경희대학교
KYUNG HEE UNIVERSITY

Git / GitHub 2주차

2014년 1월 5일

"누군가가 나의 등잔의 심지에서 불을 붙여가도
내 등잔의 불은 여전히 빛나고 있습니다."

Mobile Convergence Laboratory

목차

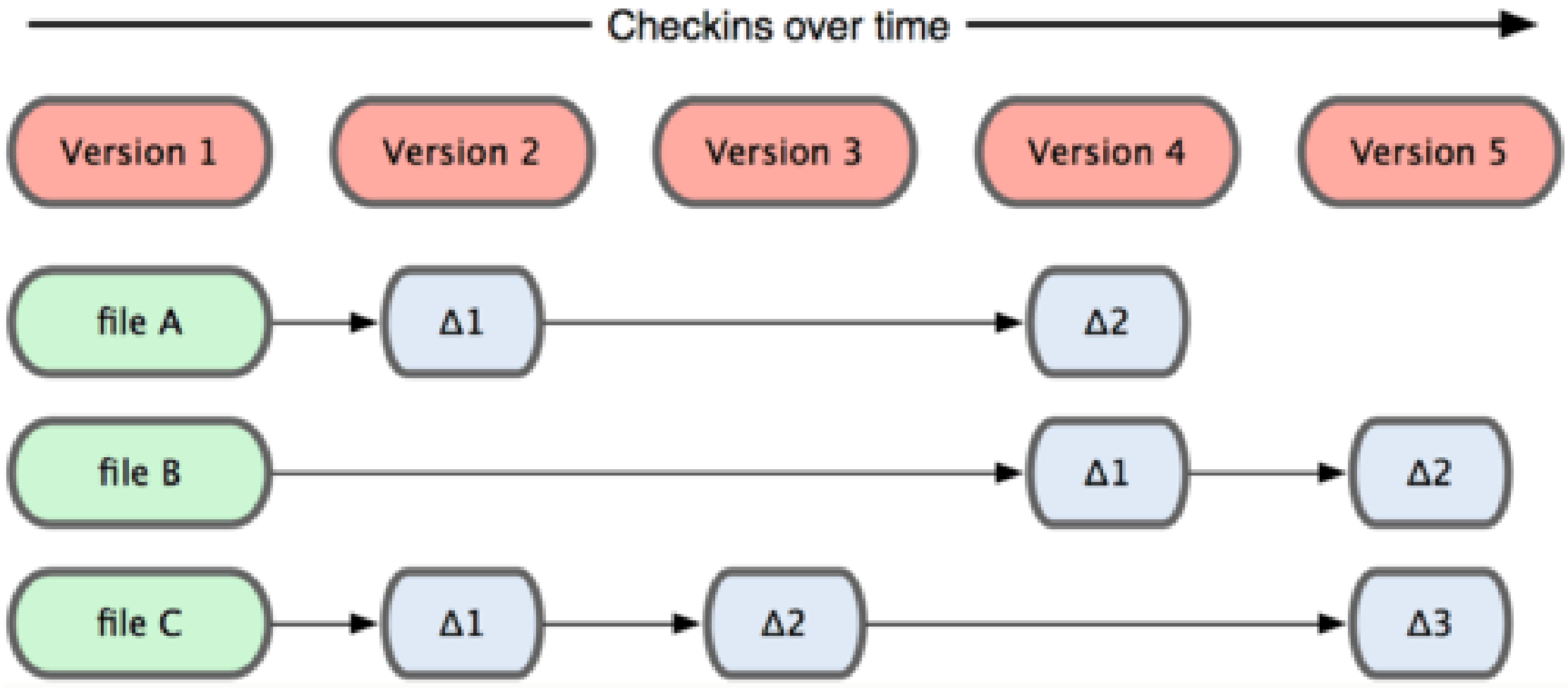
- 지난 시간 궁금증 – git 파일 구조
- branch 소개
- merge 소개
- 충돌 다루기 – diff / blame 소개
- 사용 시나리오

- GitHub 사용 – 원격저장소 push/pull
- GitHub 사용 – Introducing GitHub

Git의 파일 구조

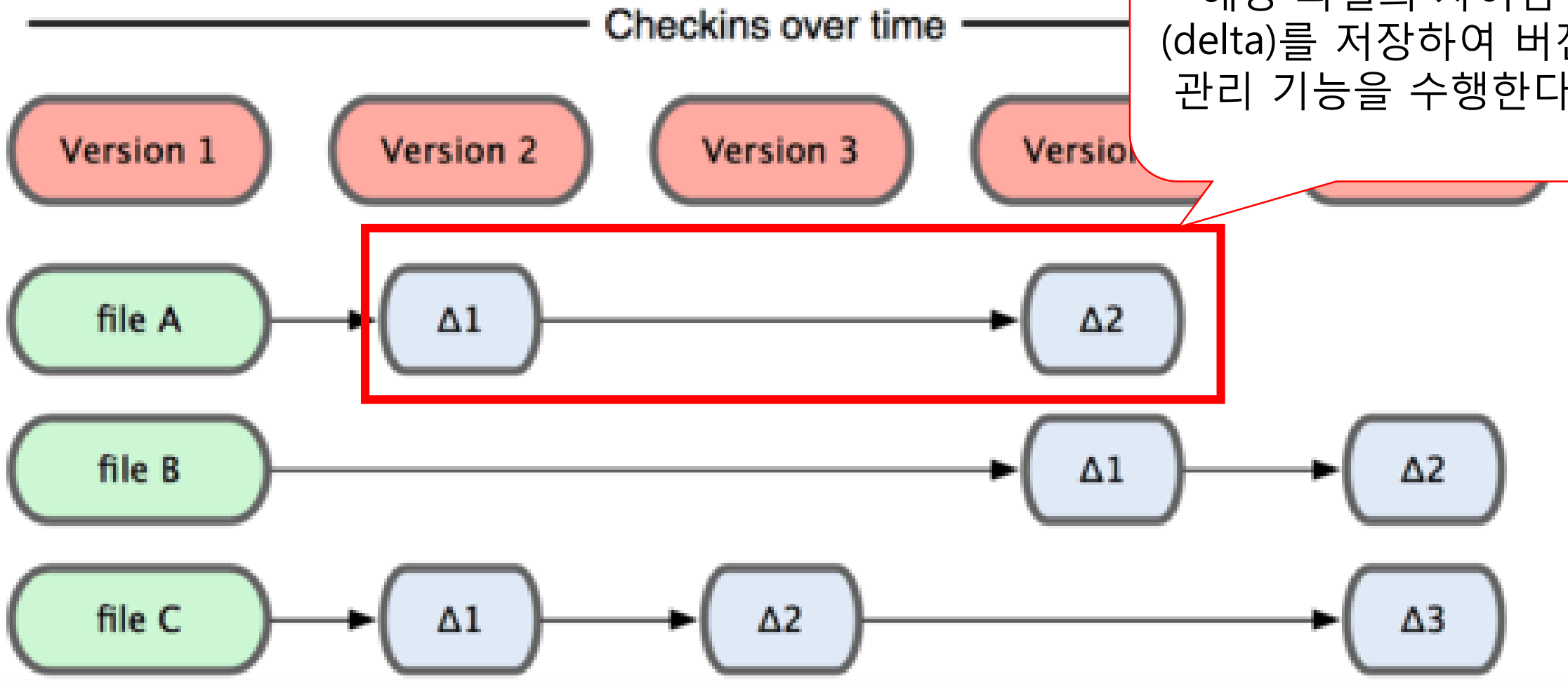
Git은 파일 간의 차이점을
저장하는 것이 아니다!.

Subversion (SVN, CVS…)



CVS(Concurrnt Version System
SVN(Subversion)
형상관리 = 버전 관리

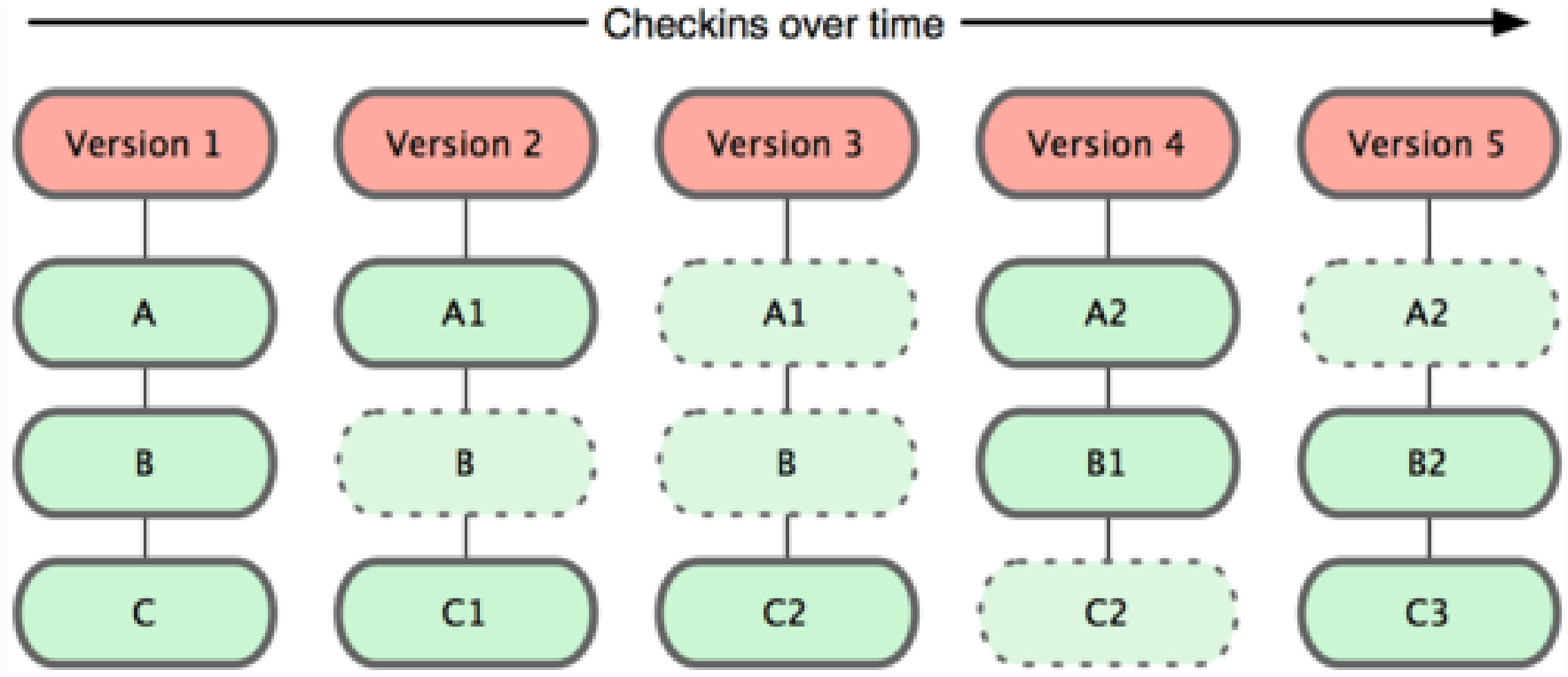
Subversion (SVN, CVS…)



해당 파일의 차이점 (delta)를 저장하여 버전 관리 기능을 수행한다.

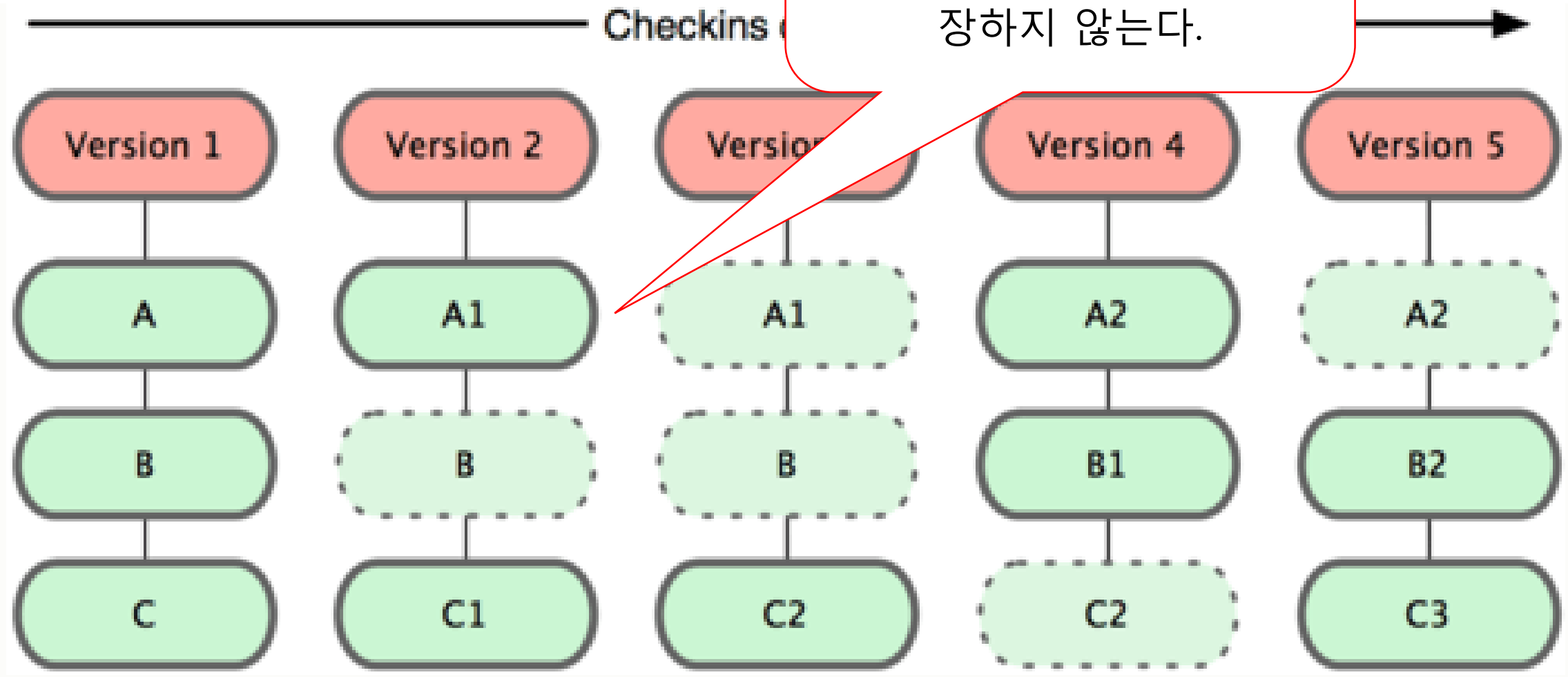
CVS(Concurrnt Version System
SVN(Subversion)
형상관리 = 버전 관리

