

Git/GitHub Lab

Adapted from "오픈소스 참여를 위한
Git/GitHub 기본실습" by 송태웅

'24W

송 인 식

실습 환경

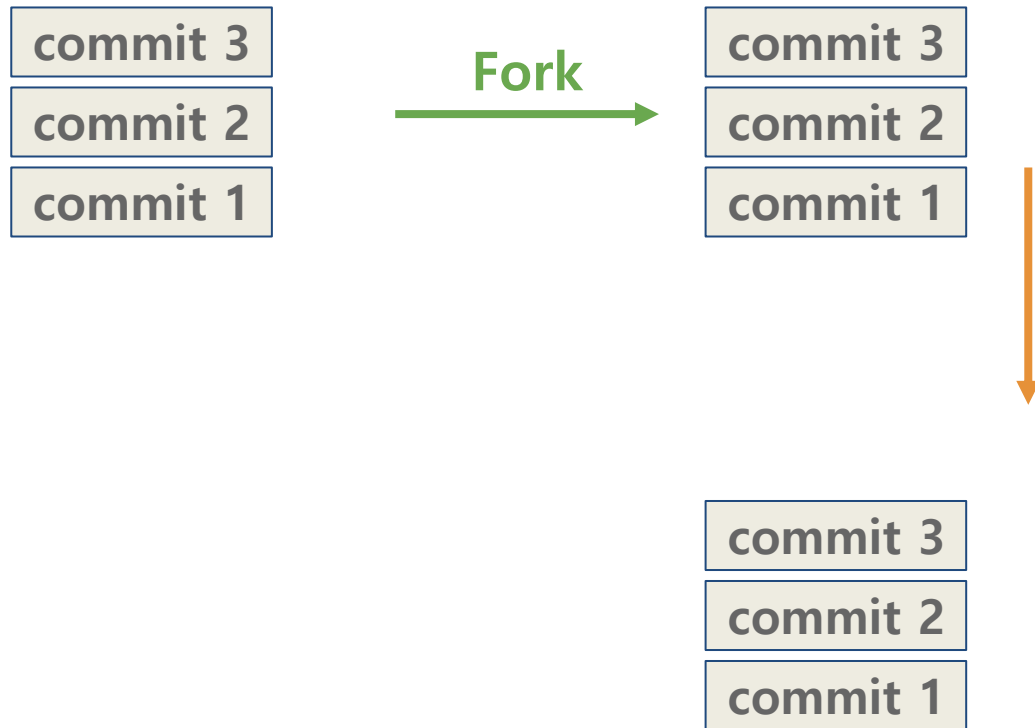
- GitHub 계정 등록
- 구름 IDE에서 컨테이너 생성
 - PyTorch 타입 컨테이너

Outline

- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급

공식 오픈소스 프로젝트에서 내 작업 공간으로 프로젝트 복사

공식 오픈소스 프로젝트
pytorch/examples



pytorch 예제 프로젝트 Fork

<https://github.com/issong-DKU/pytorch-example>

The screenshot shows the GitHub interface for the repository 'issong-DKU / pytorch-example'. The repository is public and was forked from 'Taeung/pytorch-example'. The 'Fork' button is highlighted with a red box, showing 728 forks. The repository has 0 stars and 0 watches. The 'Code' button is green. The repository is on the 'master' branch, with 1 branch and 0 tags. A message states 'This branch is up to date with Taeung/pytorch-example:master.' Below this, there is a pull request from 'Taeung' to 'gentlelinuxer/pr_test' with 502 commits. A file named '.github/workflows' is listed with the description 'Add CI to run examples via github action' and was added 2 years ago. The 'About' section on the right indicates no description, website, or topics are provided, and lists the license as BSD-3-Clause.

Search or jump to... Pull requests Issues Marketplace Explore

issong-DKU / pytorch-example Public

forked from Taeung/pytorch-example

Watch 0 Fork 728 Star 0

<> Code Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Add file Code

About

No description, website, or topics provided.

Readme

BSD-3-Clause License

0 stars

0 watching

728 forks

This branch is up to date with Taeung/pytorch-example:master.

Contribute

Taeung Merge pull request Taeung#632 from gentlelinuxer/pr_test 113447f on 22 Feb 502 commits

.github/workflows Add CI to run examples via github action 2 years ago

Local machine(구름 IDE)로 복사 (Clone)

The screenshot shows the GitHub interface for the repository 'inshiksong-dku / pytorch-example'. The repository is public and forked from 'issong-DKU/pytorch-example'. The 'Code' button is highlighted, and the dropdown menu is open, showing the 'Clone' option. The URL 'https://github.com/inshiksong-dku/pytorch-' is highlighted in a red box. The 'Clone' menu also shows options for 'HTTPS', 'SSH', 'GitHub CLI', and 'New'. Below the URL, it says 'Use Git or checkout with SVN using the web URL.' Other options in the menu include 'Open with GitHub Desktop' and 'Download ZIP'. The repository page also shows a message 'This branch is up to date with issong-DKU/pytorch-example:master.' and a list of files: '.github/workflows', 'PR_TEST', and 'cpp'.

inshiksong-dku / pytorch-example Public

forked from issong-DKU/pytorch-example

<> Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is up to date with issong-DKU/pytorch-example:master.

Taeung Merge pull request Taeung#632 from gentlelinuxer/pr_test ...

.github/workflows Add CI to run examples via github

PR_TEST PR test

cpp Fix cpp/mnist CMakeList build. (#7

Clone

HTTPS SSH GitHub CLI New

https://github.com/inshiksong-dku/pytorch-

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Local machine에서 복사된 프로젝트 확인

```
root@goorm:/workspace/pytorch# git clone https://github.com/inshiksong-dku/pytorch-example.git
Cloning into 'pytorch-example'...
remote: Enumerating objects: 2459, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 2459 (delta 17), reused 26 (delta 11), pack-reused 2420
Receiving objects: 100% (2459/2459), 16.83 MiB | 4.69 MiB/s, done.
Resolving deltas: 100% (1246/1246), done.
root@goorm:/workspace/pytorch# mv pytorch-example examples
root@goorm:/workspace/pytorch# cd examples
root@goorm:/workspace/pytorch/examples(master)# cd mnist
root@goorm:/workspace/pytorch/examples/mnist(master)# cat README.md | less
root@goorm:/workspace/pytorch/examples/mnist(master)#
```

