

PORTFOLIO

졸업작품(졸업설계)
20170686이원태





포트폴리오를 들어가기 전에

이 포트폴리오를 써 내려가기 전에 내 마음가짐과 어떠한 요소를 중점적으로 다루고 어떤 부분을 중요하게 생각했는지 말하고자 한다. 우선 포트폴리오란 자신의 실력을 한 눈에 볼 수 있도록 관련 내용 등을 모아 놓는 자료다.

그래서 나는 깃과 깃허브를 주제로 정하고 그에 대한 나의 지식 등을 압축해 총 30페이지 정도에 달하는 PPT를 만들었다. 졸업작품과 수업을 준비하는 과정에서 깃과 깃허브 그리고 안드로이드 스튜디오에 관련된 나의 성장을 보여주는 것에 중점을 두었다.

취업 포트폴리오가 아닌 교과목 포트폴리오기 때문에 자유로운 형식에 기반을 두었고 자기주도적으로 학습을 한다는 마음가짐으로 자료를 수집해 작성하였다. 이렇게 포트폴리오를 작성하면서 깃허브에 대해 공부를 더 할 수 있는 좋은 기회가 되었다고 생각한다.

Part 0-2 강의계획서



강 의 계 획 서

아시아 직업교육 허브대학

2021 학년도 2학기	전공	컴퓨터정보공학과	학부	컴퓨터공학부
과 목 명	물입작품(종합설계)(2018086-PA)			
강의실 과 강의시간	화·5(3-218), 6(3-218), 7(3-218), 8(3-218)			학점 4
교과분류	실습			사수 4

담당 교수	장환수 + 연구실 : 2호관-706 + 전 화 : 02-2610-1941 + E-MAIL : hskang@dongyang.ac.kr + 민담가능기간 : 화 11:00-11:50
-------	--

학과 교육목표	
------------	--

교육 개요	○ 수업운영방안 : 비대면(코로나19 방역상황 호전시 대면으로 전환) ○ 온라인 수업 운영방법 : 녹화형/실시간 화상 수업으로 병행하여 진행 ○ 과목 개요 : 컴퓨터정보공학과에서 학습한 내용들을 종합적으로 활용하여 실제로 프로젝트를 구현해 보는 과목으로서 팀별로 하나의 프로젝트를 완성하는 것을 목표로한다. 이러한 프로젝트 개발 활동은 졸업 후 사회에 진출했을 때 실제적으로 평가될 일의 실전준비가 될 것이다. 또한 하나의 프로젝트를 처음부터 시작하여 끝까지 스스로 완성하여 봄으로써 학습내용들을 체계화하고 실용적인 실력을 키우는 데 그 목적이 있다. 팀별로 완성된 작품은 졸업작품 최종 발표와 학내대회 등을 통해 공개되어 평가된다.
-------	--

학습목표 및 성취수준	대학 교육목표 및 학과 교육목표를 달성하기 위하여 본 과목은 컴퓨터정보공학과 교육과정에서 배운 다양한 기술을 이용하여 팀별로 하나의 프로젝트를 수행해 프로젝트를 작품을 개발하는 것을 목표로 한다. 본 과목을 통해 다음을 성취할 수 있다. - 팀별로 자신의 역할을 정하고 팀 프로젝트 작품을 기획하고 설계할 수 있다. - 작성된 설계서를 기반으로 지금까지 배운 과목별 학습내용을 이용해서 팀 프로젝트 작품을 개발할 수 있다.
----------------	---

	도시별	제자	출판사	비고
주교재	유안물	-	-	

수업시 사용도구	각 팀별로 프로젝트 개발 시 필요한 재료 및 도구
-------------	-----------------------------

평가방법	출석 20% + 중간고사(수행과정 평가) 30%(프로젝트보고서 1차발표/5, 2차발표/10, 중간개원/15) + 기말고사(과정 및 최종발표 평가) 30%(프로젝트 최종발표/15, 최종보고서/5, 개인 일대기/10) + 교내외 학습대회 및 작품전 참가 및 수상 10% + 취업활동 및 기타 10%
------	---

수강인내	강의완료 후 학생들은 팀 프로젝트의 일환으로서 하나의 작품을 개발하기 위해 작품의 기획부터 개발, 그리고 발표까지 프로그램의 개발 프로세스 모든 과정을 경험할 수 있으며 자신과 역할을 충분히 수행할 수 있다. 전수과목으로 1학기의 시스템분석설계 과목을 학습한 학생들이 수강해야 한다
------	---



강 의 계 획 서

아시아 직업교육 허브대학

1 주차	[개강일(9/1), 수강평일(9/6 ~ 9/7)]
학습주제	<강의 소개> 첫 시간의 수강 학생들 확인과 이 과목에서 학습하게 되는 졸업작품에 대한 개요 및 강의 방법, 평가 방법 등에 대해 설명한다. 특히, 시스템분석설계 과목과 연계된 졸업작품의 진행 방법과 완성된 졸업작품을 평가하기 위한 졸업작품평가회와 전산학부학내대회의 출품에 대해서 소개한다. 더불어 강의 목표와 강의 시 유의사항에 대해서도 설명한다.
목표 및 내용	1. 학습대상자 출석 및 전수과목 이수 여부 확인 2. 졸업작품 강의 내용 및 목표 소개 3. 학습 방법 및 학습 평가 소개 4. 팀구성 및 확인
미리읽어오기	1. 팀별 작품 진행 상황 파악 2. 방학 중 팀별 활동 보고 준비
과제, 시험, 기타	
2 주차	[2주]
학습주제	<팀별 작품진행도 점검 및 지도> 각 팀별로 현재 진행 중인 팀 프로젝트의 작품에 대한 작품 진행 상황 점검과 방학 중 팀별 활동 내역 파악
목표 및 내용	1. 각 팀별 프로젝트 구성 원리를 확인한다. 2. 방학 중 팀별 활동 내역을 발표하여 각 팀별 프로젝트의 진행상황을 파악한다. 3. 각 팀별 프로젝트의 진행 상황에 따라 앞으로의 프로젝트 개발 계획을 수립한다. 4. 팀별 프로젝트의 진행 상황에 따라 자신이 맡은 역할의 세부 계획을 수립한다.
미리읽어오기	
과제, 시험, 기타	졸업작품 프로젝트보고서 1차발표
3 주차	[3주, 추석연휴(9/20-9/21) -> 보강(9/20, 12/14)]
학습주제	<팀별 작품 변경 사항 확인> 팀별 프로젝트의 작품의 기획과 달리 변경 또는 수정된 기능들을 확인하고 변경된 사용자 인터페이스 등을 점검한다. 이를 통해 최종적으로 개발해야 하는 팀 프로젝트의 사용자 인터페이스 및 기능들을 확정한다.
목표 및 내용	1. 팀 프로젝트의 시작 시 기획 및 설계했던 내용과 변경된 기능 및 구현 내역을 확인한다. 2. 변경 및 수정된 부분에 대한 사용자 인터페이스와 기능에 대해 재설계를 진행한다. 3. 이에 따라 팀별 또는 개인별 개발 계획을 변경하고 수정한다.
미리읽어오기	
과제, 시험, 기타	변경된 최종 설계서
4 주차	[4주, 추석연휴(9/22) -> 보강(12/9)]
학습주제	<팀별 작품 개발을 위한 관련 기술 학습> 수업과 팀별 작품을 위해 필요한 관련 작품 개발 자료 및 기술들을 수집하고 이를 학습한다.
목표 및 내용	1. 수정 및 추가된 팀 프로젝트의 기능들을 위해 관련 기술 자료들을 수집한다. 2. 수집된 작품 관련 소스들을 분석하고 이를 작품에 반영할 수 있도록 학습한다. 3. 작품을 작품개발서버에 설치하기 위한 관련 자료들을 학습하고 이해할 수 있다.
미리읽어오기	
과제, 시험, 기타	



강 의 계 획 서

아시아 직업교육 허브대학

5 주차	[5주, 개원평(10/9) -> 보강(10/4)]
학습주제	<팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트의 작품의 개발을 진행하여 이를 통해 팀 프로젝트의 각 역할을 학습한다.
목표 및 내용	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트의 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론했고 해결할 수 있다. 4. 다음주차까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.
미리읽어오기	
과제, 시험, 기타	졸업작품 프로젝트보고서 2차발표
6 주차	[6주, 한글날(10/9) -> 보강(12/10)]
학습주제	<팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트의 작품의 개발을 진행하여 이를 통해 팀 프로젝트의 각 역할을 학습한다.
목표 및 내용	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트의 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론했고 해결할 수 있다. 4. 다음주차까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.
미리읽어오기	
과제, 시험, 기타	
7 주차	[7주]
학습주제	<팀별 작품 서버 설치> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트의 작품의 개발을 진행하여 개발된 작품을 서버에 설치하여 동작여부를 확인한다.
목표 및 내용	1. 현재까지 개발한 작품을 서버에 설치하고 이를 동작시킬 수 있다. 2. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 3. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 4. 팀 프로젝트의 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론했고 해결할 수 있다. 5. 다음주차까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.
미리읽어오기	
과제, 시험, 기타	각 팀 프로젝트 구성원별 기능별 개발 진행 사항 전공동아리 콘텐츠작업전 참가
8 주차	[8주]
학습주제	<종강고사> 지금까지 개발한 팀 프로젝트의 작품의 종강 평가를 위해 개발 내용을 시연 및 발표한다.
목표 및 내용	1. 지금까지 개발한 팀 프로젝트의 작품의 종강 평가를 위해 개발 내용을 발표 자료로 만들고 이를 프리젠테이션한다. 2. 지금까지 개발한 팀 프로젝트의 작품의 종강 평가를 위해 작품의 동작 과정을 시연한다.
미리읽어오기	
과제, 시험, 기타	종강고사 또는 보고서 발표

Part 0-2 강의계획서



강 의 계 획 서

아시아 직업교육 허브대학

9 주차	[출발교사 기간 : 10/27(목)~11/2(목)]
학습주제	<팀별 작품 개발 진행> 최종 출품작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝트의 각 역할을 학습한다.
목표 및 내용	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주차까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.
미리읽어오기	
과제, 시험, 기타	각 팀 프로젝트 구성원별 기능별 개발 진행 사항
10 주차	[10주]
학습주제	<학부 학술대회 및 경진대회 준비> 학부에서 개최하는 학술대회 및 경진대회에 각 팀별 프로젝트 작품을 출품하기 위한 작업을 진행한다.
목표 및 내용	1. 각 팀별 현재까지 개발 진행 상황을 파악하고 학부 학술대회 및 경진대회 출품여부를 지도교수와 협의한다. 2. 학부 학술대회 및 경진대회 브로셔를 위한 작품 소개 자료를 작성한다. 3. 학부 학술대회 및 경진대회 발표자료 및 대모 동영상 등을 준비한다.
미리읽어오기	
과제, 시험, 기타	학부 학술대회/경진대회 브로셔 학부 학술대회/경진대회 발표자료 학부 학술대회 대모 동영상
11 주차	[11주]
학습주제	<작품 설치 및 최종 테스트> 개발된 작품을 서버에 설치하고 각 모듈별 동작 과정과 전체 동작 과정을 테스트한다.
목표 및 내용	1. 개발된 작품을 서버에 최종적으로 설치 완료한다. 2. 각 모듈별(웹, 앱, 서버 등) 동작 과정을 테스트한다. 3. 전체 작품의 동작 시나리오를 만들고 이에 따라 테스트를 완료한다.
미리읽어오기	
과제, 시험, 기타	작품 동작 시나리오
12 주차	[12주]
학습주제	<최종출품작품 발표> 학과 교수님들을 평가위원으로 한 최종출품작품발표회를 개최하여 각 팀별로 개발된 작품을 발표하고 시연한다. 이를 통해 출품작품 과목을 평가한다.
목표 및 내용	1. 각 팀별로 팀 프로젝트 작품 발표자료를 만들어 출품작품평가전에서 발표하고 개발된 작품의 동작 여부를 시연한다. 2. 평가위원들로부터 지적받은 내용들을 체크리스트로 만들어 수정 보완 사항을 파악한다.
미리읽어오기	
과제, 시험, 기타	



