

DESIGN VERSIONING GUIDE

Contents

01. Git

- Git Install
- Git Basic

02. GitLab

- GitLab Information (Address, Login)

03. GitLab Team Setting

- Fork Parent Repository
- Add Members
- Merge Request After Project

04. Git for Designer

- Software Install (Desktop)
- SourceTree Setting
- Clone Repository
- Commit
- Push & Pull
- Branch
- Merge
- Checkout

01 Git

02 GitLab

03 GitLab Team

04 Git for Designer

GIT

01-1

Git Install

깃 설치

1. Git Install

2. Git Basic

Download Address

<https://git-scm.com/>

Git Document

<https://git-scm.com/book/ko/v2>

Git Install Check

Terminal > Enter "git"

* Normal Screen, if it is not, install again

```
[HOY:~ Hoy$ git
usage: git [--version] [-h] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [-m <path>] [--info-path]
           [-p | --paginate | --no-pager] [-n <path>] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv        Move or rename a file, a directory, or a symlink
  reset     Reset current HEAD to the specified state
  rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  branch   List, create, or delete branches
  checkout Switch branches or restore working tree files
  commit   Record changes to the repository
  diff     Show changes between commits, commit and working tree, etc
  merge   Join two or more development histories together
  rebase   Reapply commits on top of another base tip
  tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch   Download objects and refs from another repository
  pull    Fetch from and integrate with another repository or a local branch
  push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```

Learn More Git : <https://git-scm.com/>

01-2

Git Basic

Learn More Git : <https://git-scm.com/>

깃 기본

1. Git Install

2. Git Basic

Repository [= repo]

프로젝트의 디렉토리나 저장 공간. 당신의 컴퓨터 안의 로컬 폴더가 될 수도 있고,
깃허브나 다른 온라인 호스트의 저장 공간이 될 수도 있다.

Fork vs Merge Request

포크(Fork)는 저장소에 있는 UI Kit이나 다른 도메인의 디자인 파일을 그대로 복제하는 기능이다.
복제된 파일은 원본에 영향을 주지 않고 자유롭게 프로젝트를 진행 할 수 있다.
작업이 끝난 파일을 원본 파일 반영하기 위해서는 병합 요청(Merge Reuqest)해야 한다.

*3-1 참조

Commit

커밋(Commit)하면, 진행되는 프로젝트 시점의 저장소 상태를 기록해 프로젝트를 이전으로 돌아가거나
복원할 수 있는 체크포인트를 가질 수 있다.

*4-4 참조

Pull & Push

풀(Pull)은 다운로드(Download), 푸시(Push)는 업로드(Upload)로 이해하면 쉽다.
커밋(Commit)을 통해 작업된 기록들을 원격저장소인 깃랩(Gitlab)에 올리기 위해서 푸시(Push)를 한다.
