

# 분산 버전 관리 도구: Git

## Advanced #1

---

Hyunchan, Park

<http://oslab.jbnu.ac.kr>

Division of Computer Science and Engineering

Jeonbuk National University



# 학습 내용

---

- GUI로 Git 관리: GitHub Desktop (#1)
  - 개인실습 3
- Git: Branch 관리 (#2)
  - 개인실습 4
- Git을 이용한 팀플레이: 1. Start!
  - 팀 실습 1

# 참고 자료

---

- 장병진, “오픈소스 개발을 위한 GIT 사용법 실습”
- 조성수, “GitHub으로 학교 팀프로젝트 하기”
- 신승엽, “GitHub 실습”
  
- Git 홈페이지 Documentation
  - <https://git-scm.com/book/ko/v2>
- 누구나 쉽게 이해하는 Git 입문
  - <https://backlogtool.com/git-guide/kr/>
  - (꼭 살펴볼 것)

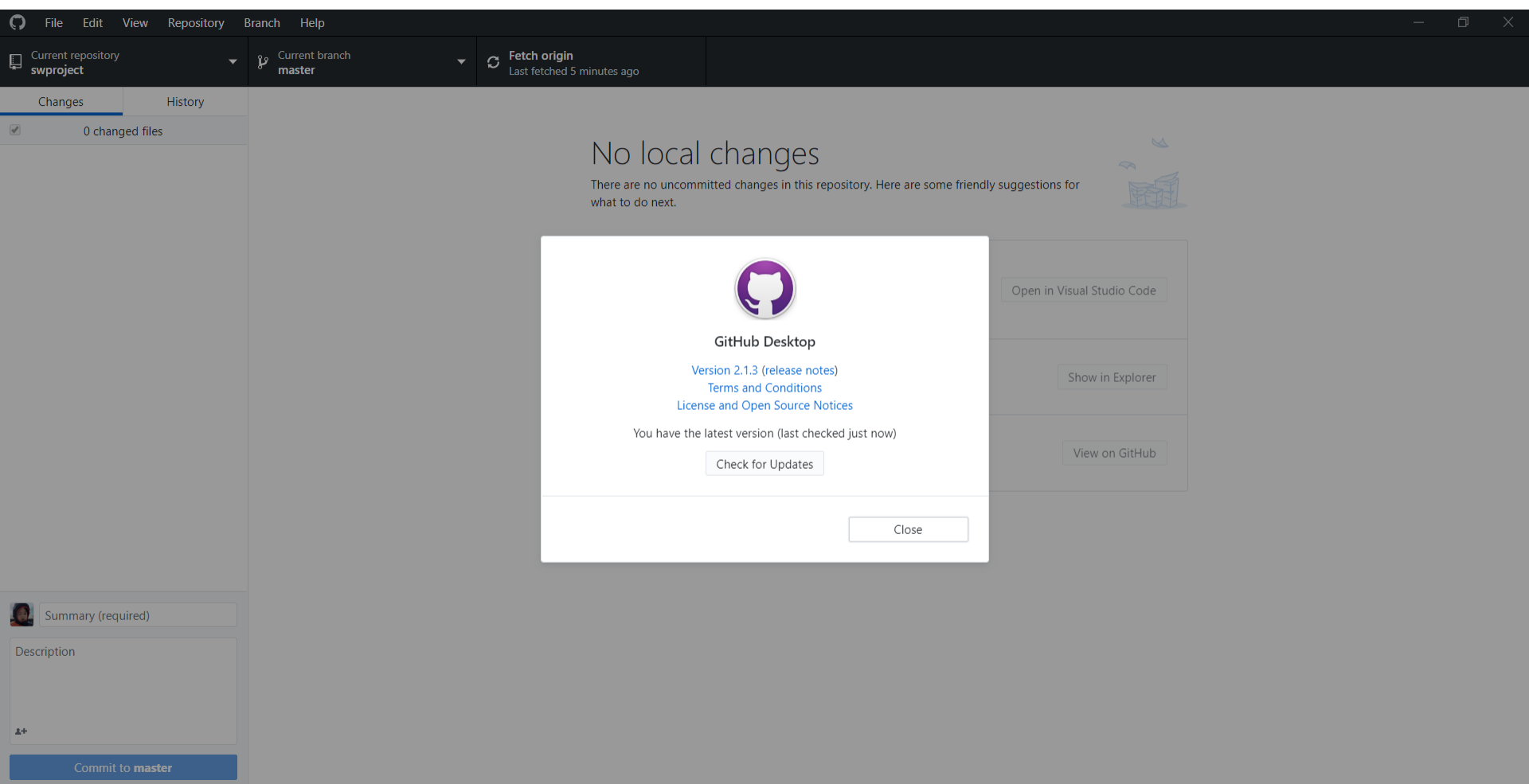
---

# GUI로 Git 관리:

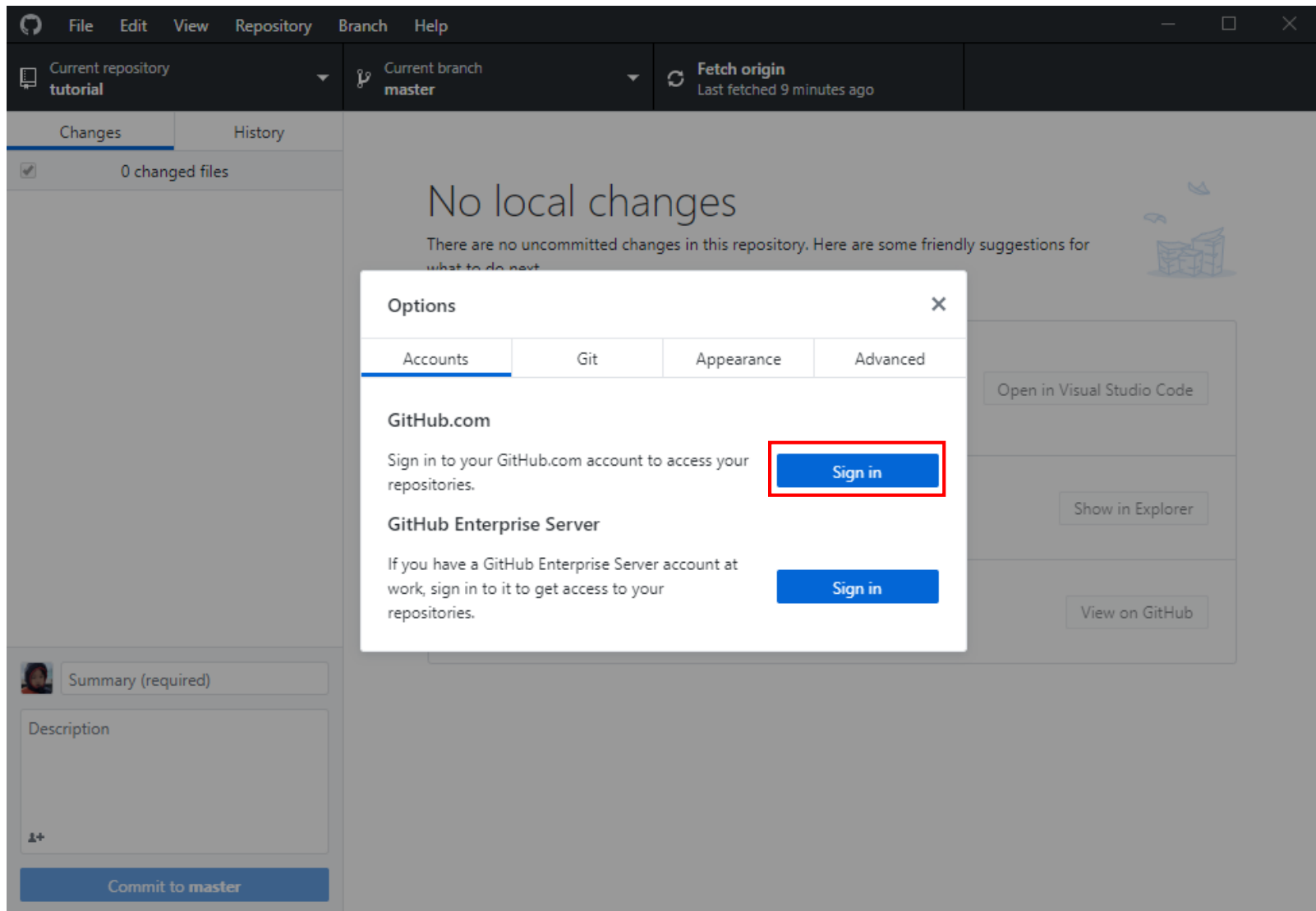
## GitHub Desktop

# GitHub Desktop: Download and Install

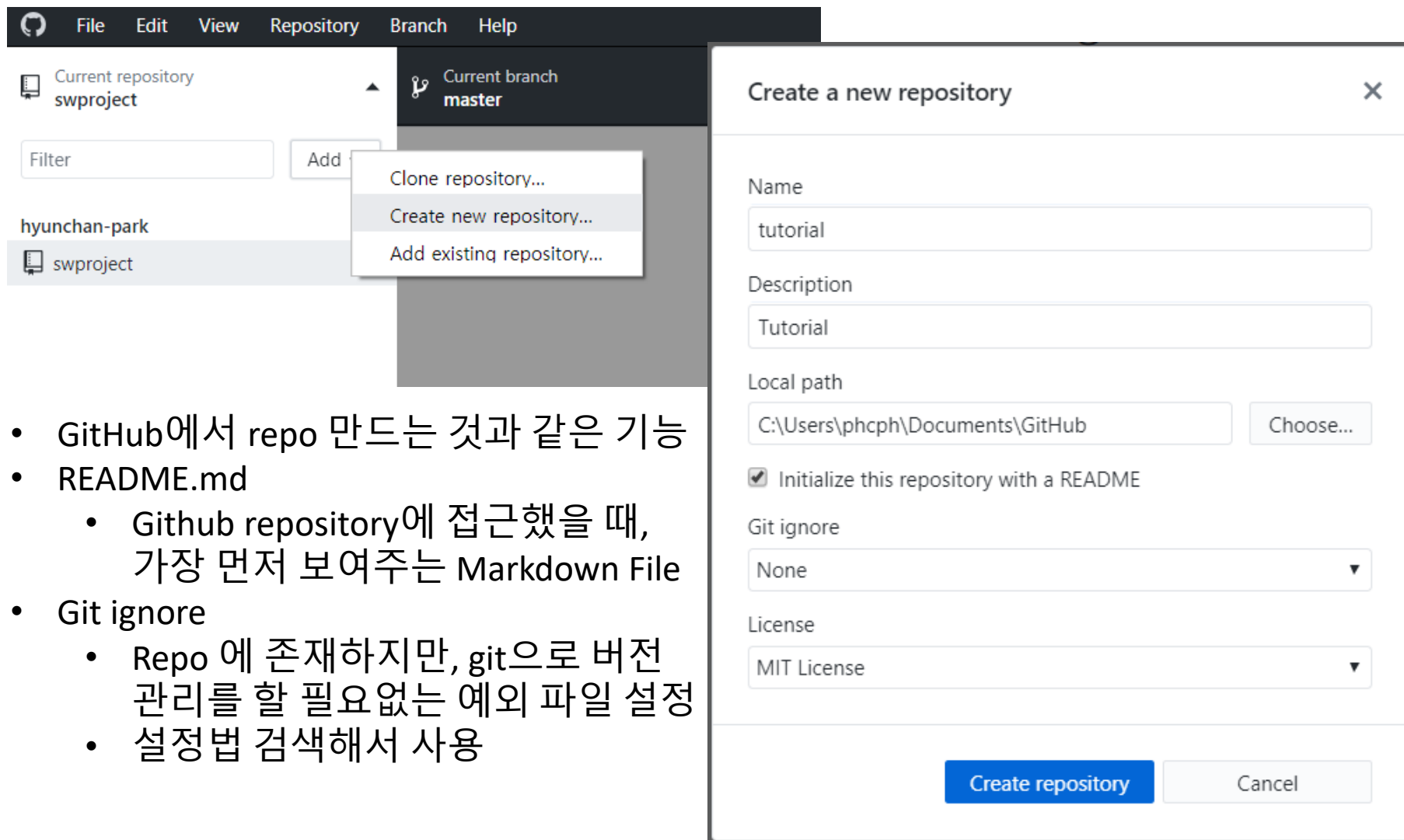
<https://desktop.github.com/>



