수의 commit 을 만들어 낸다 . (; 로 명령어들을 연속적 실행가능 )

```
# git add test; git commit -sm "test"
```

3) 그리고 push 까지해서 Github 에 있는 tree 까지 실수 commit 을 넣어버린다 .

```
# git push origin master
```

4) 그리고 가장 최근 commit 을 지운다 .

```
# git reset HEAD~1
```

5) 강제로 Github 에 있는 tree 도 밀어넣어서 수정한다 .

```
# git push origin master - -force
```

\* Github 에 있는 commit 을 수정할 길은 Local 에서 수정후 -f 옵션으로 push 하는방법뿐이다 .

# Git 자유실습 반복연습 or Keep going

---



# Git 자유실습 반복연습 or Keep going

---

다시 Stage 1 부터 새롭게 반복연습 하거나

조금은 어려울 수 있으나 고급과정을 계속해서 진행 하거나

# Git 실습

## Github 의 기능 pull-request 실습

---

Pull-request 한다는 의미

다른프로젝트에 내가 만든 commit 을 제출한다는 의미 (실제 전송단위는 branch)

상대방 프로젝트를 fork( 복사 ) 해서

내 계정에서 관리되는 프로젝트로 새롭게 만들어 두고

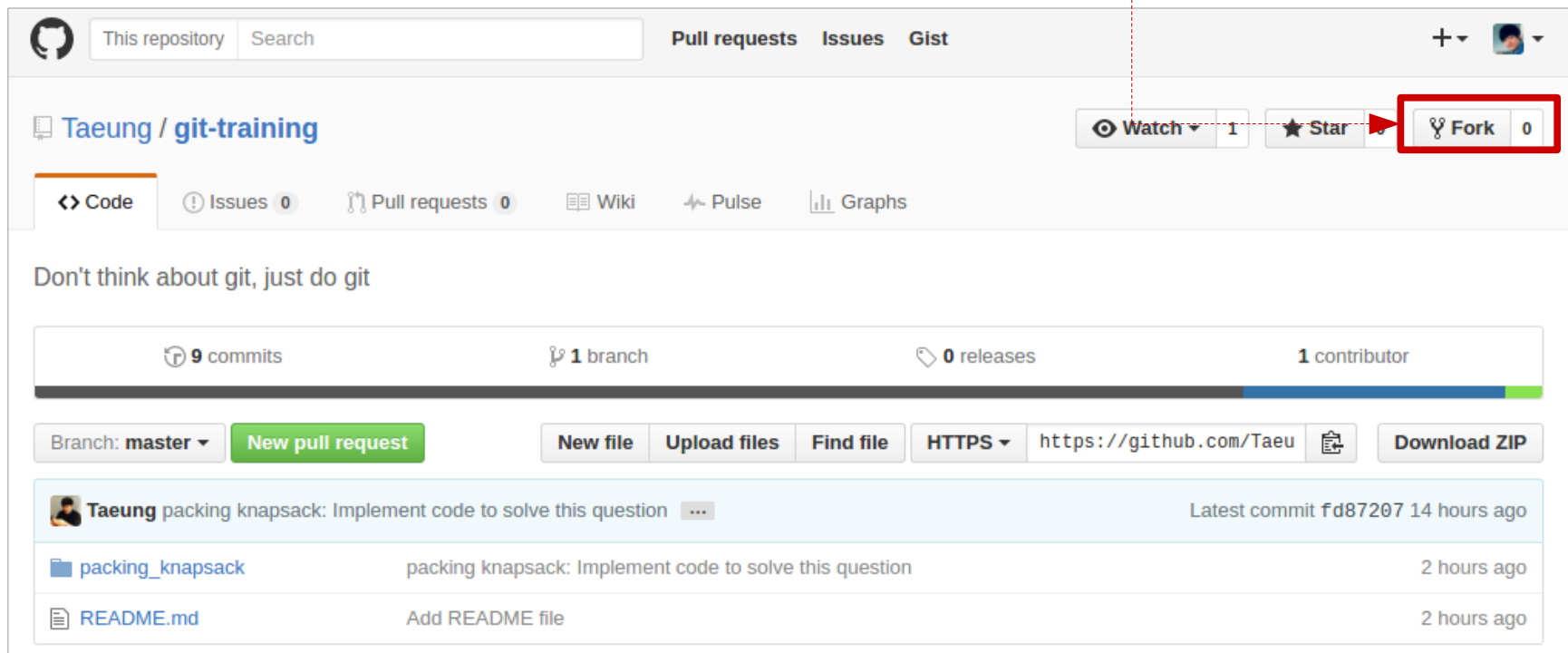
그 fork 한 github 프로젝트를 토대로 새로운 commit 내용들을 pull-regeust 제출 할 수 있다 .



# Git 실습 Github 에서 fork 하기

\* 주의 fork 는 본인 프로젝트를 대상으로 하는게 아니다 .

<http://github.com/taeung/git-training> 가서 fork 버튼 누르자



# Git 실습 Github 에서 fork 하기

fork 가 되면 내 원격저장소가 추가 된다

The screenshot shows the GitHub repository page for 'yukult400 / git-training', which is a fork of 'Taeung/git-training'. The repository has 12 commits, 3 branches, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted with a red box. A dropdown menu is open, showing the 'Clone with HTTPS' option, which provides the URL 'https://github.com/yukult400/git-training'. A red arrow points from the 'Clone or download' button to the 'Clone with HTTPS' option. Another red arrow points from the 'Fork' button to the repository name 'yukult400 / git-training'.

Clone or download 초록색 버튼 클릭 !!  
fork 해서 만들어진 **본인** repo 의 URL 복사  
\* 주의 ) Taeung/git-training 원본프로젝트 URL 사용하면 안됨

필수 ) forked from Taeung/git-training 표시 확인하기

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

1) (git-bash/ 터미널에서 ) 최초경로 HOME 경로로 이동하자 . (report-card 작업하던 폴더에서 벗어나기 )

```
# cd ~
```

2) clone 으로 fork 한 repo 받아오기 ('<' 와 '>' 두 기호는 생략하고 적는다 )

```
# git clone <아까 fork 한 repo 에서 복사한 URL>
```

3) clone 한 프로젝트 폴더로 이동하기 ( 만약 프로젝트명이 git-training-1 이면 그 이름으로 이동 )

```
# cd git-training
```

4) pull-request 작업할 브랜치 (develop) 따로 만들기

```
# git checkout -b develop
```

\* 브랜치 생성이란  
간단한 비유로 설명하면

“ 같은 폴더에 또다른 세상열기”

5) pull\_request\_test 폴더로 이동해서 \*pull\_request\_test 가 아닌 pull\_reqeust\_test 오타이지만 그대로 진행

```
# cd pull_reqeust_test
```

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

---

6) 내 이름으로된 (taeung 대신 ) 폴더 만들고

```
# mkdir taeung; cd taeung
```

7) 내이름으로된 (taeung 대신 ) 폴더에 작업하던 repot\_card.c 소스파일 또는 아무파일 복사해서 넣기

8) 추가한 폴더 ( 내가작업한 소스 내용 ) 통째로 add ('<' 와 '>' 은 생략한다 .)

```
# git add <나의 소스작업폴더>
```

9) 준비된 파일들 commit

```
# git commit -sm "test pull request"
```

10) 내가 fork 한 repo 의 develop 브랜치로 push ( 주의 : master 아님 )

```
# git push origin develop
```

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

\* 나의 프로필에서 fork 해서 만들어진 프로젝트 페이지로 이동

The screenshot shows the GitHub interface for a repository named 'yukult400 / git-training-1', which is a fork of 'Taeung/git-training'. The repository has 9 commits, 4 branches (highlighted with a red box and an arrow), 0 releases, and 1 contributor. The 'Code' tab is selected, showing the 'master' branch. Below the repository information, there is a section for the 'master' branch, indicating it is 9 commits ahead and 12 commits behind the upstream 'Taeung:master'. A list of files is shown, including 'packing\_knapsack' and 'README.md', with their commit messages and dates.

방법 1-1) 방금 Push 했던 브랜치를 확인하기 위해서 Branch 탭 클릭

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

The screenshot shows the GitHub interface for the repository 'yukult400 / git-training-1', which is a fork of 'Taeung/git-training'. The top navigation bar includes 'Pull requests', 'Issues', and 'Gist'. The repository has 1 Unwatch, 0 Stars, and 66 Forks. The 'Code' tab is selected, and the 'Overview' sub-tab is active. The 'Default branch' is 'master', updated 4 months ago by Taeung. The 'Your branches' section shows the 'develop' branch, updated 3 months ago by Taeung. A red box highlights the 'New pull request' button, which is located next to the 'develop' branch. A red arrow points from a text box below to this button.

방법 1-2) Pull-request 하려는 브랜치에서 New pull-request 버튼을 클릭

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

The screenshot shows the GitHub interface for a repository named 'yukult400 / git-training-1', which is a fork of 'Taeung/git-training'. The repository has 9 commits, 2 branches, 0 releases, and 1 contributor. The 'Compare & pull request' button is highlighted with a red box. Below the button, there is a table showing the commit history of the 'knapsack-v2' branch, including the latest commit 'fd87207' and two other commits, all dated 14 hours ago.