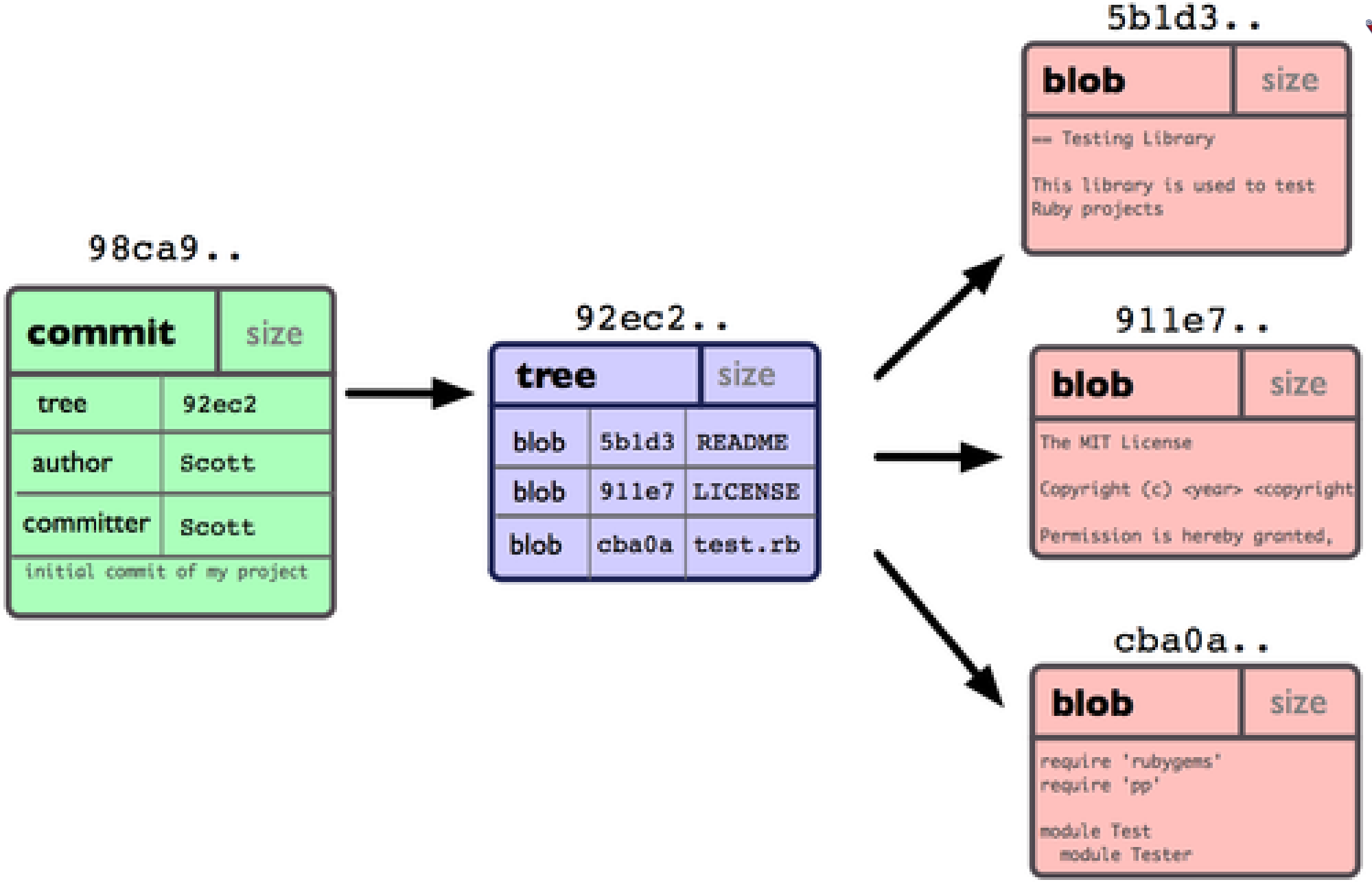
Git의 방식


Checkins over time
Version 1
Version 2
Version 3
Version 4
Version 5
A
A1
A1
A2
A2
B
B
B
B1
B2
C
C1
C2
C2
C3

Git의 방식

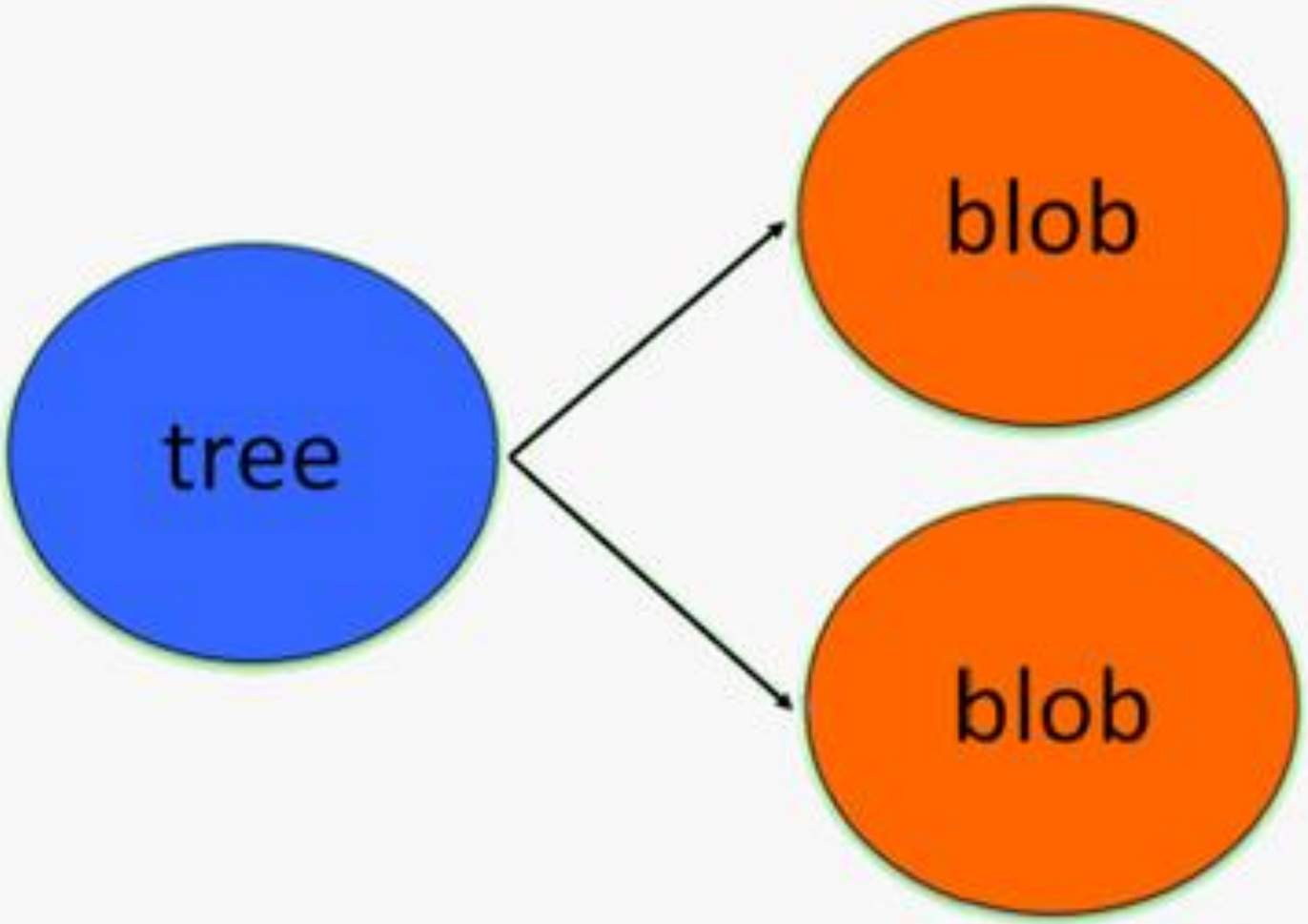
시간순으로 프로젝트의 스냅샷을 저장한다.
파일이 달라지지 않으면 저장하지 않는다.

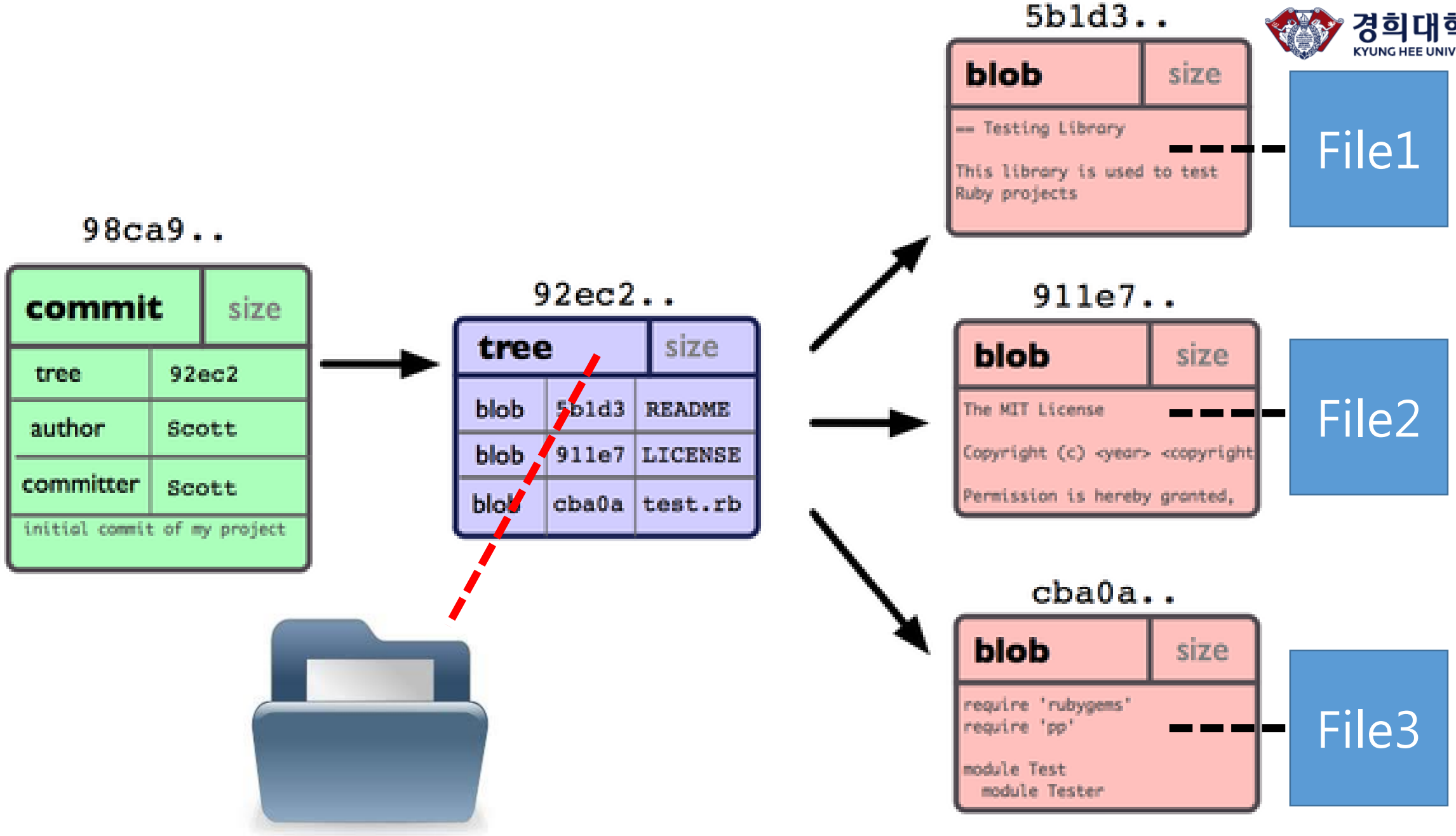



98ca9..
commit
size
tree
92ec2
author
Scott
committer
Scott
initial commit of my project
92ec2..
tree
size
blob
5b1d3
README
blob
911e7
LICENSE
blob
cba0a
test.rb
5b1d3..
blob
size
== Testing Library
This library is used to test
Ruby projects
911e7..
blob
size
The MIT License
Copyright (c) <year> <copyright
Permission is hereby granted,
cba0a..
blob
size
require 'rubygems'
require 'pp'
module Test
module Tester