강 의 계 획 서

아시아 직업교육 허브대학

13 주차	[13주]
학습주제	<개발된 작품 보완 작업> 출품작품발표회에서 평가위원들로부터 지적받은 수정사항 및 추가사항들의 개발을 보완한다. 이에 대한 체크리스트를 만들어 보완여부를 확인한다.
목표 및 내용	1. 평가위원들로부터 지적받거나 부족한 기능들에 대한 체크리스트를 만든 후 만들어진 체크리스트에 따라 보완사항들을 개발한다. 2. 보완된 내용들을 테스트하여 작품 개발을 완료한다.
미리읽어오기	
과제, 시험, 기타	
14 주차	[14주]
학습주제	<개발된 작품 보완 작업 확인> 지도교수를 통해 출품작품발표회에서 지적된 사항들의 수정 및 보완이 완료되었는지 확인한다.
목표 및 내용	1. 작품평가회에서 지적사항 보완 여부를 확인한다. 2. 아직 부족한 부분에 대해서는 재확인이 필요한지 여부를 판단하고 이에 따라 최종 작품 개발을 완료한다.
미리읽어오기	
과제, 시험, 기타	
15 주차	[기말고사 기간 : 12/16(수)~21(목)]
학습주제	<최종 완성된 작품 제출> 출입 작품 과목을 통해 개발된 모든 내용을 제출하고 개발된 작품을 장시 확인할 수 있도록 서버에 용지관리 작업을 수행한다.
목표 및 내용	1. 최종적으로 완성된 작품을 디지털 형식(CD, USB 등) 으로 제작하여 제출한다. 2. 개발된 작품을 서버에 설치 완료하여 장시 개발된 작품이 동작할 수 있도록 유지한다. 3. 작품 개발을 위해 작성한 관련 자료(기획서, 설계서, 발표자료 등)를 정리하여 제출한다.
미리읽어오기	
과제, 시험, 기타	최종작품 관련 일체 자료
수업지침 안내	장애학생을 위한 별도의 수강 지원을 받을 수 있습니다. 언어가 문제가 되는 학생은 글로 된 과제 안내, 확대문자 시험지 제공 등의 지원을 드립니다.