# Sign in: File -> options



# Create new repository



The screenshot shows the Visual Studio Code interface with the 'Create a new repository' dialog box open. The dialog box has a title bar with a close button (X). It contains the following fields and options:

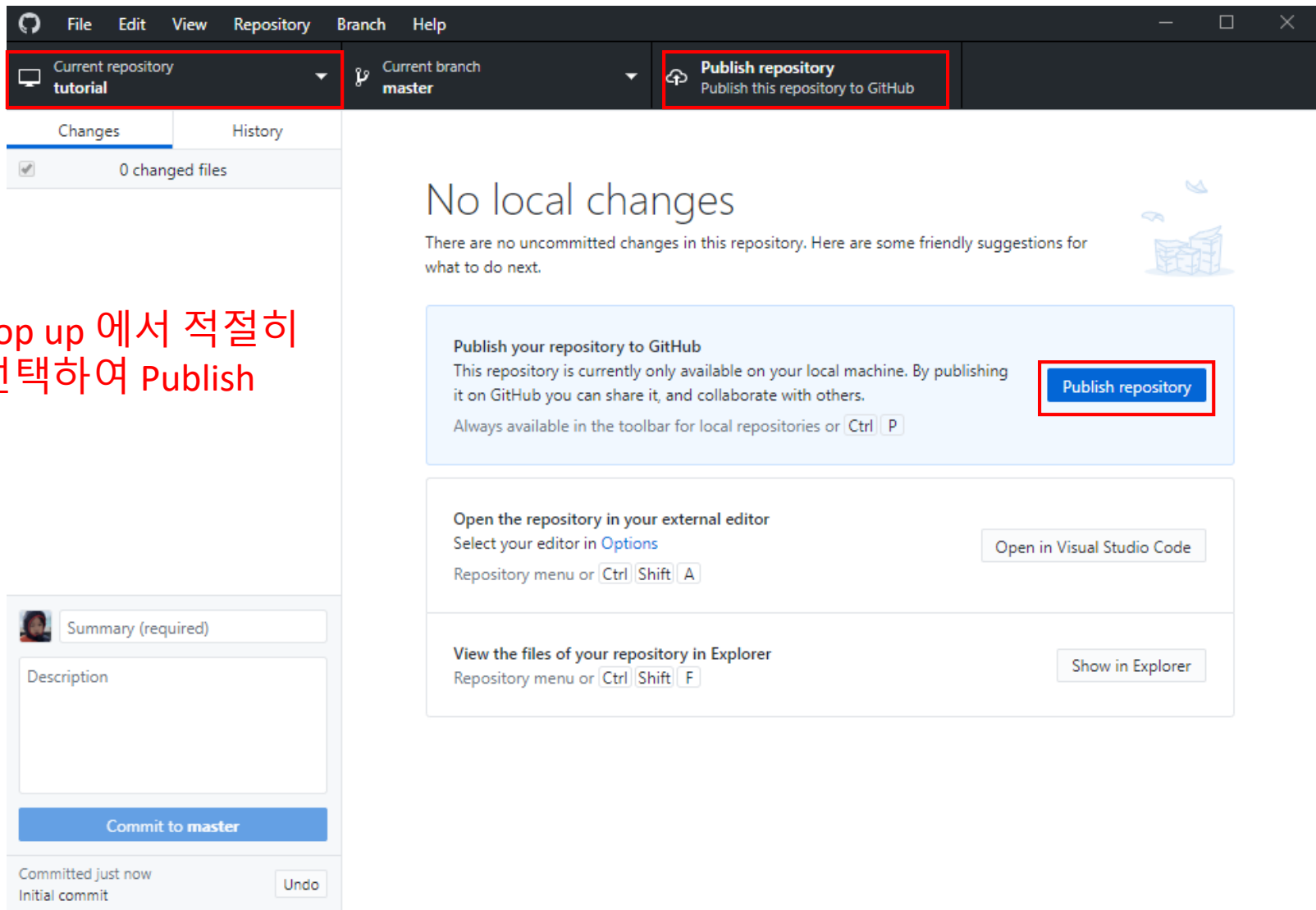
- Name:** A text input field containing 'tutorial'.
- Description:** A text input field containing 'Tutorial'.
- Local path:** A text input field containing 'C:\Users\phcph\Documents\GitHub' and a 'Choose...' button.
- Initialize this repository with a README:** A checked checkbox.
- Git ignore:** A dropdown menu showing 'None'.
- License:** A dropdown menu showing 'MIT License'.
- Buttons:** 'Create repository' (blue) and 'Cancel' (gray).