미리 업로드한 작업물을 내려받거나 다른 작업자가 작업한 작업물을 내려 받기 위해서는 풀(Pull)을 하면 된다.

*4-5, 6, 7 참조

Branch

여러 명이 하나의 프로젝트에서 작업할 때, 파일을 받아서 이름을 변경하고 자신만의 버전을 만들어 작업한다.
일반적으로 브런치(Branch)는 이럴때 미리 자신 만의 브런치(Branch)를 만들 수 있는 기능이다.
작업을 끝낸 후, 프로젝트의 메인 디렉토리인 “마스터(Master)”에 브랜치를 다시 “병합(Merge)”한다.

*4-8 참조

01 Git

02 GitLab

03 GitLab Team

04 Git for Designer

GITLAB

02-1

GitLab Infomation

모든 프로젝트, 소스들의 버전 관리 저장소

Address

gitlab.coupang.net

Login Account

Coupaing Groupware Account

GitLab UI Repository

gitlab.coupang.net/release/coupang-design-convention

01 Git

02 GitLab

03 GitLab Team

04 Git for Designer

GITLAB TEAM

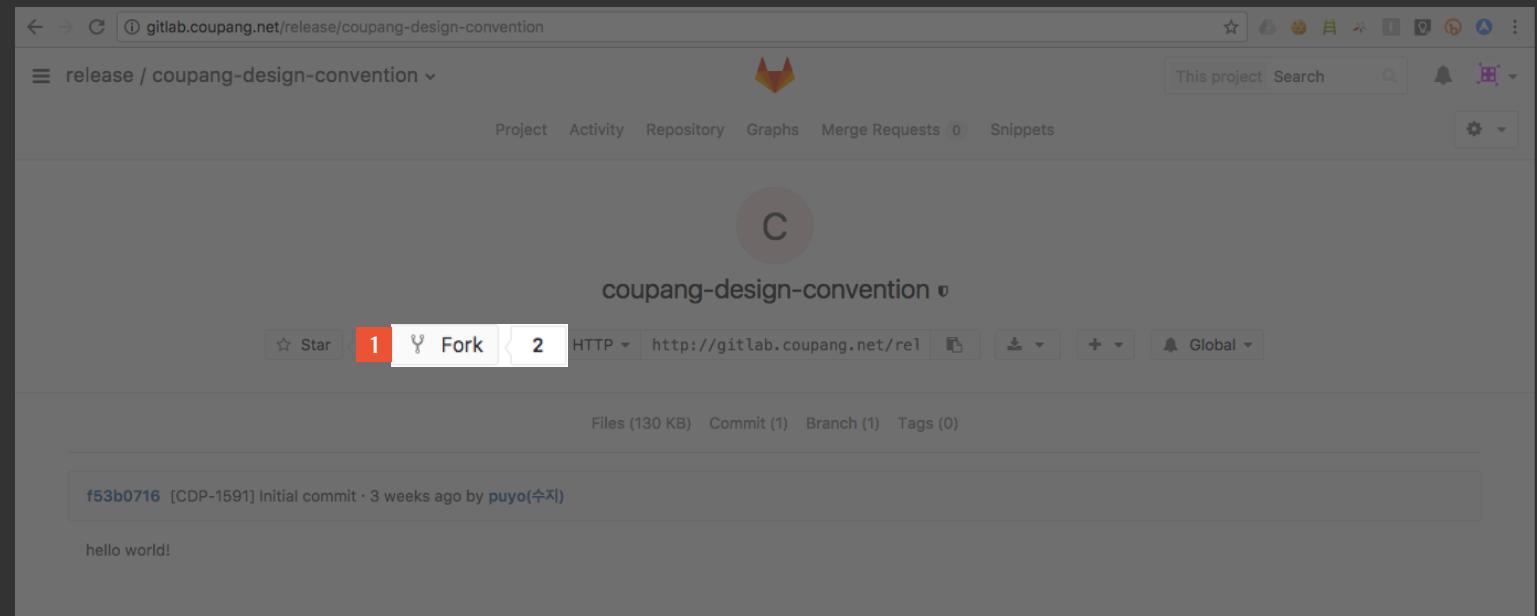
03-1

Fork Parent Repository

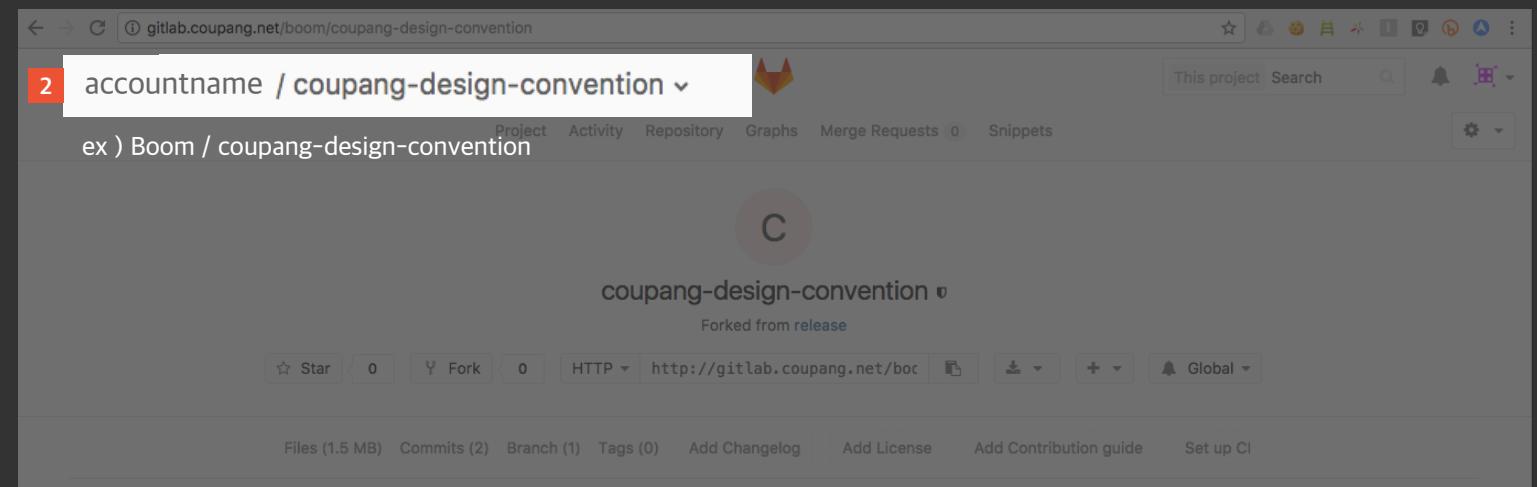
Team Setting

1. Fork
2. Add Members
3. Merge Request

Fork GitLab UI Repository > Check Address



The screenshot shows a GitLab repository page for 'coupaing-design-convention'. At the top, there's a navigation bar with links for Project, Activity, Repository, Graphs, Merge Requests (0), and Snippets. Below the navigation, there's a large circular profile picture with a letter 'C' and the repository name 'coupaing-design-convention' with a dropdown arrow. A prominent red box highlights the 'Fork' button, which is currently active with the number '1' next to it. Other buttons include 'Star' (0), '2' (likely forks or clones), and 'HTTP' with a URL. Below the buttons, there are links for 'Files (130 KB)', 'Commit (1)', 'Branch (1)', and 'Tags (0)'. A commit history box shows a single commit from 'f53b0716 [CDP-1591] Initial commit · 3 weeks ago by puyo(수지)'. The commit message is 'hello world!'. The overall interface is dark-themed.



The screenshot shows a forked repository page for 'accountname / coupaing-design-convention'. The top navigation bar is identical to the first screenshot. The repository name 'coupaing-design-convention' is displayed with a dropdown arrow, and the word 'Forked from release' is visible below it. The 'Fork' button is highlighted with a red box, showing the number '2' next to it. Other buttons include 'Star' (0), '0' (forks or clones), and 'HTTP' with a URL. Below the buttons, there are links for 'Files (1.5 MB)', 'Commits (2)', 'Branch (1)', 'Tags (0)', 'Add Changelog', 'Add License', 'Add Contribution guide', and 'Set up CI'. The overall interface is dark-themed.