Commit Hash	Commit Message	Time Ago
fd87207	Latest commit	14 hours ago
		3 hours ago
		3 hours ago

방법 2) Github 들어가서 fork 한 저장소에서 pull-request 버튼 누르기  
(아니면 pull-requests 탭에서 Create pull-request 누르기)

# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

This repository Search Pull requests Issues Gist

Taeung / git-training Watch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Wiki Pulse Graphs

## 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 fork: Taeung/git-training base: master ... head fork: yukult400/git-training-1 compare: knapsack

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

test git-pull-request

Write Preview

Signed-off-by: Taeung Song <treeze.taeung@gmail.com>

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported

Create pull request

Pull-request 보내기



# Git 자유실습 Stage 8 pull-request 하기 (Advanced)

The screenshot shows a GitHub pull request page for the repository 'Taeung / git-training'. The page title is 'test git-pull-request #1'. The pull request is from 'yukult400:knapsack' to 'Taeung:master'. The page shows a comment from 'yukult400' signed off by 'Taeung Song treeze.taeung@gmail.com'. A green checkmark indicates 'This branch has no conflicts with the base branch'. The page also shows a comment box and a 'Write' button. The repository has 1 watch, 0 stars, and 1 fork. The pull request has 1 commit and 4 files changed. The page also shows a 'Labels' section with 'None yet', a 'Milestone' section with 'No milestone', an 'Assignee' section with 'No one assigned', and a 'Notifications' section with an 'Unsubscribe' button. The page also shows '2 participants'.

Taeung/git-training 페이지 ( 본래 프로젝트 ) 에서 만들어진 pull-request 확인하기

\* 주의 ) 나의 프로필에서 fork 해서 만들어진 프로젝트 페이지에서 확인 하는게 아님

# Git 자유실습

## Stage 9 merge 로 2 개 브랜치 합치기 (Advanced)

1) 방금작업한 develop 브랜치가 현재 브랜치인지 확인하자 (status 로도 확인가능 )

```
# git branch
```

2) 추가 브랜치 만들어보자

```
# git checkout -b test
```

3) touch 로 빈파일하나 만들어서 commit 만들어보자 (; 로 명령어들을 연속적 실행가능 )

```
# touch test; git add test; git commit -sm "test"
```

4) 현재브랜치 (develop) 을 기준으로 추가브랜치 (test) 을 합치자

```
# git checkout develop; git status; git merge test
```

# Git 자유실습 Stage 10 rebase 하기 (Advanced)

---

Rebase 사용하는 시나리오 !!

commit 을 역사의 한단위 '블럭' 이라 하고 블럭들의 모임을 'tree' 라 할때

내가 쌓은 블럭을 잠시 빼고

(뺀 나머지) 기준이 되는 tree 를 최신 업데이트 한 후에

그 위에 다시 내 블럭을 쌓아 올릴때 쓸수있다 .

# Git 자유실습 Stage 10 rebase 하기 (Advanced)

---

- 1) Chapter 4. 의 Stage 8. 에서 fork 후 clone 했던 프로젝트경로로 이동하여 upstream 을 추가하자 \* 주의 ) 본인프로젝트 URL 이 아니다

```
# git remote add upstream https://github.com/Taeung/git-training.git
```

- 2) upstream 의 dev 브랜치를 가져오자

```
# git fetch upstream dev
```

- 3) 현재 내 브랜치가 develop 인지 확인하자

```
# git status
```

- 4) rebase 하자

```
# git rebase upstream/dev
```

# Git 자유실습 Stage 11 중간에 낀 commit 수정하기 (Advanced)

고난이도 기능 중 하나 **rebase -interactive**

1) commit 최초기록 부터 2 번째 commit 을 수정해보자

```
# git rebase -i --root
```

2) vi 에디터가 열리면 수정하려는 commit(2 번째 ) 맨앞에  
“pick” 을 지우고 대신에 “edit” 을 적고 저장하고 에디터를 끄자

\* --root 를 사용한 이유는 최초의 commit 까지 수정할수있도록 하기위함

3) 상태확인해서 rebase 진행 정상적인지 보고

```
# git status
```

4) commit 정보 수정 (“packing knapsack:” commit 메시지에 추가 ) 하고 --continue 로 마무리

```
# git commit --amend -sm “packing knapsack: Add knapsack problem PDF”
```

```
# git rebase --continue
```

# Git 자유실습

Bonus Stage blame 으로 추적하기 (Advanced)

---

1) 어떤 파일이든 누가 어느라인을 수정했는지 파악해보자

```
# git blame report_card.c
```

2) 해당 commit ID 를이용하여 그 당시 commit 정보확인

```
# git show <commit ID>
```

# Git 이해하기 (Why, What)

---



# Git 을 쓰는 진짜이유

---

협업 때문에 Git 을 쓴다 . ( 집단지성의 극대화 )

현대적인 , 선진화된 소스코드 개발과정의 필수도구로 Git 을 쓴다 (Needs)

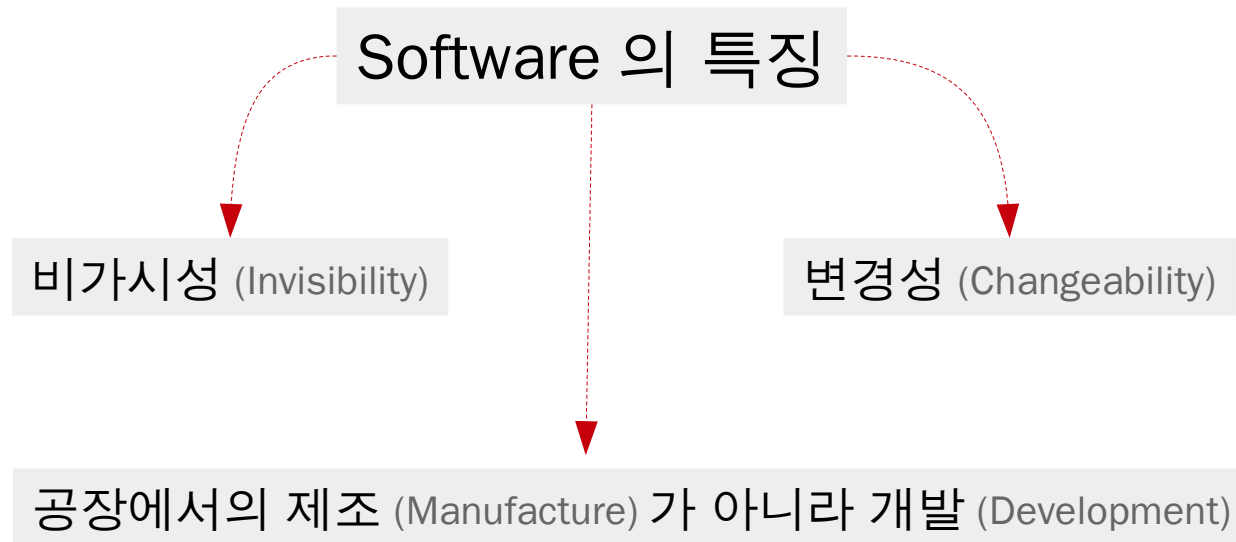


<https://geektimes.ru>



# VCS(Git, svn, etc) 을 쓰는 배경

---



# Commit 단위개발 Idea

---

Commit 단위로 코딩하고 리뷰하고 토론하고 적용한다. (집단지성의 극대화)