CONTENTS



001 Git 버전 확인

- 안드로이드 스튜디오와 깃허브 계정 연동하기
- Test 버튼을 이용해 Git버전을 확인



002 GitHub 계정과 연동

- GitHub에서 Token 발급
- Token으로 GitHub 계정 추가



003 Git Repository 생성

- GitHub에 Repository 생성 후 연동



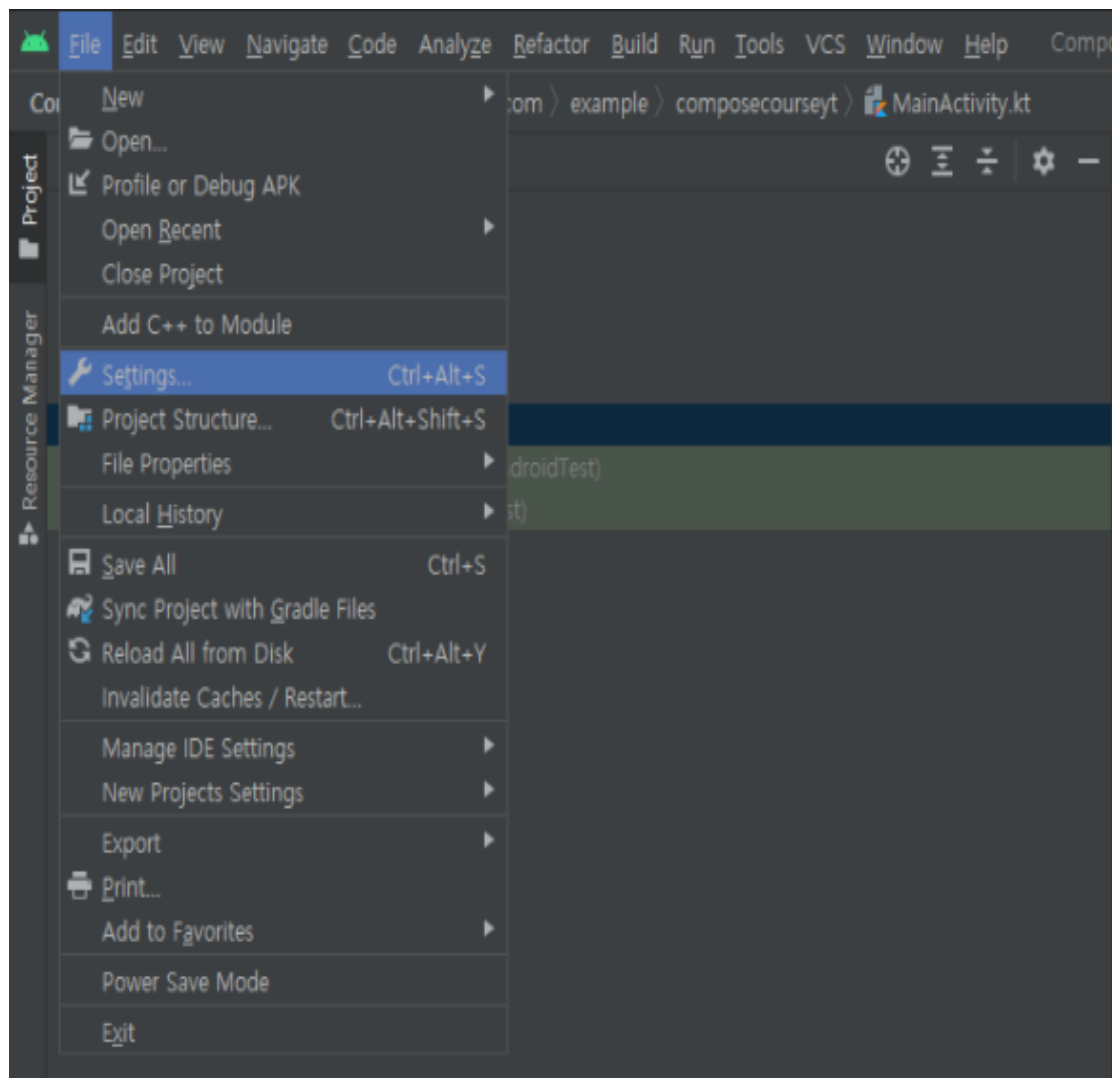
004 Commit, Push, Pull 사용

Part 1

Git 버전 확인



Git 버전 확인



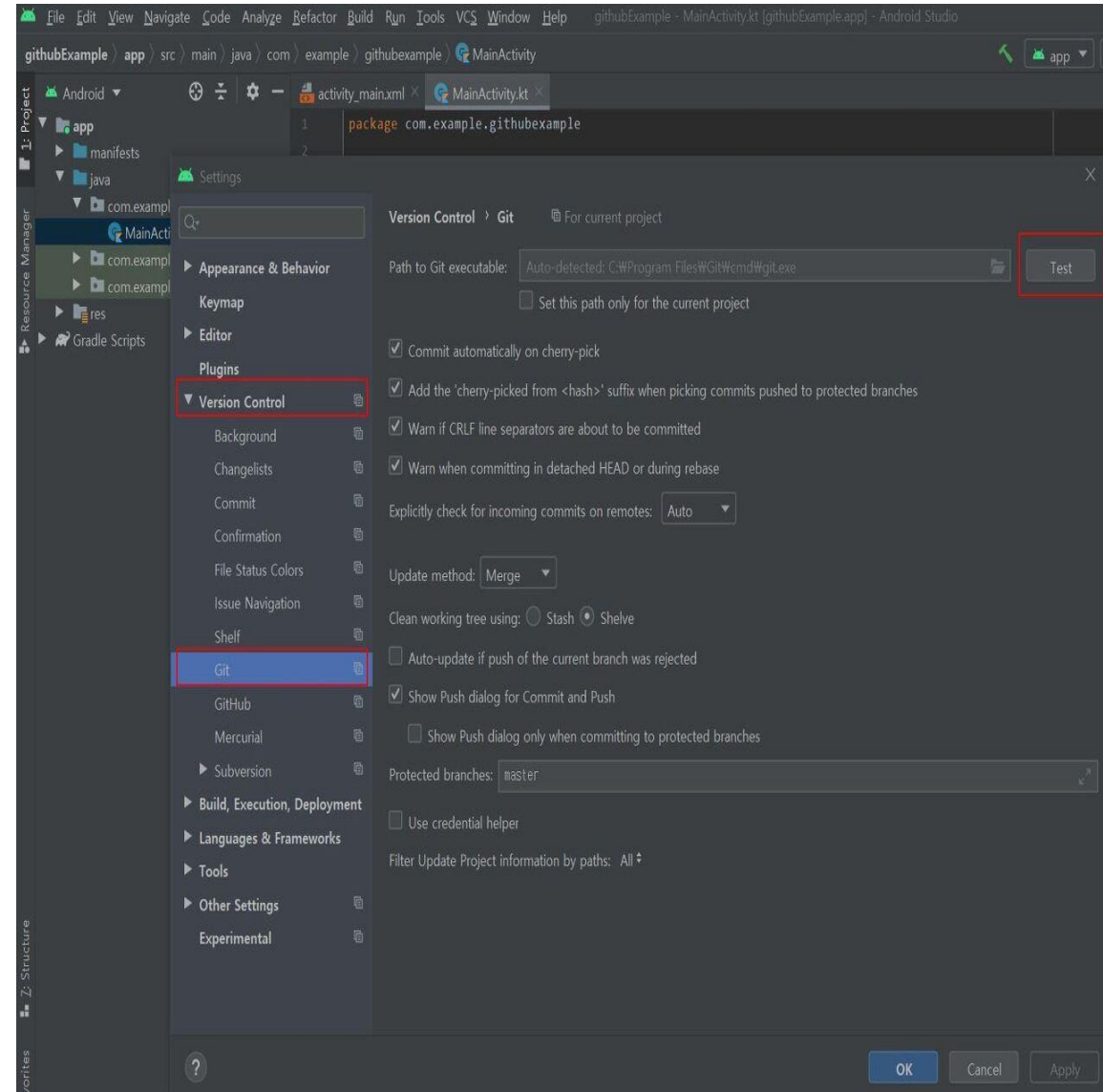
GIT 버전 확인

안드로이드 스튜디오 좌측 상단 File > Settings.... 클릭
Settings 창 뜨면 Version Control에서 Git 클릭

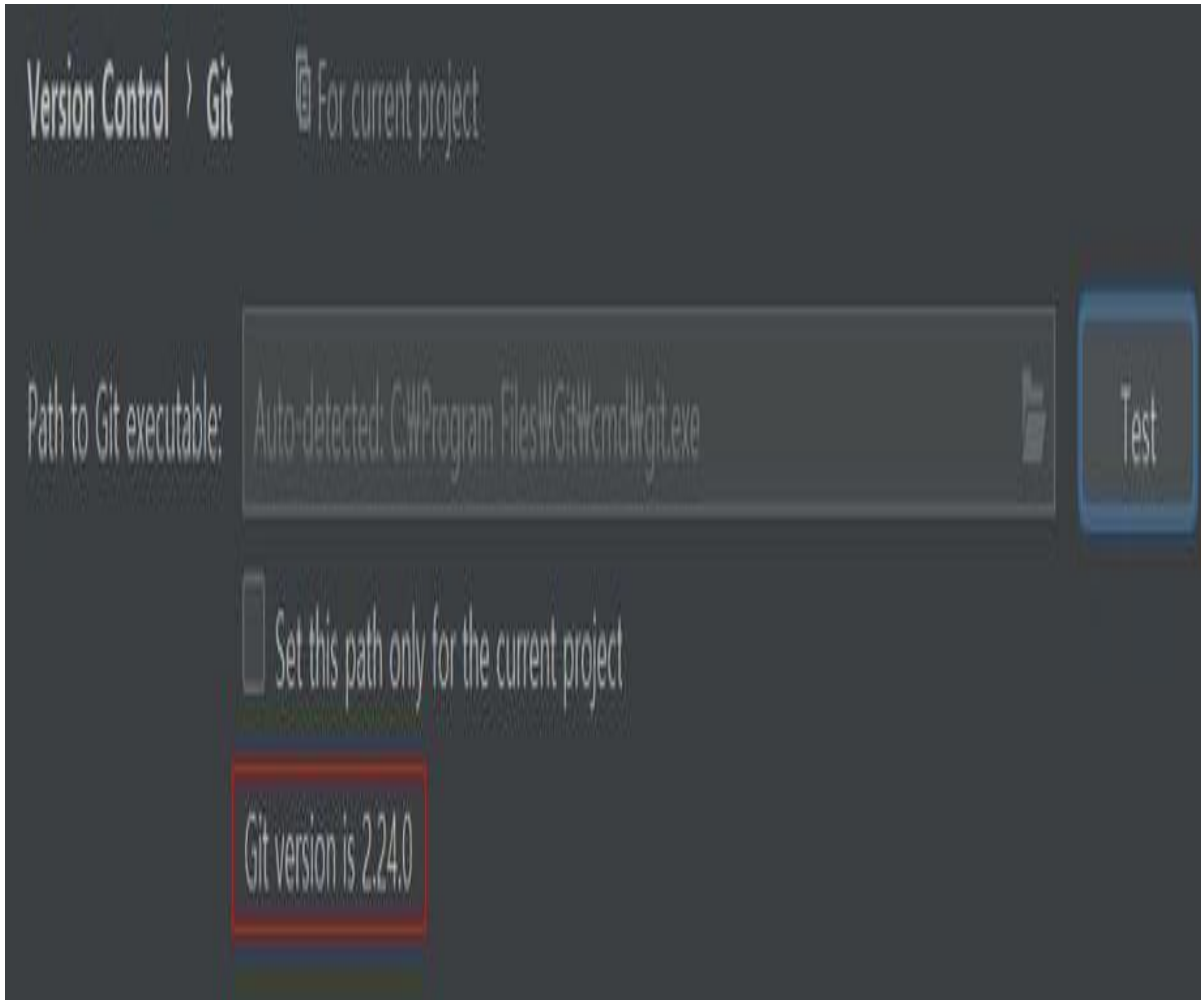
안드로이드 스튜디오와 깃허브 계정 연동하기

Settings 창 뜨면 Version Control에서 Git 클릭

왼쪽 사진에서 빨간 네모로 표시해 둔 Test를 클릭하면 현재의 깃 버전을 알 수 있고 실행 테스트를 할 수 있게 된다.



Git 버전 확인



Test 버튼을 이용해 Git 버전을 확인

왼쪽 사진에서 빨간 네모로 표시해 둔 Test를 클릭하면 현재의 깃 버전을 알 수 있고 실행 테스트를 할 수 있게 된다.

Part 2

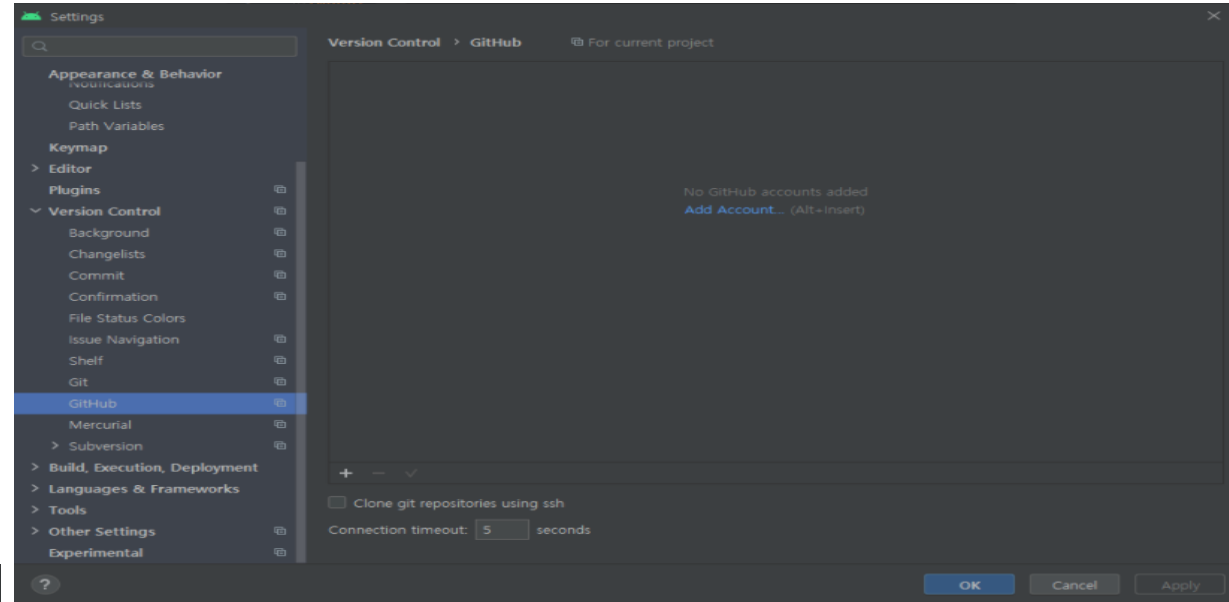
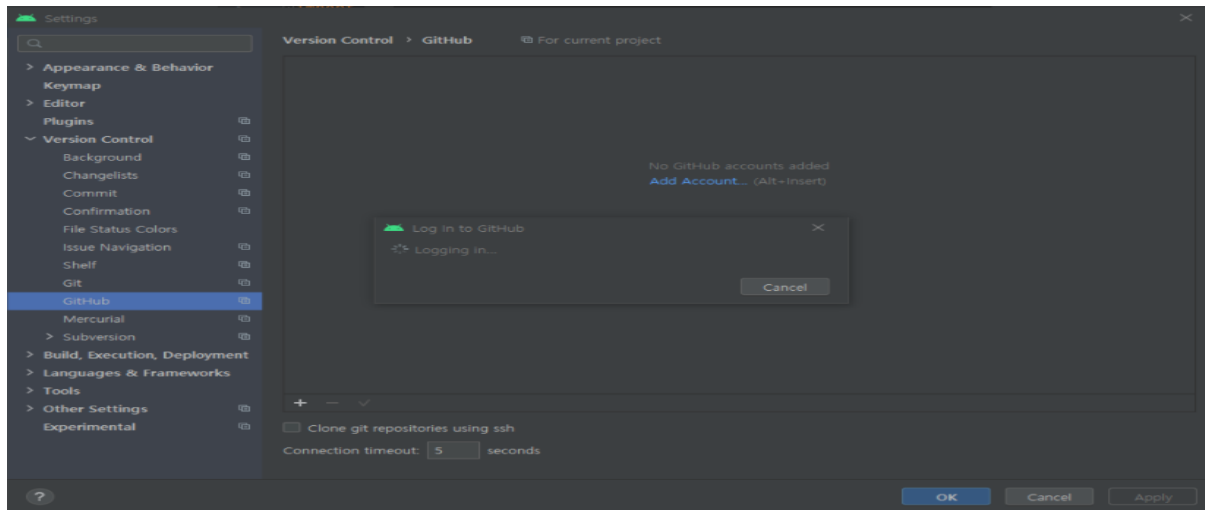
GitHub 계정과 연동



Part 2

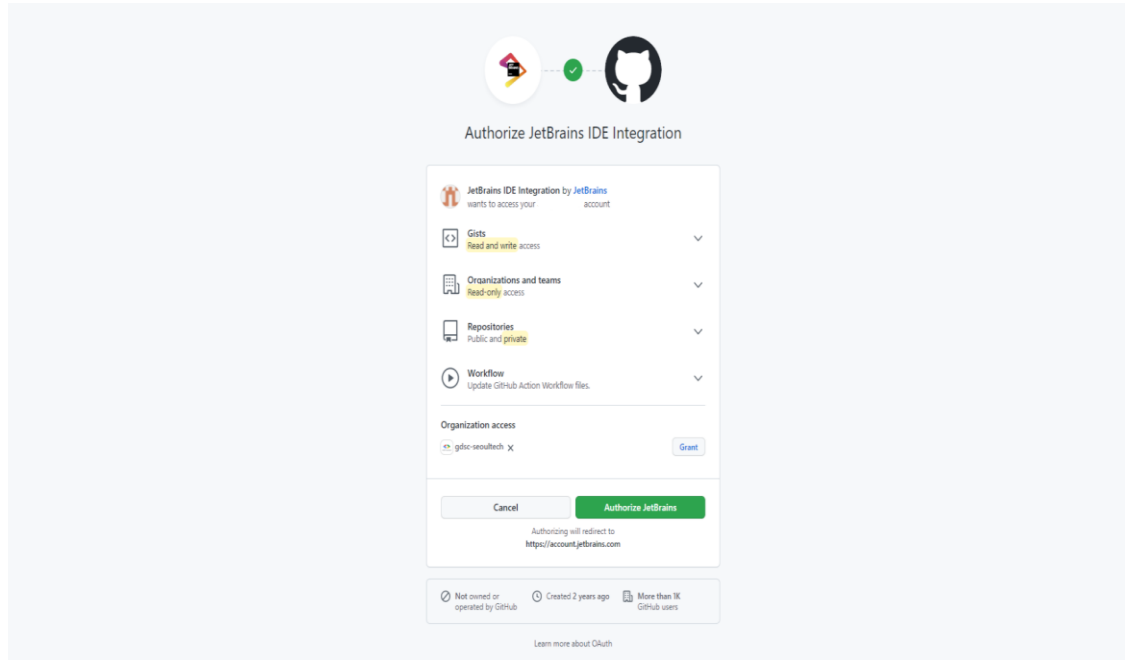
GitHub 계정과 연동

"Version Control"에서
"GitHub"를 클릭한 뒤 "Add
Account.."를 클릭한다.