Outline

- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급

전체 소스파일수정 내역(commit) 리스트

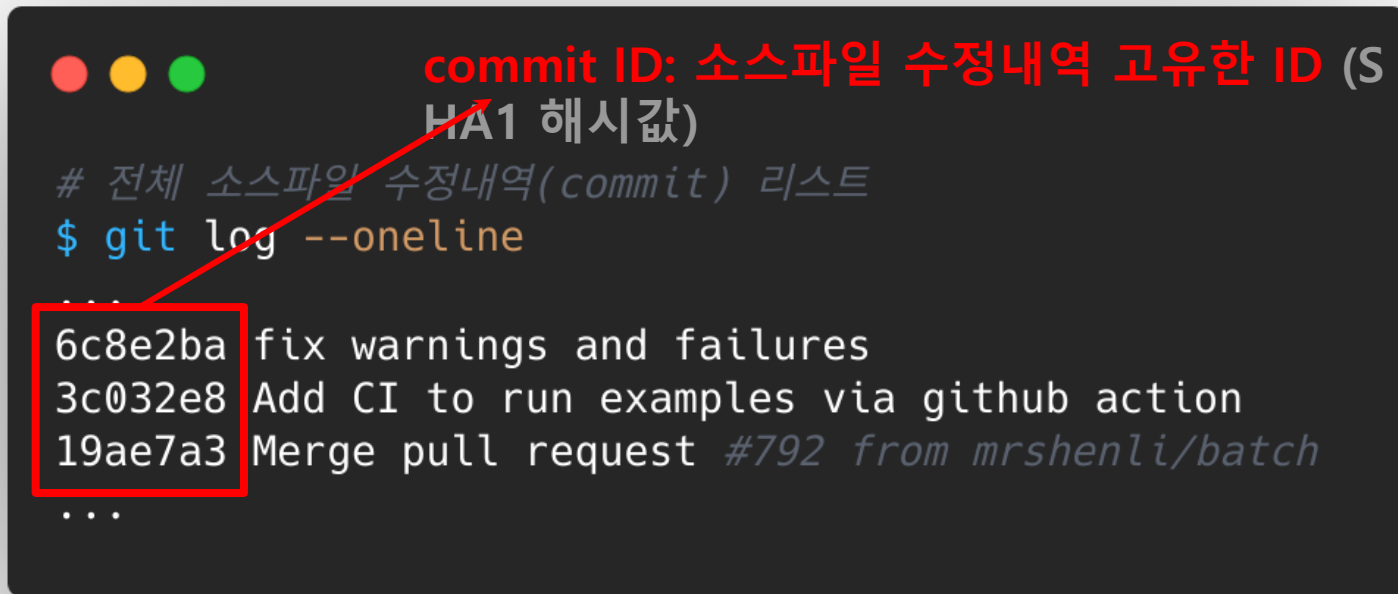


전체 소스파일 수정내역(commit) 리스트

```
$ git log --oneline
```

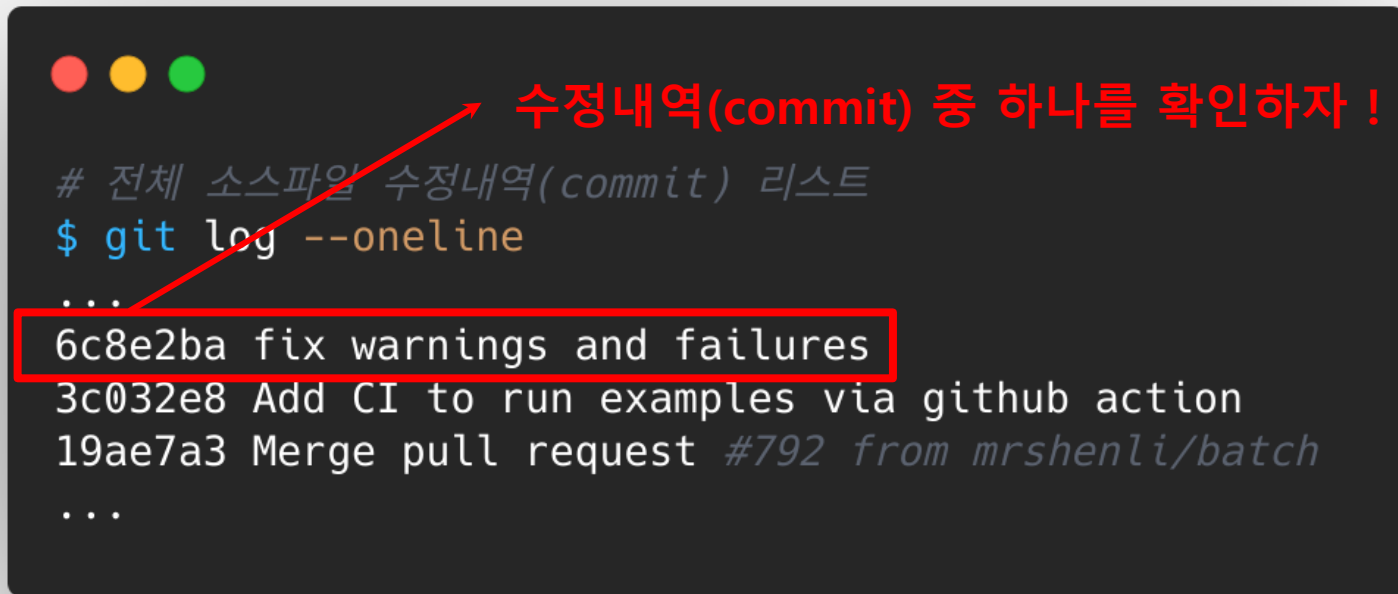
참고: 'q' 키 눌러서 나가기

소스파일 수정내역(commit)의 ID 란?



```
commit ID: 소스파일 수정내역 고유한 ID (SHA1 해시값)
# 전체 소스파일 수정내역(commit) 리스트
$ git log --oneline
...
6c8e2ba fix warnings and failures
3c032e8 Add CI to run examples via github action
19ae7a3 Merge pull request #792 from mrshenli/batch
...
```

소스파일 수정내역(commit) 내용확인



```
● ● ● 수정내역(commit) 중 하나를 확인하자 !  
# 전체 소스파일 수정내역(commit) 리스트  
$ git log --oneline  
...  
6c8e2ba fix warnings and failures  
3c032e8 Add CI to run examples via github action  
19ae7a3 Merge pull request #792 from mrshenli/batch  
...
```

소스파일 수정내역(commit) 내용확인

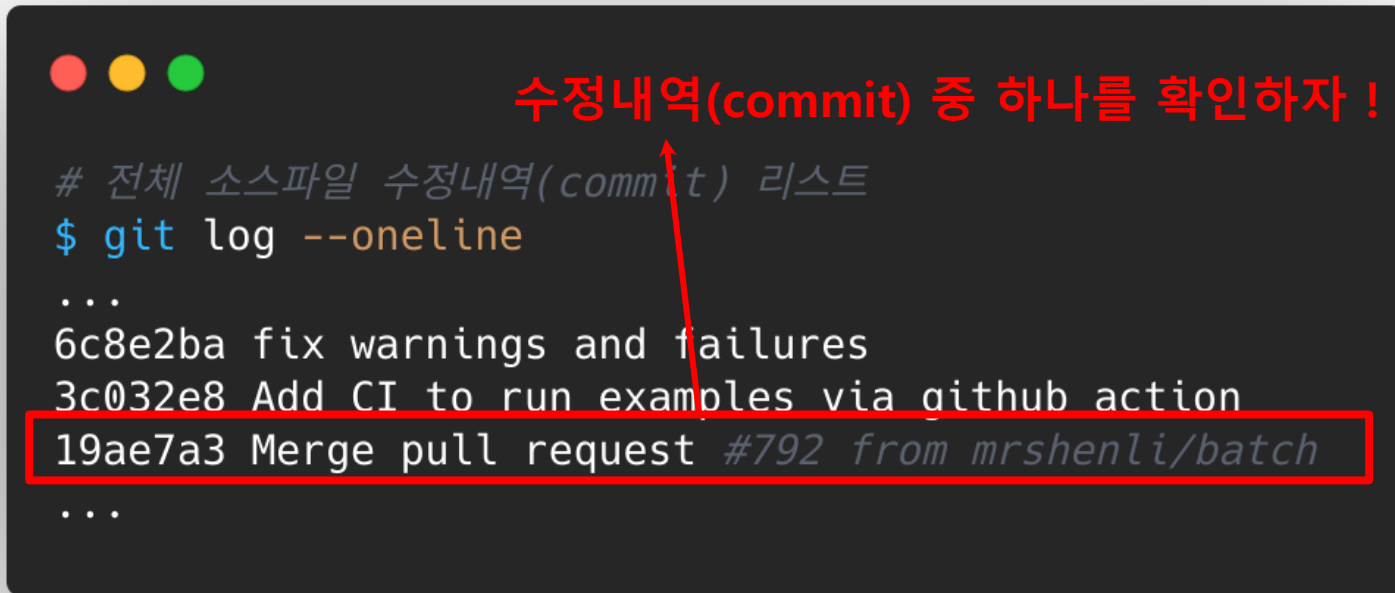


소스수정 내역(commit) 한가지 확인하기

```
$ git show 6c8e2ba
```

참고: 'q' 키 눌러서 나가기

Merge commit 이란 ?



```
● ● ●  
수정내역(commit) 중 하나를 확인하자 !  
# 전체 소스파일 수정내역(commit) 리스트  
$ git log --oneline  
...  
6c8e2ba fix warnings and failures  
3c032e8 Add CI to run examples via github action  
19ae7a3 Merge pull request #792 from mrshenli/batch  
...
```