소프트웨어의 취약점을 극복하는 전략



# Commit 단위개발의 정신

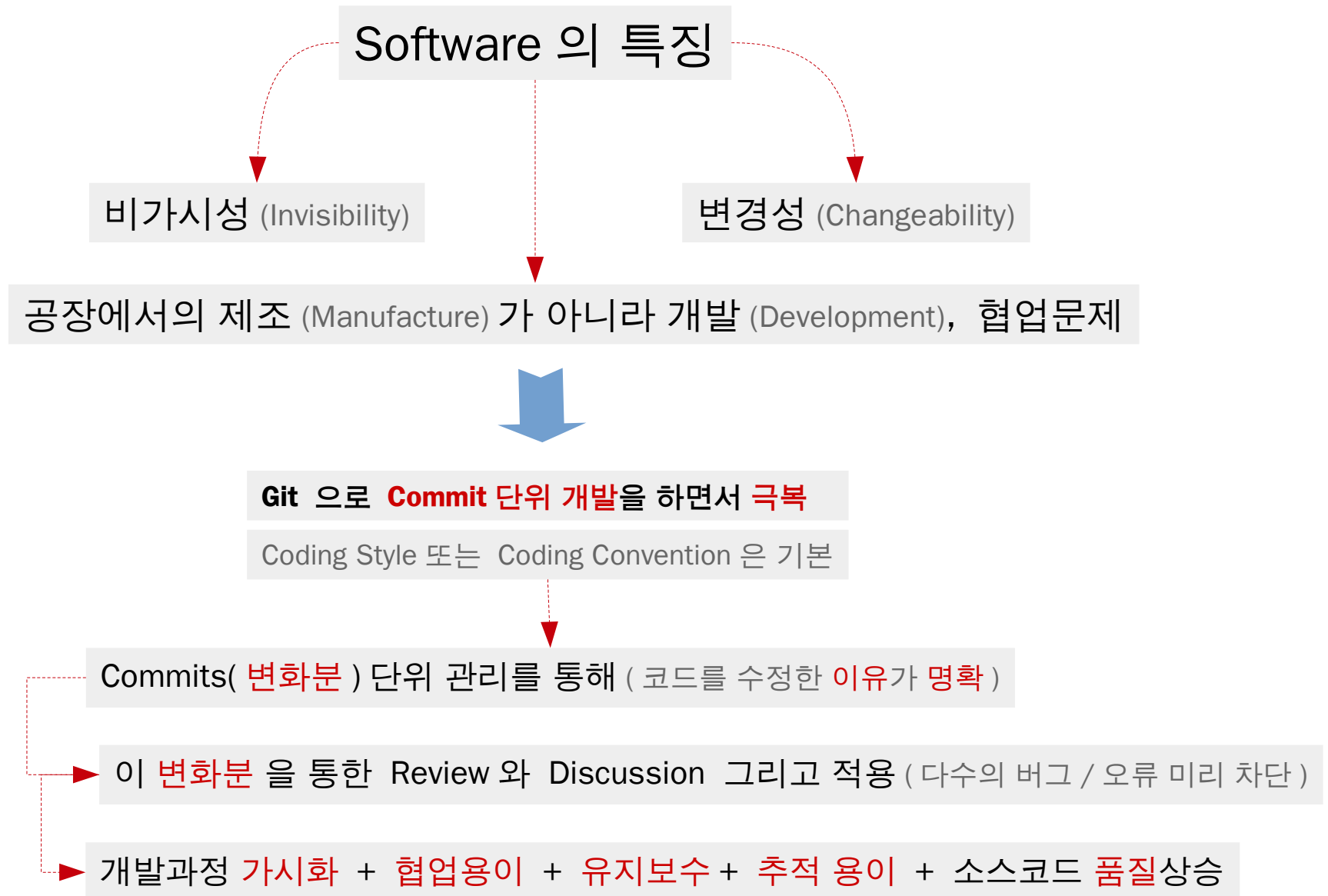
---

혼자가면 빨리가지만

함께가면 멀리간다



# Git 을 통한 현명한 협업 ( 집단지성 극대화 )



# Commit 단위 개발 (with Github)

\* 참고 : Linux kernel 은 Github 에서 mirror 까지는 되지만 다음 Repository 가 공식  
<http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/>

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유가 명확** )

torvalds / linux

Watch 4,576 Unstar 30,768 Fork 12,172

Code Pull requests 96 Pulse Graphs

Linux kernel source tree

588,144 commits 1 branch 457 releases ∞ contributors

Branch: master New pull request New file Upload files Find file HTTPS https://github.com/torvalds/linux Download ZIP

torvalds Merge tag 'asm-generic-4.6' of git://git.kernel.org/pub/scm/linux/kernel... Latest commit 11caf57 9 hours ago













Failed to load latest commit information.

Documentation	Merge tag 'pm+acpi-4.6-rc1-2' of git://git.kernel.org/pub/scm/linux/kernel...	9 hours ago
arch	Merge tag 'asm-generic-4.6' of git://git.kernel.org/pub/scm/linux/kernel...	9 hours ago
block	Merge branch 'for-linus' of git://git.kernel.dk/linux-block	12 hours ago
certs	certs: Fix misaligned data in extra certificate list	25 days ago
crypto	Merge branch 'akpm' (patches from Andrew)	7 days ago
drivers	Merge tag 'asm-generic-4.6' of git://git.kernel.org/pub/scm/linux/kernel...	9 hours ago
firmware	WHENCE: use https://linuxtv.org for LinuxTV URLs	4 months ago

<https://github.com/torvalds/linux>

# Commit 단위 개발 (with Github)

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유**가 **명확** )

	<b>Merge tag 'for-v4.5-rc' of git://git.kernel.org/pub/scm/linux/kernel/...</b> ... torvalds committed 29 days ago		1ebe383	
	<b>perf script: Remove duplicated code and needless script_spec__findnew()</b> ... Taeung committed with Arnaldo Carvalho de Melo 29 days ago		8560bae	
	<b>x86/mm: Fix slow_virt_to_phys() for X86_PAE again</b> ... dcui committed with Thomas Gleixner 29 days ago		bf70e55	
	<b>perf script: Exception handling when the print fmt is empty</b> ... Taeung committed with Arnaldo Carvalho de Melo 29 days ago		8579aca	




Commits on Feb 25, 2016

<https://github.com/torvalds/linux>

# Commit 단위 개발 (with Github)

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유**가 **명확** )



**Merge tag 'for-v4.5-rc' of git://git.kernel.org/pub/scm/linux/kernel/...** **1e3e383** **<>**  
torvalds committed 29 days ago

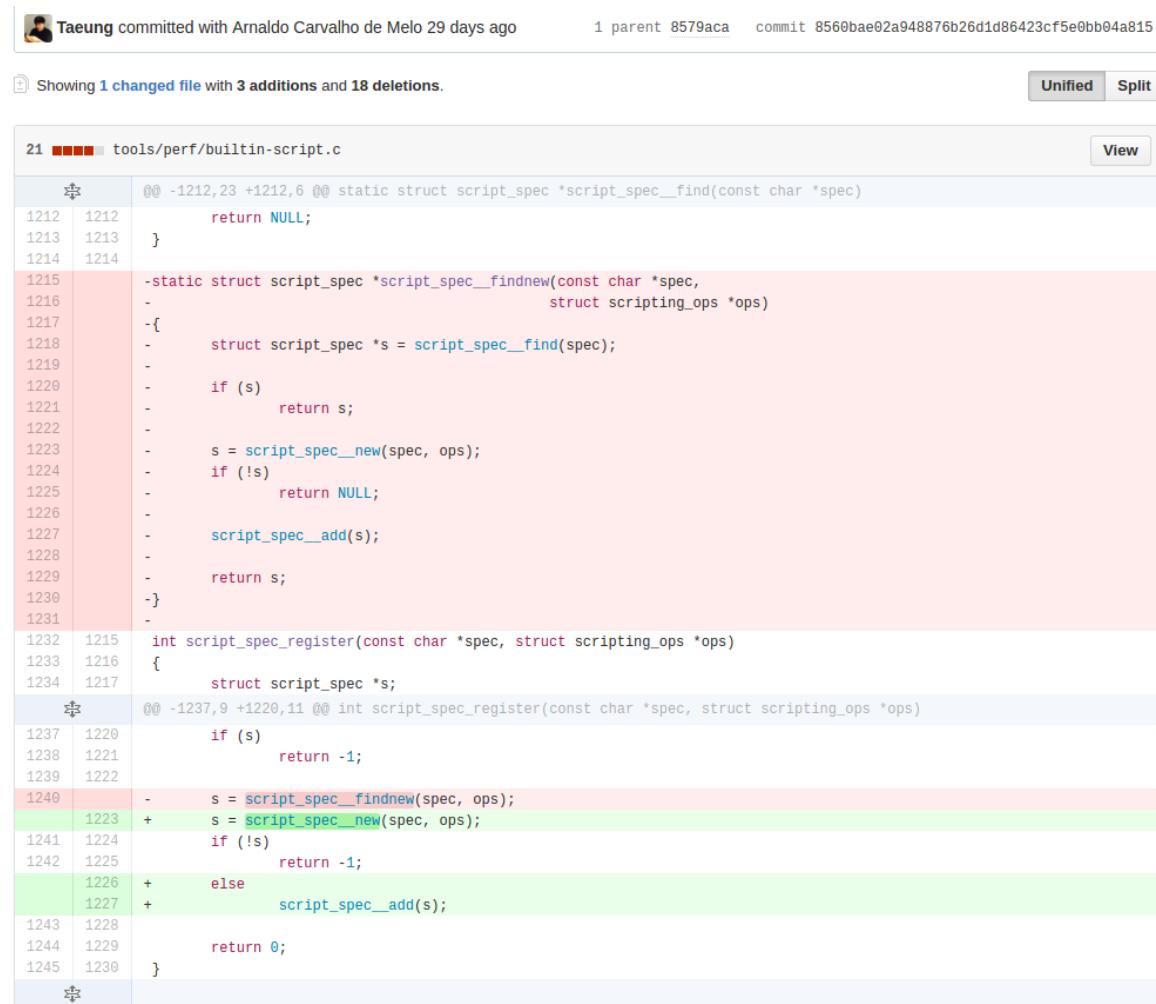
**perf script: Remove duplicated code and needless script\_spec\_\_findnew()** **8560bae** **<>**  
Taeung committed with Arnaldo Carvalho de Melo 29 days ago  
script\_spec\_register() called two functions: script\_spec\_find() and script\_spec\_\_findnew(). But this way script\_spec\_find() gets called two times, directly and via script\_spec\_\_findnew().  
  
So remove script\_spec\_\_findnew() and make script\_spec\_register() only call once script\_spec\_find().  
  