The background shows the VS Code interface with the 'Repository' menu open, highlighting 'Create new repository...'. The 'Current repository' is 'swproject' and the 'Current branch' is 'master'.

- GitHub에서 repo 만드는 것과 같은 기능
- README.md
  - Github repository에 접근했을 때, 가장 먼저 보여주는 Markdown File
- Git ignore
  - Repo 에 존재하지만, git으로 버전 관리를 할 필요없는 예외 파일 설정
  - 설정법 검색해서 사용

# Publish repository to GitHub

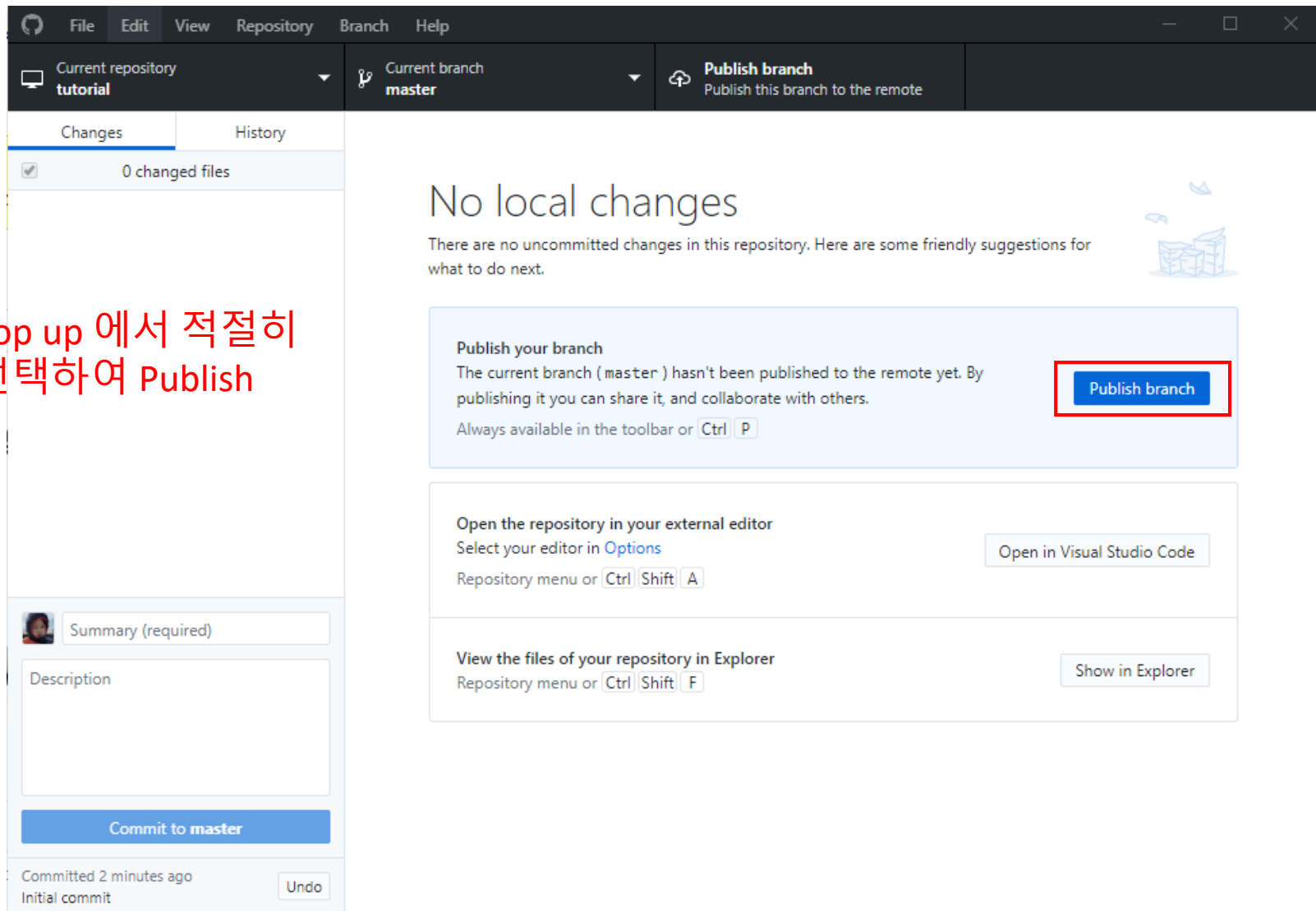
- Pop up 에서 적절히 선택하여 Publish





# Publish Master branch to GitHub

- Pop up 에서 적절히 선택하여 Publish



# GitHub 에서 확인

The screenshot shows the GitHub Desktop application window. The top bar includes a menu (File, Edit, View, Repository, Branch, Help) and status information: 'Current repository: tutorial', 'Current branch: master', and 'Fetch origin: Last fetched 5 minutes ago'. The left sidebar has tabs for 'Changes' and 'History', with 'Changes' selected, showing '0 changed files'. The main area displays 'No local changes' with a message: 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' Below this, three suggestions are listed: 'Open the repository in your external editor' (with a button 'Open in Visual Studio Code'), 'View the files of your repository in Explorer' (with a button 'Show in Explorer'), and 'Open the repository page on GitHub in your browser' (with a button 'View on GitHub'). The third suggestion and its button are highlighted with a red rectangular box. At the bottom left, there is a commit summary section with a 'Summary (required)' field, a 'Description' field, and a 'Commit to master' button.

# GitHub 에서 확인

[hyunchan-park / tutorial](#)

Watch 0Star 0Fork 0


CodeIssues 0Pull requests 0Projects 0WikiSecurityInsightsSettings

TutorialEdit

Manage topics

1 commit1 branch0 releases1 contributorMIT

Branch: master ▼New pull requestCreate new fileUpload filesFind FileClone or download ▼

 hyunchan-park Initial commitLatest commit 4b46c1a 5 minutes ago

<a href="#">.gitattributes</a>	Initial commit	5 minutes ago
<a href="#">LICENSE</a>	Initial commit	5 minutes ago
<a href="#">README.md</a>	Initial commit	5 minutes ago