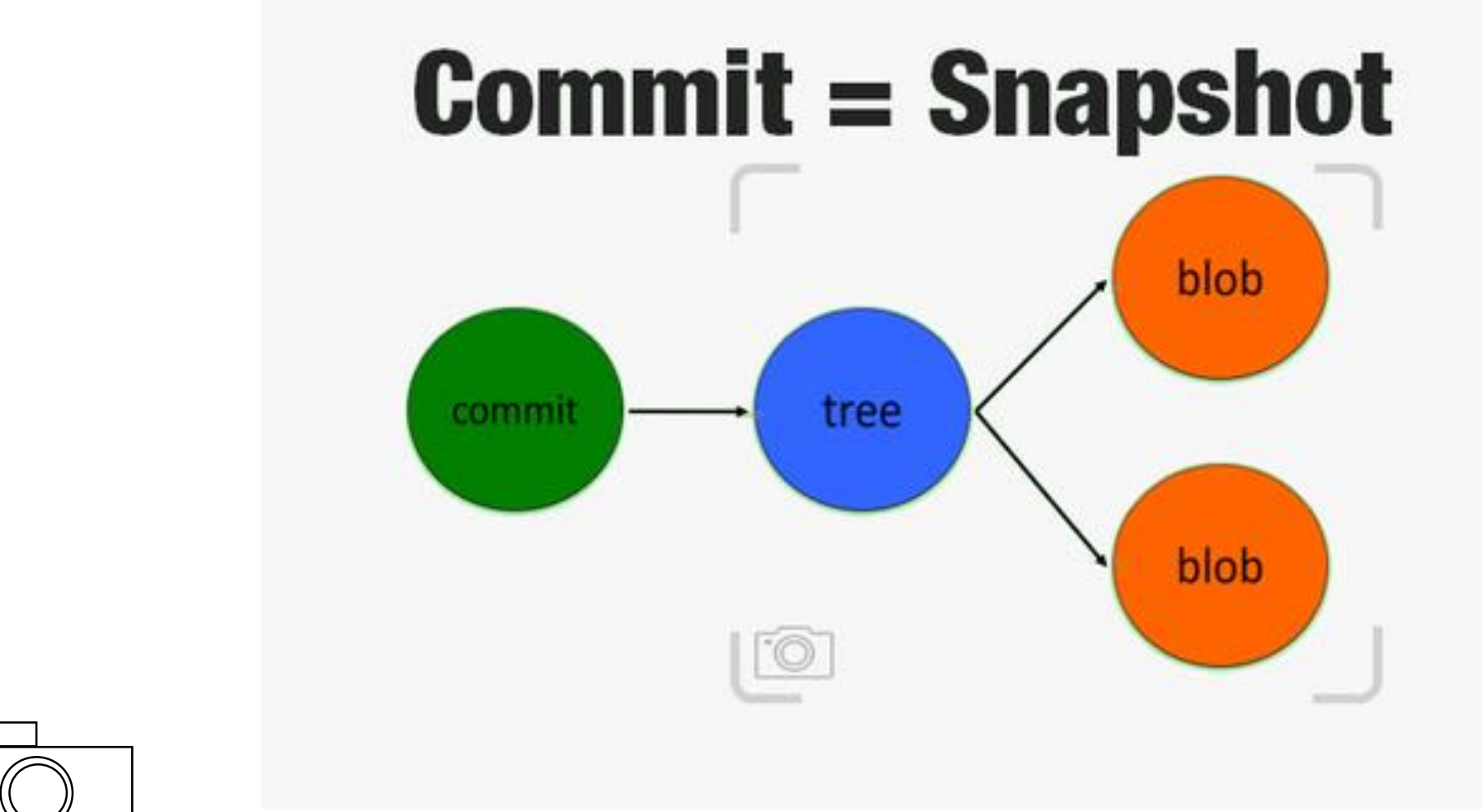
Blob = File

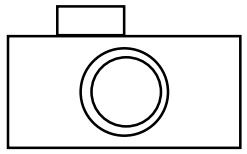
Tree = Folder

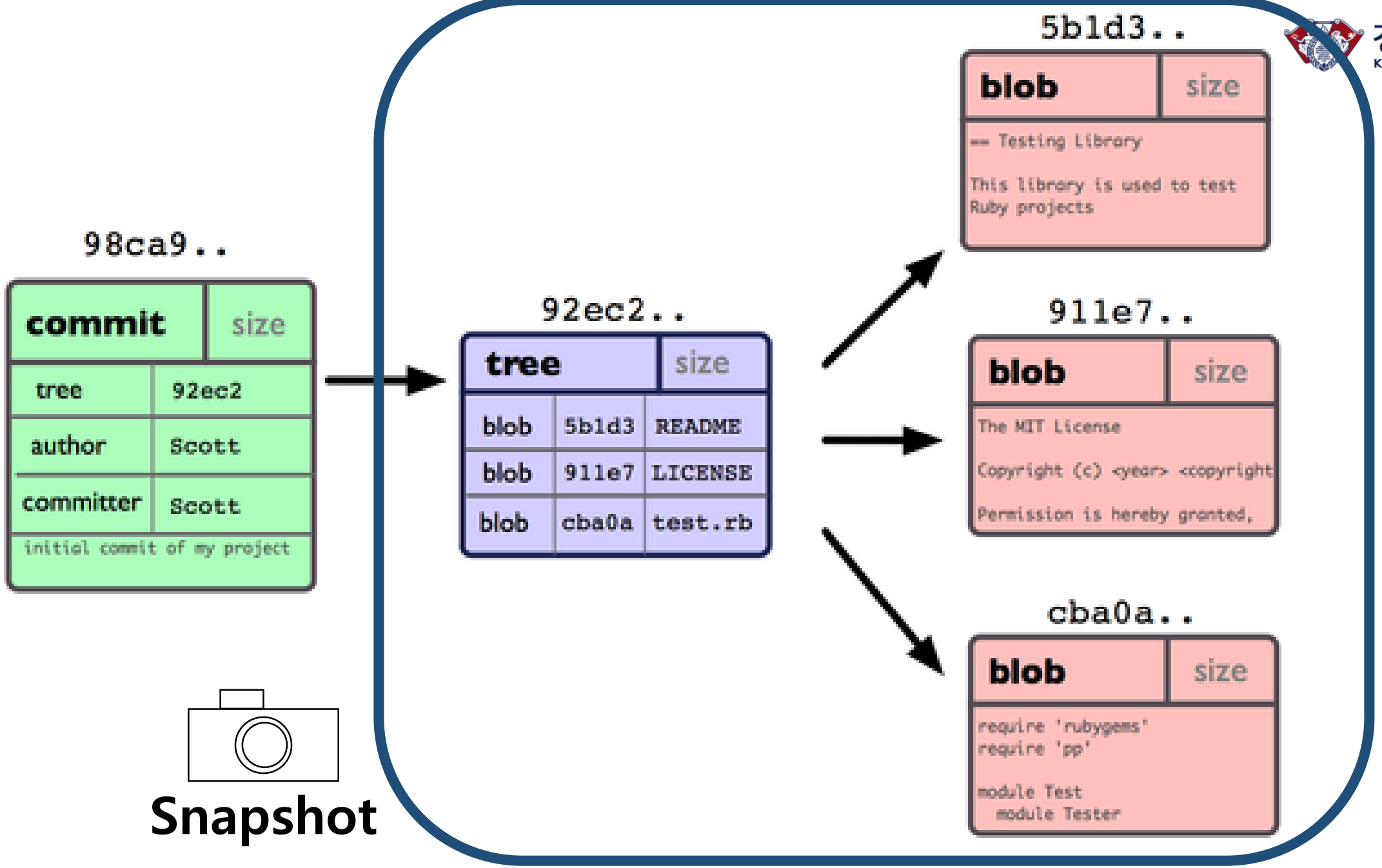



98ca9..
commit
size
tree
92ec2
author
Scott
committer
Scott
initial commit of my project
92ec2..
tree
size
blob
5b1d3
README
blob
911e7
LICENSE
blob
cba0a
test.rb
5b1d3..
blob
size
== Testing Library
This library is used to test
Ruby projects
File1
911e7..
blob
size
The MIT License
Copyright (c) <year> <copyright
Permission is hereby granted,
File2
cba0a..
blob
size
require 'rubygems'
require 'pp'
module Test
module Tester
File3

커밋의 의미






98ca9..
commit
size
tree
92ec2
author
Scott
committer
Scott
initial commit of my project
92ec2..
tree
size
blob
5b1d3
README
blob
911e7
LICENSE
blob
cba0a
test.rb
5b1d3..
blob
size
== Testing Library
This library is used to test
Ruby projects
911e7..
blob
size
The MIT License
Copyright (c) <year> <copyright
Permission is hereby granted,
cba0a..
blob
size
require 'rubygems'
require 'pp'
module Test
module Tester
Snapshot

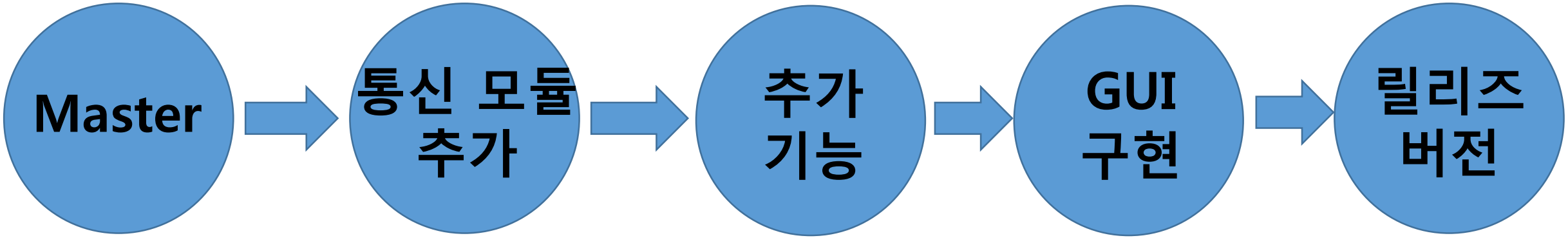
branch / merge

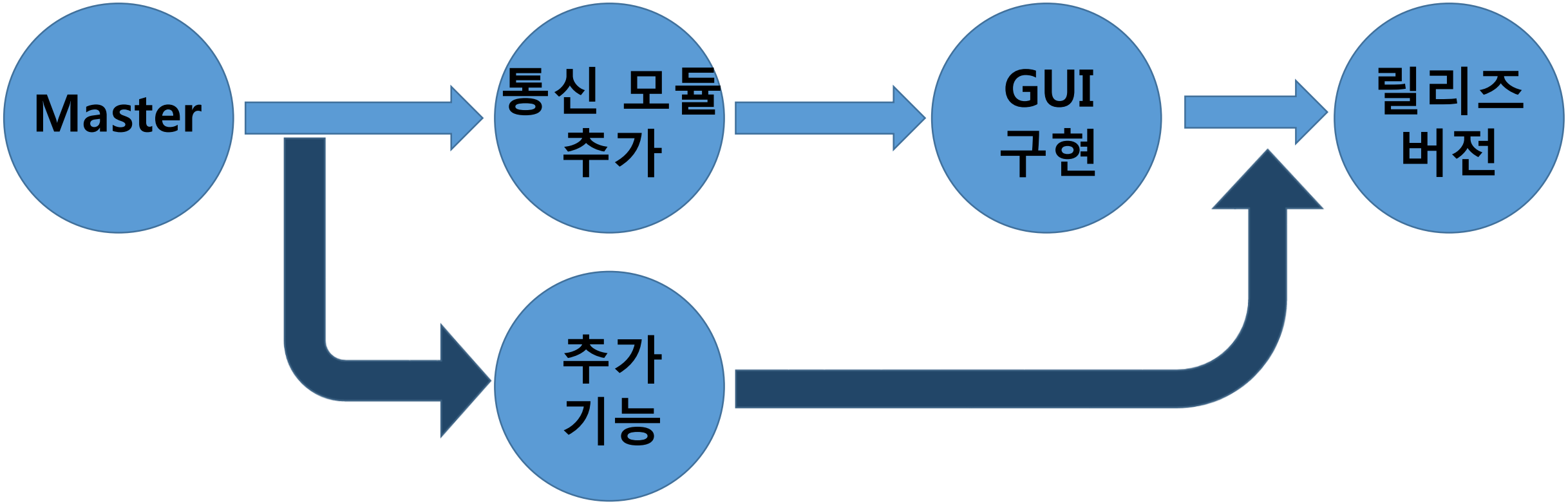
branch

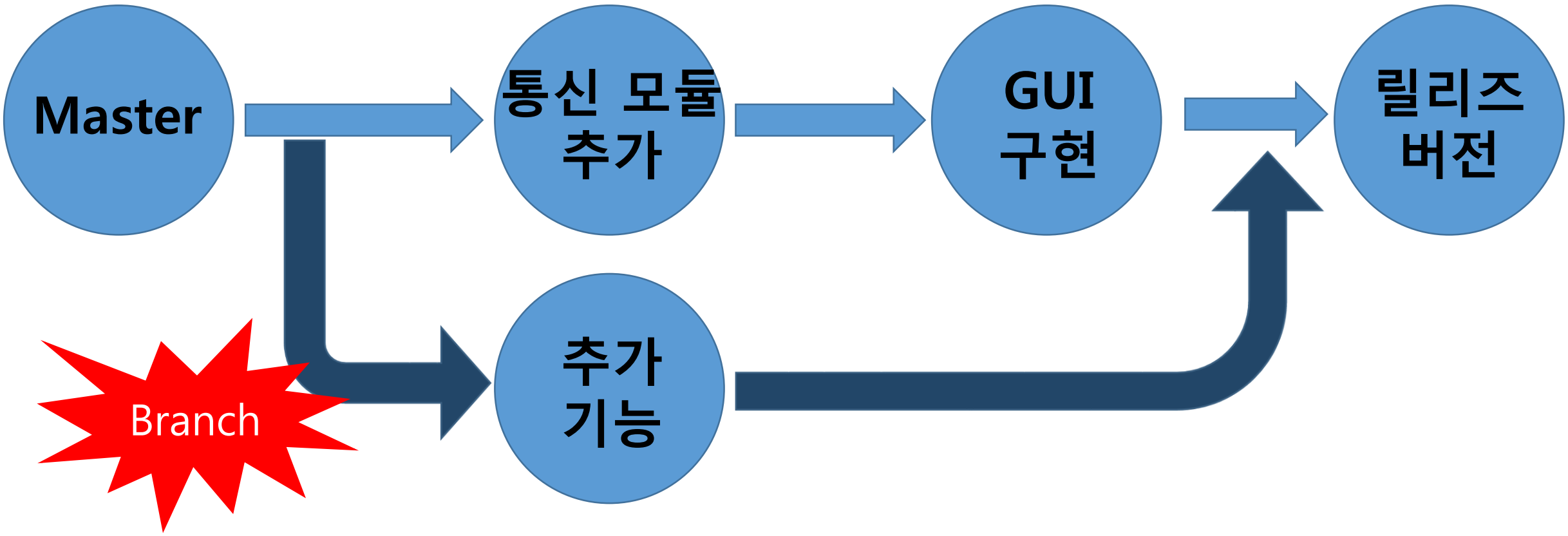
- 상태를 저장하는 작업공간을 만든다.
- 안전하게 격리된 상태에서 추가 및 수정할 때 사용
- 상태?
  파일, 파일의 내용, 커밋 정보 등 git이 관리하는 데이터.