Merge commit 이란 ?



```
# Merge commit (병합 커밋) 수정내역은 없음을 확인하자  
$ git show 19ae7a3
```

전체 소스파일 수정내역(commit) 자세히 보기



전체 소스파일 수정내역(commit) 자세히 보기

```
$ git log -p
```

참고: 'q' 키 눌러서 나가기

특정 소스파일 기준 수정내역(commit) 리스트



오픈소스 작업 폴더로 이동

```
$ cd /workspace/pytorch/examples/
```

특정 폴더를 기준으로 소스 수정내역(commit) 리스트 확인하기

```
$ git log --online -- mnist/
```


특정 날짜 기준 수정내역(commit) 리스트



```
# 2020년 1월 부터 2020년 6월 30일까지 소스 수정내역(commit) 리스트 확인  
$ git log --oneline --after=2020-01-01 --before=2020-06-30
```

특정 날짜+파일 기준 수정내역(commit) 리스트



```
# 2020년 6월 한달간 mnist/ 폴더기준 소스파일 수정내역(commit) 확인하기  
$ git log --oneline --after=2020-06-01 --before=2020-06-30 -- mnist/  
59423c5 Delete unnecessary blank lines (#785)
```

옛날 것부터 소스파일 수정내역(commit) 보기




```
# 소스파일 수정내역(commit) 옛날것부터 살펴보기  
$ git log --reverse
```

Outline

- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급

Git 설정하기



```
# GitHub ID/PW 캐싱데이터 삭제 (삭제시 문제없음)
# 다른(사람) GitHub 계정과의 충돌방지
$ git config --global --unset credential.helper
$ git config --system --unset credential.helper

# GitHub 계정 이메일 주소 및 본인영문이름
# 차후 소스코드 파일수정 내역(commit) 저자(author)정보
$ git config --global user.email "본인메일적으세요"
$ git config --global user.name "본인이름적으세요"
```

Git 기본 편집기 설정



```
# Git commit(소스파일 수정내역) message(설명글) 수정할 기본 편집기 설정  
# 원하는 편집기 설정가능 (vim, emacs, nano, notepad 등)
```

```
$ git config --global core.editor nano
```

```
# nano 편집기 사용시 설치 필요
```

```
$ sudo apt install -y nano
```

Git 설정내용 확인하기



```
# Git 설정 내용 확인하기  
# 참고: Git 설정수정은 이전 실습명령 동일하게 입력시 수정  
$ git config --list
```

Outline

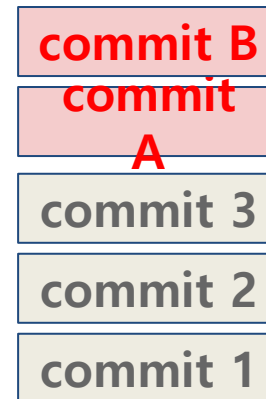
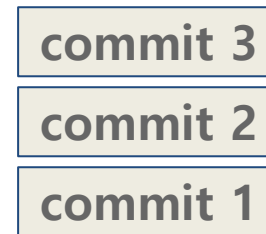
- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급

개발 작업 참여

공식 오픈소스 프로젝트
pytorch/examples



Fork



작업의 시작 "브랜치(branch) 생성"



```
# 오픈소스 프로젝트 폴더로 이동
$ cd /workspace/pytorch/examples/

# Branch 생성
# 작업내용을 대표하는 키워드로 Branch 명 생성추전
$ git checkout -b fix-mnist
```

브랜치(branch) 란 ? "같은 폴더 다른 세상"

```
# "같은 폴더 다른세상" 브랜치 테스트
$ touch hi.txt

# 새롭게 생성한 fix-mnist 브랜치에서
# 새로운 파일 hi.txt 수정내역(commit) 만들기
$ git add hi.txt
$ git commit -m "test: add hi.txt file"

# 브랜치를 master 브랜치로 변경
$ git checkout master

# 같은 폴더내 파일 내용확인하기
# hi.txt 파일 존재여부 확인
$ ls

# 브랜치 fix-mnist 로 변경 후
# hi.txt 파일 존재여부 확인
$ git checkout fix-mnist

# 다시 master 브랜치로 변경후 확인
$ git checkout master
$ ls
```

브랜치(branch) 삭제하기



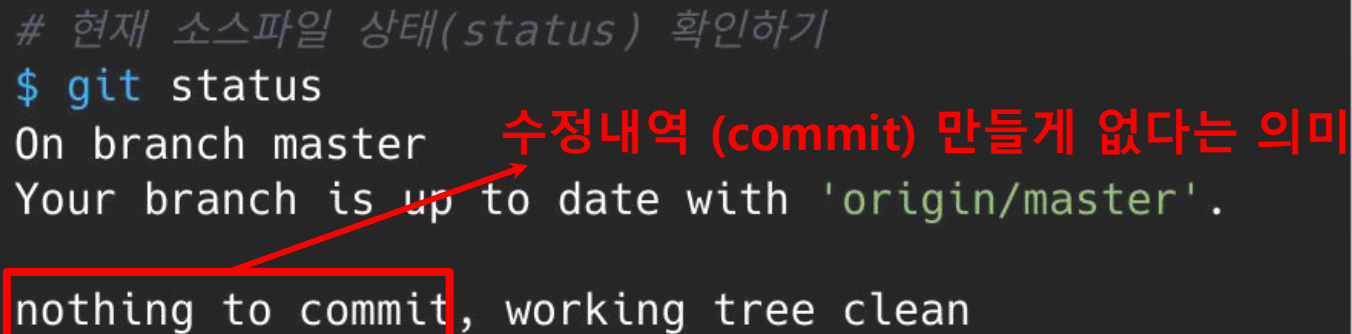
master 브랜치로 변경후

```
$ git checkout master
```

Branch 삭제

```
$ git branch -D fix-mnist
```

현재 소스파일 상태(status) 확인 하기



```
# 현재 소스파일 상태(status) 확인하기
$ git status
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
```

수정내역 (commit) 만들게 없다는 의미

nothing to commit, working tree clean

작업의 시작 "브랜치(branch) 생성"



오픈소스 프로젝트 폴더로 이동

```
$ cd /workspace/pytorch/examples/
```

Branch 생성

작업내용을 대표하는 키워드로 Branch 명 생성추전

```
$ git checkout -b fix-mnist
```

소스파일 수정 후 최신역사와의 차이점 확인하기

원하는 편집기(vim, emacs 등)로 mnist/main.py 파일 수정하기

\$ nano mnist/main.py

mnist/main.py 소스 파일 수정한 내용 확인하기

\$ git diff

diff --git a/mnist/main.py b/mnist/main.py

index 7d7899d..1bee55c 100644

--- a/mnist/main.py

+++ b/mnist/main.py

@@ -77,7 +77,7 @@ def main():

parser.add_argument('--test-batch-size', type=int, default=1000, metavar='N',
help='input batch size for testing (default: 1000)')

parser.add_argument('--epochs', type=int, default=14, metavar='N',

- help='number of epochs to train (default: 10)')