Signed-off-by: Taeung Song <treeze.taeung@gmail.com>  
Acked-by: Jiri Olsa <jolsa@kernel.org>  
Cc: Namhyung Kim <namhyung@kernel.org>  
Link: <http://lkml.kernel.org/r/1456413190-12378-1-git-send-email-treeze.taeung@gmail.com>  
Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

**x86/mm: Fix slow\_virt\_to\_phys() for X86\_PAE again** **bf70e55** **<>**  
dcui committed with Thomas Gleixner 29 days ago

**perf script: Exception handling when the print fmt is empty** **8579aca** **<>**  
Taeung committed with Arnaldo Carvalho de Melo 29 days ago

# Commit 단위 개발 (with Github)

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유가 명확** )



Taeung committed with Arnaldo Carvalho de Melo 29 days ago 1 parent 8579aca commit 8560bae02a948876b26d1d86423cf5e0bb04a815

Showing 1 changed file with 3 additions and 18 deletions. Unified Split

21 tools/perf/builtin-script.c View

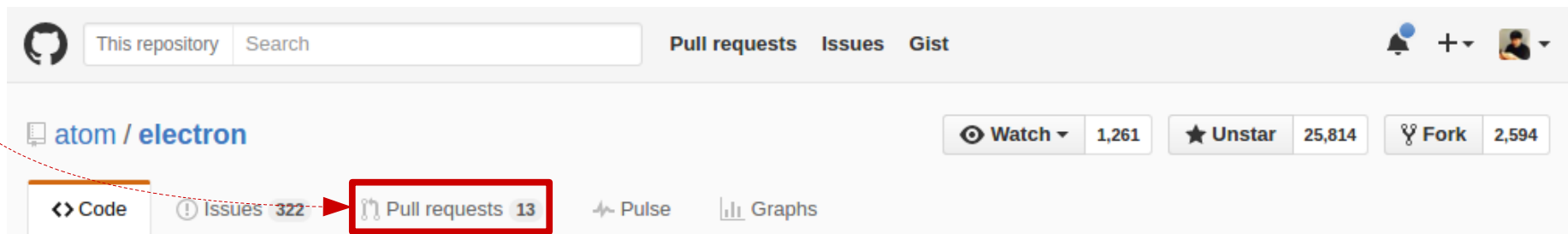
```
@@ -1212,23 +1212,6 @@ static struct script_spec *script_spec__find(const char *spec)
1212 1212     return NULL;
1213 1213 }
1214 1214
1215 -static struct script_spec *script_spec__findnew(const char *spec,
1216 - struct scripting_ops *ops)
1217 -{
1218     struct script_spec *s = script_spec__find(spec);
1219     if (s)
1220         return s;
1221     s = script_spec__new(spec, ops);
1222     if (!s)
1223         return NULL;
1224     script_spec__add(s);
1225     return s;
1226 -}
1227 -
1228 int script_spec_register(const char *spec, struct scripting_ops *ops)
1229 {
1230     struct script_spec *s;
1231     @@ -1237,9 +1220,11 @@ int script_spec_register(const char *spec, struct scripting_ops *ops)
1237 1220     if (s)
1238 1221         return -1;
1239 1222     s = script_spec__findnew(spec, ops);
1240 - s = script_spec__new(spec, ops);
1241 + s = script_spec__new(spec, ops);
1242     if (!s)
1243         return -1;
1244 + else
1245 +     script_spec__add(s);
1246     return 0;
1247 }
```

<https://github.com/torvalds/linux>



# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 (다수의 버그 / 오류 미리 차단)



atom / electron

Watch 1,261 Unstar 25,814 Fork 2,594

Code Issues 322 **Pull requests 13** Pulse Graphs

Build cross platform desktop apps with web technologies <http://electron.atom.io>

7,832 commits 4 branches 216 releases 331 contributors











Branch: master New pull request New file Upload files Find file HTTPS <https://github.com/atom> Download ZIP

kevin sawicki Merge pull request #4883 from atom/repl Latest commit 1993139 12 hours ago

atom	Set _setDeprecatedOptionsCheck on exports	7 days ago
chromium_src	devtools: fix filesystem api usage and use prefs to track filesystem ...	11 days ago
default_app	Don't quit on window-all-closed when in repl mode	16 hours ago
docs-translations	fix link	6 days ago
...	...	...

# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 (다수의 버그 / 오류 미리 차단)

#4892 opened 2 days ago by deepak1556	
 <b>Improve error reporting when passing invalid argument types for dialog API methods</b> ✓	 0
#4887 opened 2 days ago by sergeybekrin	
 <b>Added electron-release-server link to the docs</b> ✓	 2
#4885 opened 3 days ago by ArekSredzki	
 <b>Run callback of setDestructor immediately when GC happens</b> ✓	 3
#4869 opened 4 days ago by zcbenz	
 <b>Fix alert()</b> ✓	 5
#4861 opened 5 days ago by OctoHuman	
 <b>Docs: Update Korean docs as upstream</b> <span>documentation</span>	 0
#4854 opened 6 days ago by preco21	

# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 ( 다수의 버그 / 오류 미리 차단 )

atom / electron

Watch 1,261 Unstar 25,814 Fork 2,594

Code Issues 322 Pull requests 13 Pulse Graphs

## Fix alert() #4861

Open OctoHuman wants to merge 2 commits into atom:master from OctoHuman:master

Conversation 5 Commits 2 Files changed 1 +4 -1

**OctoHuman** commented 5 days ago

Stop `alert()` from throwing an error when passing no parameters or when passing null or undefined as the message. For example if you run `alert()` in a browser it will display a blank dialog box, but in Electron it will throw a error because there is no `.toString()` method on `undefined`. Changing `alert()` like this will also make it more consistent with the major browsers implementations.

**OctoHuman** added some commits 5 days ago

- Fix alert() 9f65412
- Refix alert() 116d611

**DerNivel** commented on the diff 3 days ago

```
lib/renderer/override.js
... -161,11 +161,14 @@ window.open = function(url, frameName, features) {
161 161 // Use the dialog API to implement alert().
162 162 window.alert = function(message, title) {
```

**DerNivel** added a note 3 days ago

Labels: None yet

Milestone: No milestone

Assignee: No one assigned

Notifications: [Subscribe](#)

You're not receiving notifications from this thread.


3 participants


# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 (다수의 버그 / 오류 미리 차단)


Showing changes from 1 commit ▾ 1 changed file ▾ +1 -1 Diff options ▾

**Refix alert()** < Prev Next >

 OctoHuman committed 5 days ago commit 116d61185a887b50c2a214c0736e07dd6b806868


2 lib/renderer/override.js Show notes View 

```
@@ -161,7 +161,7 @@ window.open = function(url, frameName, features) {  
161 161 // Use the dialog API to implement alert().  
162 162 window.alert = function(message, title) {
```


 DerNivel added a note 3 days ago +😊

Would it be possible to use default parameters. For example:


```
window.alert = function(message = '', title = '') {  
  // ...  
}
```

 OctoHuman added a note 3 days ago +😊


The problem with your code is if you passed `undefined` it still would be replaced by the default. So in your example passing `undefined` as the message would give me a blank `alert()` dialog. In a browser `alert()` will give me a blank dialog, but `alert(undefined)` will give me a dialog that says `undefined`

 DerNivel added a note 3 days ago +😊

I think no one would pass `undefined` to an `alert()` dialog.

 hermannplatz added a note 3 days ago +😊

you never know :)

 OctoHuman added a note 2 days ago +😊

# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 (다수의 버그 / 오류 미리 차단)

atom / electron

Watch 1,261 Unstar 25,814 Fork 2,594

Code Issues 322 Pull requests 13 Pulse Graphs

## Fix alert() #4861

Open OctoHuman wants to merge 2 commits into atom:master from OctoHuman:master

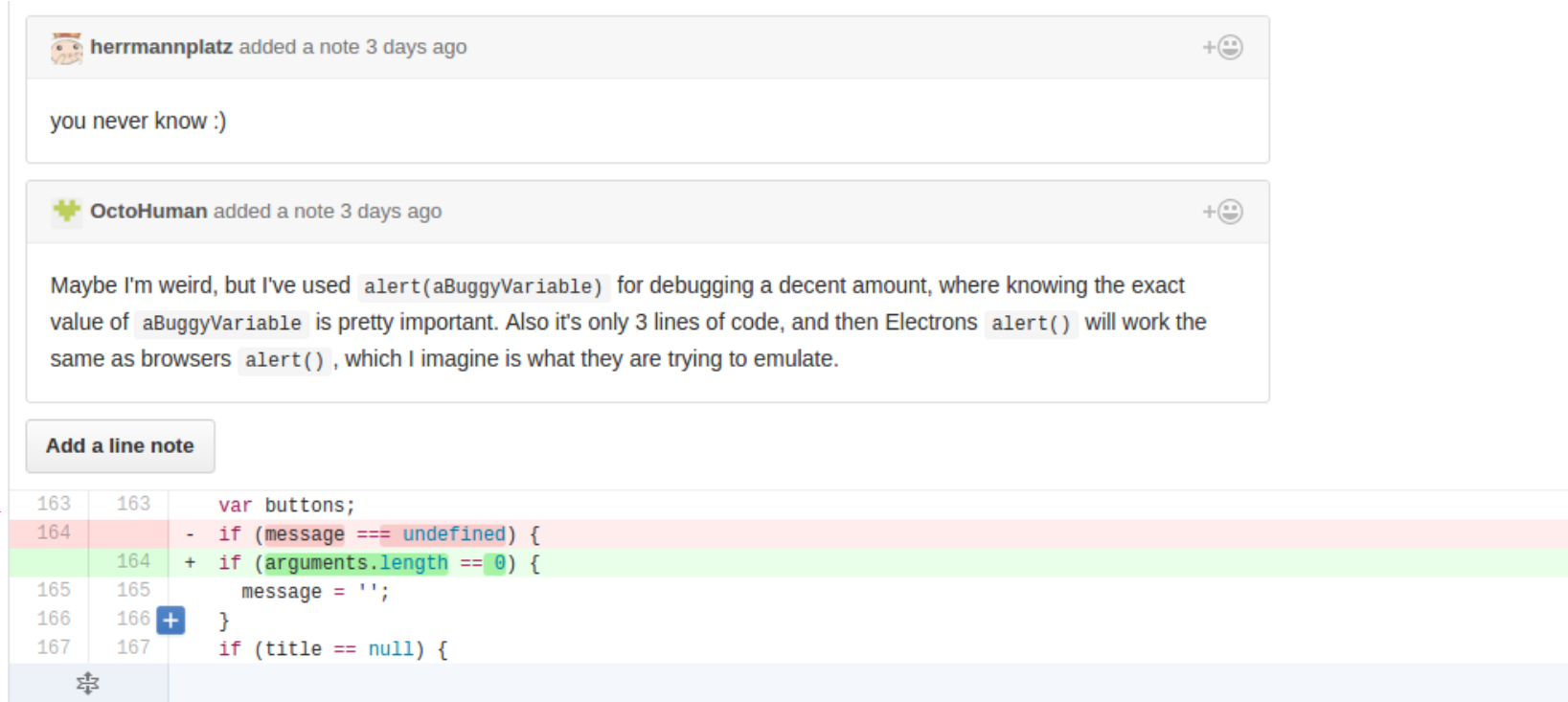
Conversation 5 Commits 2 Files changed 1 +4 -1