03-2

Add Members

Repository Setting : Members > Search Members > Role Setting > Add to project

Members

Add new user to **coupang-design-convention**

Developer Expiration date Add to project 3

Read more about role permissions

On this date, the user(s) will automatically lose access to this project.

Find existing members by name Q Name, ascending v

Avatar	Name	Role	Actions
541dba	541dba		
aarora	aarora		
abh0518	abh0518		
abhishek	abhishek		
abhyodaya	abhyodaya		

Existing users and groups

Users with access to **coupang-design-convention** 2

boom @boom It's you Joined a week ago Developer Expiration date Master

jsy1005 @jsy1005 Joined less than a minute ago Developer Expiration date Master

Find existing members by name Q Name, ascending v

03-3

Merge Request after Project

Merge Requests (Original Repository) > Select Branch > Compare branches and continue

1. Fork
2. Add Members
3. Merge Request
(Project End or Source Update)

The screenshot shows the 'Merge Requests' section of a GitLab project. The 'Merge Requests' tab is active, indicated by a red box labeled '1'. Below it, the 'New Merge Request' form is displayed. The 'Source branch' dropdown is set to 'boom/coupa...nvention' and the 'Target branch' dropdown is set to 'master' for both the source and target branches. The 'Compare branches and continue' button at the bottom is highlighted with a red box labeled '3'.

01 Git

02 GitLab

03 GitLab Team

04 Git for Designer

GIT FOR DESIGNER

04-1

Software Install

깃 관리 프로그램 설치

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Source Tree (Free)

www.sourcetreeapp.com



coupang



Folio (Fee : \$49)

<http://folioformac.com/>

Only Github Software



Github (Free)

<https://desktop.github.com/>



Kactus (Fee)

<https://desktop.github.com/>

ETC Design Version Control Software



Abstract(Fee)

www.goabstract.com

04-2

SourceTree Setting for Git

깃 관리 프로그램 설치

SourceTree > Preference > General : Enter User.Name / User.Email

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

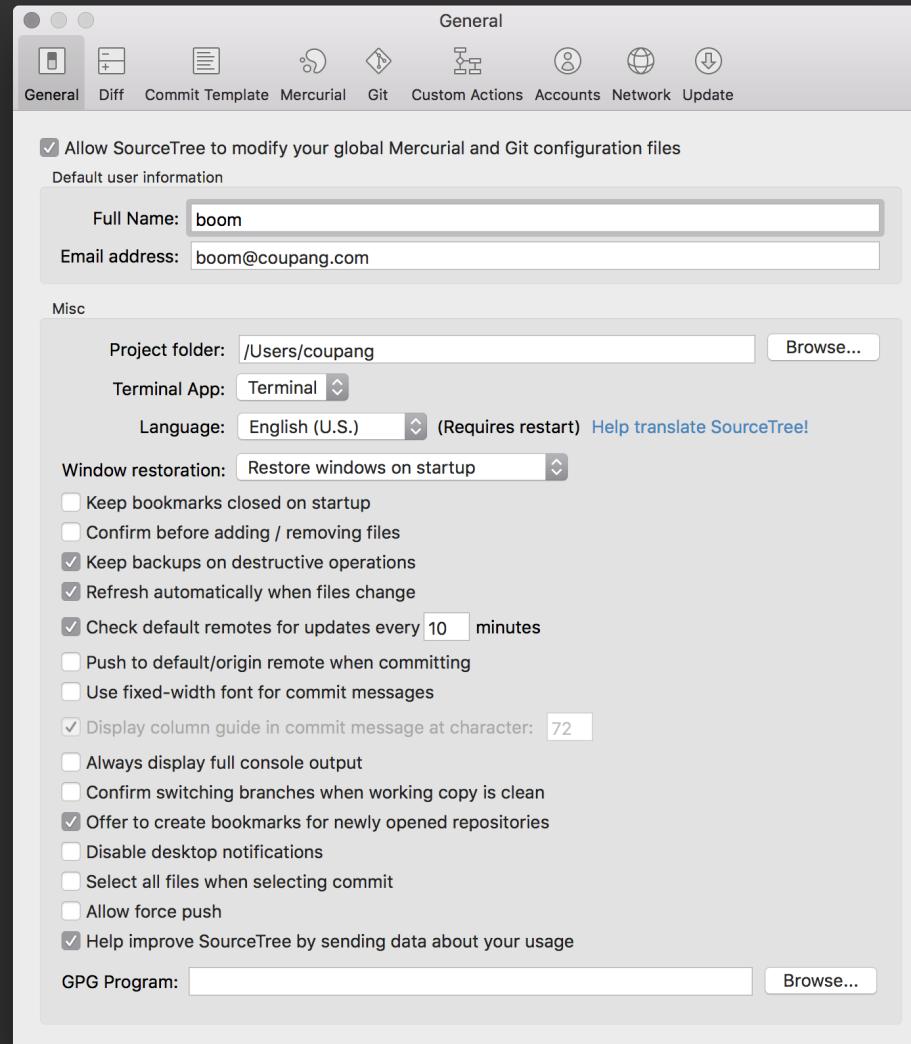
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Check Terminal

"git config --global -l"