+ help='number of epochs to train (default: 14)')

parser.add_argument('--lr', type=float, default=1.0, metavar='LR',
help='learning rate (default: 1.0)')

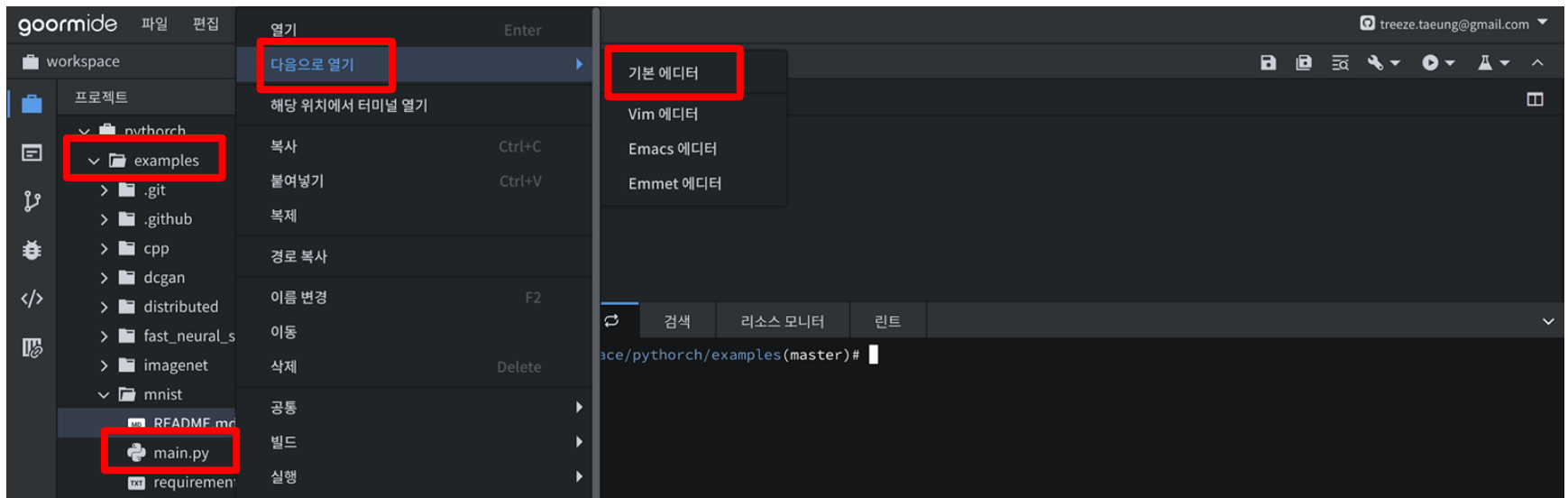
parser.add_argument('--gamma', type=float, default=0.7, metavar='M',

nano 편집기 사용법

(1) 저장하기 : Ctrl + o

(2) 나가기 : Ctrl + x

참고: 파일 수정을 위한 구름 IDE 에디터 열기 (nano 편집기 사용 안할 시)



소스파일 수정 후 최신역사와의 차이점 확인하기

```
# 원하는 편집기(vim, emacs 등)로 mnist/main.py 파일 수정하기
$ nano mnist/main.py

# mnist/main.py 소스 파일 수정한 내용 확인하기
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 7d7899d..1bee55c 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -77,7 +77,7 @@ def main():
     parser.add_argument('--test-batch-size', type=int, default=1000, metavar='N',
                        help='input batch size for testing (default: 1000)')
     parser.add_argument('--epochs', type=int, default=14, metavar='N',
-                        help='number of epochs to train (default: 10)')
+                        help='number of epochs to train (default: 14)')
     parser.add_argument('--lr', type=float, default=1.0, metavar='LR',
                        help='learning rate (default: 1.0)')
     parser.add_argument('--gamma', type=float, default=0.7, metavar='M',
```


수정내역(commit)만들기 전 준비작업 (add)

```
# 수정한 파일 git commit 할 준비하기
$ git add mnist/main.py

# 수정한 파일 git commit 준비 완료상태 확인
$ git status
On branch fix-mnist
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   mnist/main.py
```

수정내역(commit) 만들고 확인하기



```
# Git commit(수정내역) 만들고
# commit을 만든 이유 작성:
# help 내용 안에 잘못된 default 값 수정했다는 설명 적기
$ git commit -m "Correct typo in default value within help"

# 내가 작성한 commit 확인하기
$ git show

# 참고: 'q' 키 누르고 나가기
```

현재 소스파일 상태(status) 확인 하기



현재 소스폴더 상태 확인하기

\$ git status

On branch fix-mnist

nothing to commit, working tree clean

현재 변화분도 없고
commit 할것도 없는 상태

수정내역(commit) 1개 더 추가하기:

step 1 "소스파일 수정"

```
# mnist/main.py 소스파일 수정하기
$ nano mnist/main.py

# 수정한 내용 확인하기
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 1bee55c..166231a 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -22,6 +22,7 @@ class Net(nn.Module):
     x = self.conv1(x)
     x = F.relu(x)
     x = self.conv2(x)
+    x = F.relu(x)
     x = F.max_pool2d(x, 2)
     x = self.dropout1(x)
     x = torch.flatten(x, 1)
```


수정내역(commit) 1개 더 추가하기: step 2 "commit 준비(add)"

```
# 수정한 파일 git commit 할 준비하기
$ git add mnist/main.py

# 수정한 파일 git commit 준비 완료상태 확인
$ git status
On branch fix-mnist
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   mnist/main.py
```

수정내역(commit) 1개 더 추가하기: step 3 "commit 만들기"



```
# Git commit(수정내역) 만들고  
# commit을 만든 이유 작성:  
# 빠뜨린 non-linearity 비선형함수 추가하기  
$ git commit -m "added missing non-linearity"  
  
# 내가 작성한 commit 확인하기  
$ git show  
  
# 참고: 'q' 키 누르고 나가기
```