Commits on Mar 20, 2016

	<b>Fix alert()</b> OctoHuman committed 5 days ago ❌	9f65412
	<b>Refix alert()</b> OctoHuman committed 5 days ago ✅	116d611

# Commit 단위 Review (with Github)

이 **변화분** (commits) 을 통한 Review 와 Discussion 그리고 적용 ( 다수의 버그 / 오류 미리 차단 )



The screenshot displays a GitHub pull request interface. At the top, two review notes are visible:

- hermannplatz** added a note 3 days ago: "you never know :)"
- OctoHuman** added a note 3 days ago: "Maybe I'm weird, but I've used `alert(aBuggyVariable)` for debugging a decent amount, where knowing the exact value of `aBuggyVariable` is pretty important. Also it's only 3 lines of code, and then Electrons `alert()` will work the same as browsers `alert()`, which I imagine is what they are trying to emulate."

Below the notes is a button labeled "Add a line note". The bottom section shows a code diff with line numbers 163 to 167. A red arrow points from the text "이 변화분 (commits) 을 통한 Review" to the diff area.

Line	Change	Code
163		<code>var buttons;</code>
164	-	<code>if (message === undefined) {</code>
164	+	<code>if (arguments.length == 0) {</code>
165		<code>message = '';</code>
166	+	<code>}</code>
167		<code>if (title == null) {</code>

# Git 의 가치

---

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유**가 **명확** )

▶ 개발과정 **가시화** + **협업용이** + **유지보수** + **추적 용이** + 소스코드 **품질상승**



Git 을 활용하는 **오픈소스**가 대표적인 **증명** (IT 기업들의 **높은 의존** , 소프트웨어 **역사**를 이끄는 )

▶ 카카오 , NHN 엔터테인먼트 등 SW 기업들의 **pull-request** 방식도입

# Git 의 가치

---

Junior 개발자 교육 효과 ( 프로개발자의 코드를 볼수 있는 기회 )

Commits( **변화분** ) 단위 관리를 통해 ( 코드를 수정한 **이유**가 **명확** )

개발과정 **가시화** + **협업용이** + **유지보수** + **추적 용이** + 소스코드 **품질상승**



Git 을 활용하는 **오픈소스**가 대표적인 **증명** (IT 기업들의 **높은 의존** , 소프트웨어 **역사**를 이끄는 )



# Github 란

---



Git 이라는 도구를 응용한 사이트

각종 Remote repository ( 원격저장소 ) 들의 집합소

# Opensource Github / Not Github

---

## Not Github

- Apache (<http://git.apache.org>)
- Linux kernel (<http://git.kernel.org>)
- GNU (<http://git.savannah.gnu.org/cgit/>)
- Webkit (<git://git.webkit.org/WebKit.git>)

...


## Github

- Node.js (<https://github.com/nodejs/node>)
- Angular.js (<https://github.com/angular/angular.js>)
- Spring-boot (<https://github.com/spring-projects/spring-boot>)
- Rust (<https://github.com/rust-lang/rust>)
- Redis (<https://github.com/antirez/redis>)
- Flask (<https://github.com/mitsuhiko/flask>)

...

# Not Github

Gitub 가 아닌 remote repo 관리 사이트

 <b>Kernel.org git repositories</b> Git repositories hosted at kernel.org				
<a href="#">index</a> <input type="text"/> <input type="button" value="search"/>				
Name	Description	Owner	Idle	Links
<i>bluetooth</i>				
<a href="#">bluez-hcidump.git</a>	Bluetooth packet analyzer	Marcel Holtmann	3 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">bluez.git</a>	Bluetooth protocol stack for Linux	Marcel Holtmann	26 hours	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">obexd.git</a>	OBEX Server	Marcel Holtmann	3 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">sbc.git</a>	SBC library	holtmann	17 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<i>boot</i>				
<a href="#">dracut/dracut.git</a>	dracut - Initramfs generator using udev	Harald Hoyer	8 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">efilinux/efilinux.git</a>	The efilinux UEFI boot loader	Matt Fleming	21 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">syslinux/syslinux.git</a>	The Syslinux boot loader suite	Syslinux workgroup	18 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<i>devel</i>				
<a href="#">pahole/pahole.git</a>	Pahole and other DWARF utils	Arnaldo Carvalho de Melo	10 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">sparse/chrisl/sparse.git</a>	Chris Li's sparse repository.	Christopher Li	5 weeks	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">sparse/sparse.git</a>	C semantic parser	Christopher Li	13 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<i>docs</i>				
<a href="#">kernel/kernel-docs.git</a>	Kernel Documentation tree	Doc Group	2 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">kernel/ksmap.git</a>	Kernel.org keysign map source	Kernel.org users	4 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">kernel/website.git</a>	Kernel.org website source	Doc Group	13 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">man-pages/man-pages.git</a>	Linux man pages Sections 2, 3, 4, 5, and 7	Michael Timothy Kerrisk	9 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">man-pages/website.git</a>	Website files for /doc/man-pages	Michael Timothy Kerrisk	2 weeks	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<i>editors</i>				
<a href="#">uemacs/uemacs.git</a>	Micro-emacs	Linus Torvalds	16 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<i>fs</i>				
<a href="#">ext2/e2fsprogs.git</a>	Ext2/3/4 filesystem userspace utilities	Theodore T'so	3 days	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">ext2/xfstests-bld.git</a>	Build framework and autorun scripts for xfstests	Theodore T'so	4 weeks	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">fat/fatattr/fatattr.git</a>	FAT attribute set utility	H. Peter Anvin	7 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">fuse/dbfs.git</a>	FUSE fs w/ Berkeley DB backend.	Jeff Garzik	7 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">fuse/fuse-ext2.git</a>	FUSE ext2 filesystem driver.	Jeff Garzik	10 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">squashfs/squashfs-tools.git</a>	squashfs tools development	Phillip Lougher	4 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">xfs/dmapi-dev.git</a>	Data Management API runtime environment	Christoph Hellwig	5 years	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>
<a href="#">xfs/xfs-documentation.git</a>	XFS AsciiDoc Documentation tree	XFS FS Group	3 months	<a href="#">summary</a> <a href="#">log</a> <a href="#">tree</a>

<https://git.kernel.org>

# Not Github PATCH mail 방식

이 **변화분** (commits) 을 통한 Review 와 Discussion 가능 (pull-request, PATCH mail)

**Araldo Carvalho de Melo** November 23, 2015 at 11:17 PM AC

To: Taeung Song Cc: Namhyung Kim, linux-kernel@vger.kernel.org, Ingo Molnar, Jiri Olsa  
Re: [PATCH v11 RESEND 02/24] perf tools: Add perf-config document

Em Sun, Nov 22, 2015 at 07:11:56PM +0900, Taeung Song escreveu:  
Add perf-config document to describe the perf configuration and a subcommand 'list'.

Worked this time, applied, will continue processing the other patches soon.

- Araldo

[See More](#)

☆ **tip-bot for Taeung Song** Today at 5:20 PM TF

To: linux-tip-commits@vger.kernel.org Cc: Ingo Molnar, and 7 more...

[tip:perf/core] perf config: Add initial man page

Commit-ID: 7d6852432acb3b09fc3ec45dd65421d34eebe3b5  
Gitweb: <http://git.kernel.org/tip/7d6852432acb3b09fc3ec45dd65421d34eebe3b5>  
Author: Taeung Song <[treeze.taeung@gmail.com](mailto:treeze.taeung@gmail.com)>  
AuthorDate: Sun, 22 Nov 2015 19:11:56 +0900  
Committer: Araldo Carvalho de Melo <[acme@redhat.com](mailto:acme@redhat.com)>  
CommitDate: Mon, 23 Nov 2015 18:31:25 -0300

perf config: Add initial man page

Add perf-config document to describe the perf configuration and a 'list' subcommand.