```
Last login: Wed Aug 16 14:16:44 on ttys000
[HOY:~ Hoy$ git config --global -l
user.name=Hoy
user.email=boom@coupang.com
```

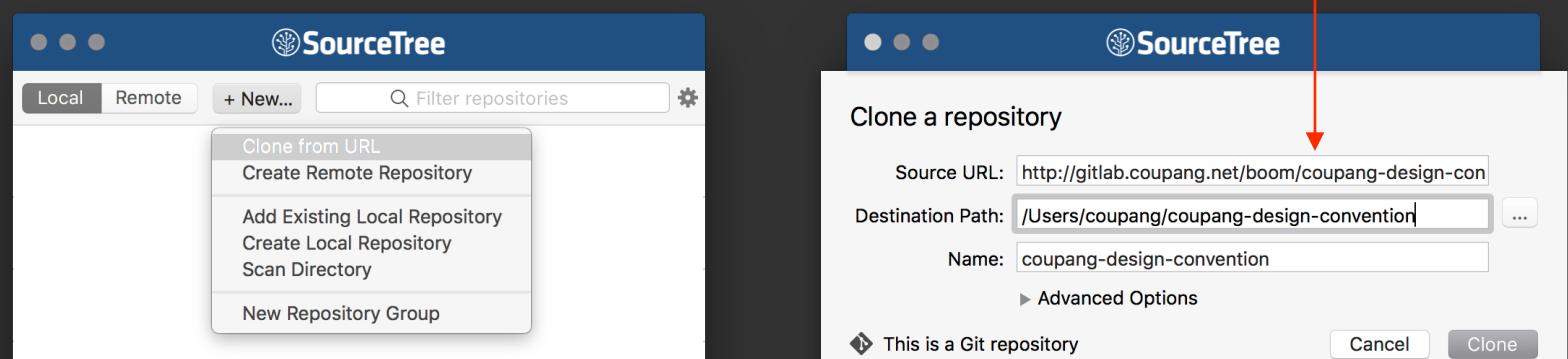
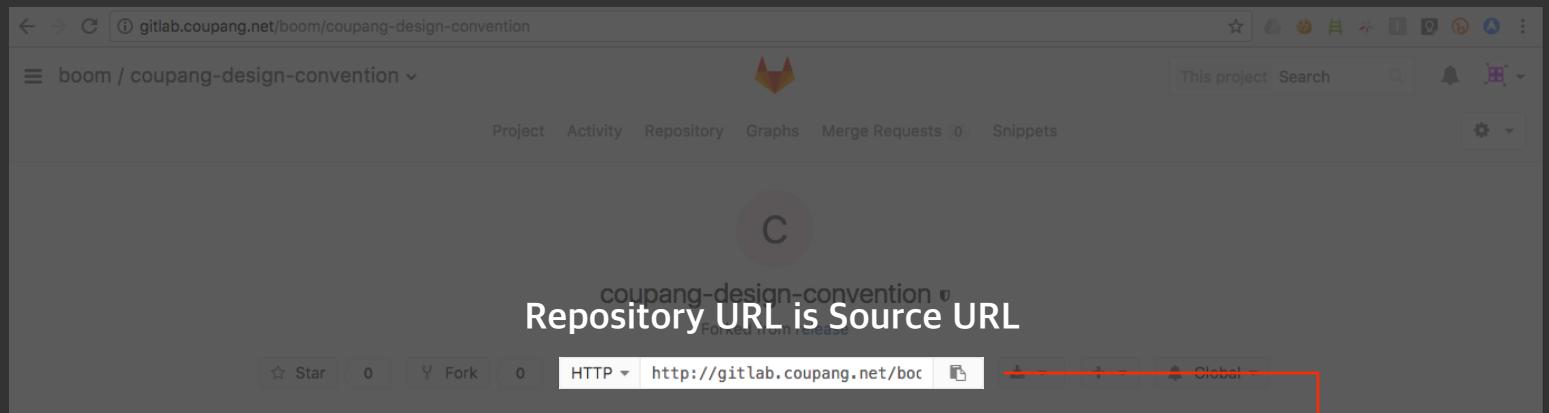
04-3

Clone Repository

원격 저장소 복제

New > Clone from URL > Copy & Paste Repository URL > Local Path Setting > Clone

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout



Destination Path = Local Path
Name = Folder Name(Please Dont change) = Repository Name

04-4

Commit after end of project

프로젝트 끝난 후 Git 관리를 위한 커밋

Commit > Check Unstaged files(or Final) to stage > Write Commit Message > Commit

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

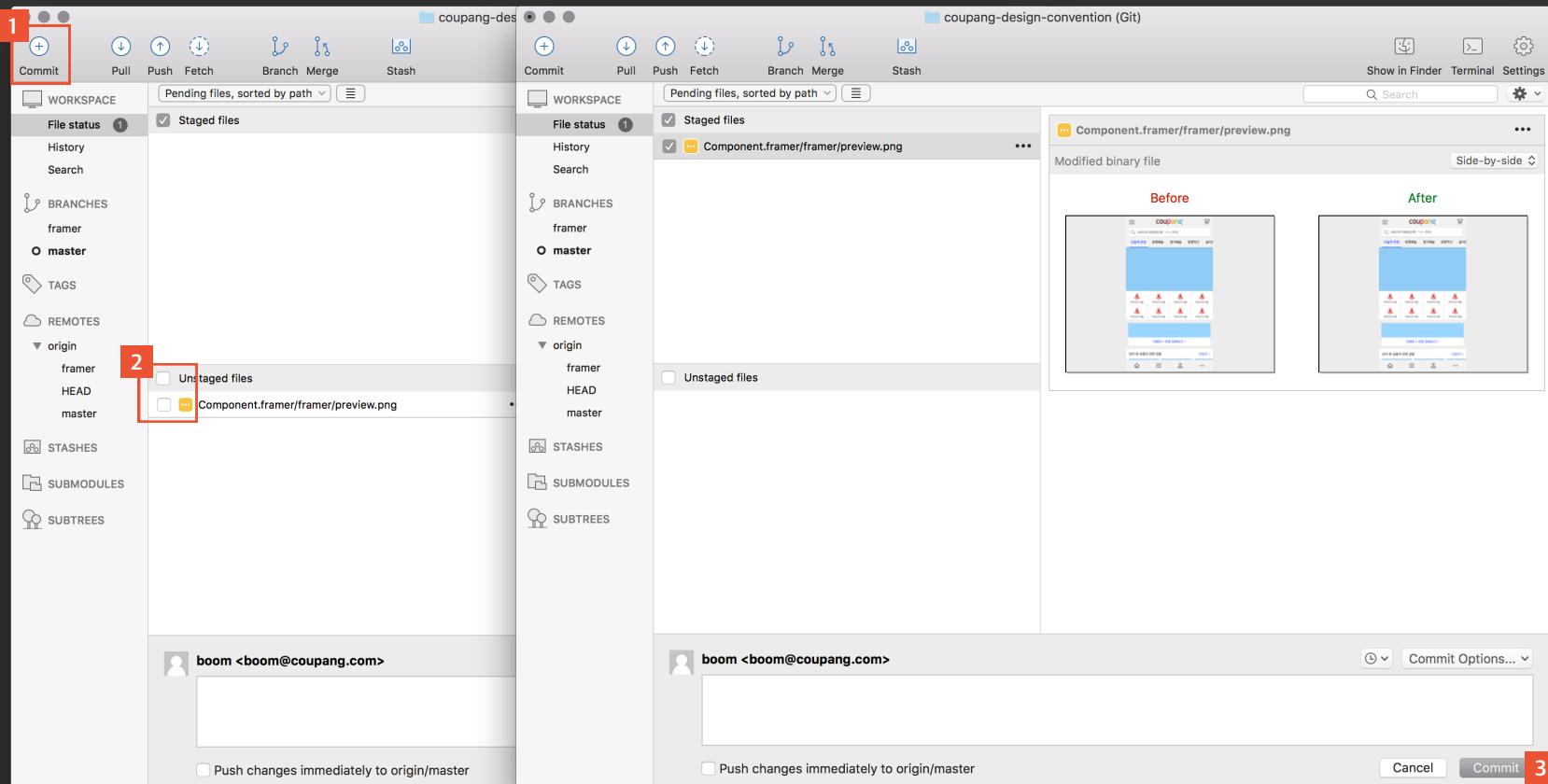
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Commit Messege Rule