Outline

- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급

작업 절차

공식 오픈소스 프로젝트

pytorch/examples

commit 3

commit 2

commit 1

Fork

Fork한 저장소

commit 3

commit 2

commit 1

commit B

commit

A

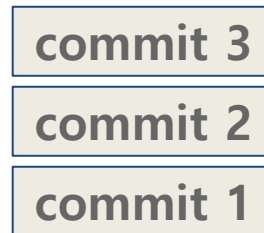
commit 3

commit 2

commit 1

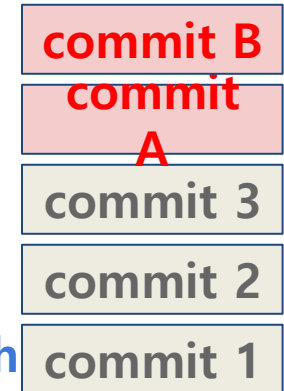
작업 절차

공식 오픈소스 프로젝트
pytorch/examples

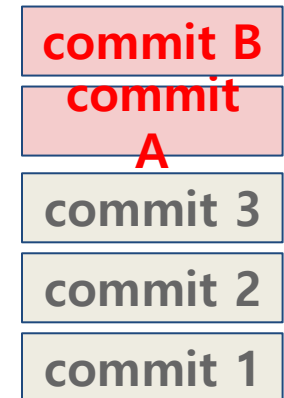


Fork

Fork한 저장소

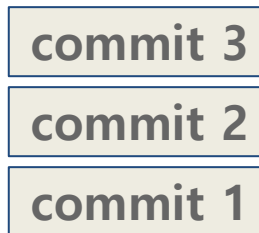


push



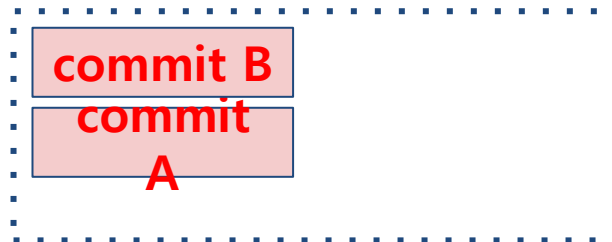
작업 절차

공식 오픈소스 프로젝트
pytorch/examples

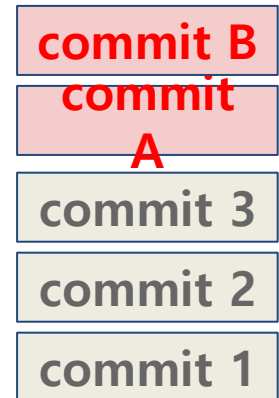
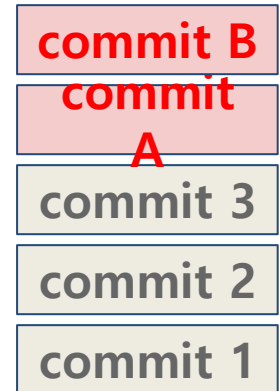


PR

Pull-Request 게시판 (Merge 되기 전)



Fork한 저장소

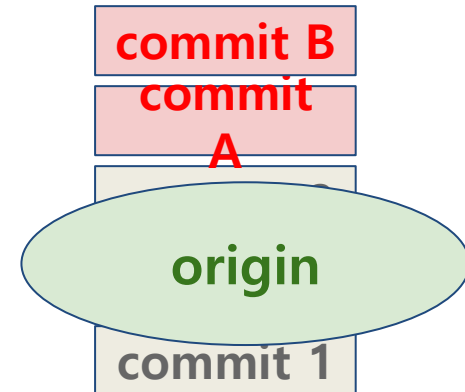


작업 절차

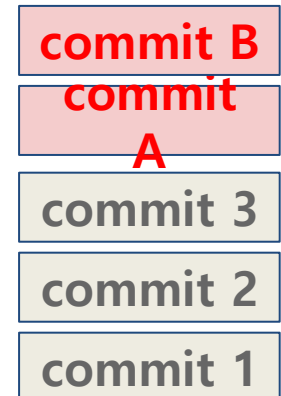
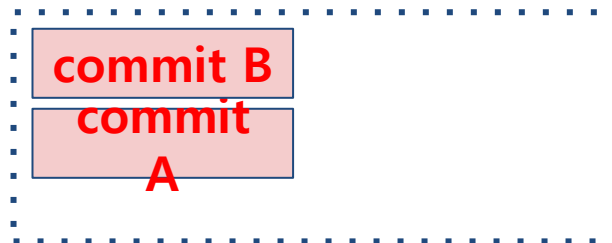
공식 오픈소스 프로젝트
pytorch/examples



Fork한 저장소



Pull-Request 게시판 (Merge 되기 전)



내가 만든 수정내역(commit) 제출 준비하기



나의 작업 브랜치 *fix-mnist* 확인하기

```
$ git branch
```

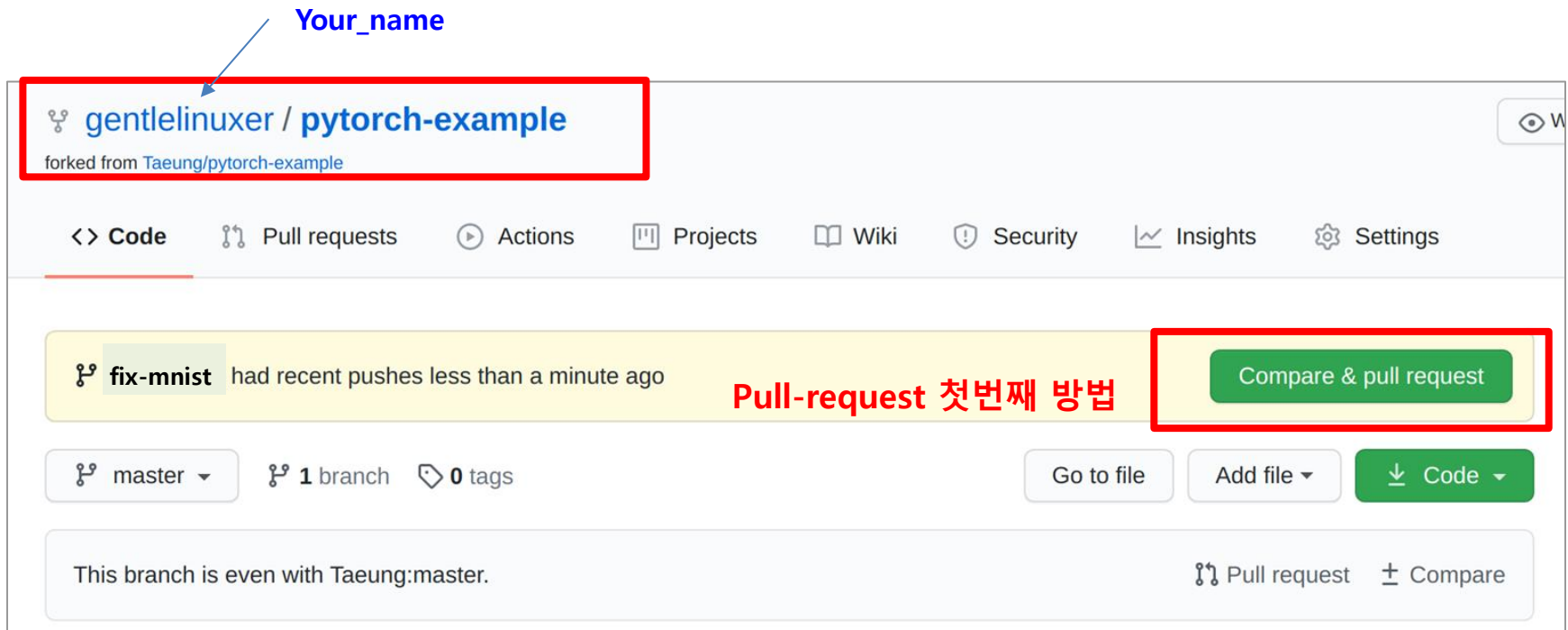
내가 작성한 2개의 *commit* 을

나의 *Fork* 저장소 *GitHub* 에 업로드해서

내가만든 수정내역(*commit*) 제출(*pull-request*) 준비하기

```
$ git push origin fix-mnist
```

Fork 한 저장소(프로젝트) GitHub 에서 commit 제출하기



Fork 한 저장소(프로젝트) GitHub 에서 commit 제출하기

Your_name

gentlelinuxer / pytorch-example
forked from Taeung/pytorch-example

<> Code Pull requests Actions Projects Wiki Security Insights

fix-mnist 3 branches 0 tags Go to file Add file Code

This branch is 2 commits ahead of Taeung:master.