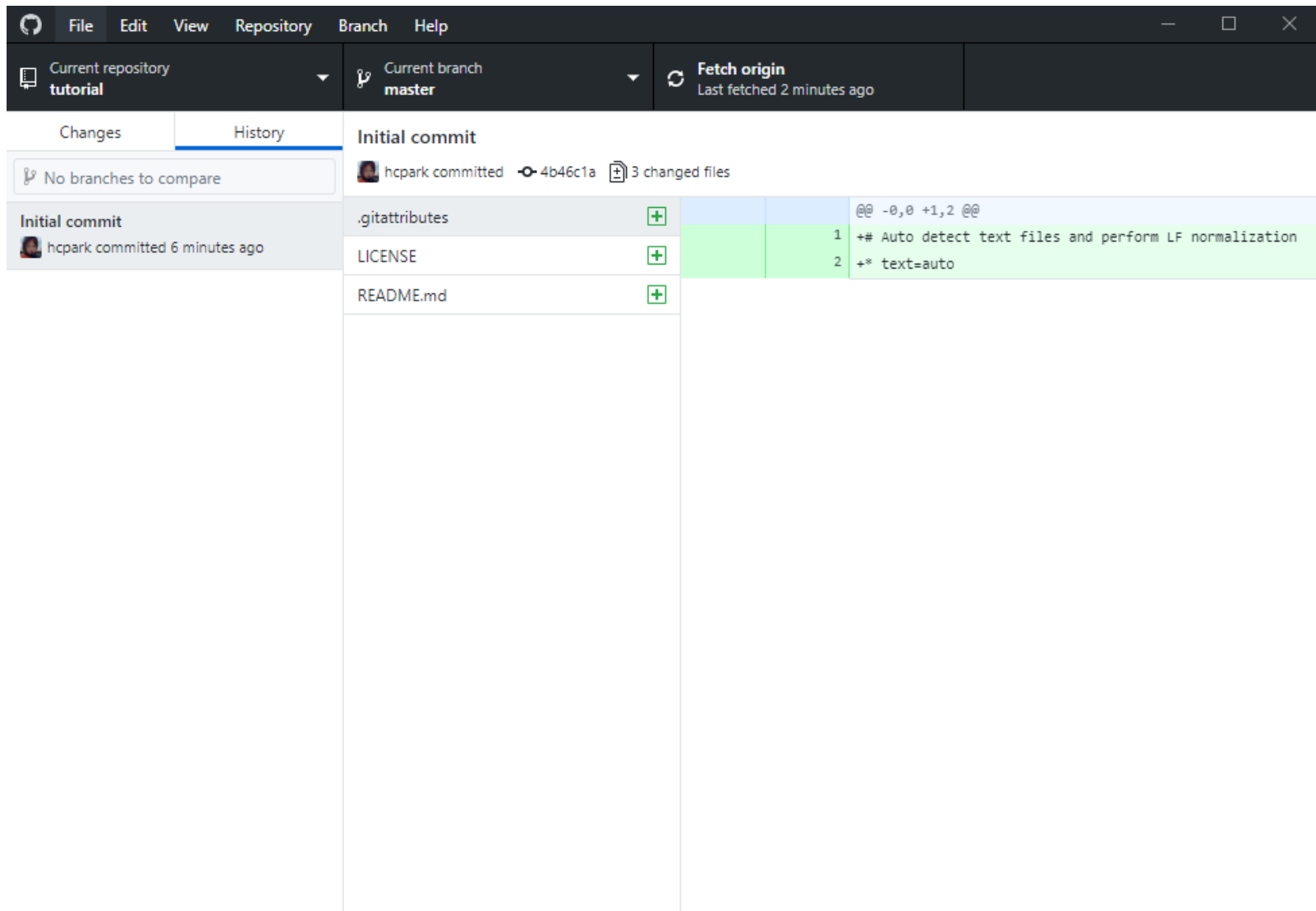
README.md

## tutorial

---

Tutorial

# GitHub Desktop: History 확인 (1 commit)



GitHub Desktop interface showing the commit history for the 'tutorial' repository on the 'master' branch. The 'History' tab is active, displaying a single commit: 'Initial commit' by hcpark, committed 6 minutes ago. The commit details show three files: .gitattributes, LICENSE, and README.md. The diff view shows changes to the LICENSE file, specifically adding lines for LF normalization and text=auto.

File	Changes
.gitattributes	+1
LICENSE	+2
README.md	+1

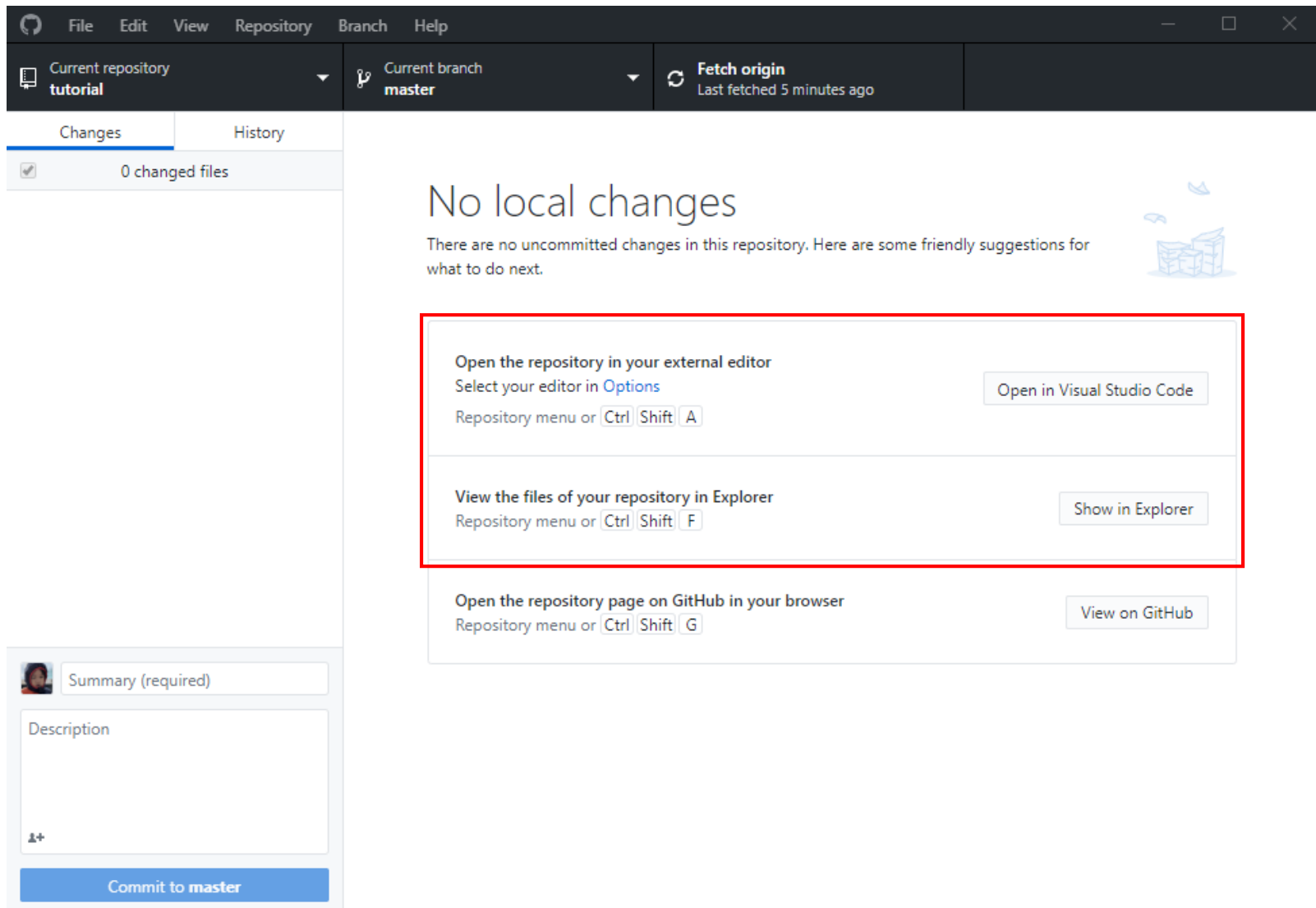
@@ -0,0 +1,2 @@  
1 +# Auto detect text files and perform LF normalization  
2 +\* text=auto