If don't follow this rule, Could not push Anything after commit.

[Nickname-00] Domain-ProjectName-Status(Update, Issues)

ex) [Hoy-00] Gateway-ThemeShop-Category-Add

04-5

Push to Repository

커밋 후 저장소 업로드(Push)

Commit > Push > Select Branch > Push

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

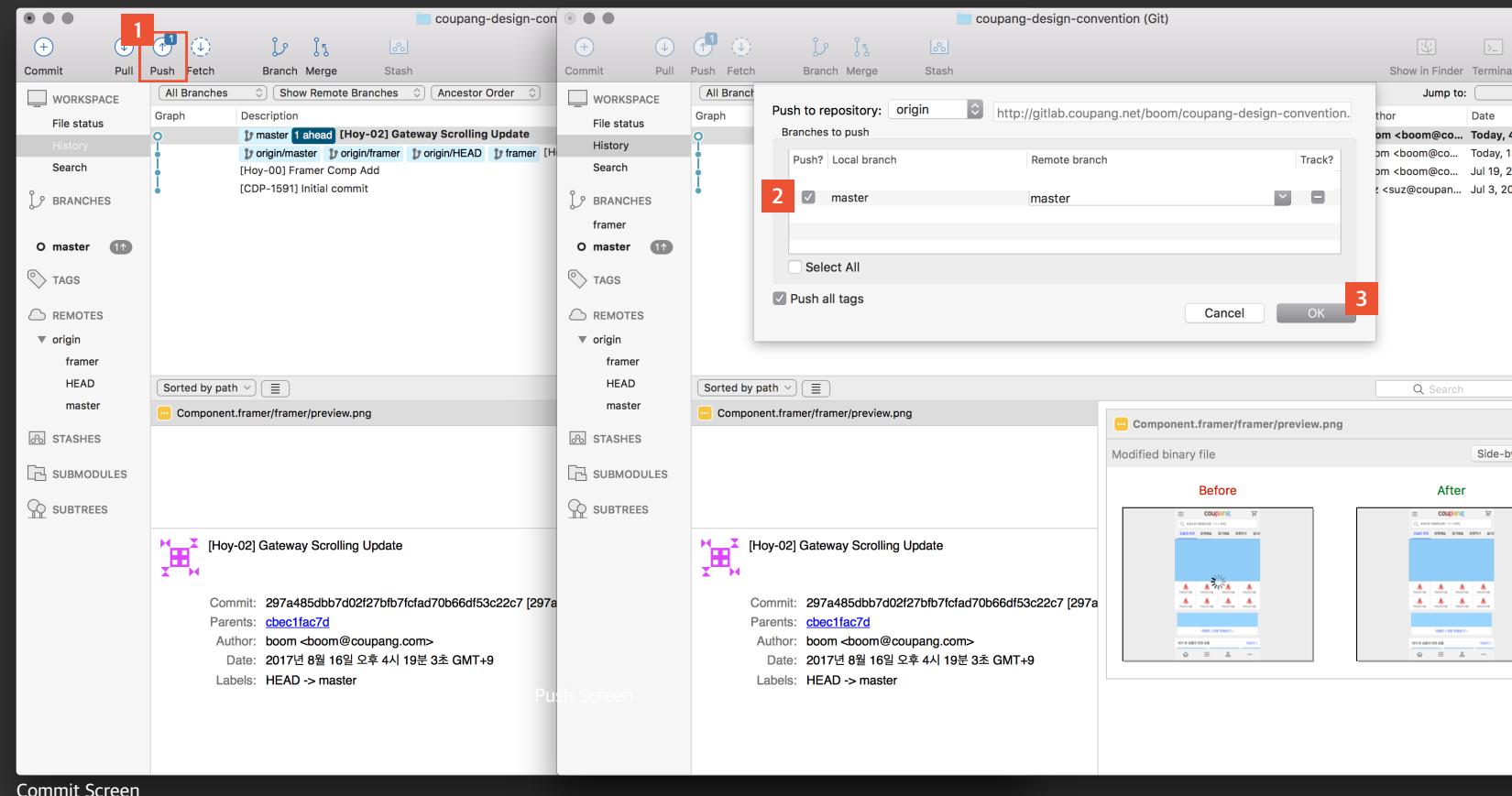
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Push Rule

If don't follow this rule, Could not push Anything.

1. Do Commit first
2. If there is(are) Pull, Have to Pull first before Push.

04-6

Check Push & Commit History

푸시 및 커밋 히스토리 확인

Gitlab : Repository > Commits

1. Software Install (Desktop)
2. SourceTree Setting
3. Clone Repository
4. Commit
5. Push & Pull
6. Branch
7. Merge
8. Checkout

**Check Terminal
"git log"**