또는 Pull-request 두번째 방법 Pull request Compare


Taeung Song added missing non-linearity c553b56 24 minutes ago 472 commits

.github/workflows Add CI to run examples via github action 11 days ago


Fork 한 저장소(프로젝트) GitHub 에서 commit 제출하기

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 repository: Taeung/pytorch-example ▼ base: master ▼ ← head repository: gentlelinuxer/pytorch-example ▼ compare: fix-readme ▼

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



Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ↶

Leave a comment

오픈소스 기여하기: GitHub 에서 commit 제출하기

Pull-Request

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

☒ Allow edits by maintainers ⓘ

Create pull request ▼

Pull-request 게시판에서 내가 제출한 commit 확인

The screenshot shows the GitHub interface for a repository named 'Taeung / pytorch-example'. The top navigation bar includes links for 'Code', 'Issues', 'Pull requests' (highlighted with a red box and a count of 1), 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. Below the navigation bar, the title 'PR test #1' is displayed. A green button labeled 'Open' is next to the text 'gentlelinuxer wants to merge 1 commit into Taeung:master from gentlelinuxer:fix-readme'. Below this, a summary bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1'. A comment by 'gentlelinuxer' is shown, with the username and profile picture (highlighted with a red box) and the text 'commented now'. Below the comment, it says 'No description provided.' At the bottom, a commit preview for 'PR test' is visible, with a commit hash '72425b9'.

Outline

- 프로젝트 복사 (Fork & Clone)
- Project 읽기
- 개발 참여 준비 (Git config)
- 수정 작업 (Git branch & commit)
- Commit 제출하기 (Git push & GitHub Pull-request)
- Git 고급


파일 수정하고 수정 내용(diff) 확인하기

```
# 원하는 편집기로 mnist/main.py 파일 수정 후
$ nano mnist/main.py

# 수정한 내용 확인하기
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 166231a..6a9607a 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -6,7 +6,7 @@ import torch.nn.functional as F
     import torch.optim as optim
     from torchvision import datasets, transforms
     from torch.optim.lr_scheduler import StepLR
-
+import json

class Net(nn.Module):
    def __init__(self):
```

수정 내용 잠시 저장(stash)해두기



```
# 수정한 파일 확인하기
$ git status

# 수정한 내용 잠시 저장(stash) 하기
$ git stash

# 현재 소스폴더 상태 확인하기: 아무 수정분 없음을 확인
$ git status

# 잠시 저장(stash)해둔 내용 복구
$ git stash pop

# 복구된 수정한 파일 확인하기
$ git status
```

수정한 파일 복구하기

```
# 수정한 내용 확인하기
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 166231a..6a9607a 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -6,7 +6,7 @@ import torch.nn.functional as F
 import torch.optim as optim
 from torchvision import datasets, transforms
 from torch.optim.lr_scheduler import StepLR
-
+import json

class Net(nn.Module):
    def __init__(self):

# 파일 수정한 내용 최신역사를 기준으로 복구하기
# checkout 의미 local git 저장소 에서 "가져오다 / 대출받다" 의미
$ git checkout -- mnist/main.py

# 최신역사 기준으로 파일내용 복구 후 내용 확인
$ git diff
$ nano mnist/main.py
```

수정한 파일 복구하기

- 두 가지의 차이점은?

파일 수정하고 수정 내용(diff) 확인하기

```
# 원하는 편집기로 mnist/main.py 파일 수정 후
$ nano mnist/main.py

# 수정한 내용 확인하기
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 166231a..6a9607a 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -6,7 +6,7 @@ import torch.nn.functional as F
     import torch.optim as optim
     from torchvision import datasets, transforms
     from torch.optim.lr_scheduler import StepLR
-
+import json

class Net(nn.Module):
    def __init__(self):
```

commit을 준비하는 add 명령 취소하기 (reset)

```
# 수정한 내용 기준으로 commit 할 준비하기
$ git add mnist/main.py

# commit 할 준비완료 상태 확인
$ git status
On branch fix-mnist
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
    modified:   mnist/main.py

# add 명령 취소 후
$ git reset

# commit 할 준비취소 후 상태 확인
$ git status
On branch fix-mnist
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
    modified:   mnist/main.py

no changes added to commit (use "git add" and/or "git commit -a")
```


소스 수정내역(commit) 삭제 하기

```
# 수정한 내용 기준으로 commit 할 준비하기
$ git add mnist/main.py

# commit 만들기
$ git commit -m "Add import json"

# 생성한 commit 정보 확인하기
$ git show
$ git log --oneline -1

# commit 정보 삭제하기
# 참고: HEAD~1 은 가장 위에서 첫번째 내용을 삭제한다는 의미
$ git reset --hard HEAD~1

# 삭제 후 가장 최신 commit 확인하기
$ git log --oneline -1
```

파일 수정하고 수정 내용(diff) 확인하기

```
# 소스 내용 수정 후 확인
$ nano mnist/main.py
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 6a9607a..56da205 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -7,6 +7,7 @@ import torch.optim as optim
     from torchvision import datasets, transforms
     from torch.optim.lr_scheduler import StepLR
     import json
+    import requests

class Net(nn.Module):
    def __init__(self):
```

라이선스 서명 넣어서 commit message 적기

```
# 수정한 내용 기준으로 commit 할 준비하기
$ git add mnist/main.py

# commit 만들기
# -s 옵션 포함시 라이선스 서명을 의미하는 Signed-off-by 내용을
# commit message 안에 포함하게 된다.
$ git commit -sm "Add import requests"


# commit message 안에 Signed-off-by 확인
$ git show
```

파일 수정하고 수정 내용(diff) 확인하기

```
# 소스 내용 수정 후 확인
$ nano mnist/main.py
$ git diff
diff --git a/mnist/main.py b/mnist/main.py
index 6a9607a..56da205 100644
--- a/mnist/main.py
+++ b/mnist/main.py
@@ -7,6 +7,7 @@ import torch.optim as optim
     from torchvision import datasets, transforms
     from torch.optim.lr_scheduler import StepLR
     import json
-    import requests
+    import requests as req

class Net(nn.Module):
    def __init__(self):
```

commit 수정하기 (amend)



```
# 최신 commit 수정이전 commit ID 확인하기
$ git log --oneline -1
106d167 (HEAD -> fix-mnist) Add import requests

# 수정한 내용 commit 할 준비 하기
$ git add mnist/main.py

# 최신 commit 수정하기
$ git commit --amend

# 최신 commit 수정 이후 commit ID 확인하기
$ git log --oneline -1
9786c82 (HEAD -> fix-mnist) Add import requests
```

commit 수정(amend) 이후 Before / After 확인하기

```
# 최신 commit 수정이전 commit ID 확인하기
$ git log --oneline -1
106d167 (HEAD -> fix-mnist) Add import requests

# 수정한 내용 commit 할 준비 하기
$ git add mnist/main.py

# 최신 commit 수정하기
$ git commit --amend

# 최신 commit 수정 이후 commit ID 확인하기
$ git log --oneline -1
9786c82 (HEAD -> fix-mnist) Add import requests
```

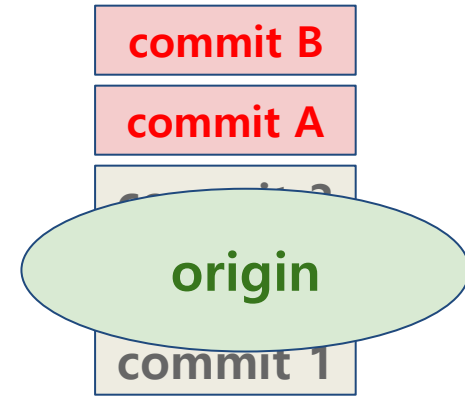
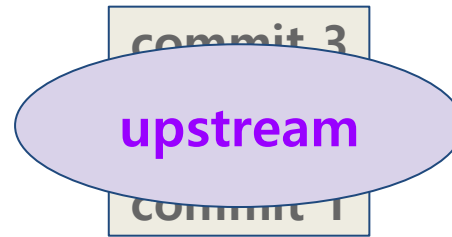
나의 Fork 저장소와 오픈소스 GitHub 저장소 설정

```
# 오픈소스 공식 GitHub 프로젝트 URL  
# upstream 으로 등록 하기
```