Signed-off-by: Taeung Song <[treeze.taeung@gmail.com](mailto:treeze.taeung@gmail.com)>  
Cc: Jiri Olsa <[jolsa@redhat.com](mailto:jolsa@redhat.com)>  
Cc: Namhyung Kim <[namhyung@kernel.org](mailto:namhyung@kernel.org)>  
Link: <http://lkml.kernel.org/r/63AD9B57-7B8C-46F8-8F18-0FFEB9A6A1BC@gmail.com>  
Signed-off-by: Araldo Carvalho de Melo <[acme@redhat.com](mailto:acme@redhat.com)>  
---

tools/perf/Documentation/perf-config.txt | 103 +++++  
1 file changed, 103 insertions(+)

**tip-bot for Taeung Song** Today at 5:20 PM TF

To: linux-tip-commits@vger.kernel.org  
Cc: tglx@linutronix.de, Jiri Olsa, linux-kernel@vger.kernel.org, Ingo Molnar, hpa@zytor.com, Taeung Song, Namhyung Kim, [acme@redhat.com](mailto:acme@redhat.com)

[tip:perf/core] perf tools: Add 'perf config' command

Commit-ID: 30862f2c5725c46afcfab5af710fdf5163bf0f81  
Gitweb: <http://git.kernel.org/tip/30862f2c5725c46afcfab5af710fdf5163bf0f81>  
Author: Taeung Song <[treeze.taeung@gmail.com](mailto:treeze.taeung@gmail.com)>  
AuthorDate: Tue, 17 Nov 2015 22:53:21 +0900  
Committer: Araldo Carvalho de Melo <[acme@redhat.com](mailto:acme@redhat.com)>  
CommitDate: Mon, 23 Nov 2015 18:31:24 -0300

perf tools: Add 'perf config' command

The perf configuration file contains many variables to change various aspects of each of its tools, including output, disk usage, etc.

But looking at the state of configuration is difficult and there's no documentation about config variables except for the variables in perfconfig.example exist.

So this patch adds a 'perf-config' command with a '--list' option.

perf config [options]

display current perf config variables.  
# perf config -l | --list

Signed-off-by: Taeung Song <[treeze.taeung@gmail.com](mailto:treeze.taeung@gmail.com)>  
Acked-by: Namhyung Kim <[namhyung@kernel.org](mailto:namhyung@kernel.org)>  
Cc: Jiri Olsa <[jolsa@redhat.com](mailto:jolsa@redhat.com)>  
Link: <http://lkml.kernel.org/r/1447768424-17327-1-git-send-email-treeze.taeung@gmail.com>  
Signed-off-by: Araldo Carvalho de Melo <[acme@redhat.com](mailto:acme@redhat.com)>  
---

tools/perf/Build | 1 +  
tools/perf/builtin-config.c | 66 +++++  
tools/perf/builtin.h | 1 +  
tools/perf/command-list.txt | 1 +  
tools/perf/perf.c | 1 +  
5 files changed, 70 insertions(+)

# Opensource 각종 오픈소스

The screenshot shows the Open Hub website interface. At the top, there's a navigation bar with links for PROJECTS, PEOPLE, ORGANIZATIONS, TOOLS, CODE, and BLOG. A search bar is prominently displayed in the center. Below the search bar, there are two main sections: 'Join Now' and 'What's New'. The 'Join Now' section includes options to 'Claim your contributions', 'Manage your project's data', and 'Highlight your use of FOSS'. The 'What's New' section features a large advertisement for the 'FREE BLACK DUCK VULNERABILITY PLUGIN'. Below these sections, there are three columns of project rankings: 'Most Popular Projects', 'Most Active Projects', and 'Most Active Contributors'. Each column lists projects with their respective user or commit counts.

**Join Now**

- Claim your contributions
- Manage your project's data
- Highlight your use of FOSS
- Join Now

**What's New**

**FREE BLACK DUCK VULNERABILITY PLUGIN**

Identify open source software in your projects and discover known vulnerabilities.

**Most Popular Projects**

Project	Users
Mozilla Firefox	13261 users
Apache HTTP Server	9466 users
MySQL	9180 users
Apache Subversion	8680 users
PHP	7789 users
Linux Kernel	7142 users

**Most Active Projects**

Project	Commits
Alfresco Content Management Community Edition	19909 commits
Arch Linux Packages	7343 commits
Gentoo Linux	6280 commits
azure-content	4740 commits
Mozilla Firefox	4558 commits

**Most Active Contributors**

Contributor	Commits
Felix Yan	7389 commits
Translation updater bot	1440 commits
luca suriano	1159 commits
Manfred Tournon	546 commits
Mischa ter Smitten	501 commits
Ori Livneh	501 commits

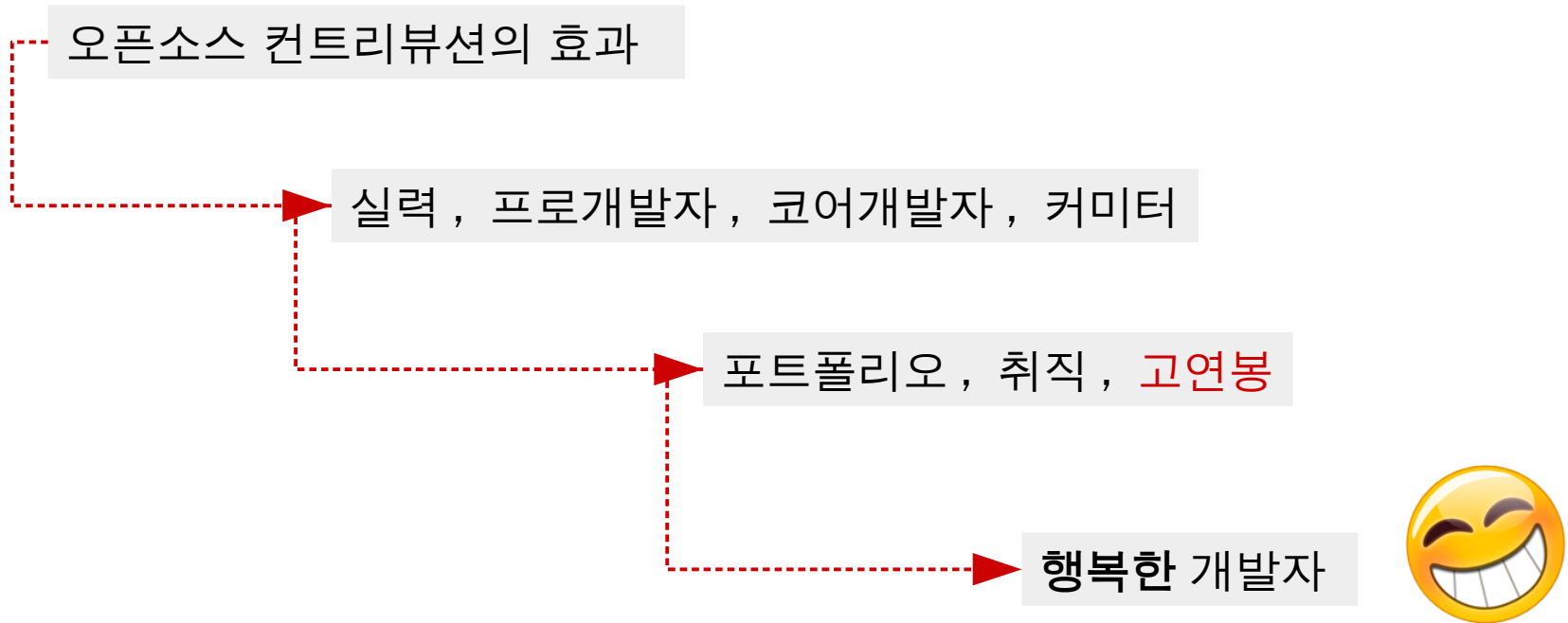
전세계 각종 오픈소스 통계 사이트 (<https://www.openhub.net/>)

# Opensource 각종 오픈소스에 기여 통계



오픈소스 커미터 개인별 통계 뷰 (<https://www.openhub.net/accounts/namhyung>)

# Opensource Contributor, Committer, Maintainer



( 참고 )

<https://wiki.kldp.org/wiki.php/KoreanOpenSourceCommitter>

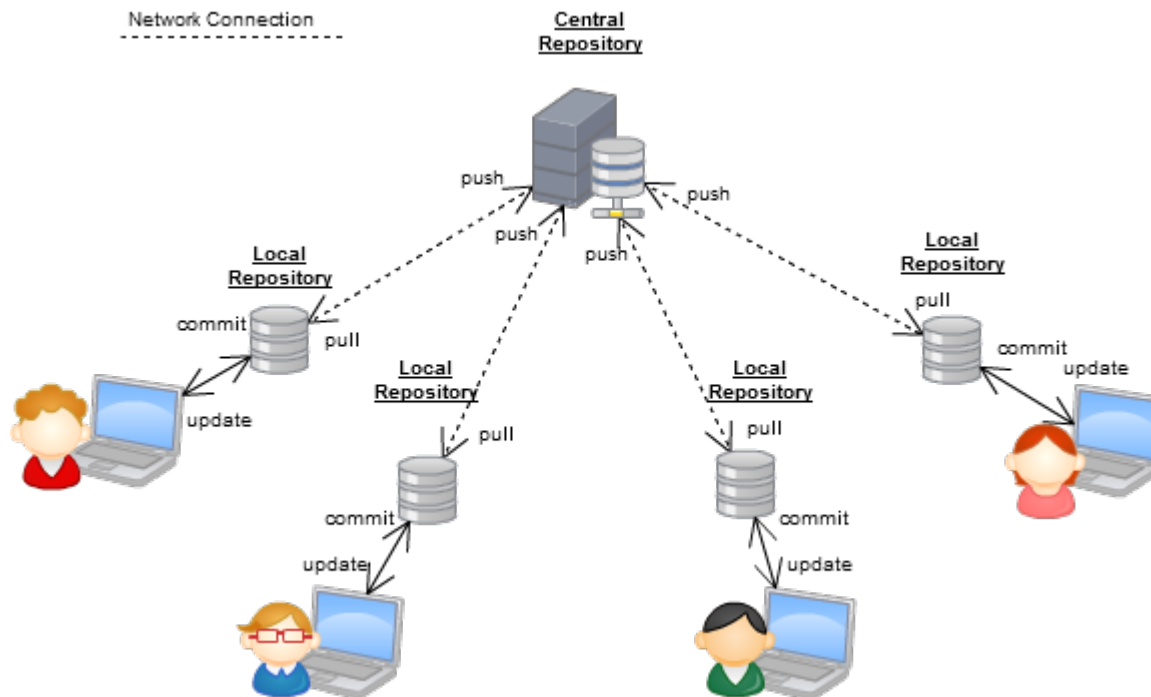
<http://dirkriehle.com/publications/2014-2/the-open-source-software-developer-career-and-its-benefits/>