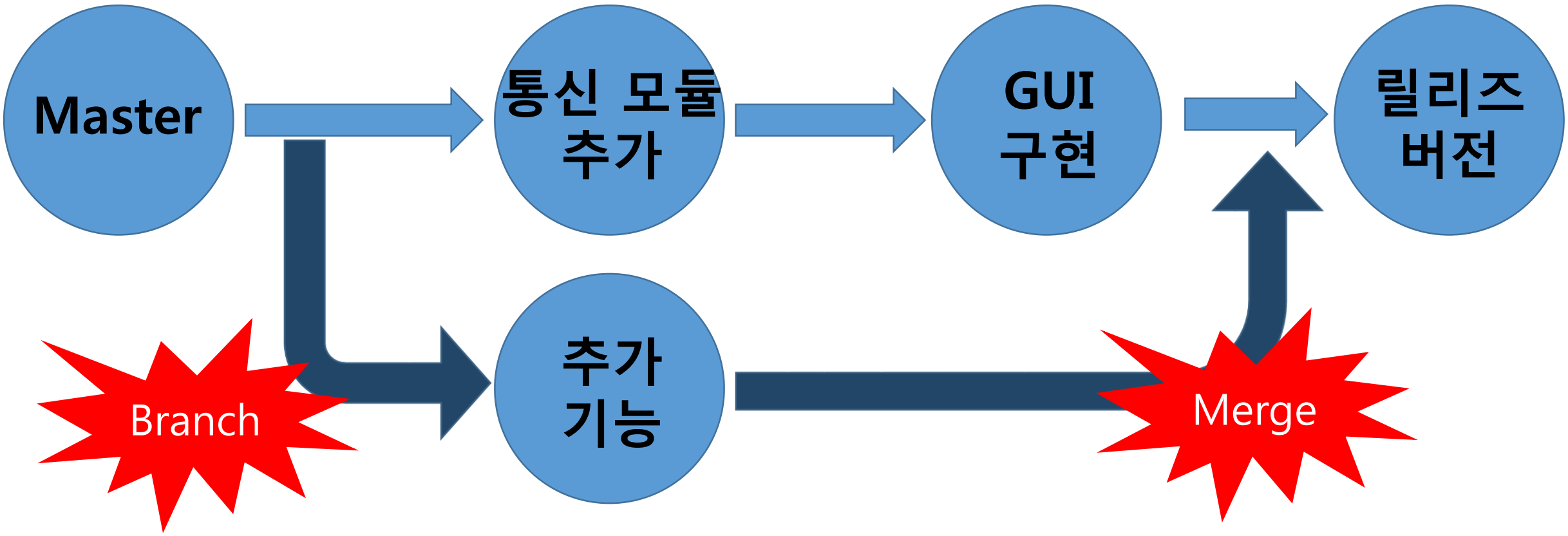
branch

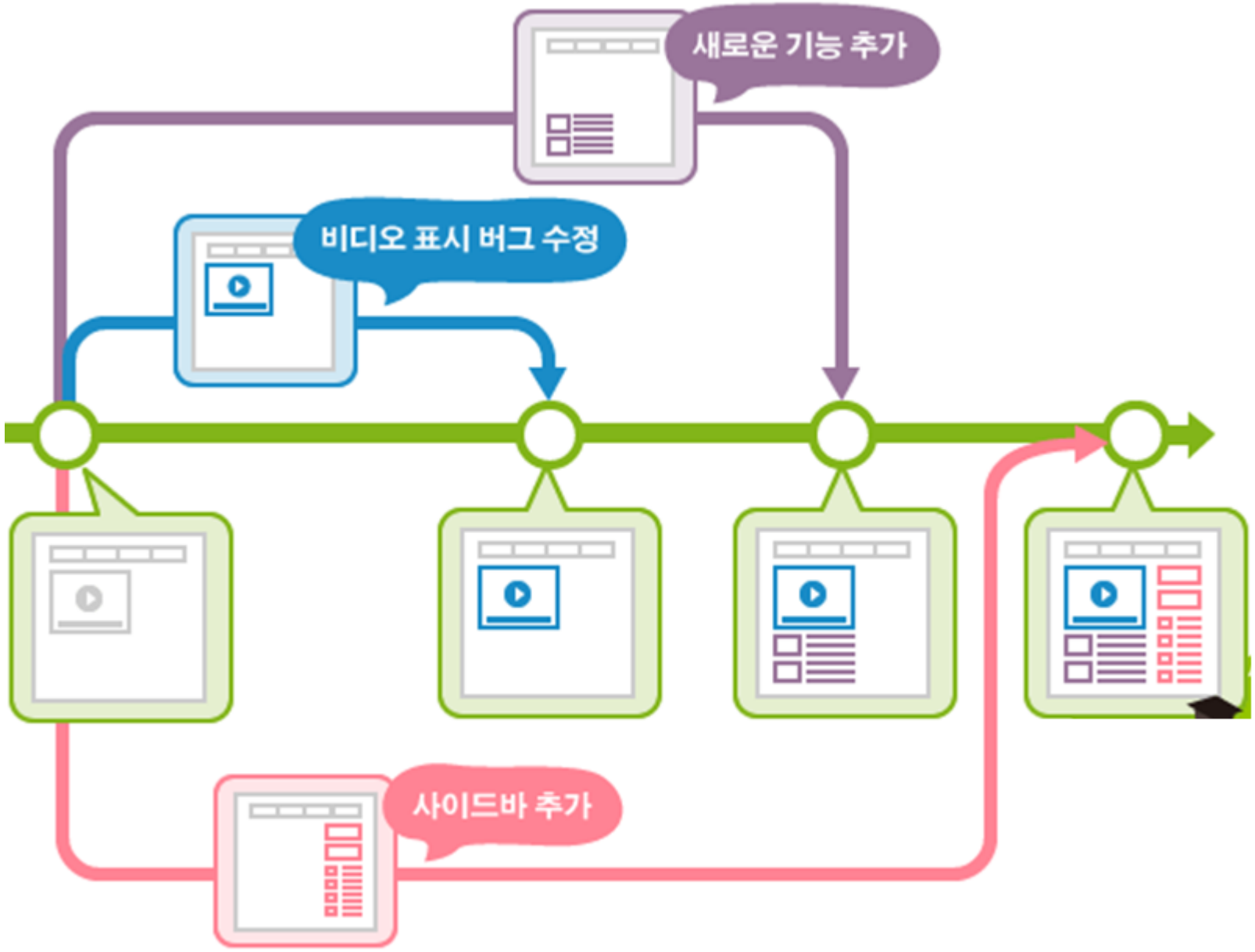
- 저장소를 새로 만들게 되면 기본적으로 master branch 생성
- 다른 branch를 이용해서 개발을 진행하고, 나중에 개발이 완료되면 master branch로 돌아와서 병합(merge)하면 된다.

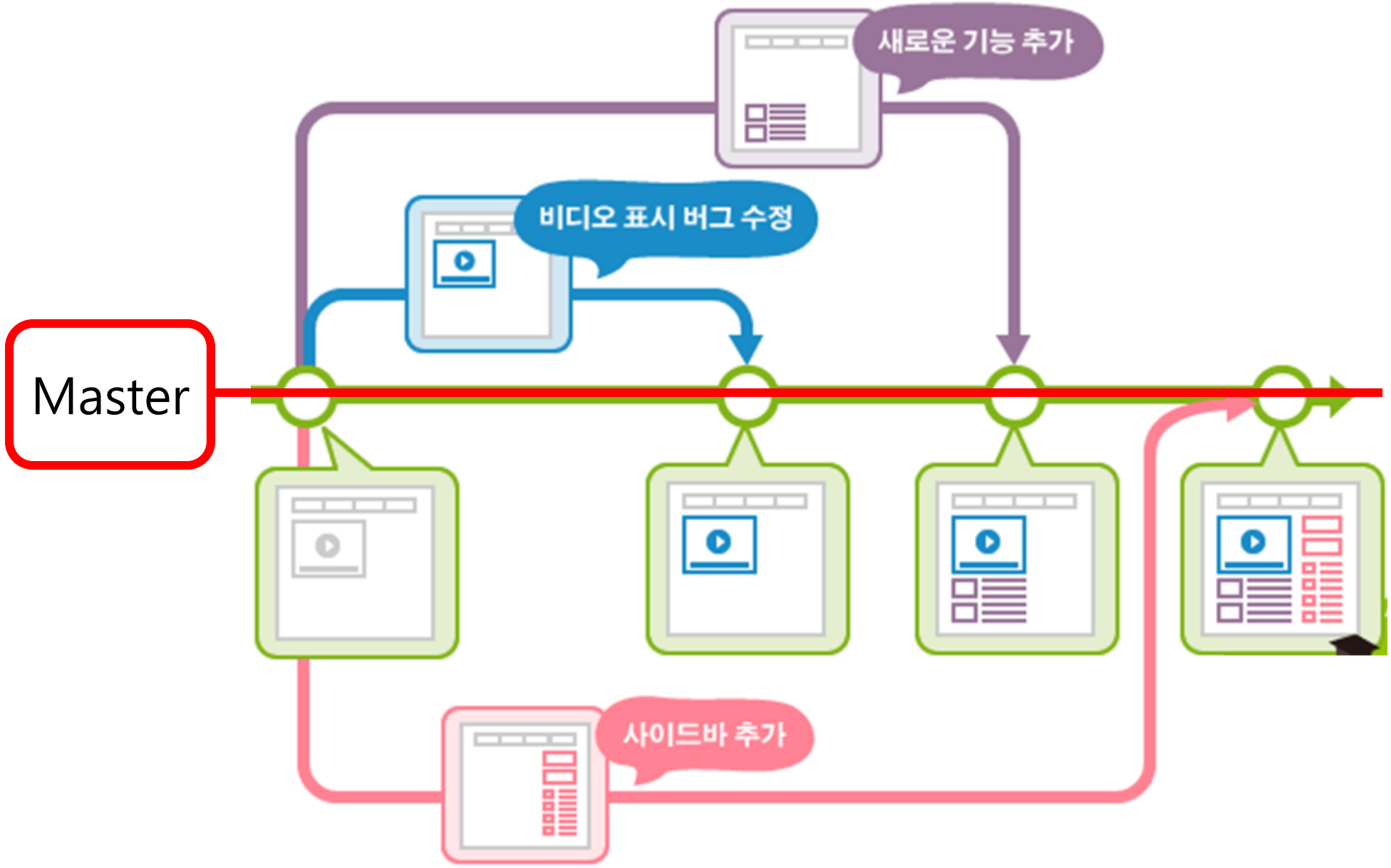

Master
통신 모듈 추가
추가 기능
GUI 구현
릴리즈 버전


Master
통신 모듈
추가
GUI
구현
릴리즈
버전
추가
기능


Master
통신 모듈
추가
GUI
구현
릴리즈
버전
Branch
추가
기능


Master
통신 모듈
추가
GUI
구현
릴리즈
버전
Branch
추가
기능
Merge


새로운 기능 추가
비디오 표시 버그 수정
사이드바 추가


새로운 기능 추가
비디오 표시 버그 수정
Master
사이드바 추가

브랜치의 종류

- 통합 브랜치(Integration Branch)
  - 언제든지 배포할 수 있는 버전
  - 안정적인 버전

- 토픽 브랜치(Topic Branch)
  - 기능 추가나 버그 수정과 같은 단위 작업을 위한 브랜치


version1.0
version1.1
version1.2
통합 브랜치
토픽 브랜치 이미지

브랜치 만들기(git branch 브랜치명)

- ex> git branch add_file2

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch add_file2
```

브랜치 목록

- git branch
- '*'가 붙어있는 것이 현재 선택된 브랜치이다.

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
  add_file2
  add_other
* master
```

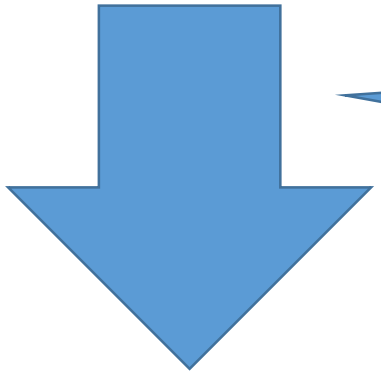
브랜치 전환하기

- git checkout 브랜치이름

```
hansangyun@hansangyun:~/바탕화면/git_example$ git checkout add_file2
D       file.c
Switched to branch 'add_file2'
```

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
* add_file2
  add_other
  master
```

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
  add_file2
  add_other
* master
```



git branch add_file2

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
* add_file2
  add_other
  master
```

브랜치 삭제하기

- git branch –d 브랜치이름

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch -d add_other
Deleted branch add_other (was b3e6874).
hansangyun@hansangyun:~/바탕화면/git_example$ git branch -d first_branch
Deleted branch first_branch (was b3e6874).
```

Merge(병합하기)

- git merge 커밋이름

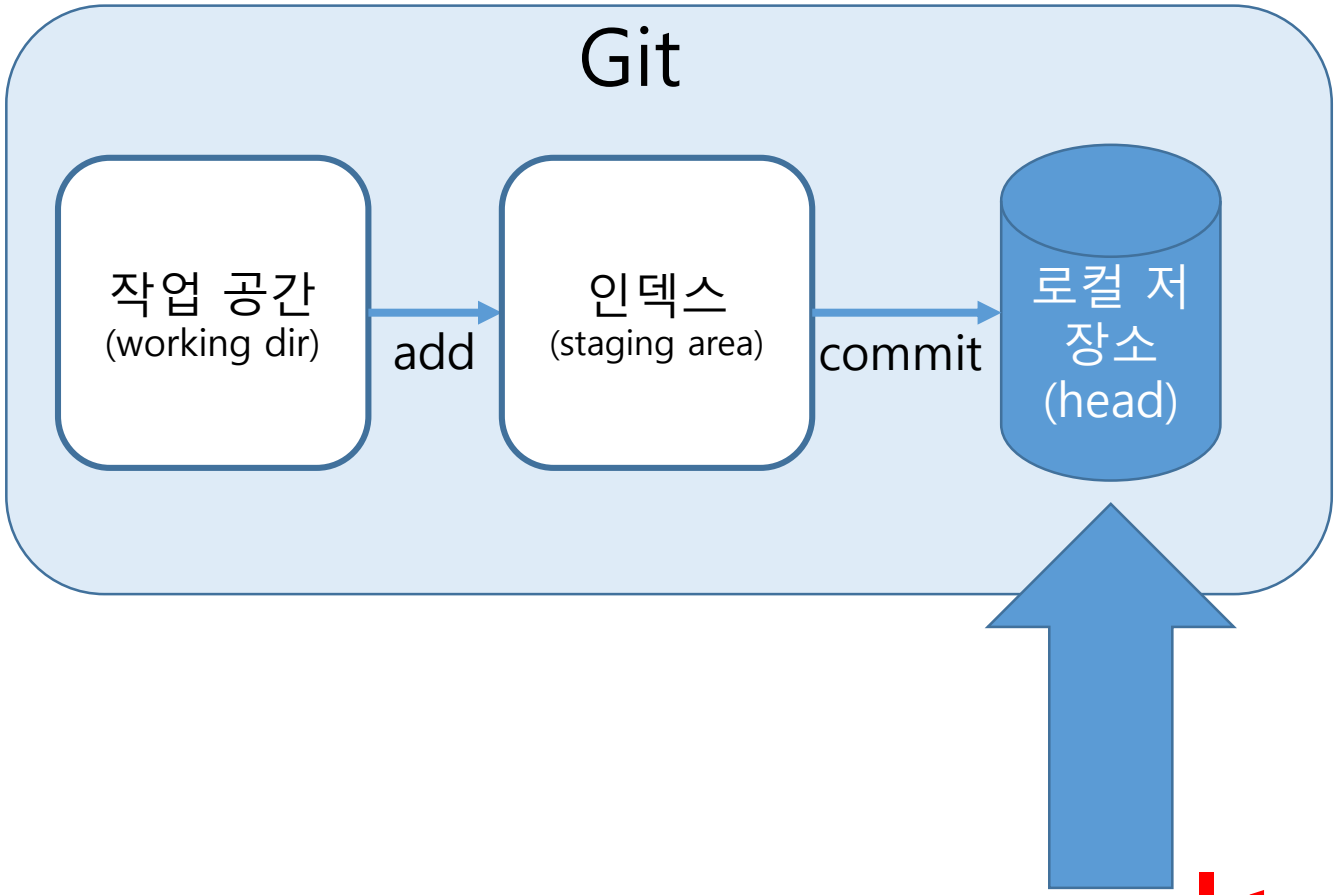
- 병합할 커밋이름을 넣어 실행하면 지정한 커밋 내용이 Head가 가리키고 있는 브랜치에 병합된다.

Merge(병합하기)

result += num

Merge(병합하기)

result += num


Git
작업 공간
(working dir)
add
인덱스
(staging area)
commit
로컬 저장소
(head)
result

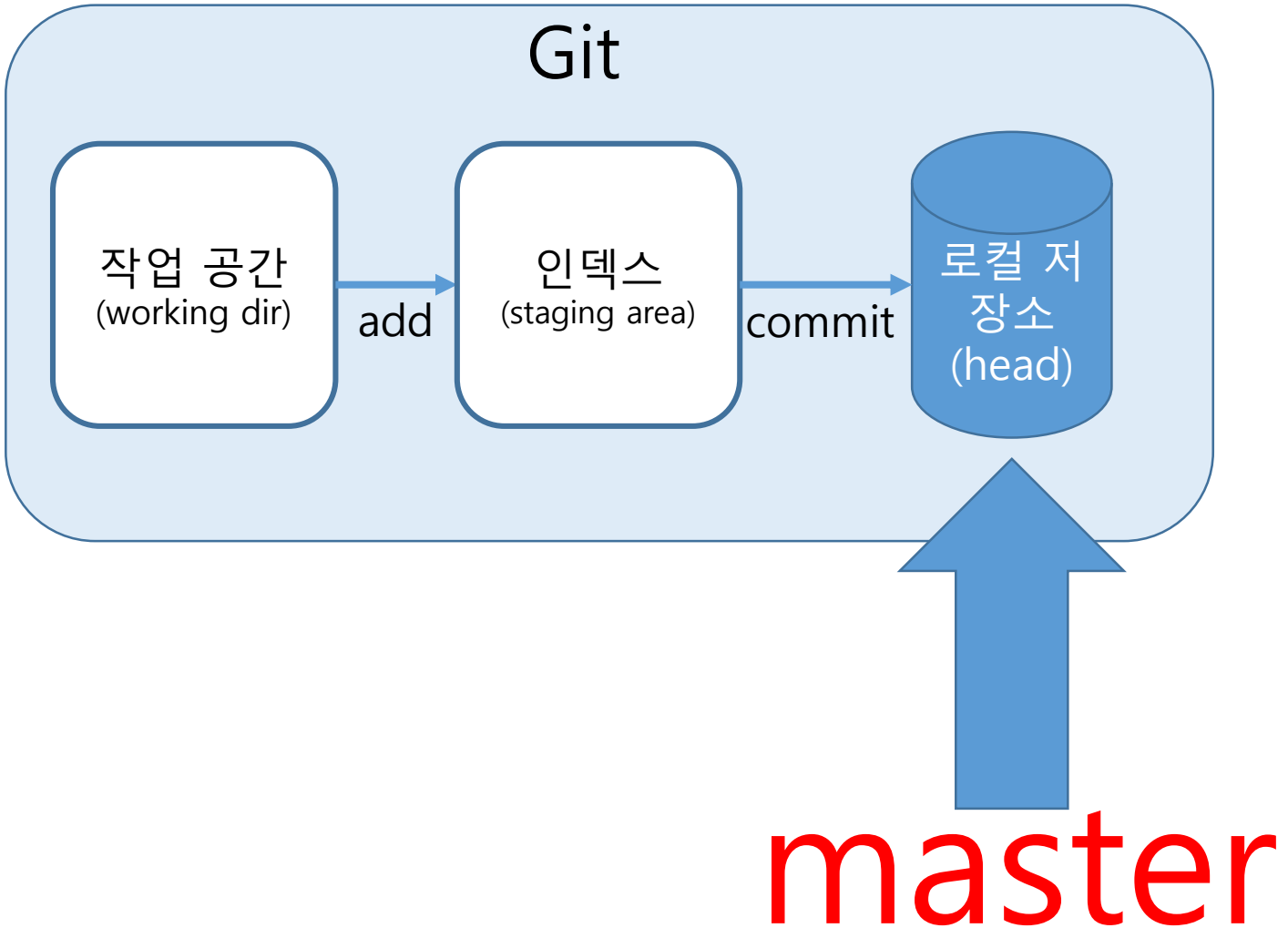
Merge(병합하기)

master += branch

Merge(병합하기)

- 병합하기 전, master 브랜치를 chechout 해서 Head에 위치시켜야 한다.