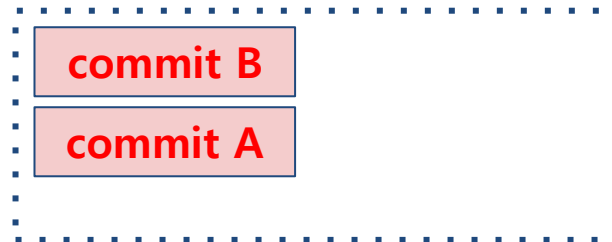
```
root@goorm:/workspace/pytorch/examples(master)# git remote add upstream https://github.com/issong-DKU/pytorch-example  
# origin: 나의 Fork 저장소 GitHub URL  
# upstream: 오픈소스 공식 GitHub URL (또는 팀프로젝트 URL)  
root@goorm:/workspace/pytorch/examples(master)# git remote -v  
origin https://github.com/inshiksong-dku/pytorch-example.git (fetch)  
origin https://github.com/inshiksong-dku/pytorch-example.git (push)  
upstream https://github.com/issong-DKU/pytorch-example (fetch)  
upstream https://github.com/issong-DKU/pytorch-example (push)  
root@goorm:/workspace/pytorch/examples(master)#
```

Rebase 가 필요한 상황

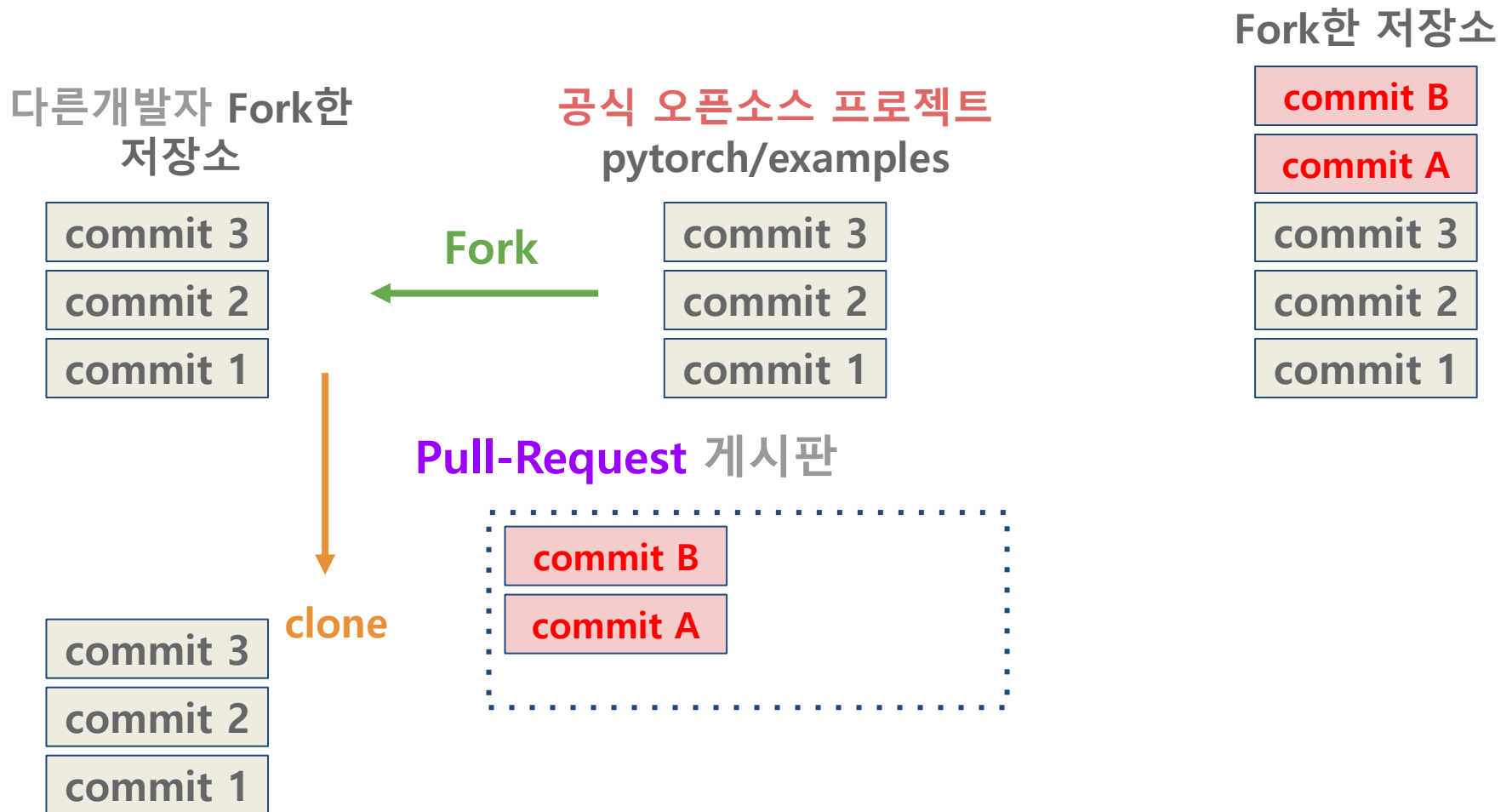
공식 오픈소스 프로젝트
pytorch/examples



Pull-Request 게시판

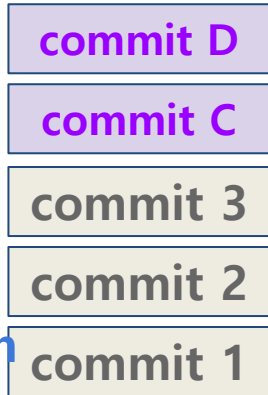


Rebase 가 필요한 상황

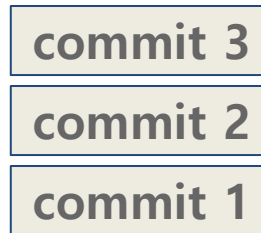


Rebase 가 필요한 상황

다른개발자 Fork한
저장소



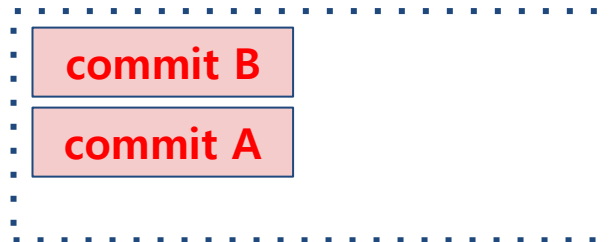
공식 오픈소스 프로젝트
pytorch/examples



Fork한 저장소



Pull-Request 게시판



push

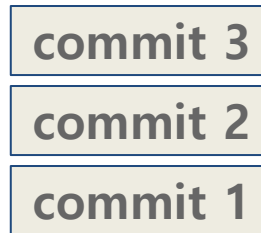


Rebase 가 필요한 상황

다른개발자 Fork한
저장소



공식 오픈소스 프로젝트
pytorch/examples

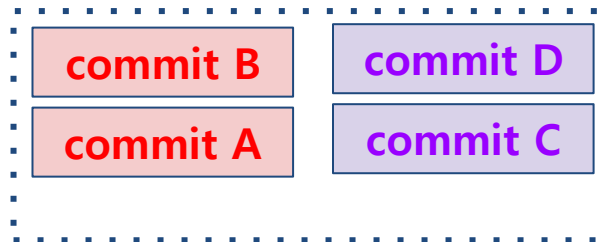


Fork한 저장소



PR

Pull-Request 게시판



Rebase 가 필요한 상황

공식 오픈소스 프로젝트

pytorch/examples



Fork한 저장소



Rebase 필요 !