---

# Commit



# Open Explorer and Create a new file



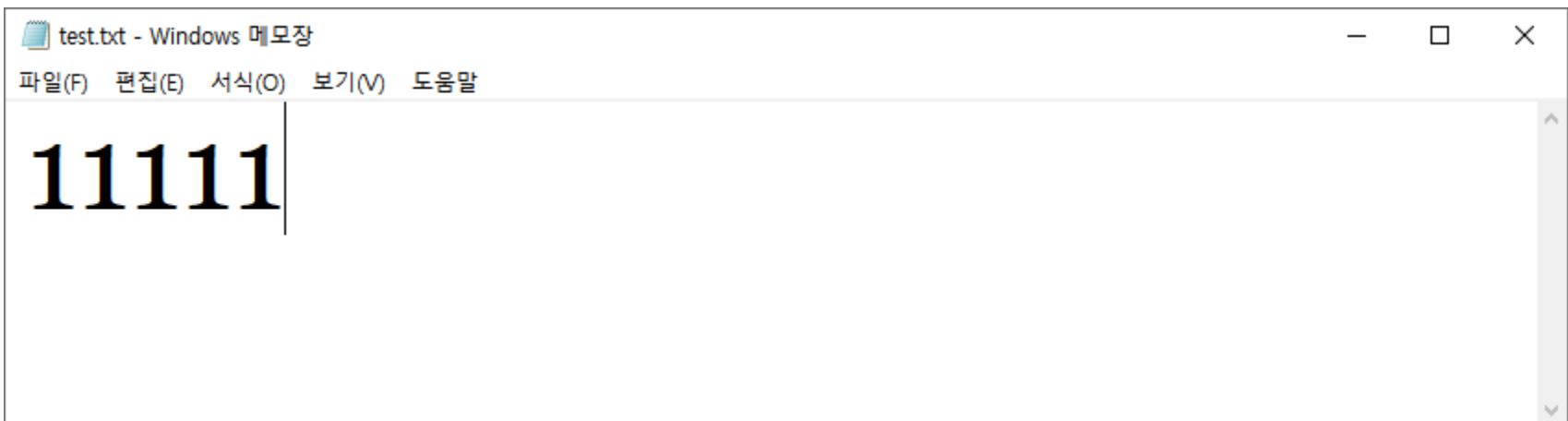
# test.txt 생성 및 편집

tutorial

공유 보기

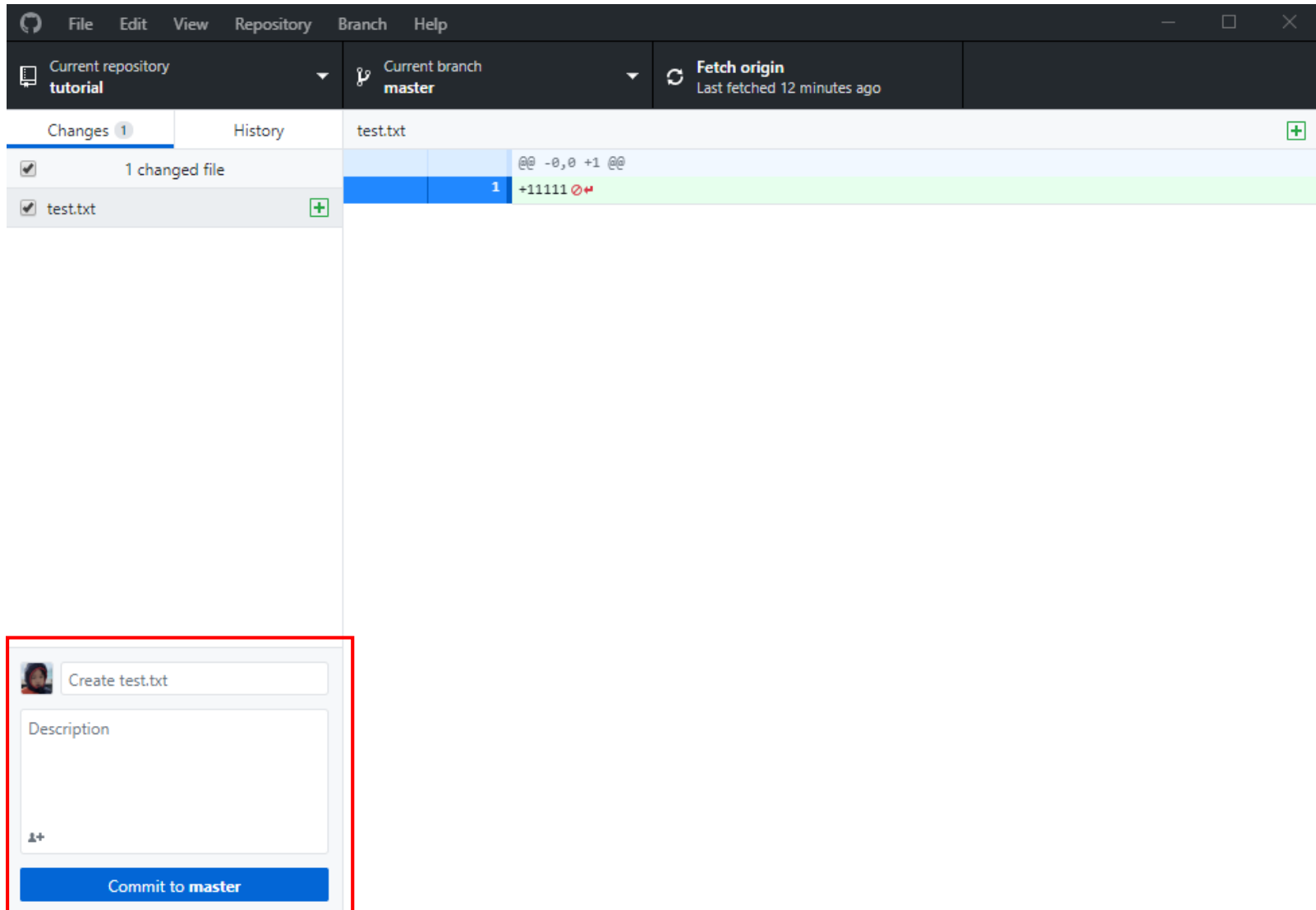
내 PC > 문서 > GitHub > tutorial

이름	수정한 날짜	유형	크기
.git	2019-09-25 오후 10:50	파일 폴더	
.gitattributes	2019-09-25 오후 10:46	텍스트 문서	1KB
LICENSE	2019-09-25 오후 10:46	파일	2KB
README.md	2019-09-25 오후 10:46	MD 파일	1KB
test.txt	2019-09-25 오후 11:00	텍스트 문서	0KB



# GitHub Desktop 확인

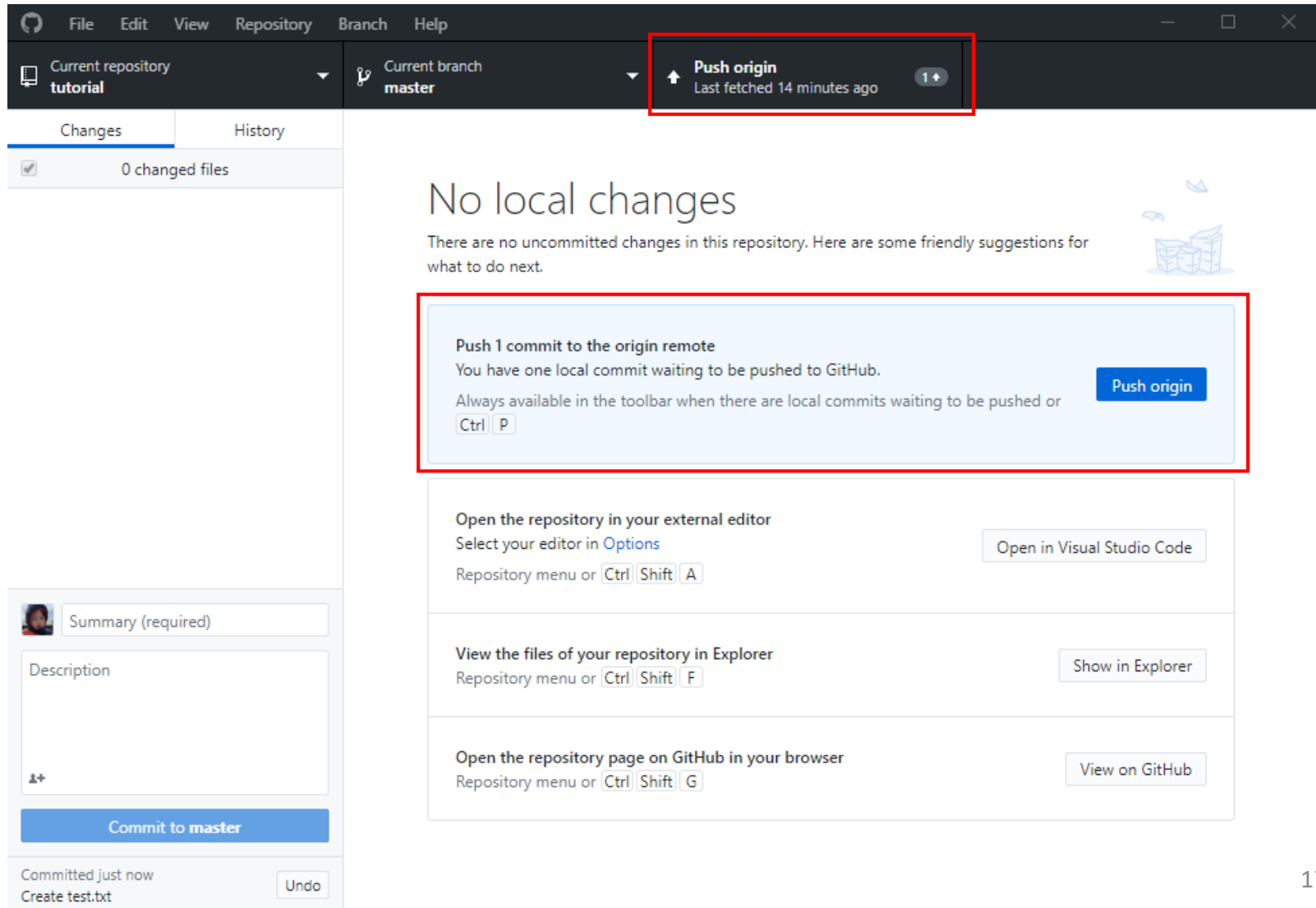
- 자동으로 Changes 에 반영된 것을 볼 수 있음. 바로 commit 해보자!





# Commit 완료

- Staging 과정 없이 즉각 commit이 완료됨. Push 버튼 활성화됨 -> PUSH!