[http://www.payscale.com/research/US/Skill=Open\\_Source/Salary](http://www.payscale.com/research/US/Skill=Open_Source/Salary)

<http://nolongernew.blogspot.kr/2010/02/economic-motivation-of-open-source.html>

# Git 이해하기

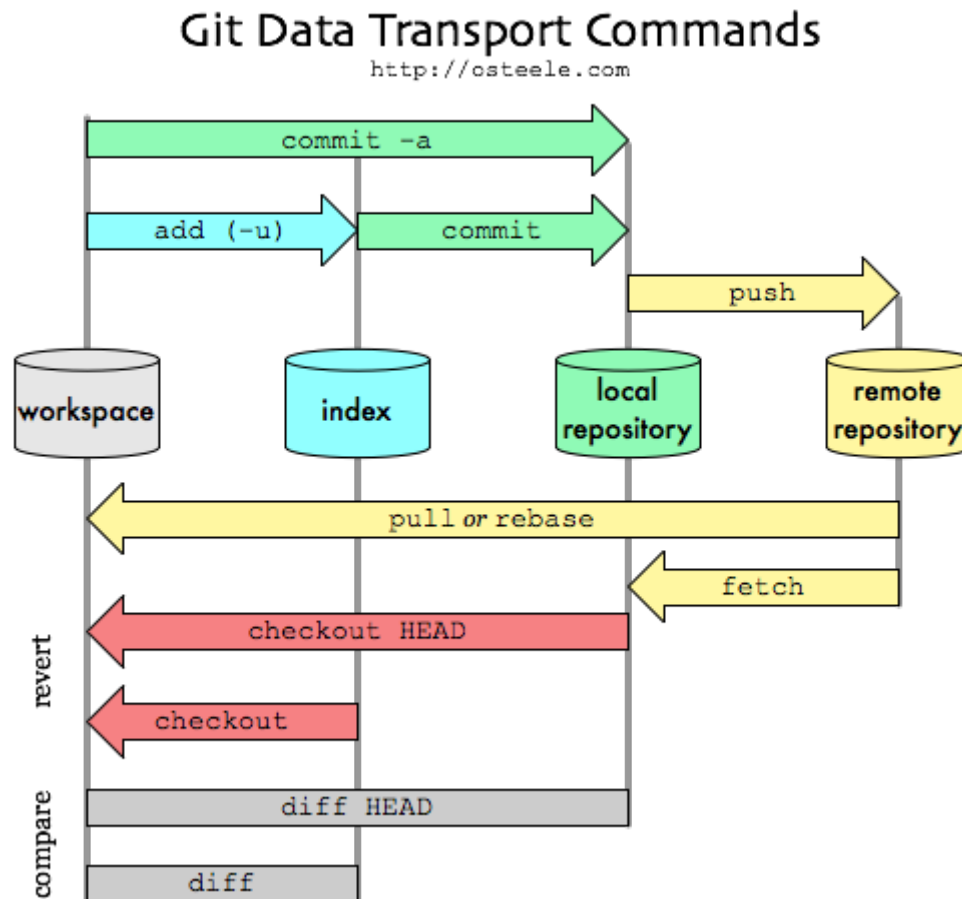
이제는 이 그림을 이해할 수 있다 . (Git 을 통한 작업 흐름 )





# Git 이해하기

이제는 이 그림도 이해할 수 있다 . (Git 을 통한 작업 흐름 )



# 우리 프로젝트와 Git 운용 전략 (프로젝트 관리방식)

---

프로젝트명 : salady

팀원 : Taeung, Irvanda ...

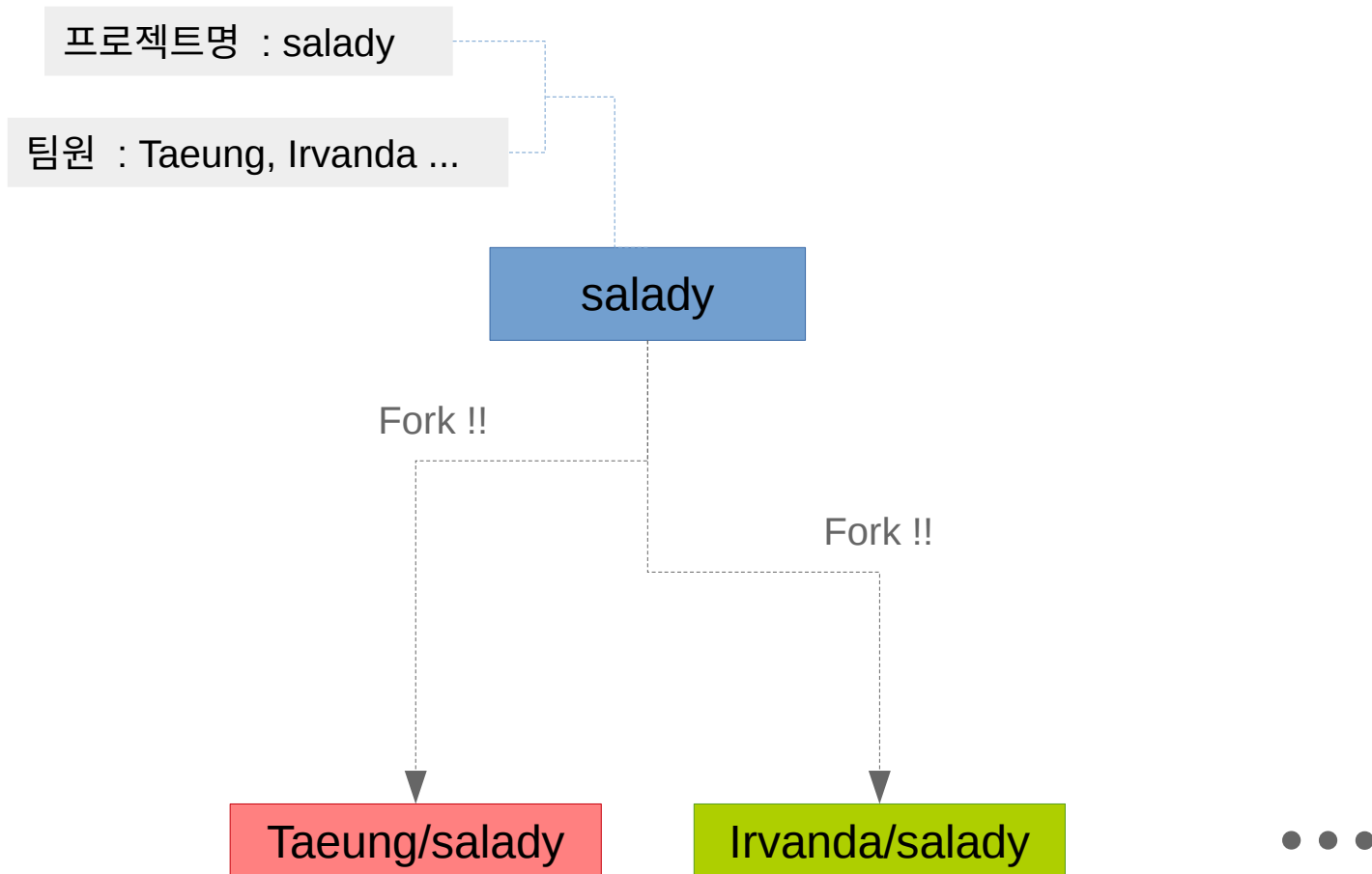
salady

```
graph TD; A[프로젝트명 : salady] -.- B[팀원 : Taeung, Irvanda ...]; B -.- C[salady];
```