(최신역사로 베이스 업데이트)

Rebase 가 필요한 상황

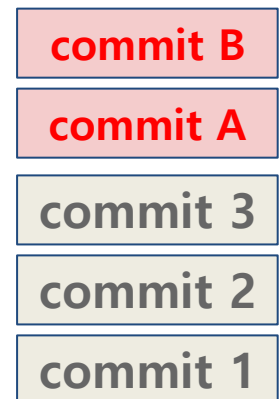
공식 오픈소스 프로젝트
pytorch/examples



Fork한 저장소



fetch

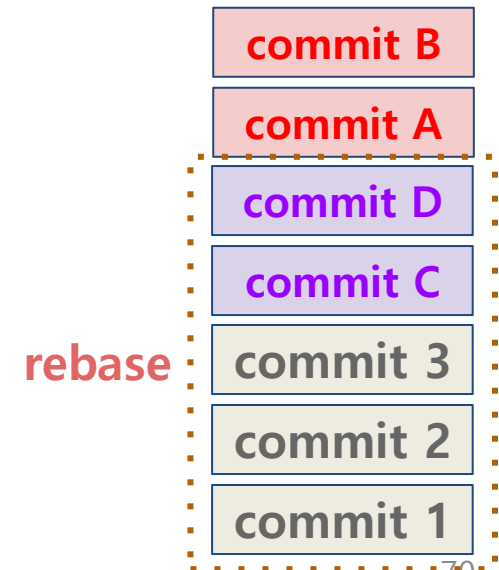


Rebase 가 필요한 상황

공식 오픈소스 프로젝트
pytorch/examples



Fork한 저장소

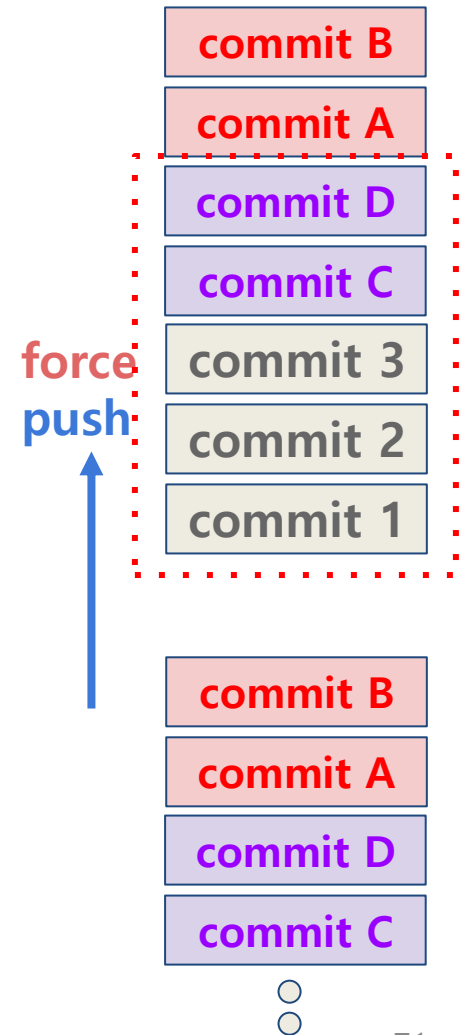


Rebase 가 필요한 상황

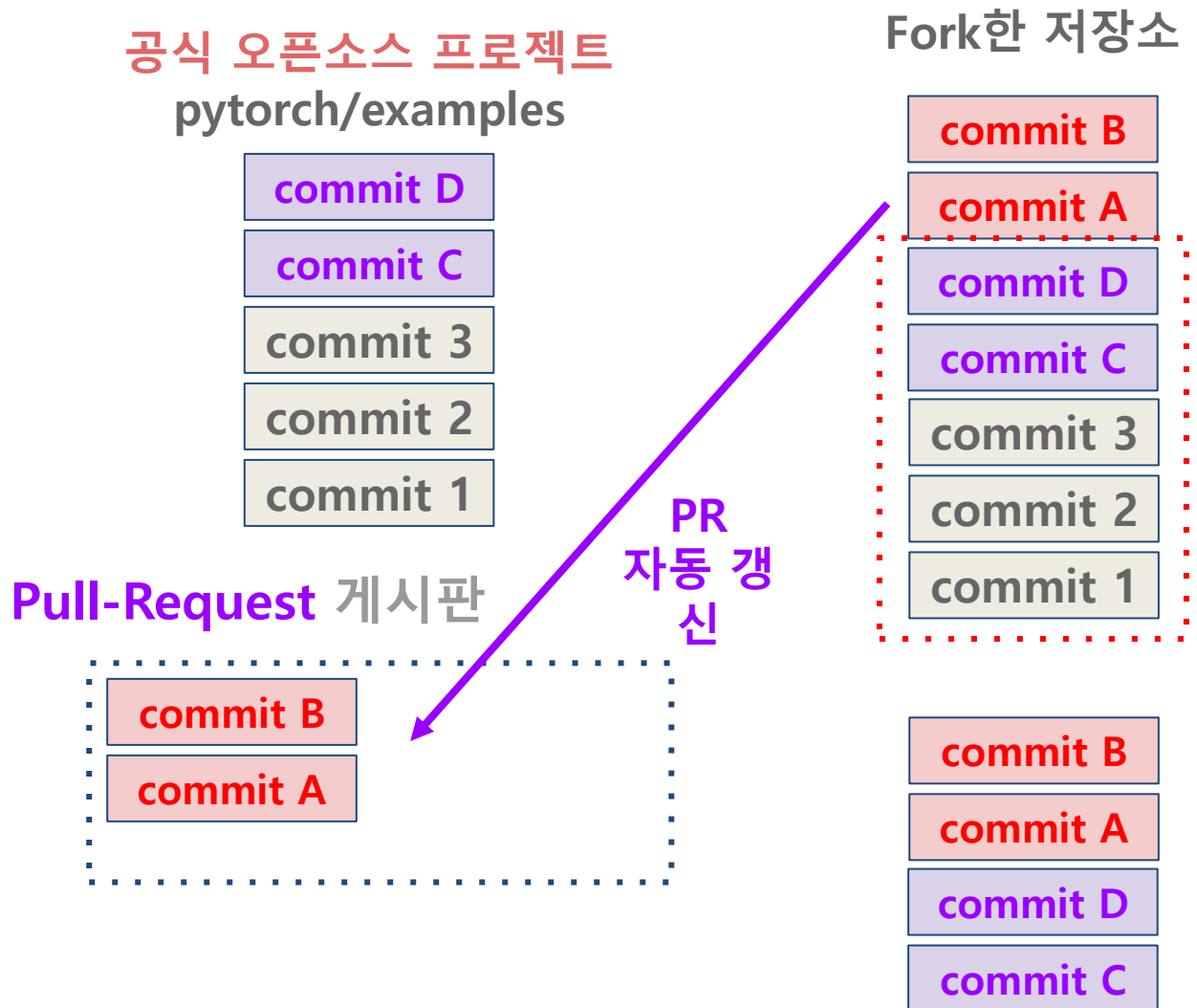
공식 오픈소스 프로젝트
pytorch/examples



Fork한 저장소



Rebase 가 필요한 상황



최신 소스수정내역(commit)로 Base 업데이트 Rebase

```
# 공식 upstream 저장소에서 최신 commit history 가져오기
$ git fetch upstream master

# 최신 commit history 기준으로 베이스 갱신 (rebase)
$ git rebase upstream/master

# Fork 한 저장소(GitHub)도 수정하기 (PR 자동 갱신)
$ git push --force origin fix-mnist
```

Questions?