# GitHub 확인

hyunchan-park / tutorial

Watch 0

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Tutorial

Edit

Manage topics

2 commits

1 branch

0 releases

1 contributor

MIT

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download



hyunchan-park Create test.txt

Latest commit cc3e11a 3 minutes ago

.gitattributes

Initial commit

21 minutes ago

LICENSE

Initial commit

21 minutes ago

README.md

Initial commit

21 minutes ago

test.txt

Create test.txt

3 minutes ago

# GitHub Desktop 의 활용

---

- Local repo 와 Remote repo 의 동기화를 위해 간단히 사용
- Pull, modification and Push
- Commit의 수정, Branch 의 세세한 조정 등은 실제 개발자가 수행할 필요가 거의 없음
  - 프로젝트 매니저, Repository 및 코드 관리자가 수행
- 많은 IDE에서 GitHub Desktop 을 대체할 수 있는 GitHub 연동 기능을 제공하고 있으므로, 그러한 경우에는 아예 불필요함
  - Visual Studio Code, Eclipse 등 널리 사용되는 IDE는 거의 모두 지원
  - 차후 배우게 될 Source Insight는 지원하지 않음
  - 이러한 경우, source code directory 를 git repo 로 설정해두고, source insight (혹은 다른 에디터)로 수정 후, GitHub Desktop 으로 push

---

# Git: Branch 관리



# 실습 과제 (git 개인 실습 #4)

---

- 두 개의 브랜치를 생성하고, 각 브랜치에서 1번씩 커밋
  - 같은 파일에 대해 서로 다른 내용으로 수정해 커밋할 것
- Merge의 충돌 관리
  - 위 두 브랜치를 merge 하며 발생하는 충돌을 해결할 것
  - 결과 화면 캡처: merge.jpg



# Branch

---

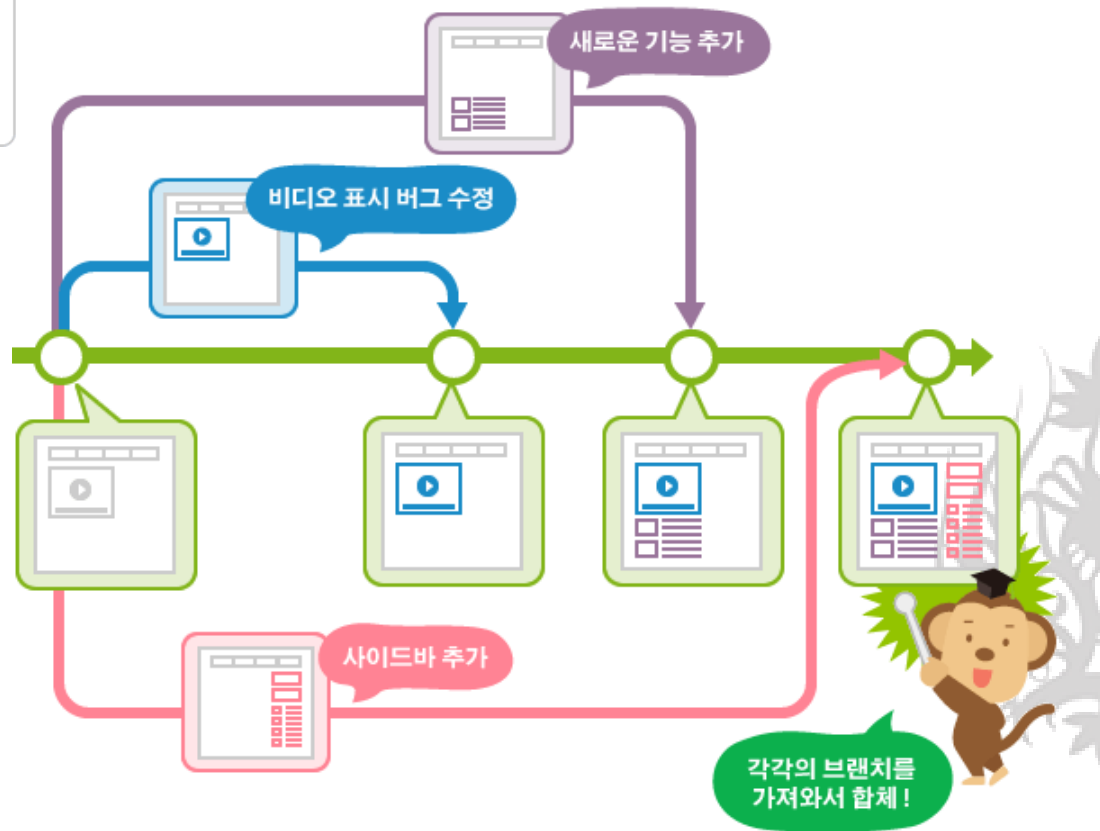
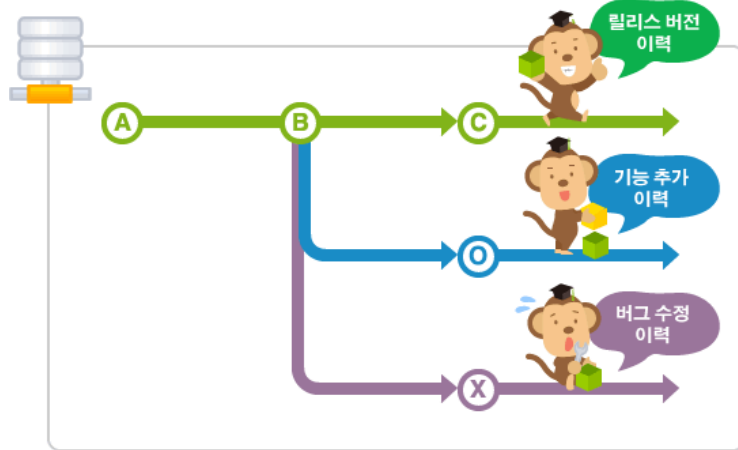
- vcs의 한계

- 충돌이 나는 파일은 수동으로 해결할 수 밖에 없음
- 따라서 최대한 분리하여 작업하도록 구성
  - 단위: 함수, 클래스, 파일, 폴더, 기능 등
- 그러나 협업 구조에서 동시 작업의 필요성은 여전히 존재
  - 기능 추가 팀, 버그 수정 팀이 동시에 작업

- Branch

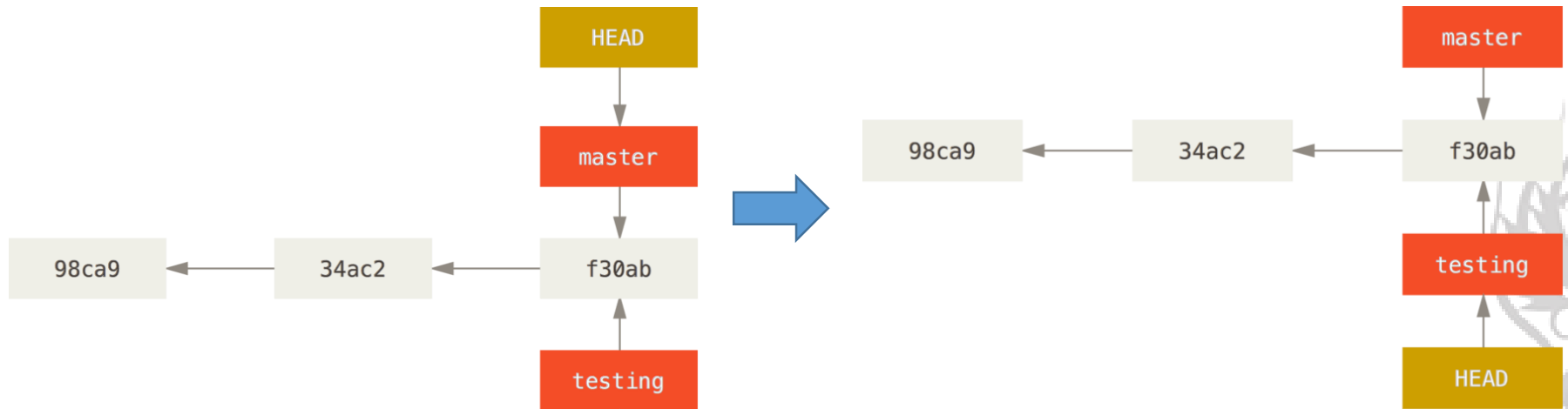
- 같은 코드를 공유하되, 작업을 독립적으로 분리해서 수행할 수 있음
  - 독립적인 개발 및 테스트, 커밋 이력 관리, 저장소 관리
  - Git은 가볍고 빠른 브랜치 동작 제공
- 분리는 쉽다, 나중에 다시 합칠 때는?
  - Merge 담당 전문가, 테스트 과정 필요