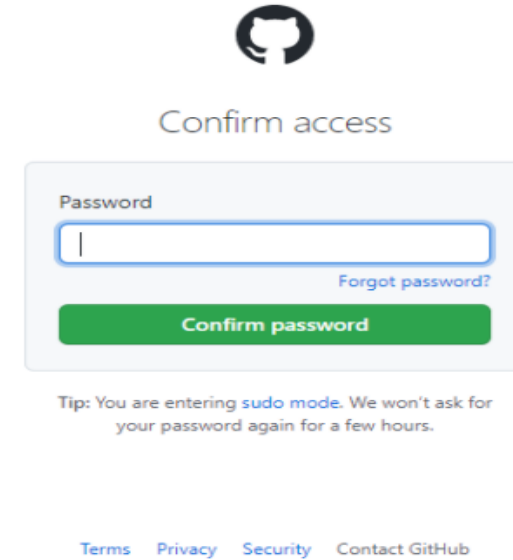


Part 2

GitHub 계정과 연동



로딩이 되면서 JetBrains 사이트 창이 뜨고, "Authorize in GitHub" 버튼을 클릭하여 GitHub 계정을 인증해야한다.

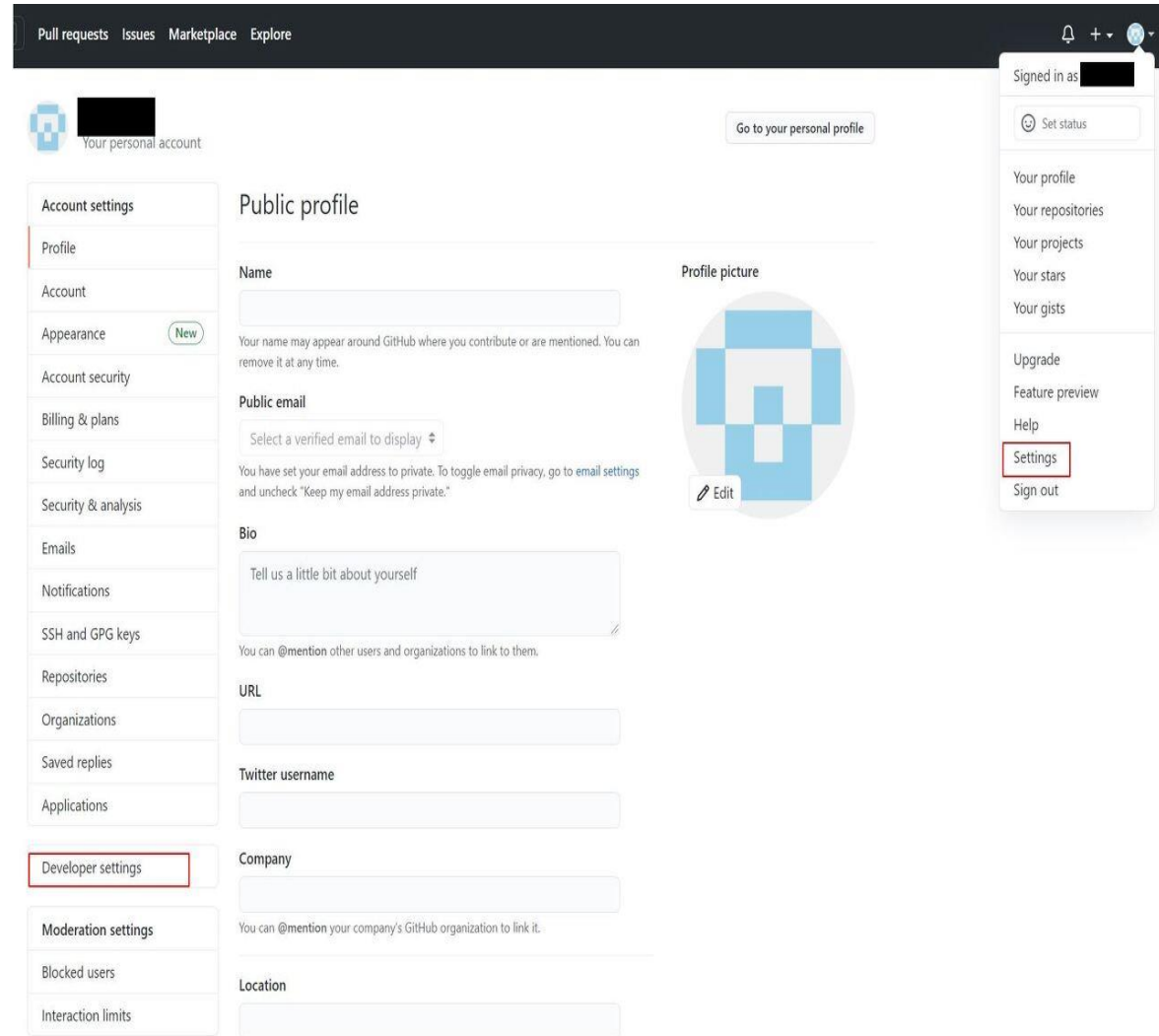


"Authorize JetBrains" 버튼을 클릭하여 GitHub 계정을 JetBrains와 통합한다.

안드로이드 스튜디오에서 Use Token으로 깃허브 계정 연동하기

먼저 깃허브에 들어가서 로그인한다.

그리고 아래와 같이 우측 상단 아이콘을 누르면 빨간 네모 박스로 표시해둔 Settings가 보이는데 클릭해준다.
Settings로 들어가면 아래 화면과 같이 Profile이 뜨고 왼쪽에 다양한 카테고리가 보입니다. 그중 Developer settings를 클릭한다.



Part 2

GitHub 계정과 연동

[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

[Redacted token]

Last used within the last week

Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

- Developer settings를 클릭하였다면 다시 왼쪽 카테고리에서 "Personal access token" 버튼을 클릭한 후 "Generate new token" 버튼을 클릭한다.

-이제 어디에 사용할 token인지 알 수 있도록 적당한 Note를 작성한다.



[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration *

30 days

The token will expire on Thu, Oct 14 2021

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repostatus	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo_migrate	Access repository migrations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write_packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read_packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete_packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin_org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write_org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read_org	Read org and team membership, read org projects
<input type="checkbox"/> admin_public_key	Full control of user public keys

Part 2

GitHub 계정과 연동

Note

android_link

What's this token for?

Expiration *

30 days

The token will expire on Thu, Oct 14 2021

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys

<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys (Developer Preview)
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys

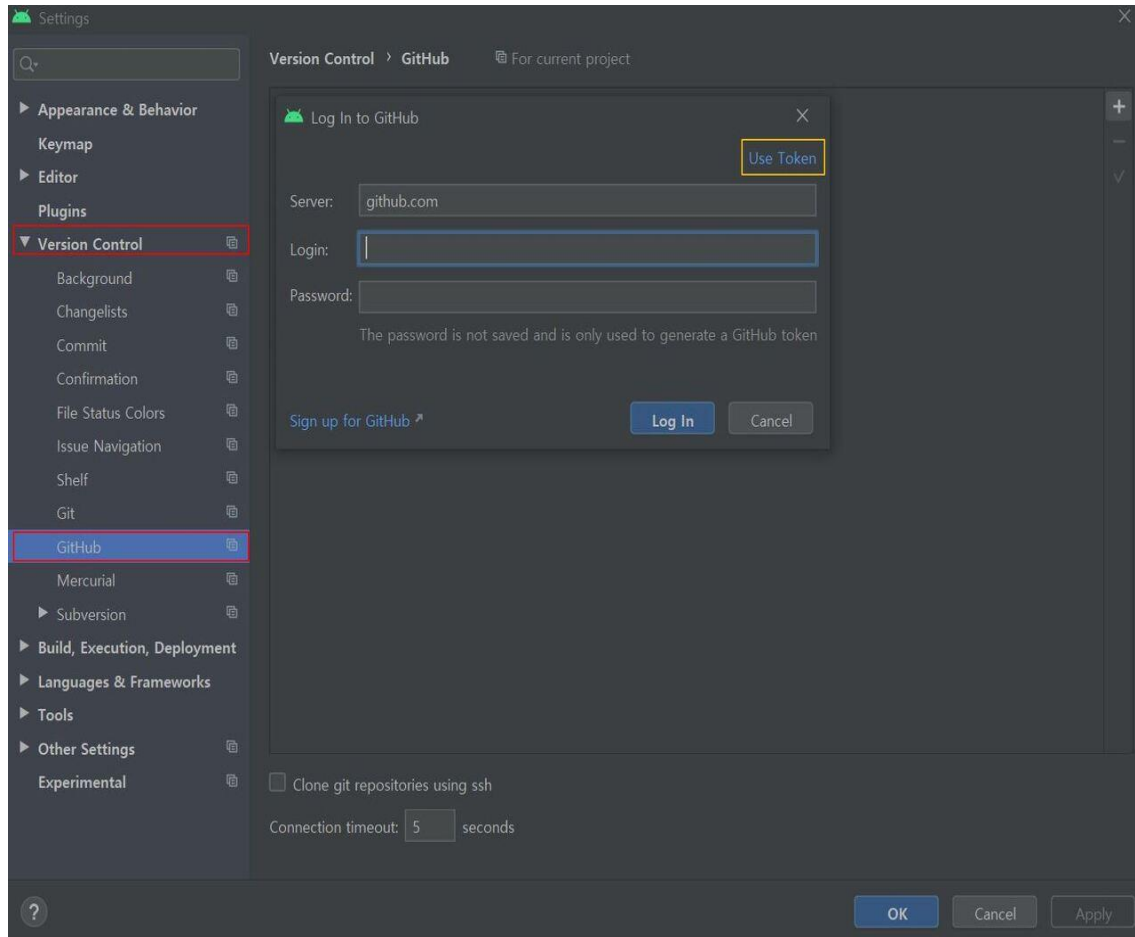
Generate token

Cancel

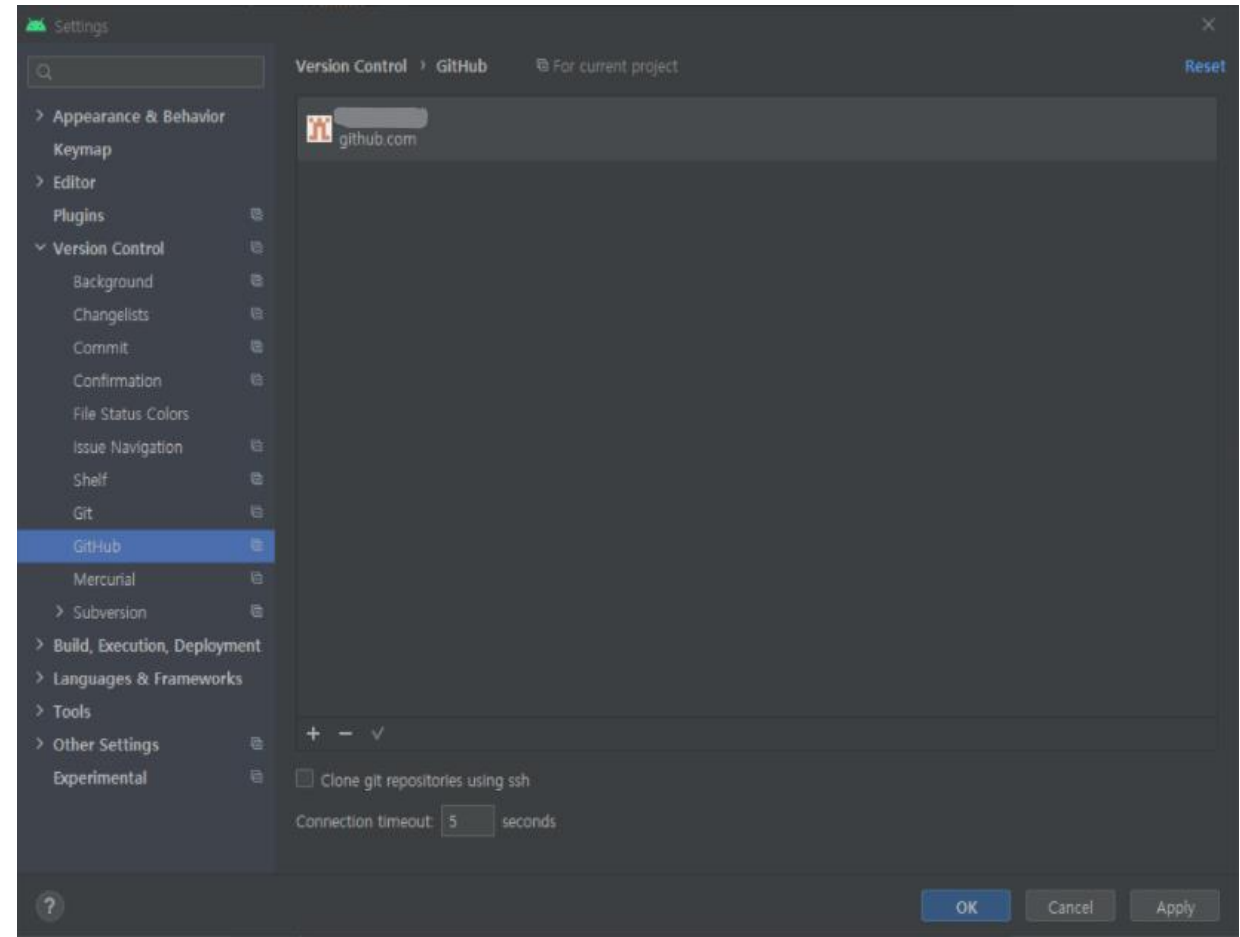
Note 입력칸에 원하는 대로 발급할 토큰에 대한 이름 또는 설명을 적는다.
이제 어디에 사용할 token인지 알 수 있도록 적당한 Note를 작성해주고 아래와 같이 체크합니다. (repo, admin:org, gist 정도만 체크해준다.)

Part 2

GitHub 계정과 연동



안드로이드 스튜디오로 돌아와서 복사한 토큰을 Token 입력칸에 붙여 넣어준 뒤에 "Add Account" 버튼을 클릭한다.



GitHub 계정 추가가 잘 된 것을 확인한 후 "OK" 버튼을 클릭한다.

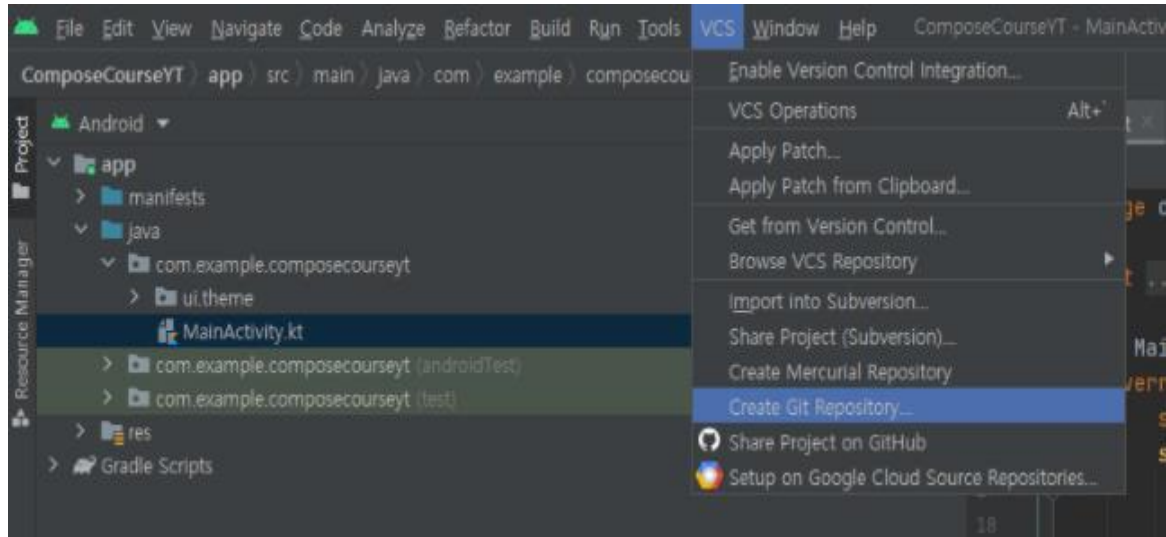
Part 3

Git Repository 생성



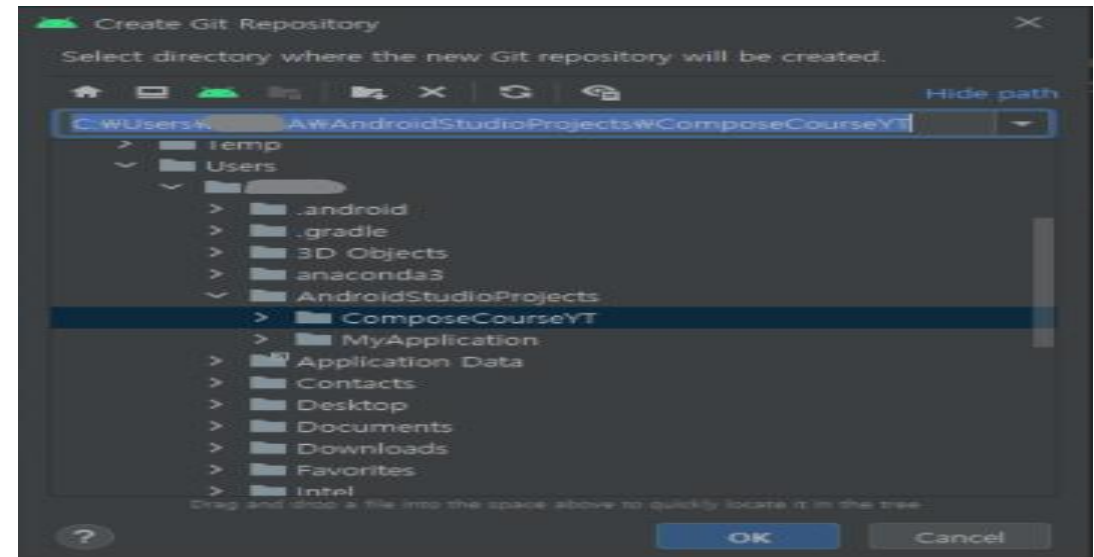
Part 3

Git Repository 생성



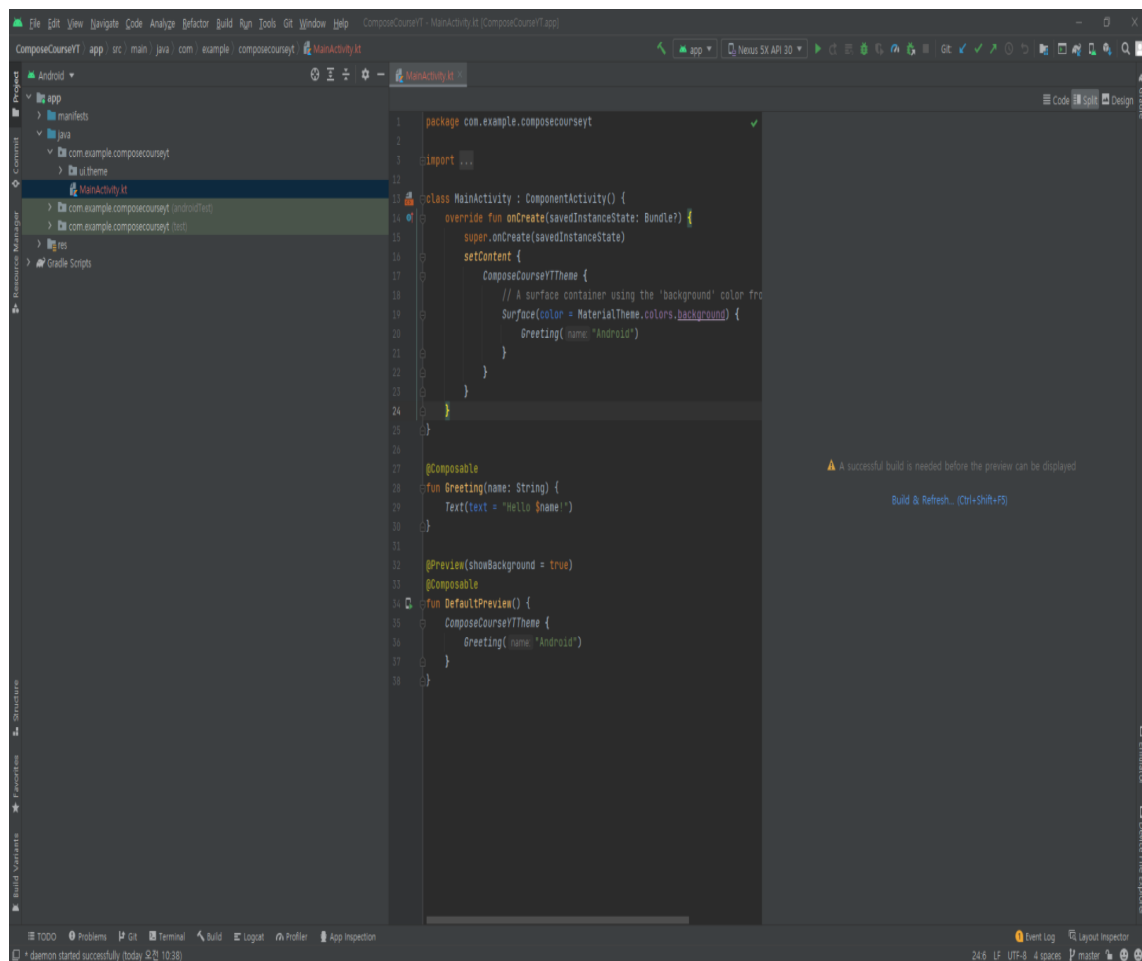
상단탭에서 "VCS"를 클릭한 뒤 "Create Git Repository..."를 클릭한다.

연결하고자 하는 현재 작업 중인 프로젝트를 클릭한 후 "OK" 버튼을 클릭한다.