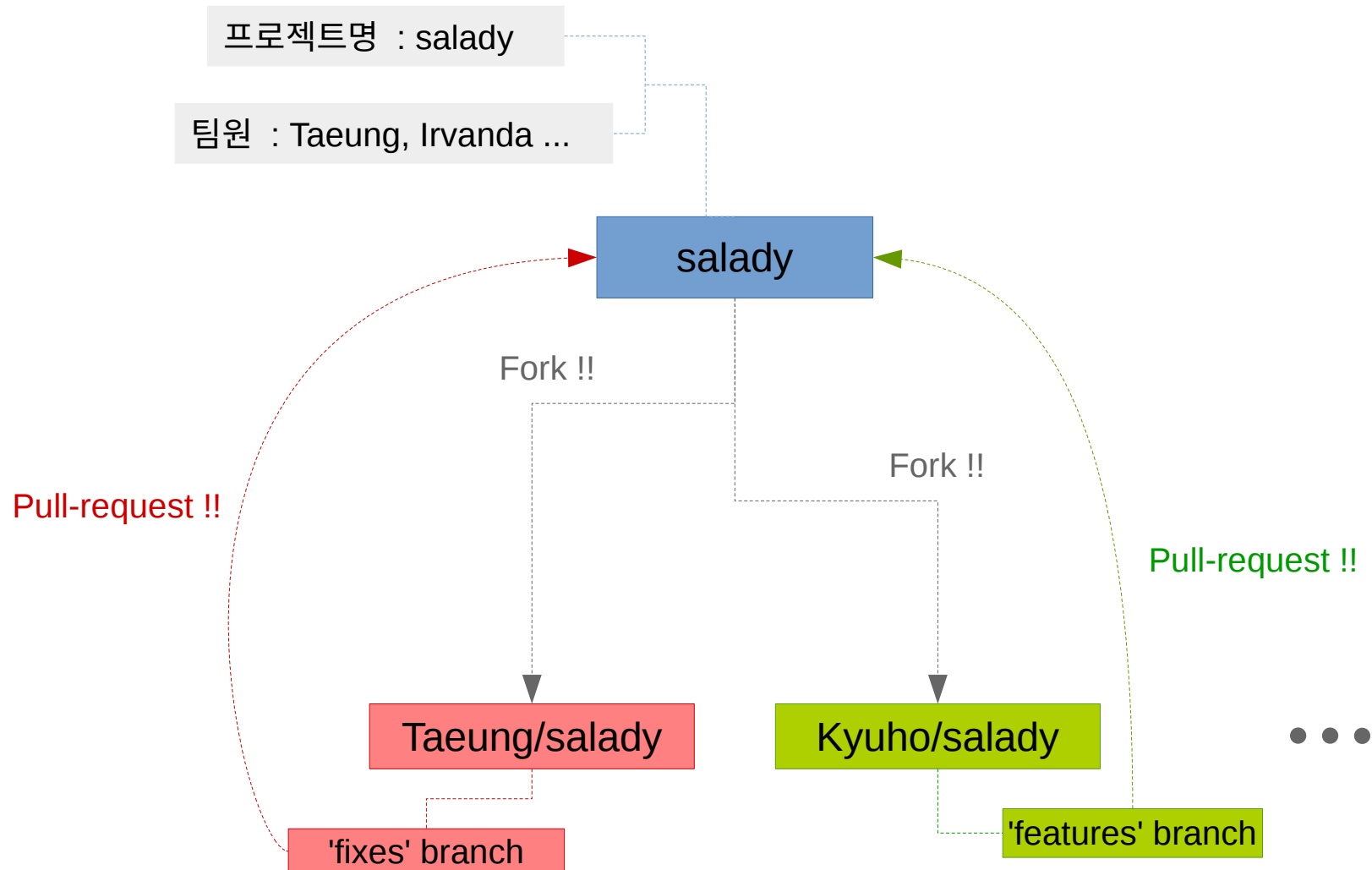
The diagram illustrates the project management structure. It features two light gray boxes at the top left: the top one contains '프로젝트명 : salady' and the bottom one contains '팀원 : Taeung, Irvanda ...'. Dashed lines connect these two boxes, and another dashed line extends from the bottom box to a solid blue box at the bottom right labeled 'salady'.

# 우리 프로젝트와 Git 운용 전략 (프로젝트 관리방식)

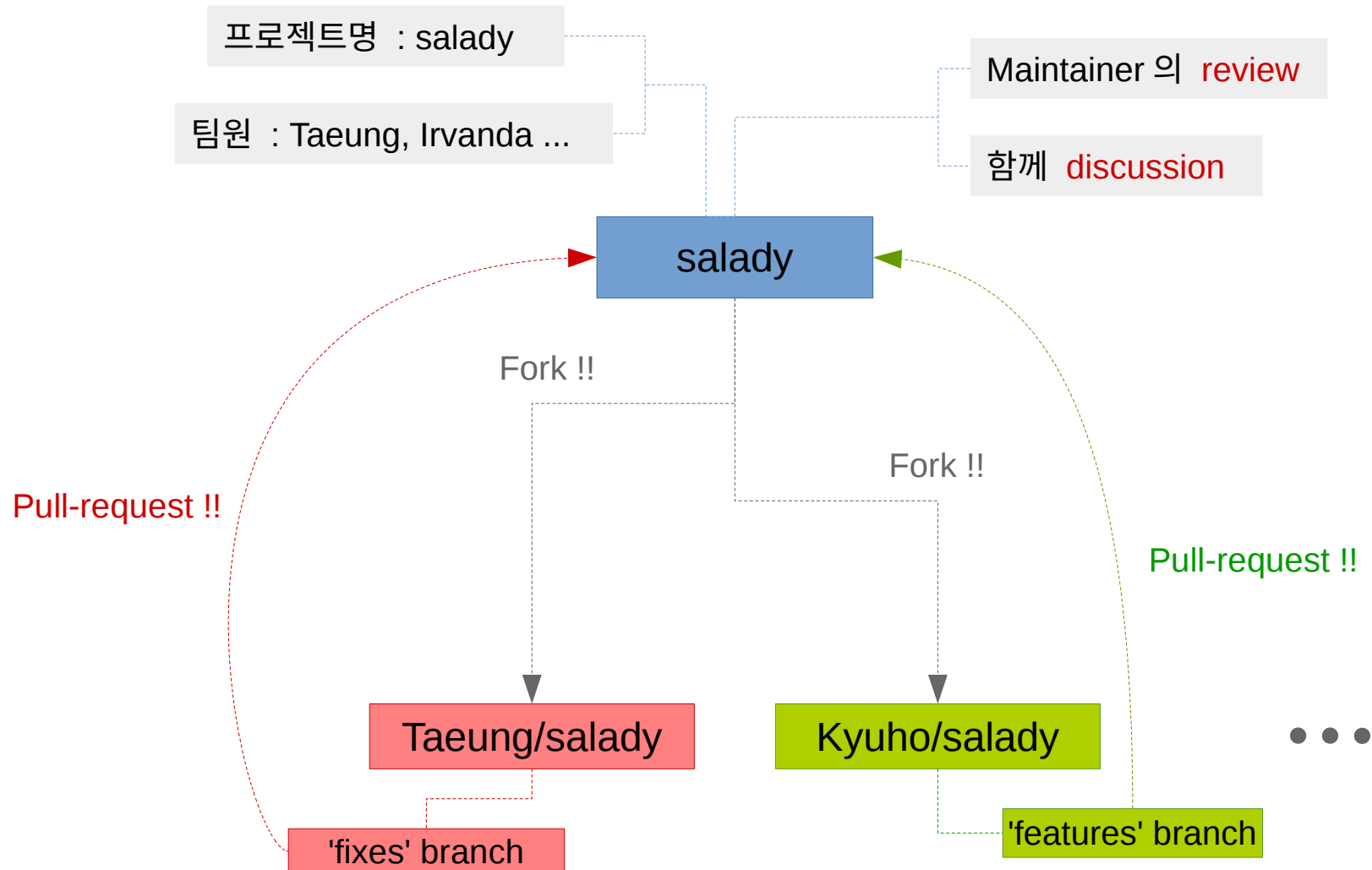
---



# 우리 프로젝트와 Git 운용 전략 (프로젝트 관리방식)



# 우리 프로젝트와 Git 운용 전략 (프로젝트 관리방식)



# Q&A - 자주 묻는 질문들과 간단한 대답

---

## 1) Git 과 Github 의 차이는 ?

Git 은 각 컴퓨터 (**local**) 에 설치되어 소스코드관리가 가능한 프로그램이고 Github 는 **remote** 저장소가 있는 외부서버를 지칭한다 .

## 2) Commit 과 Push 의 차이는 ?

commit 은 **local** 작업폴더에 history 를 쌓는것이어서 외부망 (internet) 을 안쓰고 Push 는 **remote** 저장소 (Github 등 ) 에 history 를 쌓는것이어서 외부망 (internet) 이 필요하다 .

## 3) Fetch 와 Pull 의 차이는 ?

Remote 저장소 (Github 등 ) 로 부터 최신 commit 정보들을 가져오는것은 매한가지이나 Fetch 는 가져와서 **임시폴더** (.git) 에 **저장**하고 Pull 은 바로 현재 branch 에 **merge 작업을 동반**한다 .

## 4) Rebase 와 Merge 의 차이는 ?

둘다 두 branch 의 차이점 (commits) 를 합치는것은 매한가지나 Rebase 는 합치기 전에 **되감기** (rewinding) 를 하고 Merge 는 **안하고** 합친다 .

# Assignment – 오픈소스의 Commits 분석 숙제

---

- 4 개의 commit 내용을 설명하는 간단한 해설 써서 [taeung@kosslab.kr](mailto:taeung@kosslab.kr)에 보내기

## 1. electron 프로젝트 , Jimi

<https://github.com/electron/electron/commit/8121f913cae44c8d374cb19ea6a655a670d33995>

## 2. Linux Kernel 프로젝트 , Taeung Song

<https://github.com/torvalds/linux/commit/860b8d4b3f893c97f905b978ecf62f48816dc5de>

## 3. facebook 의 nuclide 프로젝트 , Tomas Liu

<https://github.com/facebook/nuclide/commit/63ae4a4095fc5796909fbafed388f416662f7a63>

## 4. Linux Kernel 프로젝트 , Tejun Heo

<https://github.com/torvalds/linux/commit/3a06bb78ceeceacc86a1e31133a7944013f9775b>

Other questions ?