# Branch 이용 예



# Branch 생성 및 HEAD의 이동

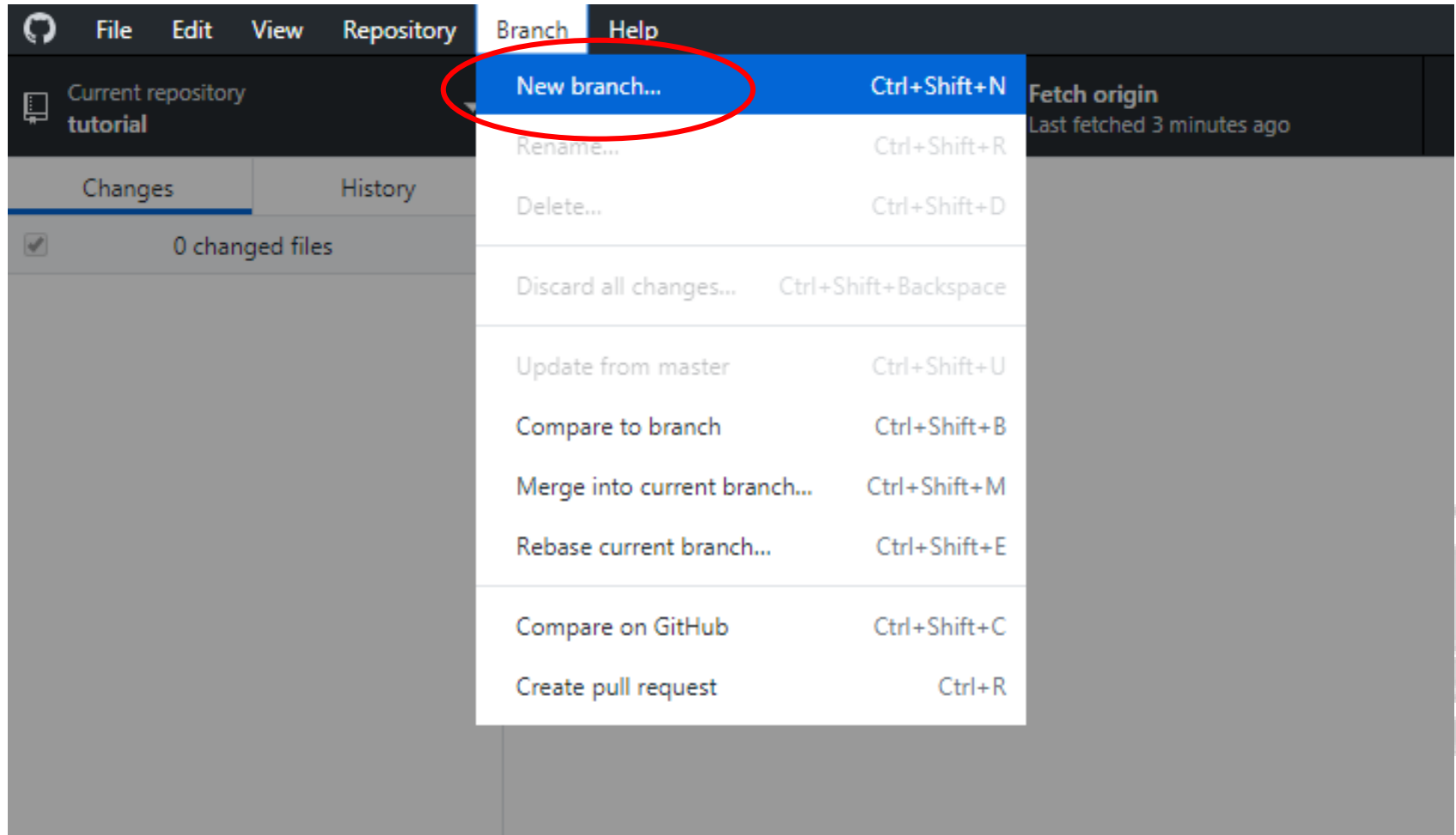
- \$ git branch testing
  - 새로운 브랜치 생성
  - HEAD의 위치는 변경되지 않음
- \$ git checkout testing
  - HEAD의 위치를 변경
  - 작업 브랜치가 변경됨. 이후 커밋은 해당 브랜치에 추가됨





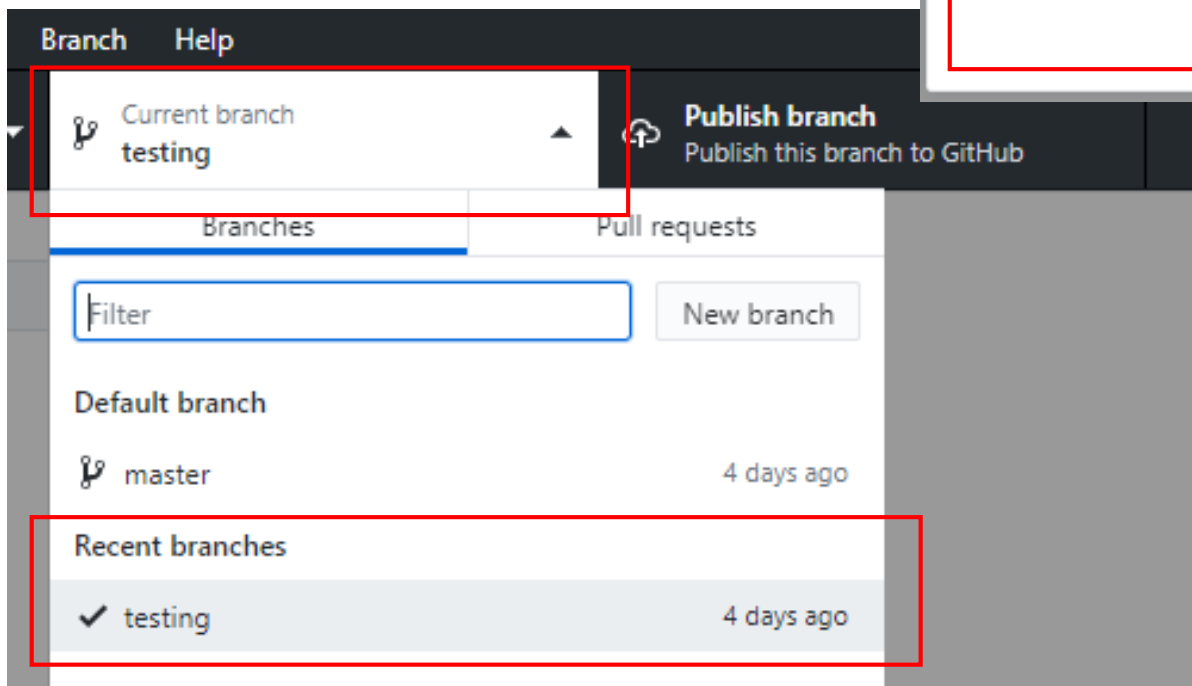
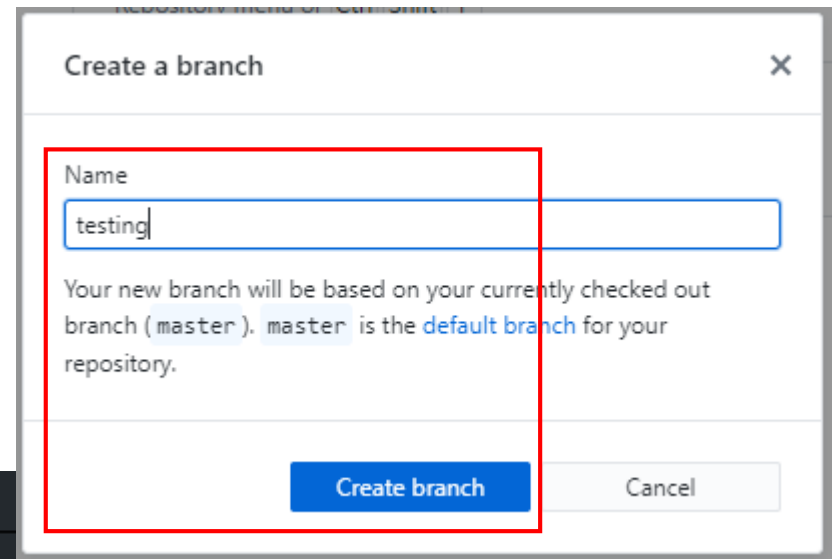
# Branch 생성 및 HEAD의 이동

- GitHub Desktop에서의 branch



# Branch 생성 및 HEAD의 이동

- Branch 생성 후,
- 자동으로 branch 변경됨
- (변경 여부 확인하고, 안됐으면 branch 에서 변경)