```

[Hoy-02] Gateway Scrolling Update
commit 297a485dbb7d02f27fb7fcfad70b66df53c22c7
Author: boom <boom@coupaning.com>
Date:   Wed Aug 16 16:19:03 2017 +0900

[Hoy-01] Comp Update
commit cbec1fac7dca98e04a1a1ab4d97e34b9089b28aa
Author: boom <boom@coupaning.com>
Date:   Wed Aug 16 13:44:56 2017 +0900

[Hoy-00] Framer Comp Add
commit 16030a5e489799a921124243ea7ca7623a3c7917
Author: boom <boom@coupaning.com>
Date:   Wed Jul 19 16:45:55 2017 +0900

[Hoy-00] Initial commit
commit f53b0716dcde05a509c2401d20b2ea86351756f5
Author: suz <suz@coupaning.com>
Date:   Mon Jul 3 17:34:03 2017 +0900

```

04-7

Pull

파일 당겨오기 : 해당 폴더(Branch)의 파일이 업데이트 됩니다.

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

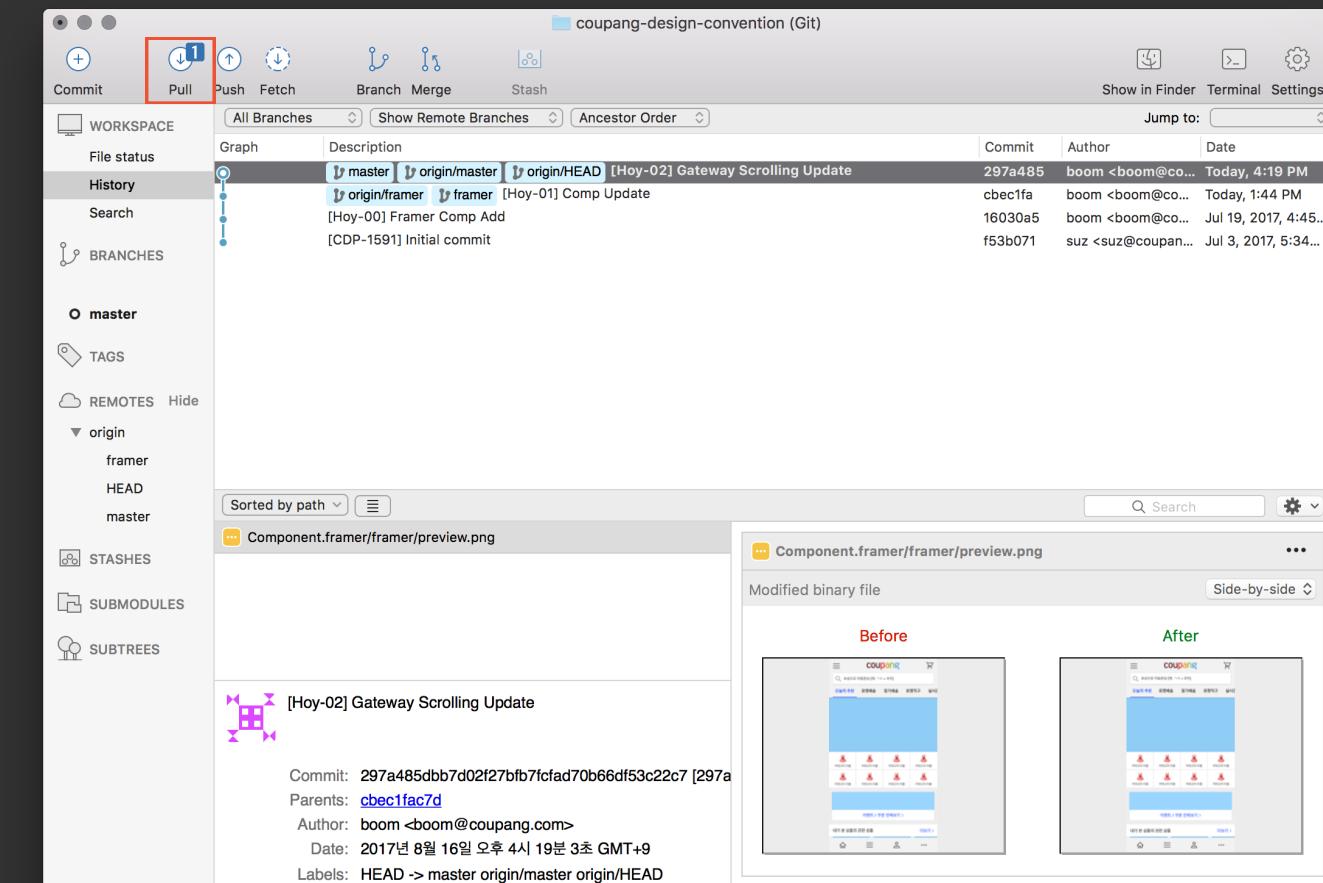
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Pull Rule