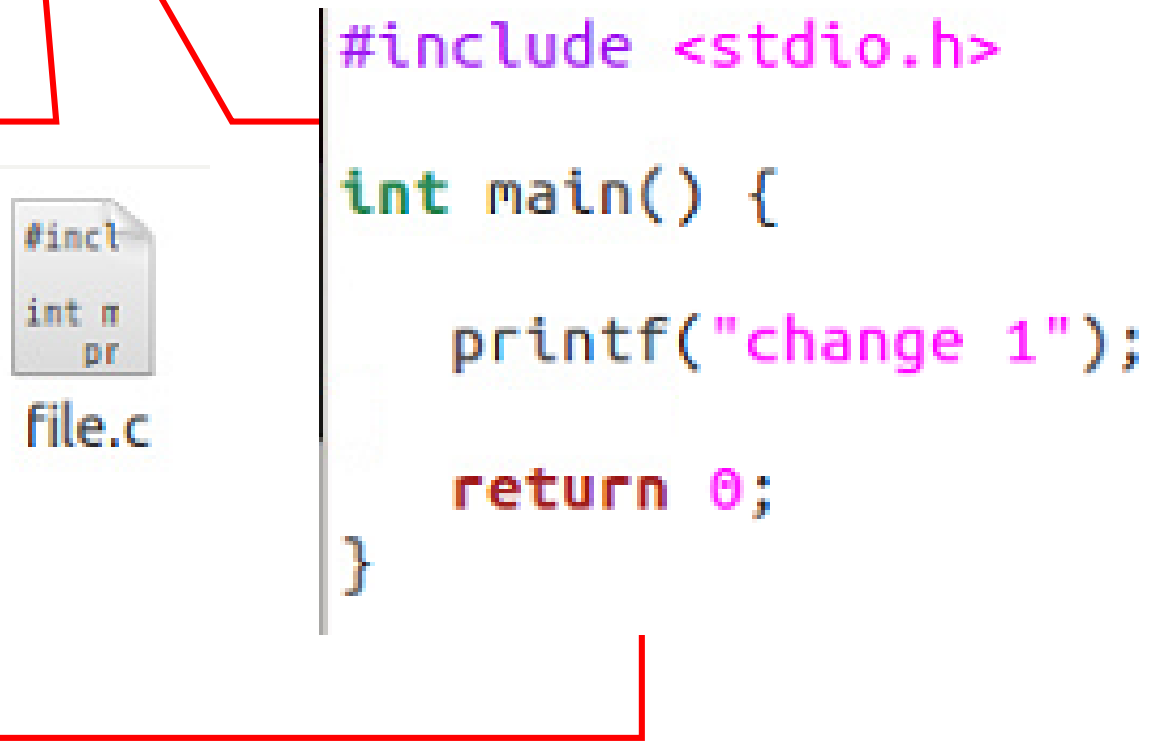
- git checkout master


Git
작업 공간
(working dir)
add
인덱스
(staging area)
commit
로컬 저장소
(head)
master

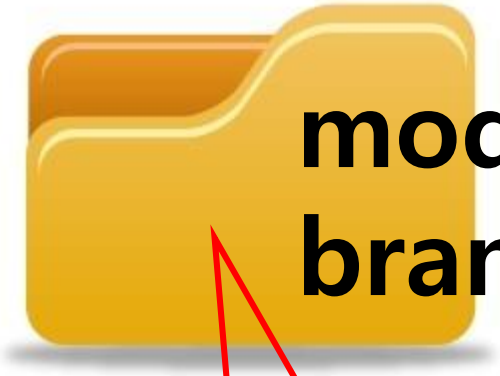
master
branch

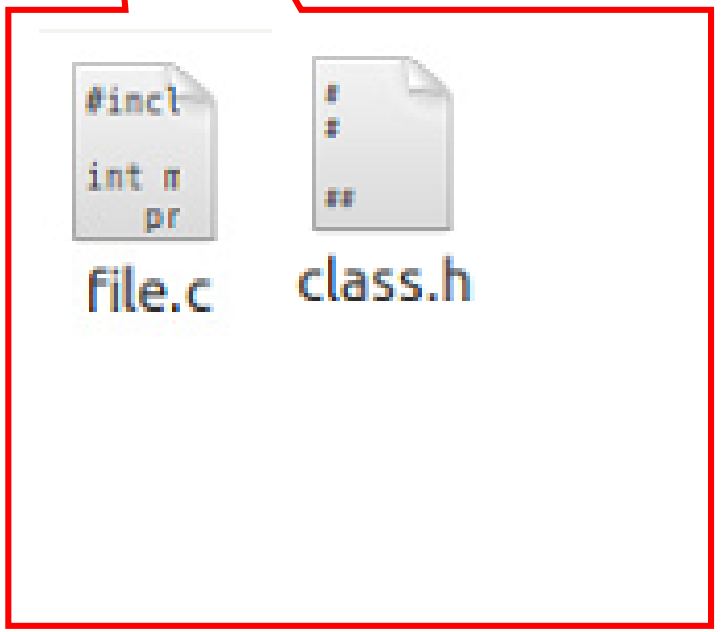
modified
branch

master branch



modified branch





master branch

file.c

```c
#include <stdio.h>

int main() {

    printf("change 1");

    return 0;
}
```

modified branch

```c
#include <stdio.h>

int main() {

    printf("change 1");
    printf("Happy New Year");

    return 0;
}
```

file.c

class.h

master branch

file.c 수정
&
class.h 추가

modified branch

```c
#include <stdio.h>

int main() {

    printf("change 1");

    return 0;
}
```

file.c

```c
#include <stdio.h>

int main() {

    printf("change 1");
    printf("Happy New Year");

    return 0;
}
```

file.c

class.h

데모

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch modified
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
* master
  modified
hansangyun@hansangyun:~/바탕화면/git_example$ git checkout modified
D       file 2.c
M       file.c
M       file.c~
D       header.h
Switched to branch 'modified'
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
  master
* modified
```

데모


hansangyun@hansangyun:~/바탕화면/git_example$ git branch modified
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
* master
modified
hansangyun@hansangyun:~/바탕화면/git_ t modified
D file 2.c
M file.c
M file.c~
D header.h
Switched to branch 'modified'
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
master
* modified
'modified' 브랜치 생성

데모

```
hansangyun@hansangyun:~/바탕화면/git_example$ git branch modified
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
* master
  modified
hansangyun@hansangyun:~/바탕화면/git_example$ git checkout modified
D       file 2.c
M       file.c
M       file.c~
D       header.h
Switched to branch 'modifi
hansangyun@hansangyun:~/바탕화면/git_example$ git branch
  master
* modified
```

'modified' 브랜치 변경

데모

```
hansangyun@hansangyun:~/바탕화면/git_example$ git commit -m "commit modified bra
nch"
[modified 2f40d53] commit modified branch
 3 files changed, 7 insertions(+), 1 deletion(-)
 create mode 100644 class.h
hansangyun@hansangyun:~/바탕화면/git_example$ git checkout master
D       file 2.c
D       header.h
Switched to branch 'master'
```

- 변경된 modified 브랜치로 변경 내역 commit
- master 브랜치를 checkout으로 가져오기

데모

```
hansangyun@hansangyun:~/바탕화면/git_example$ git merge modified
Updating 71b9cdf..2f40d53
Fast-forward
 class.h | 4 ++++
 file.c  | 3 +++
 file.c~ | 1 -
 3 files changed, 7 insertions(+), 1 deletion(-)
 create mode 100644 class.h
```

- merge

- '+' 몇 줄 추가되었는지,

Merge(병합하기)

master += modified

차이점 확인하기

```c
#include <stdio.h>

int main() {

    printf("change 1");

    return 0;
}
```

```c
#include <stdio.h>

int main() {

    printf("Happy New Year");

    return 0;
}
```

- git diff

차이점 확인하기

```
hansangyun@hansangyun:~/바탕화면/git_example$ git diff
diff --git a/file.c b/file.c
index 2959482..a245c3f 100644
--- a/file.c
+++ b/file.c
@@ -2,7 +2,7 @@

 int main() {

-    printf("change 1");
+    printf("Happy New Year");

     return 0;
 }
```

차이점 확인하기

- git diff : 현재 작업 트리와 인덱스의 차이점 보기
- git diff HEAD : 작업 트리와 저장소의 차이점 보기

- git diff --cached : 인덱스와 저장소의 차이점 보기
- git diff 시작지점 : 작업 트리와 특정 위치 간의 차이점 보기 (시작지점은 커밋명이나 브랜치명, 파일명)
- git diff 시작지점 끝지점 : 저장소의 두 지점 사이의 차이점 보기

수정사항 및 충돌 다루기

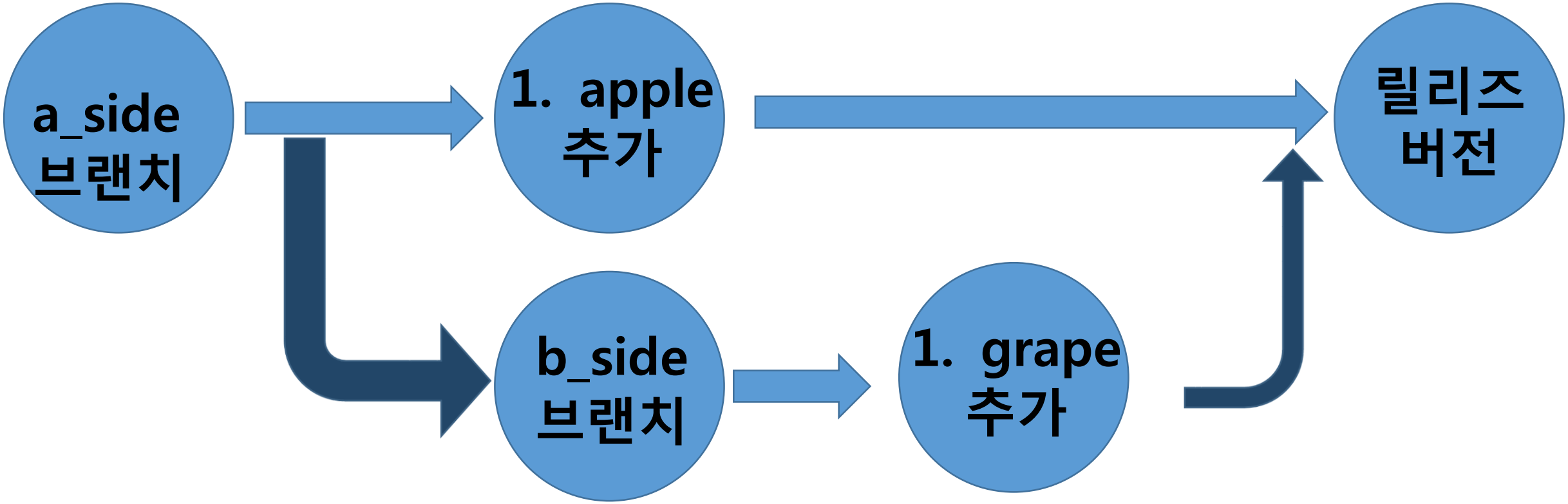
충돌 다루기

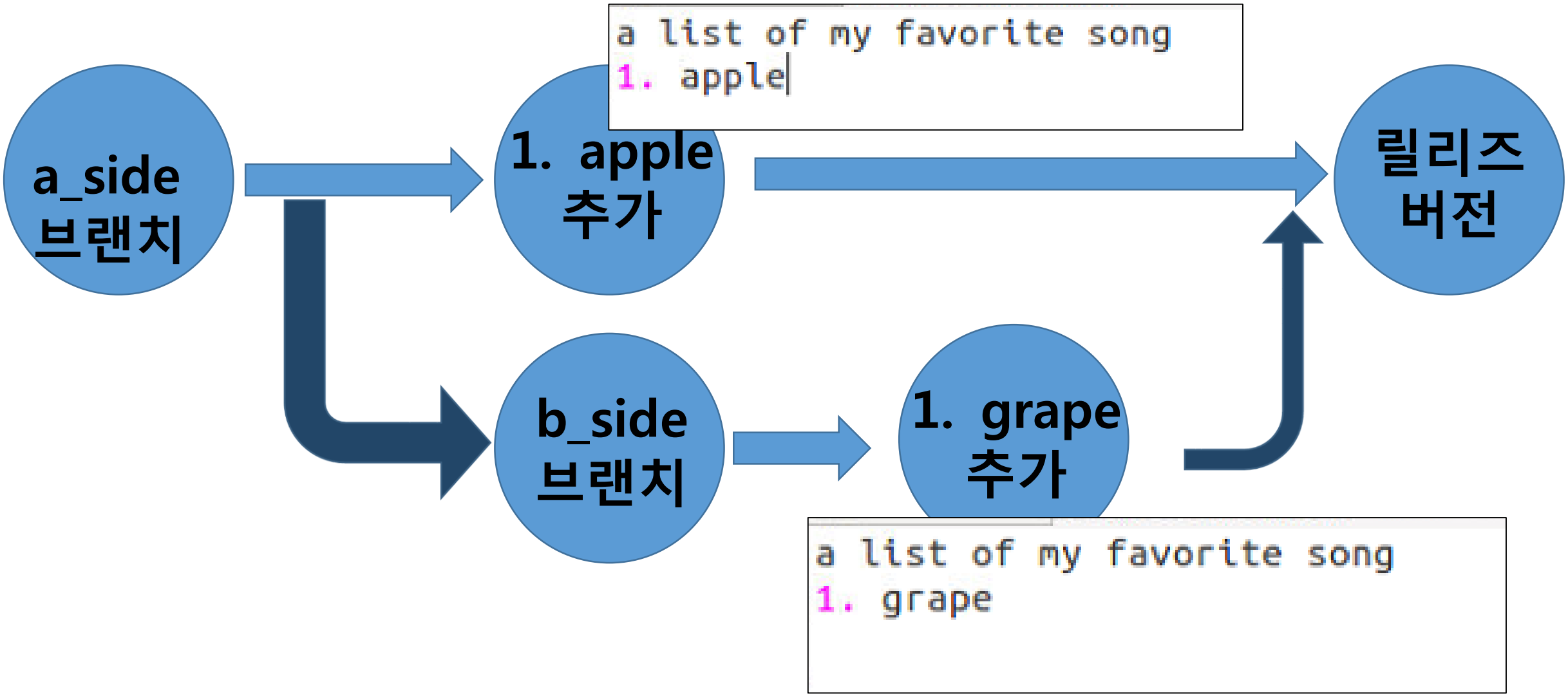
blame 명령어

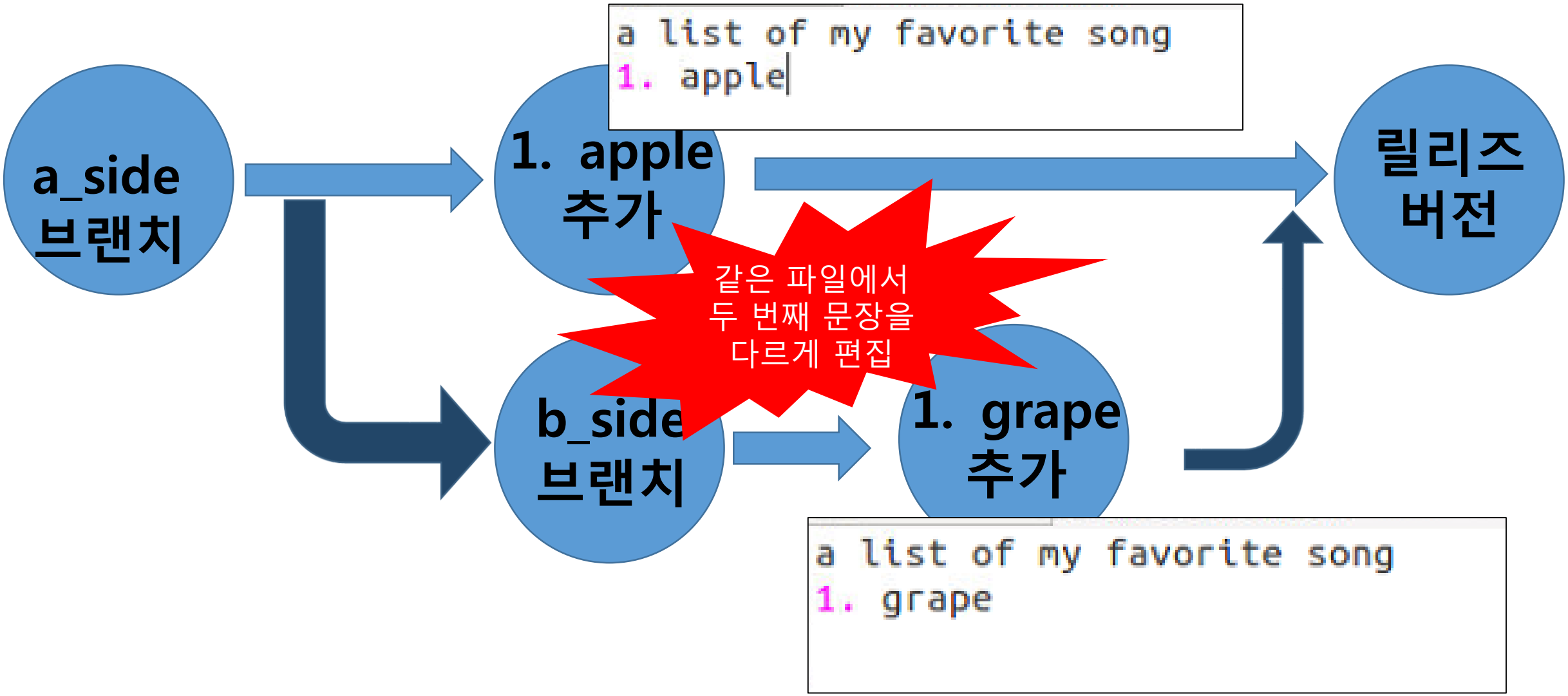
충돌 다루기

- 두 개의 다른 브랜치에서 동일한 파일을 다르게 편집한 후 합칠 때 충돌이 발생

- 이때 git에서 자동으로 merge를 할 수 없게 된다.


a_side
브랜치
1. apple
추가
릴리즈
버전
b_side
브랜치
1. grape
추가

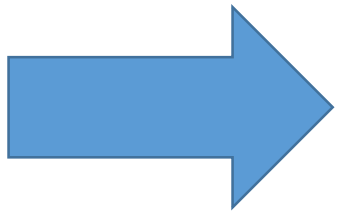

a list of my favorite song
1. apple
a_side
브랜치
1. apple
추가
릴리즈
버전
b_side
브랜치
1. grape
추가
a list of my favorite song
1. grape


a list of my favorite song
1. apple
a_side
브랜치
1. apple
추가
릴리즈
버전
같은 파일에서
두 번째 문장을
다르게 편집
b_side
브랜치
1. grape
추가
a list of my favorite song
1. grape

충돌 발생