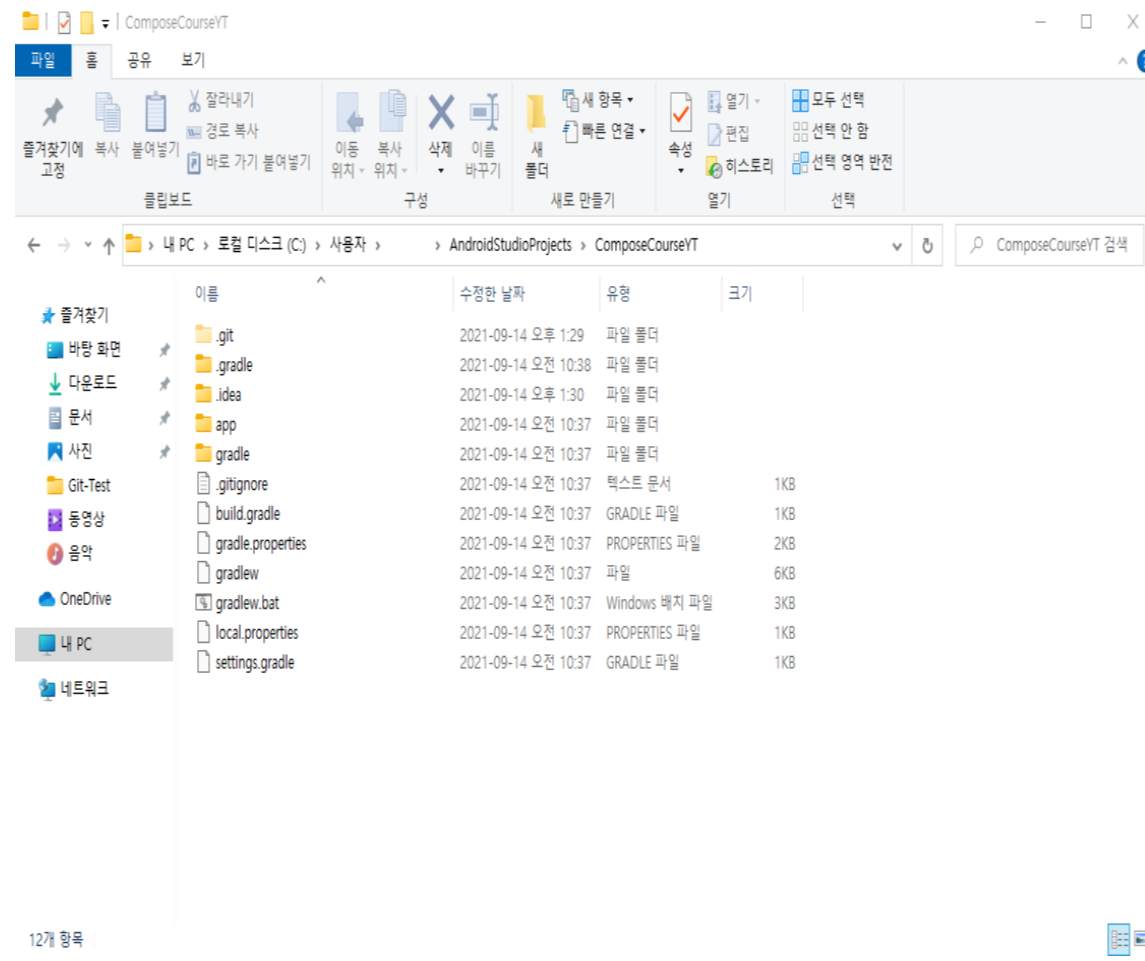


Part 3

Git Repository 생성



Git에 Repository로 생성이 되면 파일들의 이름을 빨간색으로 변한다.



프로젝트 폴더에 가보면 .git 폴더가 추가된 것을 확인할 수 있다.

Part 3

Git Repository 생성

The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

The form includes fields for 'Owner' (set to 'KangInyeong') and 'Repository name' (set to 'ComposeCourseYT' with a green checkmark). Below this, a note says 'Great repository names are short and memorable. Need inspiration? How about [congenial-train](#)?'. There's an optional 'Description' field with the text 'I study Kotlin with Jetpack Compose by watching Philipp Lackner's Jetpack Compose (https://youtube.com/play)'. The 'Visibility' section has 'Public' selected (with a note 'Anyone on the internet can see this repository. You choose who can commit.') and 'Private' as an option (with a note 'You choose who can see and commit to this repository.').

Under 'Initialize this repository with:', there are three checkboxes: 'Add a README file' (with a note 'This is where you can write a long description for your project. [Learn more.](#)'), 'Add .gitignore' (with a note 'Choose which files not to track from a list of templates. [Learn more.](#)'), and 'Choose a license' (with a note 'A license tells others what they can and can't do with your code. [Learn more.](#)'). A green 'Create repository' button is at the bottom.

The footer contains copyright information for 2021 GitHub, Inc. and links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

The screenshot shows the GitHub repository page for 'ComposeCourseYT'. The repository is public. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository has 1 Unwatch, 0 Stars, and 0 Forks.

The main content area has a light blue header with the text 'Quick setup — if you've done this kind of thing before'. It includes a 'Set up in Desktop' button, an 'or' separator, and buttons for 'HTTPS' and 'SSH'. The HTTPS URL is 'https://github.com/KangInyeong/ComposeCourseYT.git'. Below this, it says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.'

There are two sections for command-line setup:

- '...or create a new repository on the command line' with a code block:

```
echo "# ComposeCourseYT" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/KangInyeong/ComposeCourseYT.git
git push -u origin main
```
- '...or push an existing repository from the command line' with a code block:

```
git remote add origin https://github.com/KangInyeong/ComposeCourseYT.git
git branch -M main
git push -u origin main
```

There is also a section for '...or import code from another repository' with the text 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and an 'Import code' button.

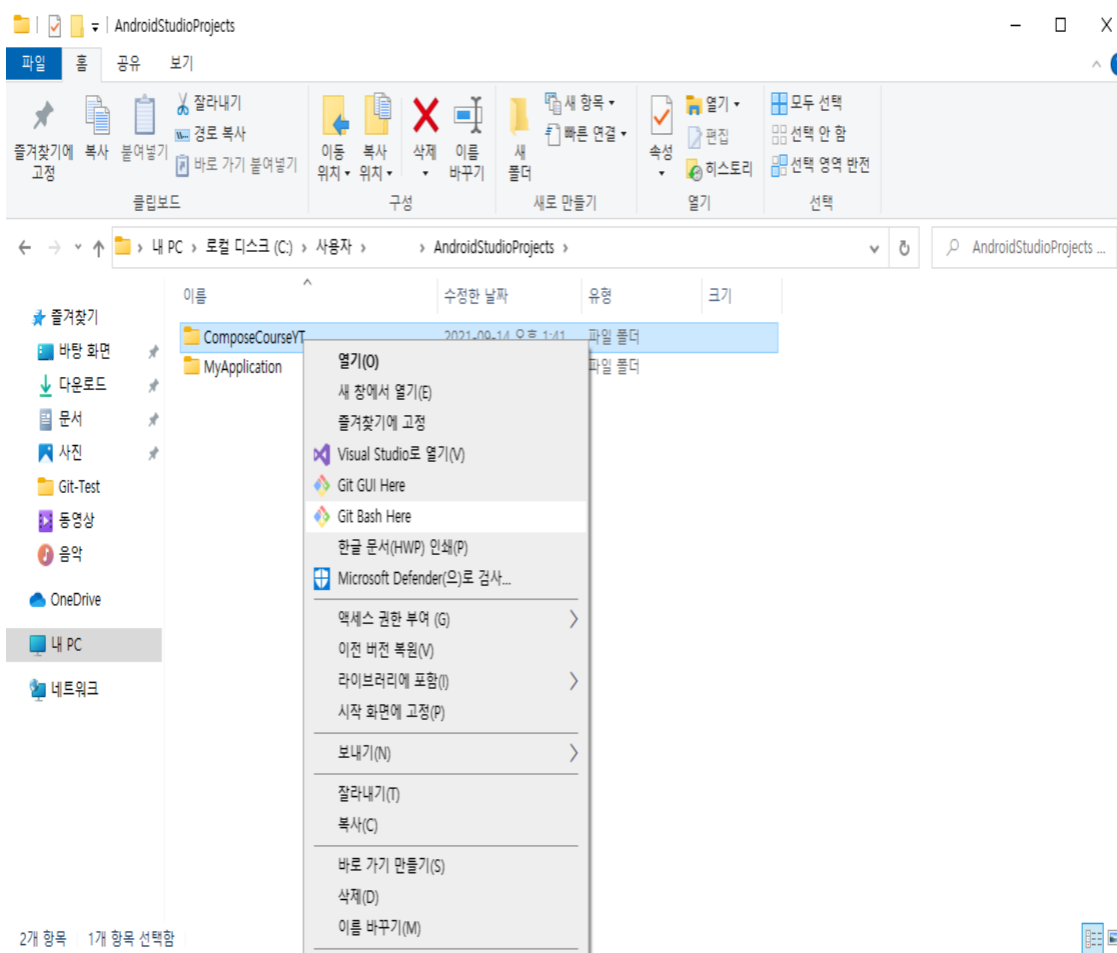
The footer includes a 'ProTip!' icon and the text 'Use the URL for this page when adding GitHub as a remote.'

GitHub에도 연결시킬 Repository를 생성한다.

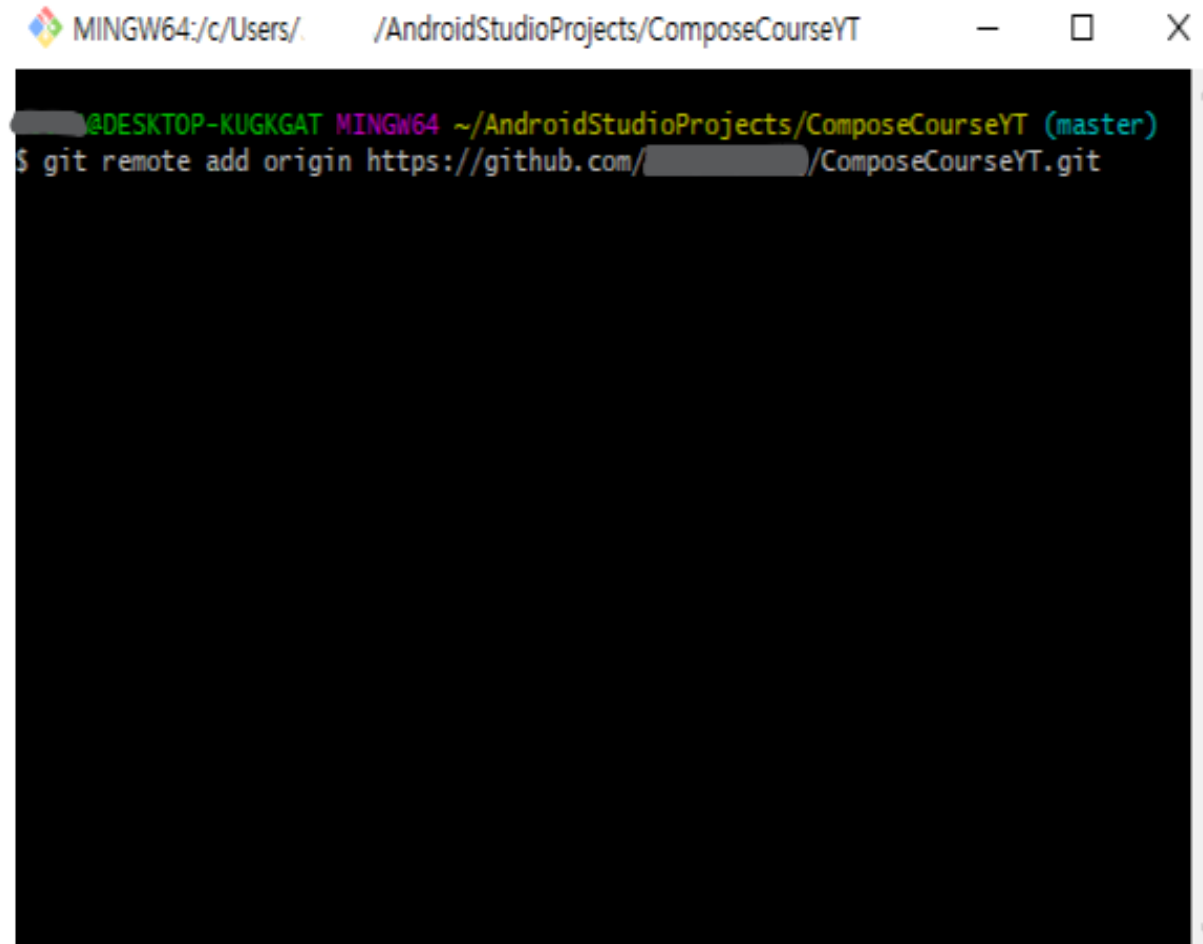
생성된 HTTPS 주소를 복사한다.