Pull first before the Push

1. If project is worked in common file(ex: UI Kit), Have to Branch first for your design files.

If you have already branched, Just Pull.

04-8 Branch

자신만의 버전 만들기

Branch > Input Branch Name (Nick or Feature) > Create Branch

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

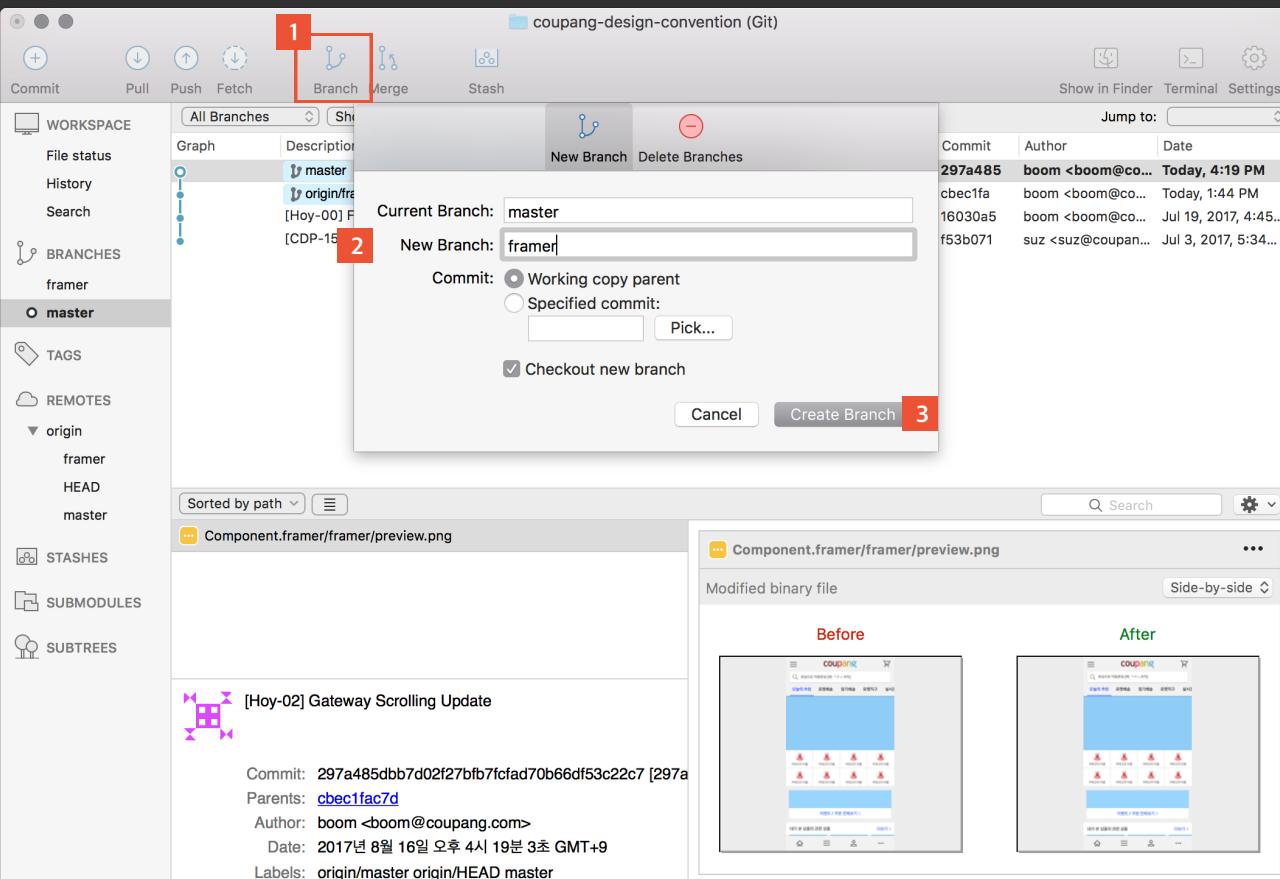
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Branch Rule

Purpose : Safe of lossing design files

Designer

1. Each Designer, Each Branch
2. Each Feature, Each Branch

Manager

1. Each Domain, Each Branch
2. Each Project, Each Branch

04-9

Branch File Merge

브랜치 파일 병합하기

Select Unchanged Branch > Merge > Select The Lastest Commit > Ok

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