```
hansangyun@hansangyun:~/바탕화면/git_example$ git merge b_side
Auto-merging side.h
CONFLICT (content): Merge conflict in side.h
Automatic merge failed; fix conflicts and then commit the result.
```

```
a list of my favorite song
<<<<<<< HEAD
1. apple
=======
1. grape
>>>>>>> b_side
```



- 충돌이 난 파일에 충돌 사항에 대해 로깅이 된다.

수정사항 추적하기(누구 책임인지 찾기)

- git blame 파일이름
- 파일의 각 줄 앞에 커밋명, 커밋한 사용자, 시간 정보 출력

```c
#include <stdio.h>

int main() {

    printf("Happy New Year");

    return 0;
}
```

수정사항 추적하기(누구 책임인지 찾기)

- git blame 파일이름
- 파일의 각 줄 앞에 커밋명, 커밋한 사용자, 시간 정보 출력

```
hansangyun@hansangyun:~/바탕화면/git_example$ git blame file.c
^614bffd (sangyunHan 2014-12-26 18:23:45 +0900 1) #include <stdio.h>
^614bffd (sangyunHan 2014-12-26 18:23:45 +0900 2)
^614bffd (sangyunHan 2014-12-26 18:23:45 +0900 3) int main() {
2f40d53a (sangyunHan 2015-01-02 15:20:08 +0900 4)
2f40d53a (sangyunHan 2015-01-02 15:20:08 +0900 5)     printf("Happy New Year");
2f40d53a (sangyunHan 2015-01-02 15:20:08 +0900 6)
^614bffd (sangyunHan 2014-12-26 18:23:45 +0900 7)     return 0;
^614bffd (sangyunHan 2014-12-26 18:23:45 +0900 8) }
```

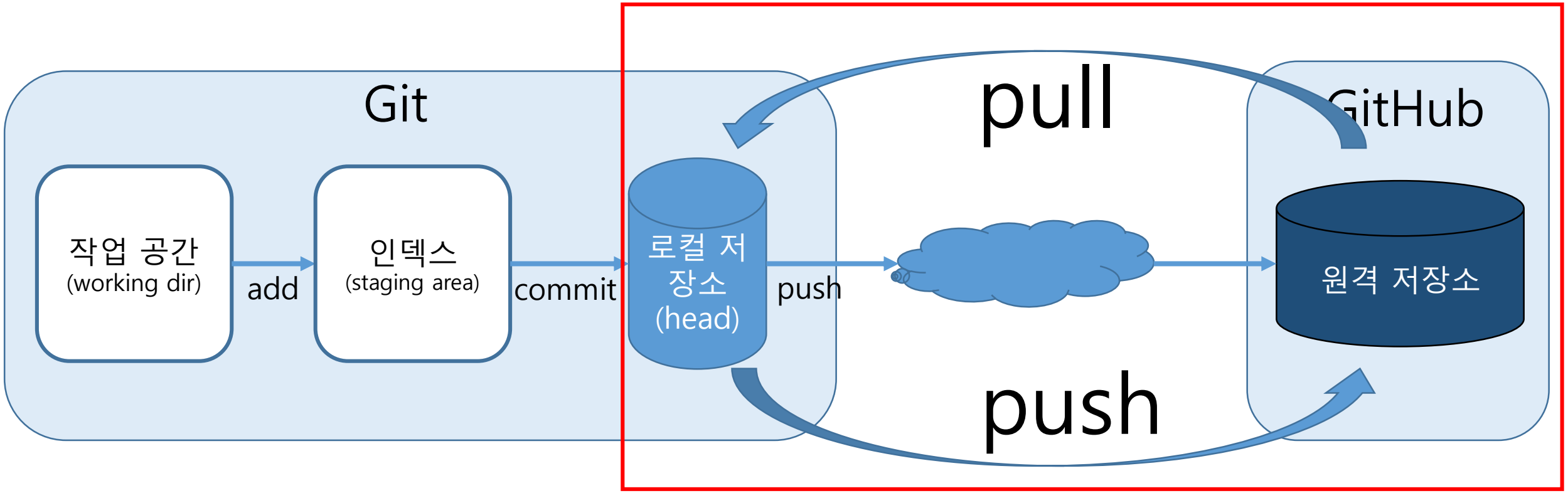
원격 저장소 – push/pull

로컬 저장소 -> 원격 저장소 연결

- git remote add origin git@github.com:사용자ID/저장소이름.git
- origin:원격저장소 / master:로컬 저장소

- 원격 저장소 주소
  - SSH : git@github.com:아이디/프로젝트.git
  - git 프로토콜 : git://github.com/아이디/프로젝트.git
  - HTTPS : https://github.com/아이디/프로젝트.git

push와 pull


Git
작업 공간
(working dir)
add
인덱스
(staging area)
commit
로컬 저장소
(head)
push
pull
push
GitHub
원격 저장소

pull

- git pull 원격저장소
  원격 저장소에서 변경 사항을 가져와 현재 브랜치에 합치기

- git pull
  origin 저장소에서 변경 사항을 가져와 현재 브랜치에 합치기

push

- git push
  지역 변경 사항을 origin 저장소에 푸싱

- git push <원격저장소> <지역브랜치>
  지역 브랜치를 동인한 이름의 원격브랜치에 푸싱

- git push <원격저장소> <지역브랜치>:<원격브랜치>
  지역 브랜치를 원격 브랜치에 푸싱

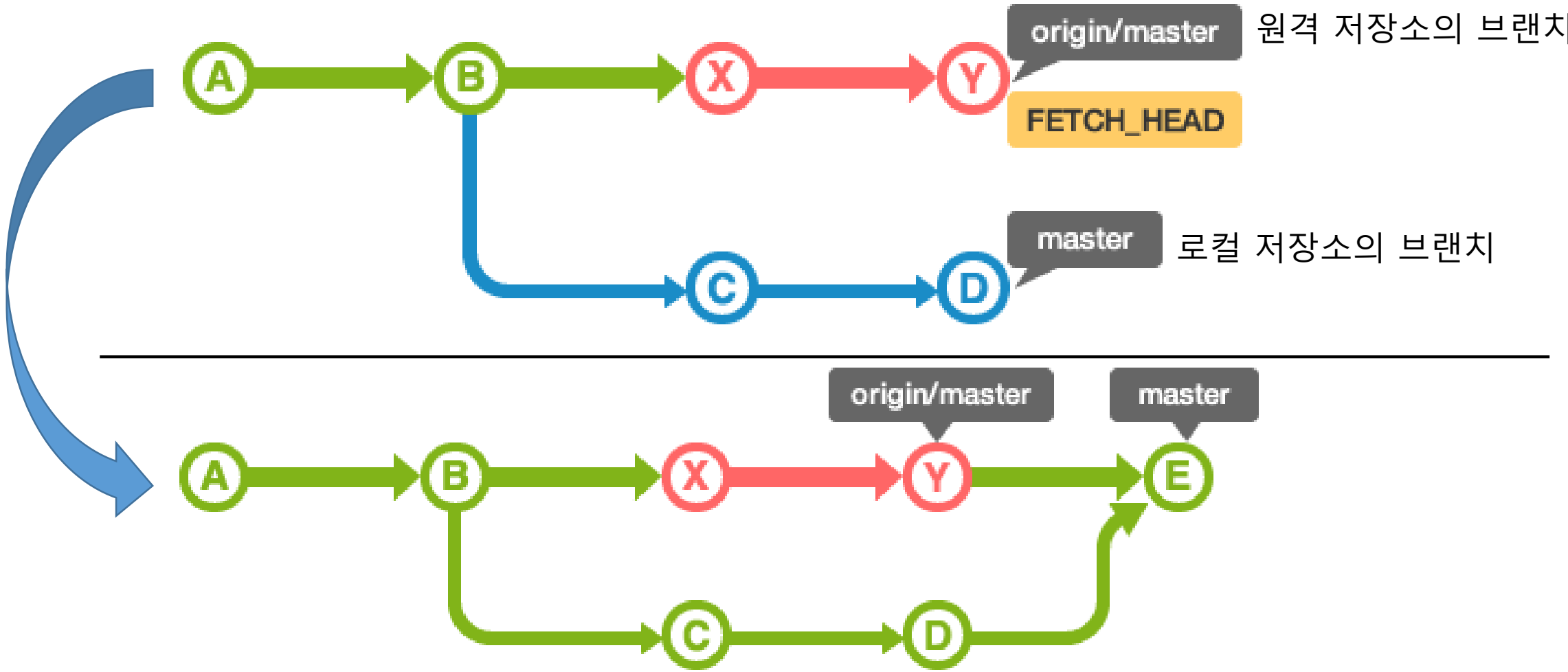
fetch

- pull 명령어 같은 경우 특정 브랜치나 커밋을 가져와서 현재 브랜치에 병합해버리는 한계가 존재

- fetch를 사용하면 특정 브랜치(FETCH_HEAD)에 가져올 수 있다.

- pull = fetch + merge

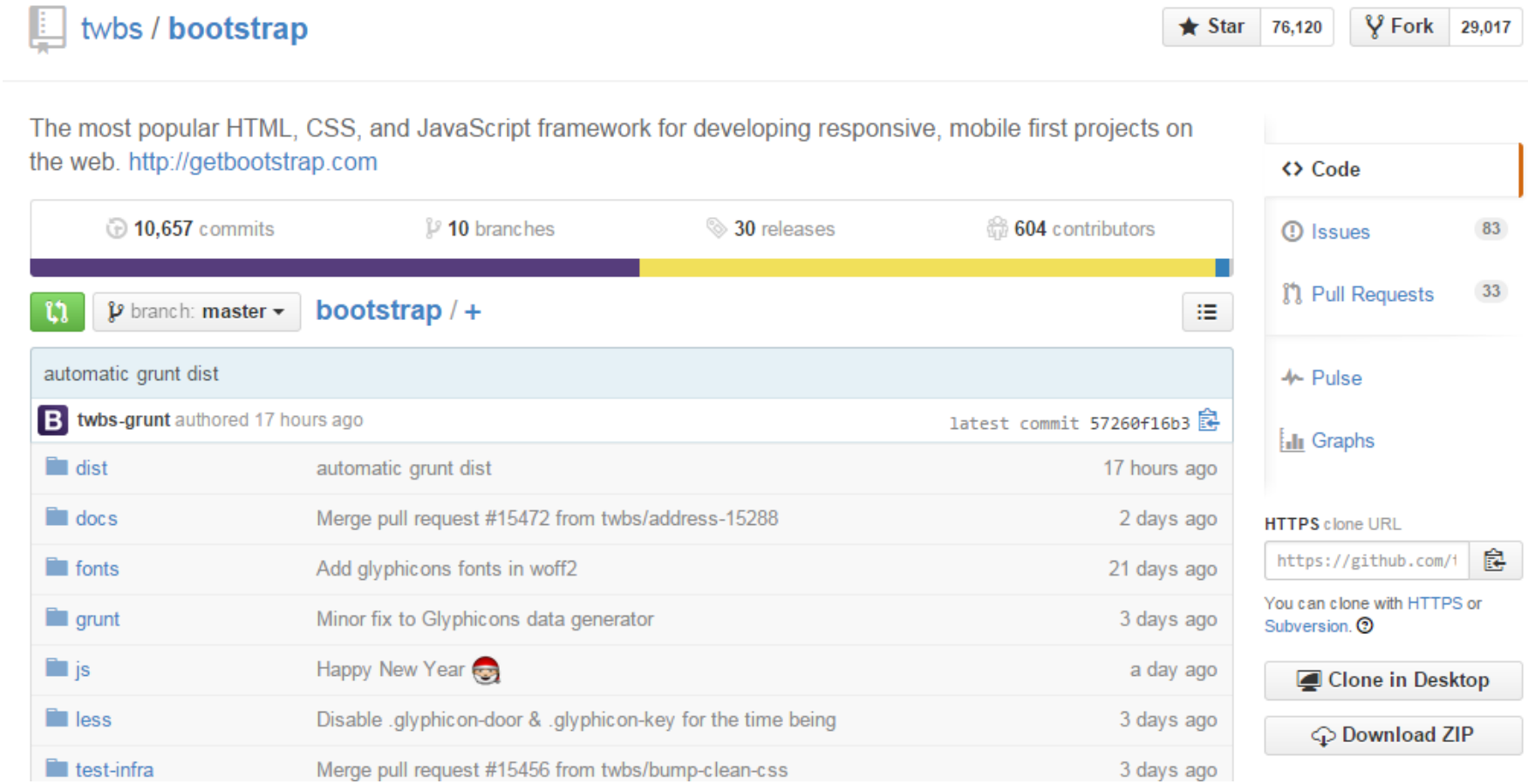
fetch


origin/master
원격 저장소의 브랜치
A
B
X
Y
FETCH_HEAD
master
로컬 저장소의 브랜치
C
D
origin/master
master
A
B
X
Y
E
C
D

GitHub 자체 기능

Introducing GitHub

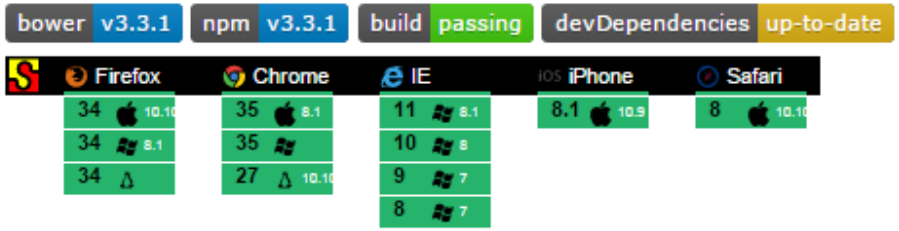
프로젝트 소개 페이지


twbs / bootstrap
Star 76,120
Fork 29,017
The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web. http://getbootstrap.com
10,657 commits
10 branches
30 releases
604 contributors
branch: master
bootstrap / +
automatic grunt dist
twbs-grunt authored 17 hours ago
latest commit 57260f16b3
dist automatic grunt dist 17 hours ago
docs Merge pull request #15472 from twbs/address-15288 2 days ago
fonts Add glyphicons fonts in woff2 21 days ago
grunt Minor fix to Glyphicons data generator 3 days ago
js Happy New Year a day ago
less Disable .glyphicon-door & .glyphicon-key for the time being 3 days ago
test-infra Merge pull request #15456 from twbs/bump-clean-css 3 days ago
Code
Issues 83
Pull Requests 33
Pulse
Graphs
HTTPS clone URL
https://github.com/t
You can clone with HTTPS or Subversion.