- 아직 publish 하지는 말자

- Branch 가 생성된 것을 remote repository에 아직 알리지 말자

## No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



### Publish your branch

The current branch ( `testing` ) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

[Publish branch](#)

Always available in the toolbar or `Ctrl` `P`

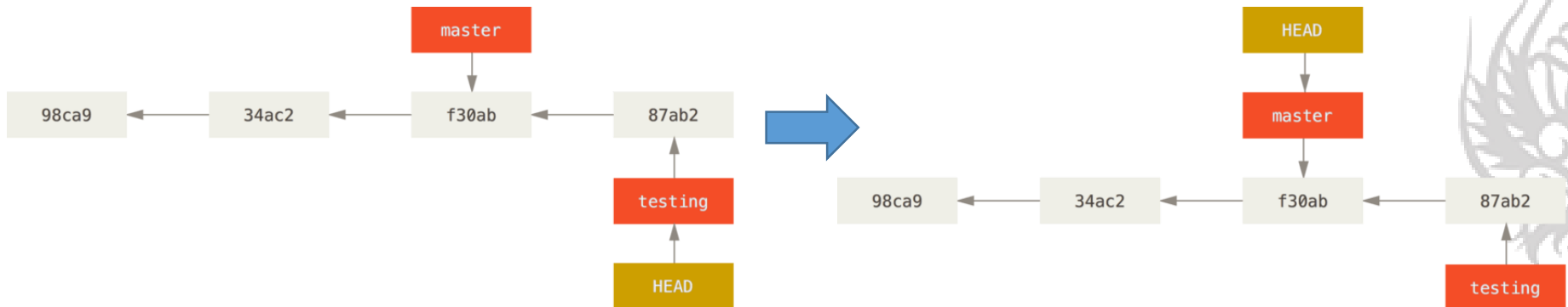


# Branch 생성 및 HEAD의 이동

- \$ notepad test.c
- \$ git commit -a -m 'made a change'
- \$ git checkout master



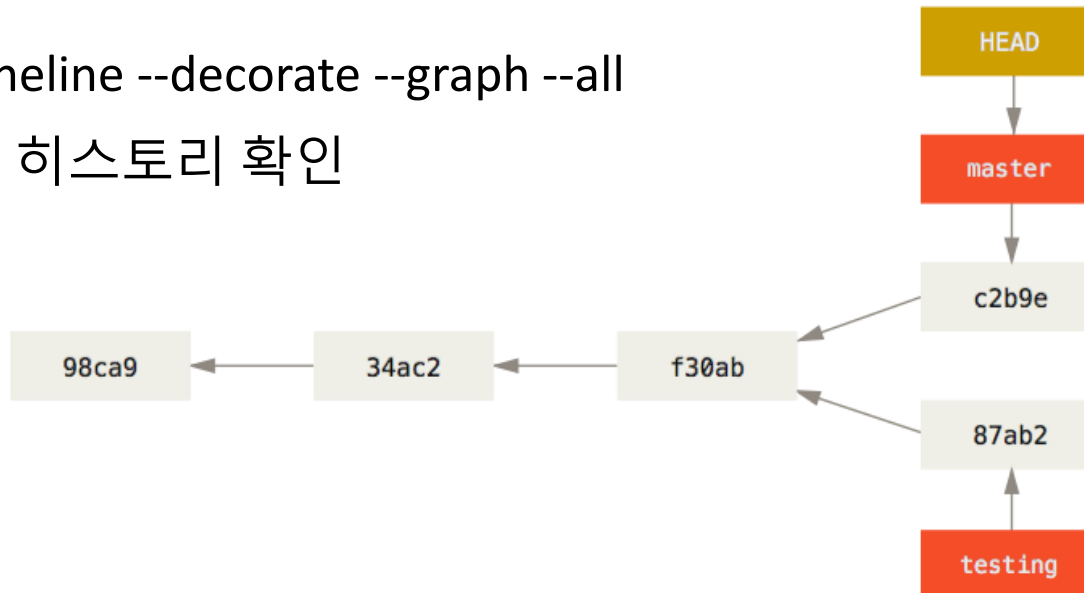
Current repository tutorial		Current branch testing	
Changes 1		History	
1 changed file		test.txt	
test.txt		1	1
		@@ -1 +1 @@	
		-11111	
		+11111@testing	



# Branch 생성 및 HEAD의 이동

- \$ notepad test.c
  - 이때 내용을 한번 확인해보자 (master 브랜치에서는 아직 변경하지 않았음)
  - 11111@master 로 변경
- \$ git commit -a -m 'made other changes'
  - 분리된 브랜치에서 각각 커밋 완성
  - 현재 HEAD는 master 브랜치에 위치
- \$ git log --oneline --decorate --graph --all
  - 브랜치 히스토리 확인

Current repository tutorial		Current branch master	
Changes 1		History	
1 changed file		test.txt	
test.txt		1	1
		@@ -1 +1 @@ -11111 +11111@master	



# Branch history @ Git Bash

---

```
phcph@LAPTOP-1PM0RJFD MINGW64 ~/Documents/GitHub/tutorial (master)
$ git log --oneline --decorate --graph --all
* 9dae91f (HEAD -> master) made other changes
| * f7368c0 (testing) made a change
|/
* cc3e11a (origin/master) Create test.txt
* 4b46c1a Initial commit

phcph@LAPTOP-1PM0RJFD MINGW64 ~/Documents/GitHub/tutorial (master)
$ |
```

# Branch 이동하여 파일 내용 확인

- GitHub Desktop에서 branch 확인 후
  - 두 개 브랜치를 번갈아 변경 (checkout)
  - 각 브랜치 checkout 후, test.c 파일 내용 확인



# Push 하고, GitHub 에서 Commit 내역 확인

- 두 브랜치에서 각각 push, publish 수행하고,
- GitHub 에서 내용 확인

The screenshot shows the GitHub interface for a repository named 'tutorial' by user 'hyunchan-park'. At the top, there are buttons for 'Watch', 'Star', and 'Fork', each with a count of 0. Below these are tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. The repository name 'Tutorial' is displayed with an 'Edit' button. Below this is a 'Manage topics' link. A summary bar shows '3 commits', '1 branch', '0 releases', '1 contributor', and the license 'MIT'. A section titled 'Your recently pushed branches:' highlights the 'testing' branch, pushed '1 minute ago', with a 'Compare & pull request' button. Below this are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. The commit history table shows four entries: '.gitattributes', 'LICENSE', and 'README.md' (all 'Initial commit', '4 days ago') and 'test.txt' ('made other changes', '34 minutes ago').

hyunchan-park / tutorial

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Tutorial Edit

Manage topics

3 commits 1 branch 0 releases 1 contributor MIT

Your recently pushed branches:

testing (1 minute ago) Compare & pull request

Branch: master New pull request Create new file Upload files Find File Clone or download

hyunchan-park made other changes	Latest commit 9dae91f 34 minutes ago
.gitattributes	Initial commit 4 days ago
LICENSE	Initial commit 4 days ago
README.md	Initial commit 4 days ago
test.txt	made other changes 34 minutes ago



# Push 하고, GitHub 에서 Commit 내역 확인

The image shows two screenshots of the GitHub commit history interface, illustrating the process of pushing and verifying commits.

**Top Screenshot (Branch: master):**

- Branch: master
- Commits on Sep 29, 2019
  - made other changes ...  
hyunchan-park committed 35 minutes ago
- Commits on Sep 25, 2019
  - Create test.txt  
hyunchan-park committed 4 days ago
  - Initial commit  
hyunchan-park committed 4 days ago

**Bottom Screenshot (Branch: testing):**

- Branch: testing
- Commits on Sep 29, 2019
  - made a change ...  
hyunchan-park committed 37 minutes ago
- Commits on Sep 25, 2019
  - Create test.txt  
hyunchan-park committed 4 days ago
  - Initial commit  
hyunchan-park committed 4 days ago

# 실습 과제 (git 개인실습 #3)

---

- GitHub Desktop, GitHub repository 생성
  - Slide #17, #18 과 유사하게, 양쪽의 repository 생성 확인
  - repo.jpg
- 추가 Branch 생성 및 양쪽의 내용 변경, GitBash 확인
  - Slide #30 -> branches1.jpg
- GitHub Desktop 에서 양쪽의 변경 내용 확인
  - Slide #31 -> branches2.jpg
- GitHub 에서 확인
  - Slide #33 -> Github.jpg
- 제출 기한:
  - 10/27 (일) 23:59
  - 지각 감점: 5%p / day (3주 내 제출해야 함)