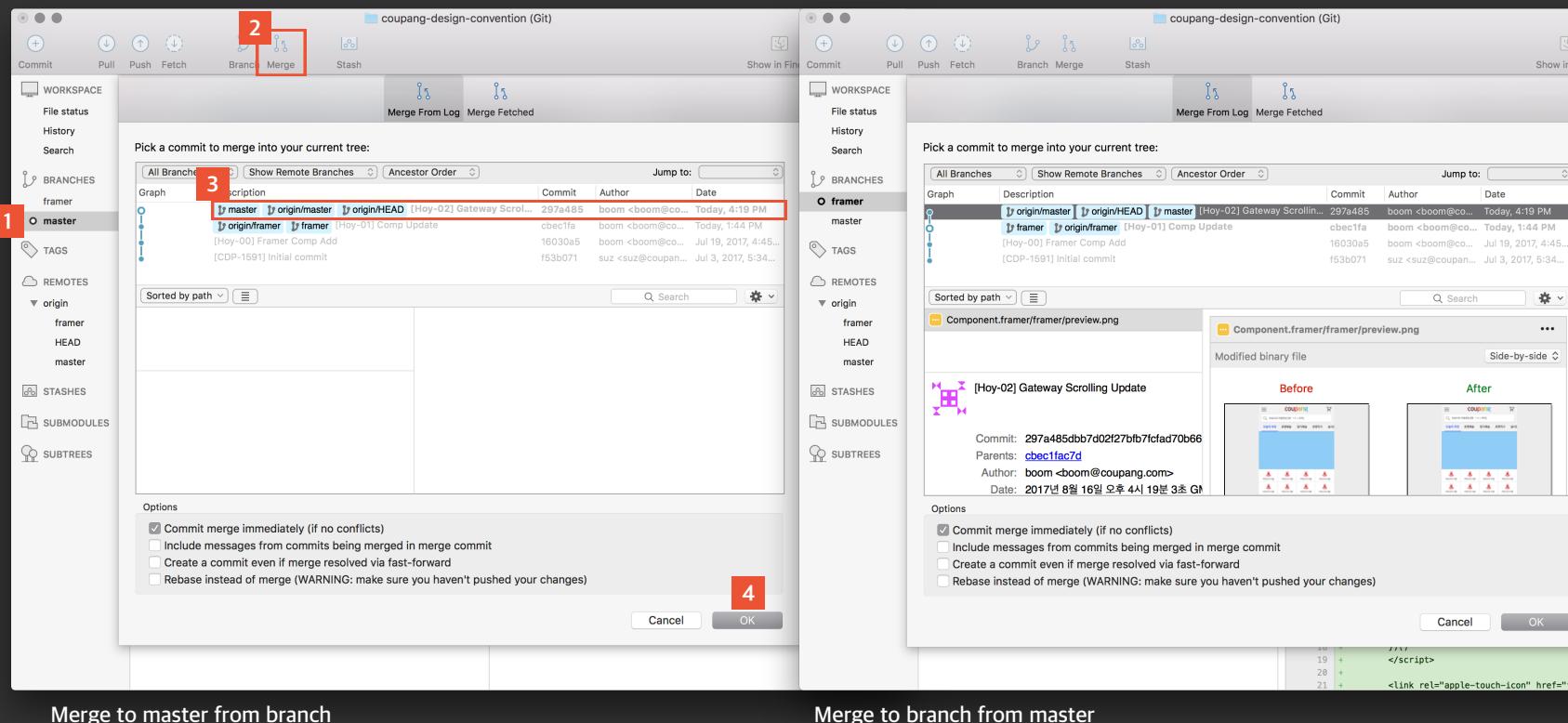
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Merge Rule

If you choose the wrong branch, files might be disappeared.
but Choose working branch is showed Working Design Files

1. Merge after Commit
2. Merge have to base from changed branch
3. When common ui is pulled in Master Branch, Base is Working Branch

04-10

Checkout

과거 작업물 확인

Mouse Right > CheckOut > Check Folder(the file is changed to this commit)

1. Software Install (Desktop)

2. SourceTree Setting

3. Clone Repository

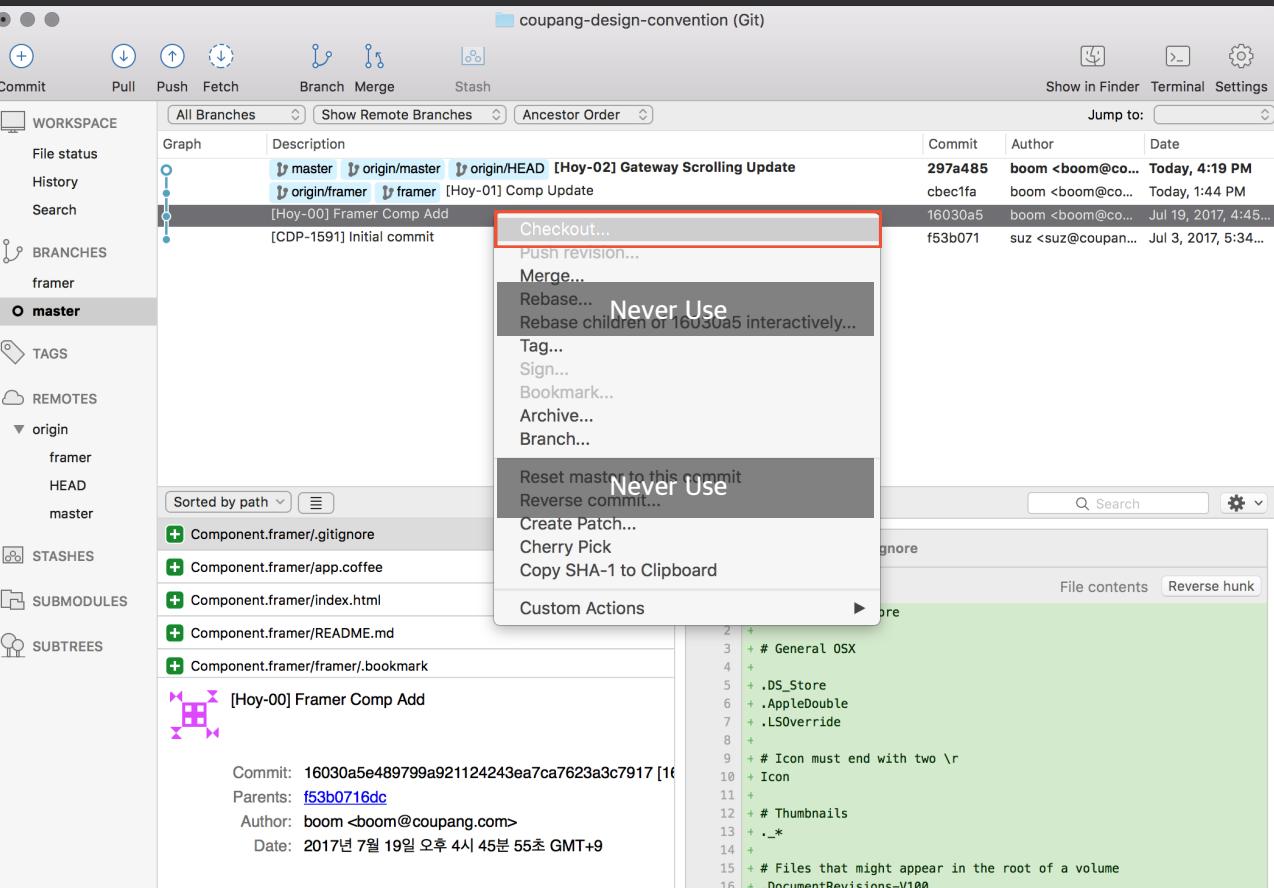
4. Commit

5. Push & Pull

6. Branch

7. Merge

8. Checkout



Checkout Rule

If you choose the wrong branch, files might be disappeared

1. if you change after selected commit, make branch first.

Please, Use only what you know