Clone in Desktop
Download ZIP

readme 웹 뷰

README.md

Bootstrap

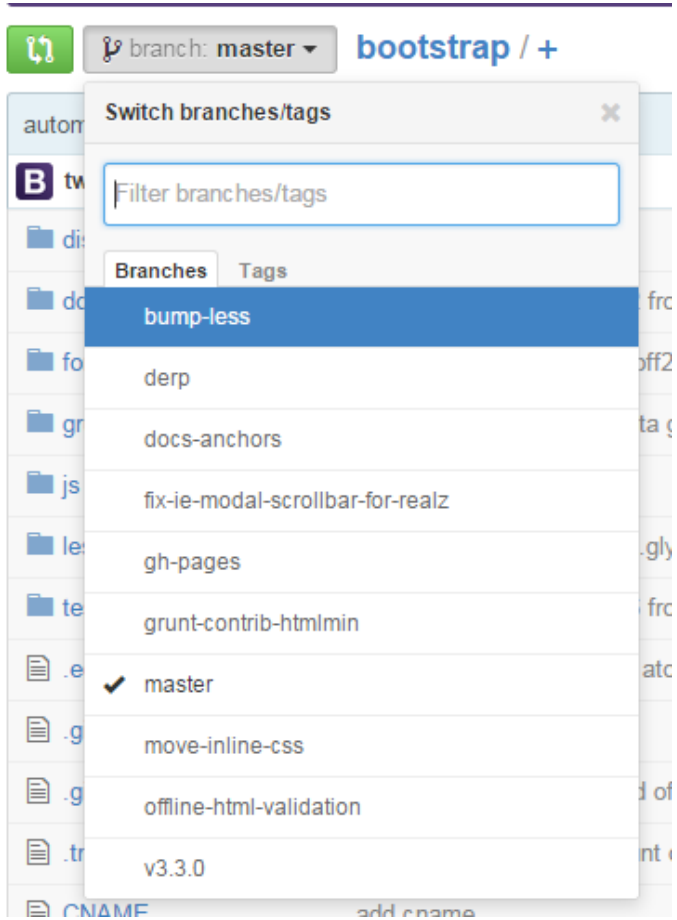


Bootstrap is a sleek, intuitive, and powerful front-end framework for faster and easier web development, created by Mark Otto and Jacob Thornton, and maintained by the core team with the massive support and involvement of the community.

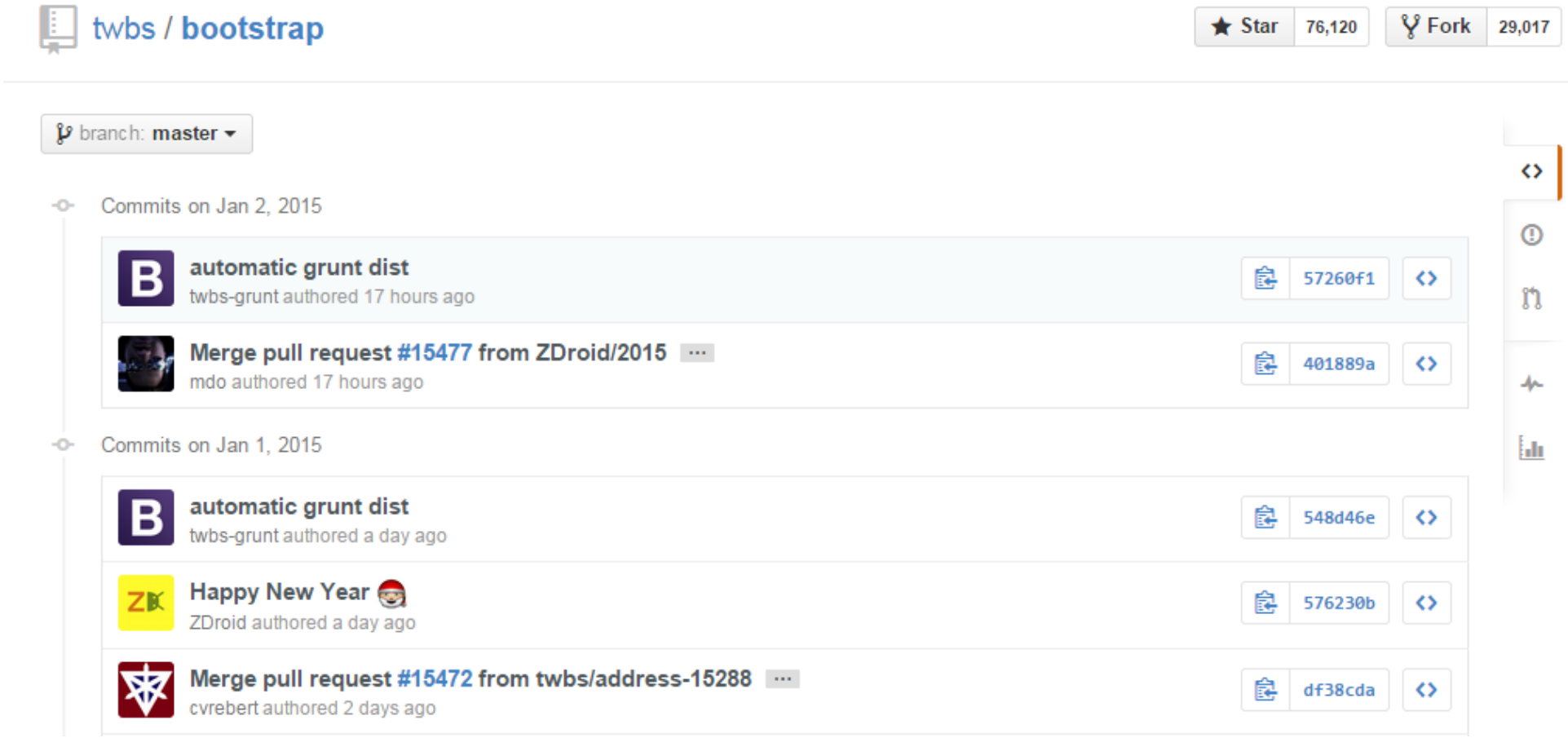
To get started, check out http://getbootstrap.com!

Table of contents

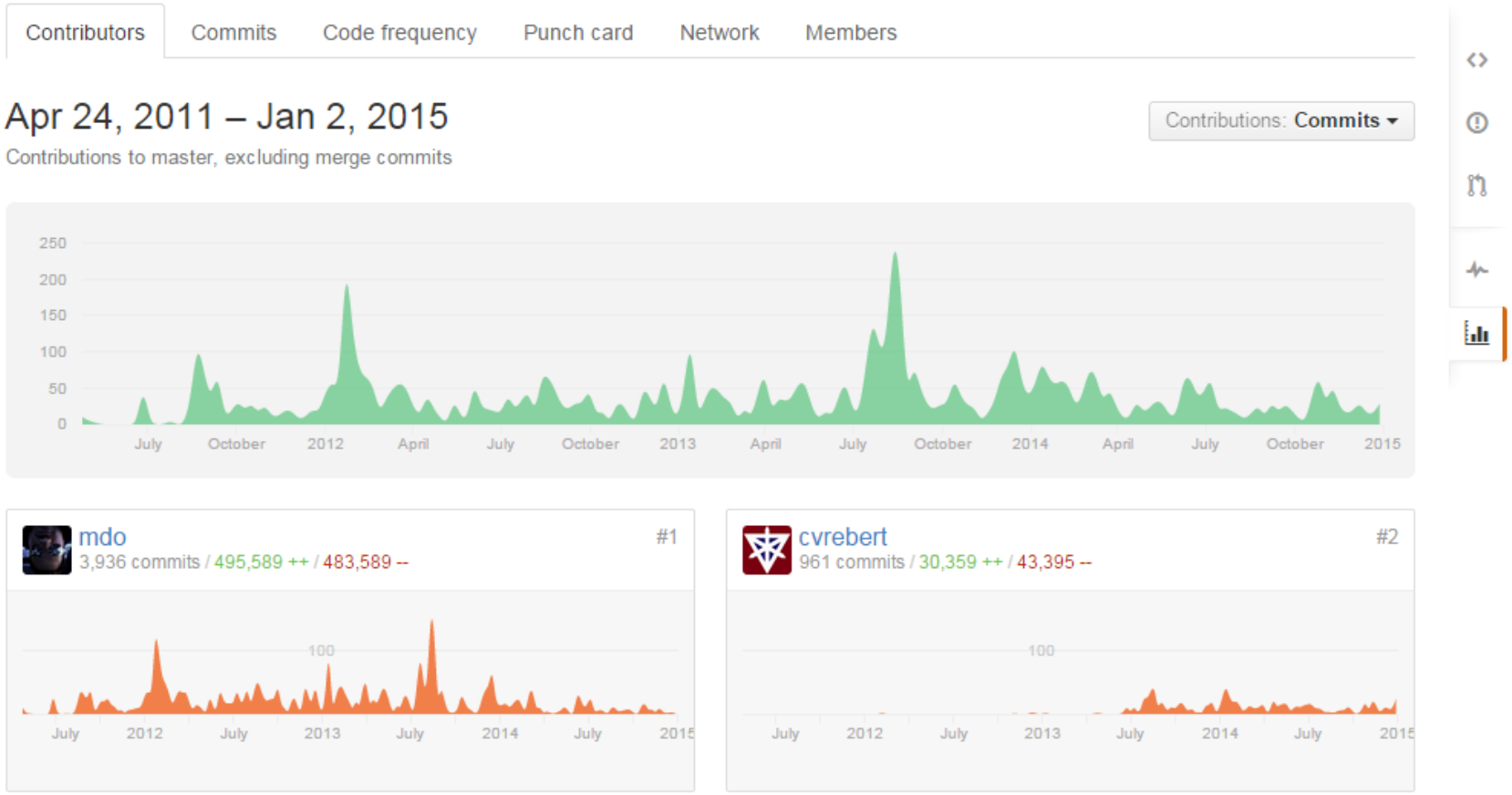
프로젝트 브랜치의 정보


branch: master
bootstrap / +
Switch branches/tags
Filter branches/tags
Branches
Tags
bump-less
derp
docs-anchors
fix-ie-modal-scrollbar-for-realz
gh-pages
grunt-contrib-htmlmin
master
move-inline-css
offline-html-validation
v3.3.0

커밋에 대한 정보(git log 기능)



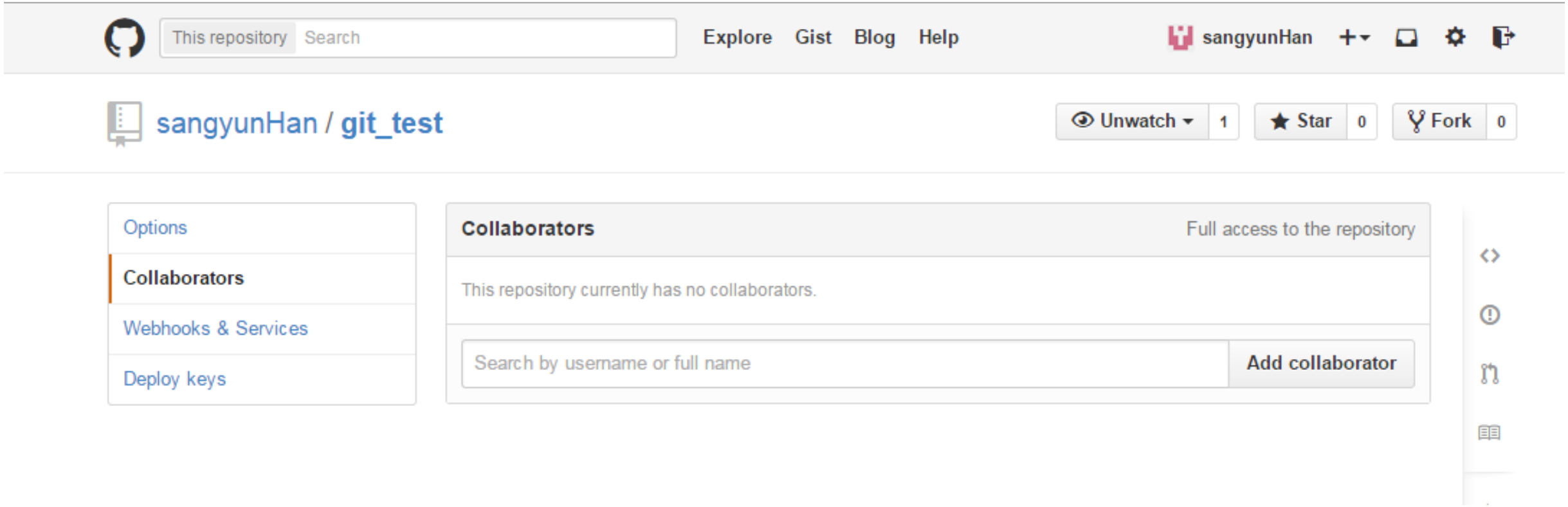
contributo별 커밋 그래프



주간 변경된 코드량


Contributors
Commits
Code frequency
Punch card
Network
Members
Additions and Deletions per week
80k
60k
40k
20k
0
20k
40k
60k
80k
100k
07/11
10/11
01/12
04/12
07/12
10/12
01/13
04/13
07/13
10/13
01/14
04/14
07/14
10/14

collaborator 추가



Open HUB 소개

OpenHUB or Ohloh

- 66만개 이상의 프로젝트들이 등록된 통계 사이트
- 개발자 정보 / 프로젝트 정보를 열람할 수 있다.
- github을 포함한 수많은 호스팅 서비스에 등록된 프로젝트와 openflow, linux와 같이 특정 단체에서 따로 관리하는 프로젝트까지 모두 볼 수 있다.

오픈소스 계의

C

Linus.Torvalds

Portland, OR, USA |

Account Summary

Analyzed 3 days ago

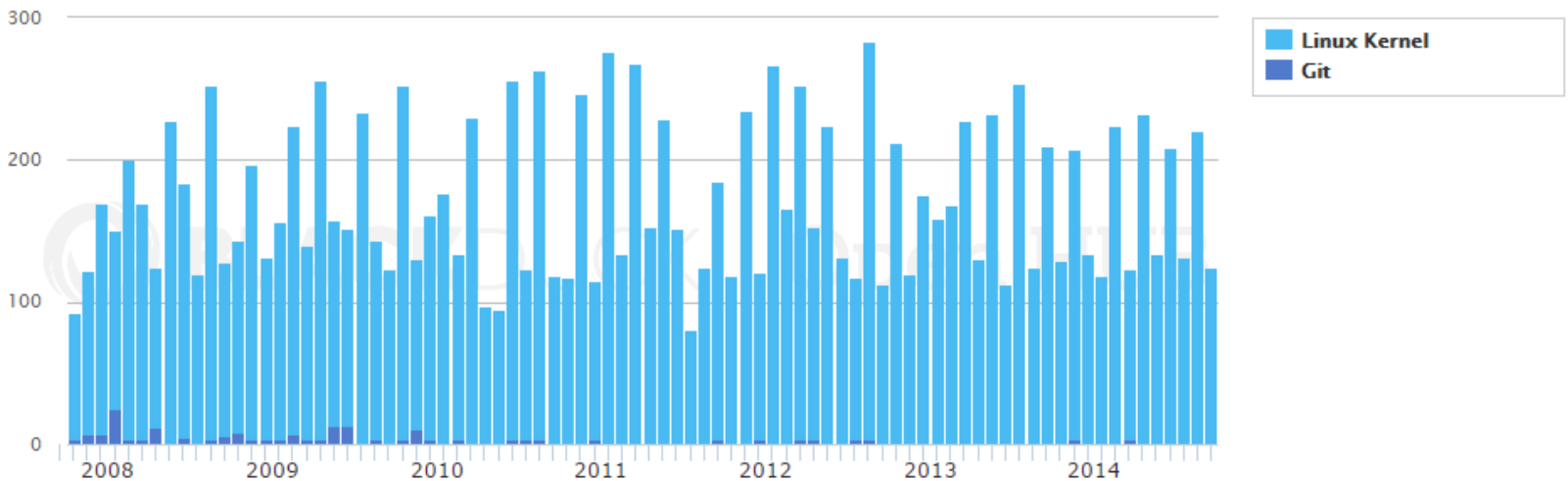
Projects Used

Most experienced in	C
First commit	almost 13 years ago
Most recent commit	12 days ago
Has made	23082 commits
Joined Open Hub	over 7 years ago
Contributed to	2 projects

Development History

Commits by Project

Jan 2008 - Present



C

Linus.Torvalds

Portland, OR, USA |

경희대학교
KYUNG HEE UNIVERSITY

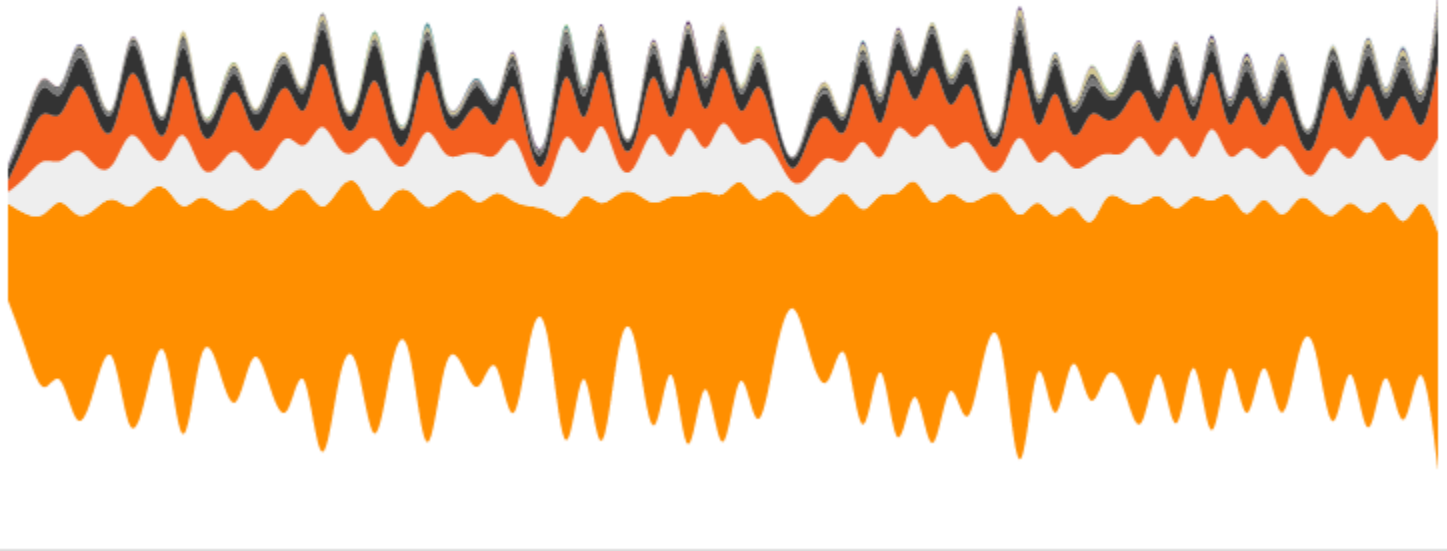
Accour

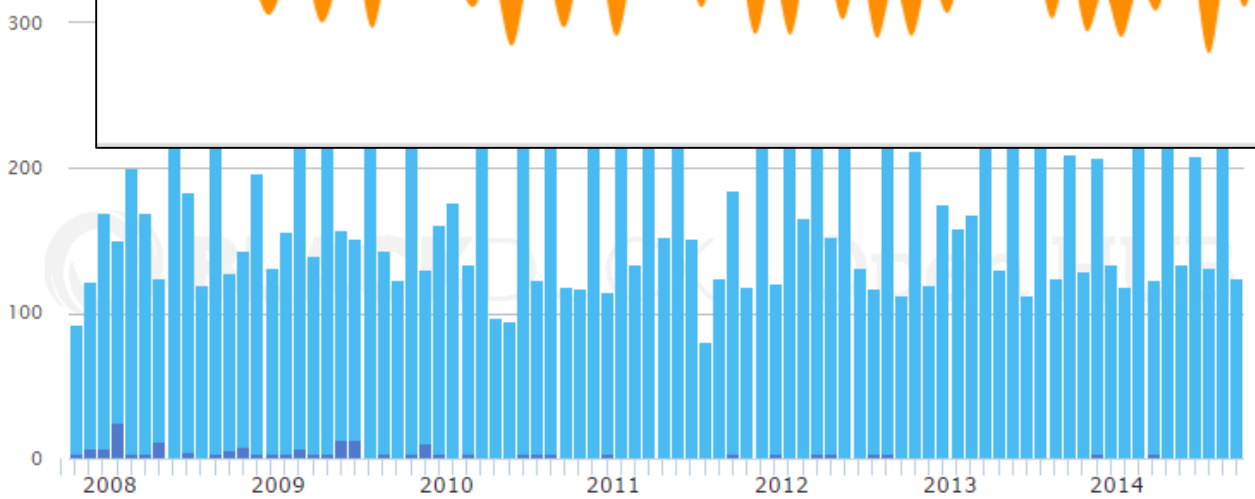
Projec

Develop

Comm

Commits by Language *Jan 2008 - Present*






OpenDaylight
Compare
Analyzed 7 days ago
1.92M lines of code
Very High Activity
The OpenDaylight Project is a collaborative open source project that aims to accelerate adoption of Software-Defined Networking (SDN) and create a solid foundation for Network Functions Virtualization (NFV) for a more transparent approach that fosters new innovation and reduces risk. Founded by industry leaders and open to all, the Ope... [More]
263 current contributors
8 days since last commit
12 users on Open Hub
0 Reviews
Mostly written in Java
Licenses: EPL-1.0
networking
sdn
network
java
nfv

OpenDaylight

Settings | Report Duplicate

Very High Activity

12 I Use This!

경희대학교 KYUNG HEE UNIVERSITY

Analyzed 7 days ago based on code collected 7 days ago.

Project Summary

The OpenDaylight Project is a collaborative open source project that aims to accelerate adoption of Software-Defined Networking (SDN) and create a solid foundation for Network Functions Virtualization (NFV) for a more transparent approach that fosters new innovation and reduces risk. Founded by industry leaders and open to all, the OpenDaylight community is developing a common, open SDN framework consisting of code and blueprints. Get involved: www.opendaylight.org.