Part 3

Git Repository 생성

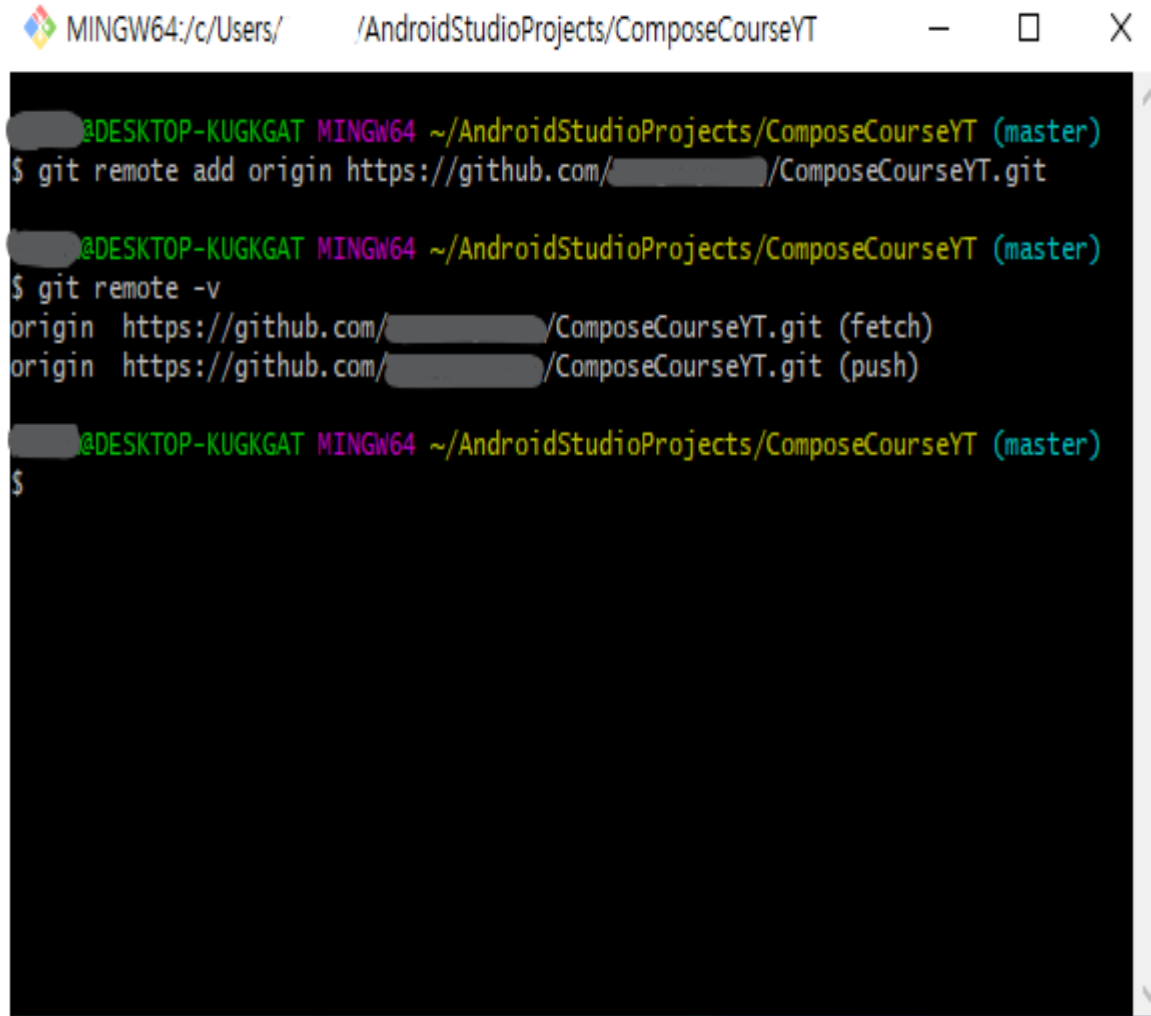


로컬에 저장된 안드로이드 프로젝트 폴더를 마우스 우클릭을 한 후 "Git Bash Here"를 클릭한다. 해당 프로젝트의 위치에서 Git Bash를 실행시키는 것이다.



remote를 위한 git 코드를 입력한다.
Git remote add origin [복사한 https 주소]

Git Repository 생성



```
MINGW64:/c/Users/ /AndroidStudioProjects/ComposeCourseYT - □ X

@DESKTOP-KUGKGAT MINGW64 ~/AndroidStudioProjects/ComposeCourseYT (master)
$ git remote add origin https://github.com/ /ComposeCourseYT.git

@DESKTOP-KUGKGAT MINGW64 ~/AndroidStudioProjects/ComposeCourseYT (master)
$ git remote -v
origin https://github.com/ /ComposeCourseYT.git (fetch)
origin https://github.com/ /ComposeCourseYT.git (push)

@DESKTOP-KUGKGAT MINGW64 ~/AndroidStudioProjects/ComposeCourseYT (master)
$
```

origin으로 연결된 repository를 확인할 수 있다.
git remote -v

Part 4

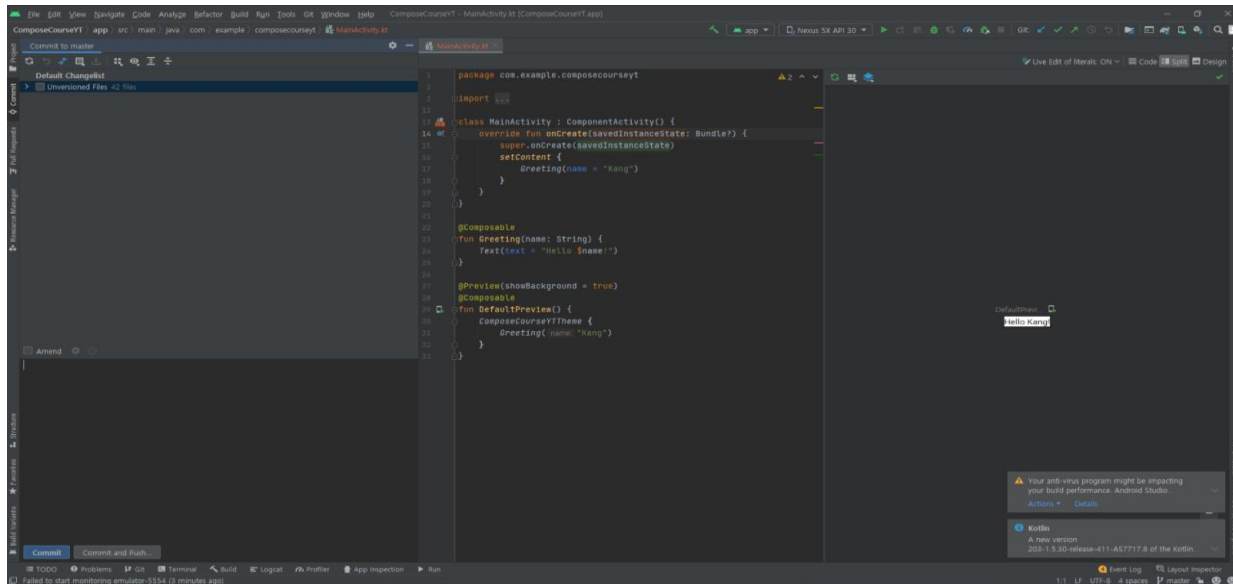
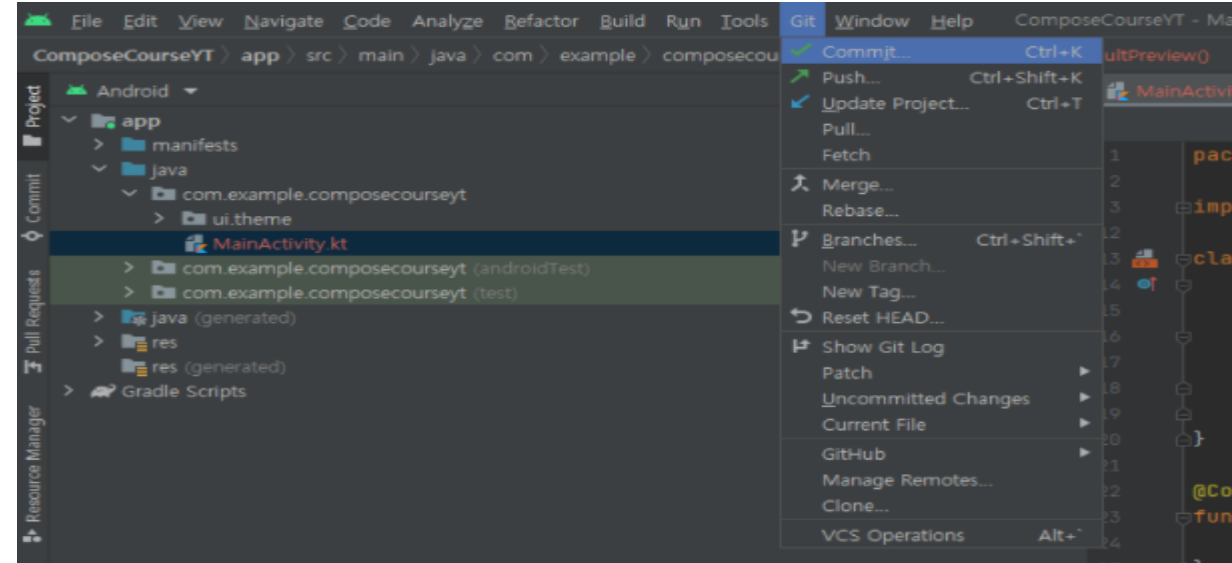
Commit, Push, Pull 사용