OpenDaylight is a Collaborative Project at The Linux Foundation. Linux Foundation Collaborative Projects are independently funded software projects that harness the power of collaborative development to fuel innovation across industries and ecosystems. www.linuxfoundation.org

Quick Reference

Project Links: Homepage

Code Locations: *(30 Locations)*

Licenses: EPL-1.0

Similar Projects:

Managers: Madhu Venugopal

Browse Code

Tags

networking sdn network java nfv

Share

좋아요 0 Tweet 0 +1 0 Share 1

In a Nutshell, OpenDaylight...

... has had 13,895 commits made by 310 contributors representing 1,922,725 lines of code

... is mostly written in Java with an average number of source code comments

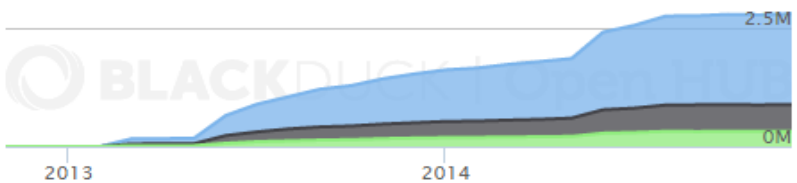
... has a young, but established codebase maintained by a very large development team with increasing Y-O-Y commits

... took an estimated 551 years of effort (COCOMO model) starting with its first commit in November, 2012 ending with its most recent commit 8 days ago

Languages

Java	52%	C++	18%
C	10%	18 Other	20%

Lines of Code



OpenDaylight

Settings | Report Duplicate

Very High Activity

12 I Use This!

경희대학교 KYUNG HEE UNIVERSITY

Analyzed 7 days ago based on code collected 7 days ago.

Project Summary

The OpenDaylight Project is a collaborative open source project that aims to accelerate adoption of Software-Defined Networking (SDN) and create a solid foundation for Network Functions Virtualization (NFV) for a more transparent approach that fosters new innovation and reduces risk. Founded by industry leaders and open to all, the OpenDaylight community is developing a common, open SDN framework consisting of code and blueprints. Get involved: www.opendaylight.org.

OpenDaylight is a Collaborative Project at The Linux Foundation. Linux Foundation Collaborative Projects are independently funded software projects that harness the power of collaborative development to fuel innovation across industries and ecosystems. www.linuxfoundation.org

Quick Reference

Project Links: Homepage
Code Locations: *(30 Locations)*
Licenses: EPL-1.0
Similar Projects:
Managers: Madhu Venugopal

Browse Code

Tags

networking sdn network java nfv

Share

좋아요 0 Tweet 0 +1 0 Share 1

In a Nutshell, OpenDaylight...

... has had 13,895 commits made by 310 contributors representing 1,922,725 lines of code

... is mostly written in Java with an average number of source code comments

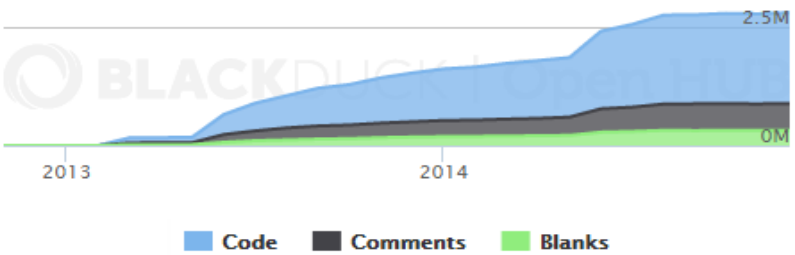
... has a young, but established codebase maintained by a very large development team with increasing Y-O-Y commits

... took an estimated 551 years of effort (COCOMO model) starting with its first commit in November, 2012 ending with its most recent commit 8 days ago

Languages

Java	52%	C++	18%
C	10%	18 Other	20%

Lines of Code



참고

- http://git-scm.com/
- Git, 분산버전 관리시스템, 트라비스 스위스굿
- Git 어떻게 동작하는가?, John Wiegley
- http://backlogtool.com/git-guide/kr/

Q&A