Commit, Push, Pull 사용

Commit 하기

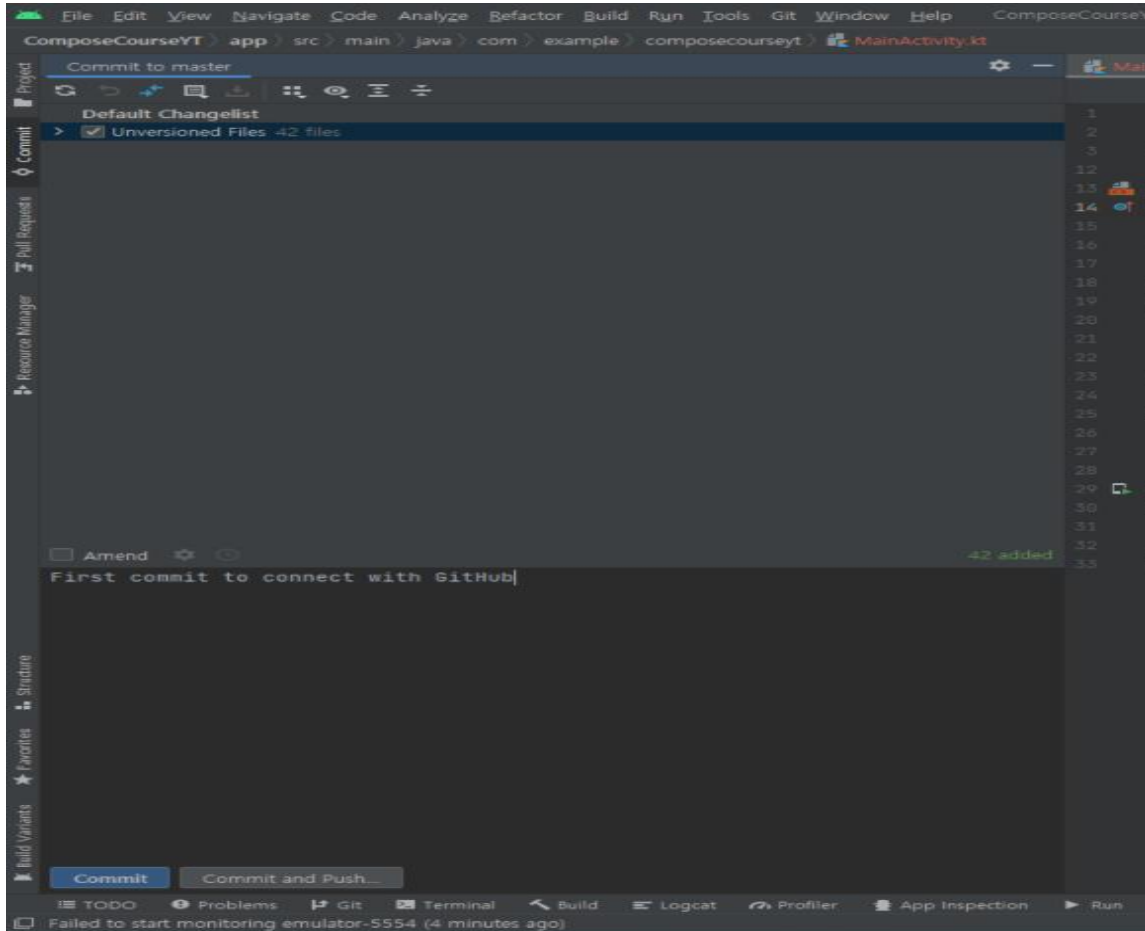
상단탭에 "Git"에서 "Commit..."을 클릭한다.



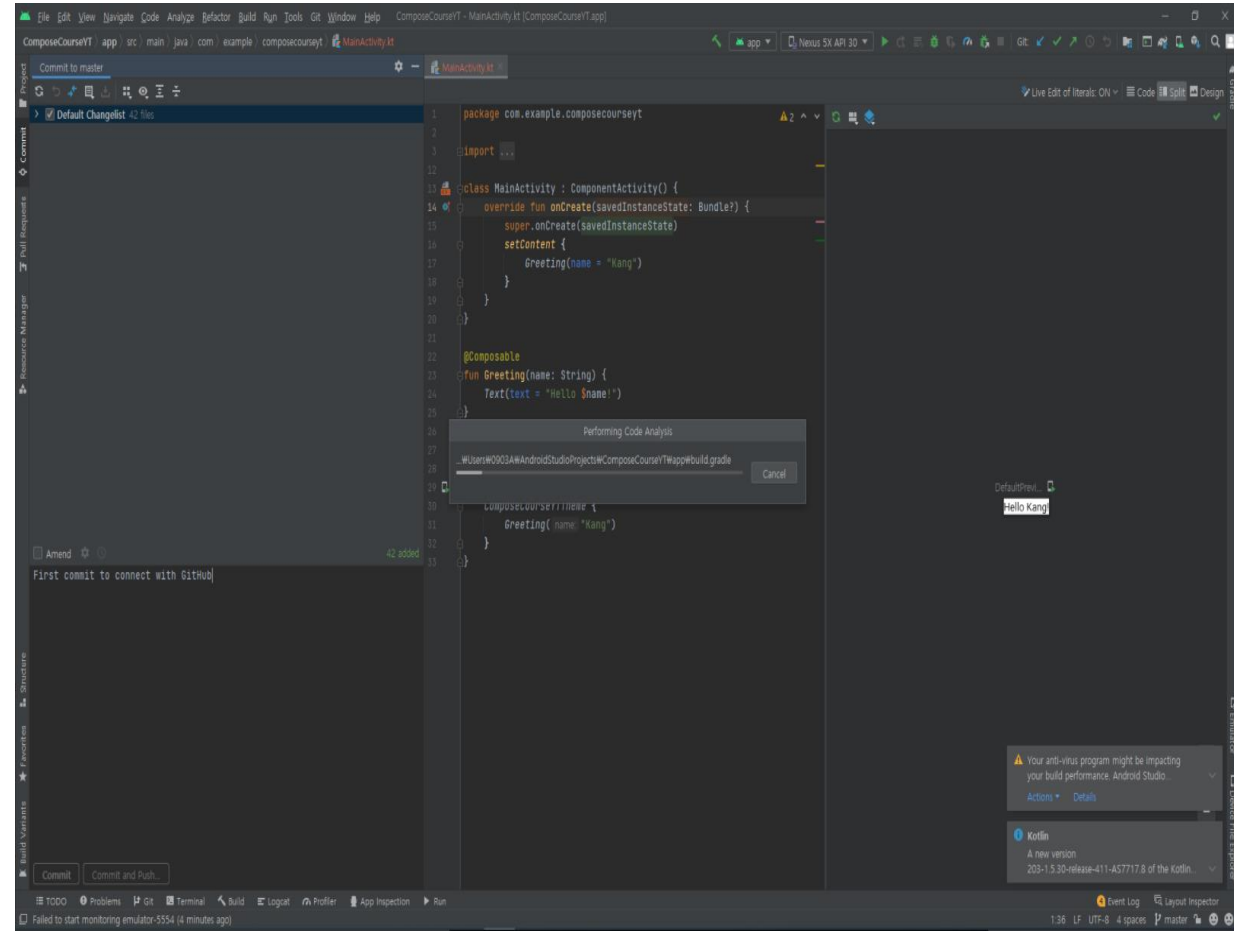
"Commit..."을 클릭한후에 왼쪽에 Commit을 위한 공간이 생성된다.

Part 4

Commit, Push, Pull 사용



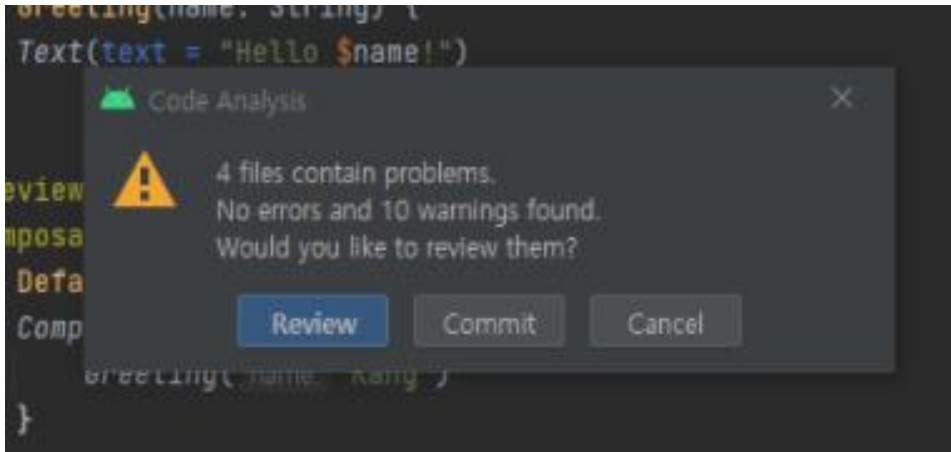
위쪽 공간에서는 Commit하고 싶은 파일을 선택할 수 있고, 아래 공간에서는 Commit 메시지를 작성할 수 있다. Commit 메시지는 원하대로 작성한다.



그 후 "Commit" 버튼을 클릭하면 잠간의 로딩이 생긴다.

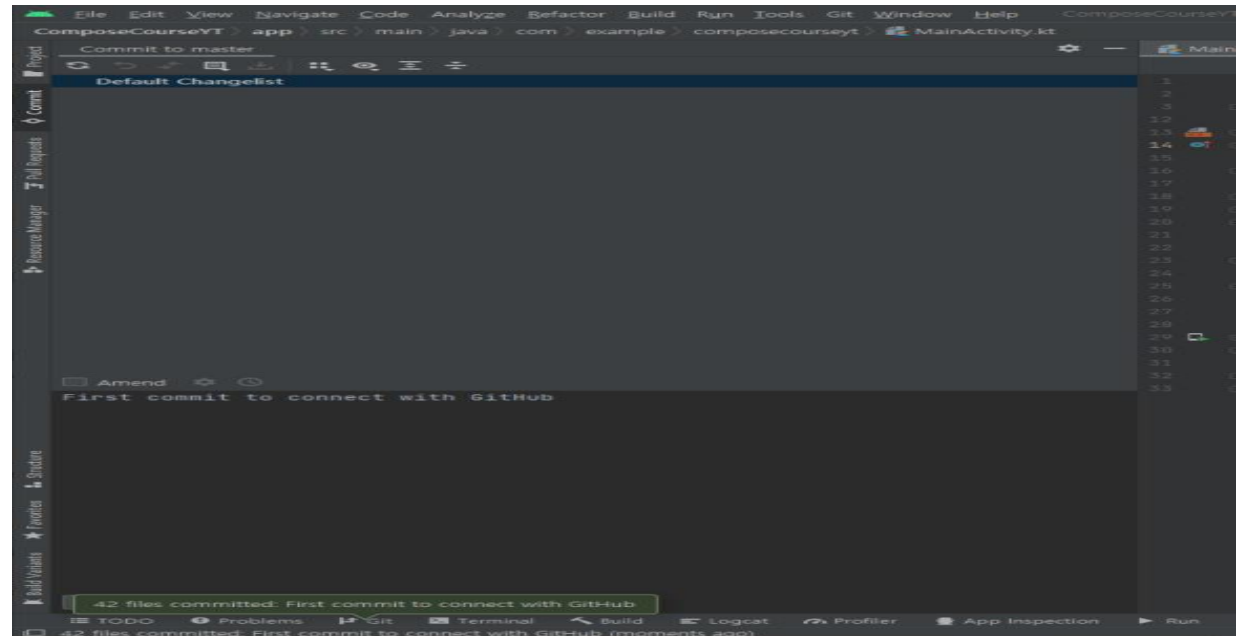
Part 4

Commit, Push, Pull 사용



이러한 경고창이 뜰 수 있으나 에러가 생길 것이 염려되어 파일을 다시 확인하고 싶다면 Review를 하고, 그냥 이대로 Commit을 진행하려면 "Commit" 버튼을 클릭한다.

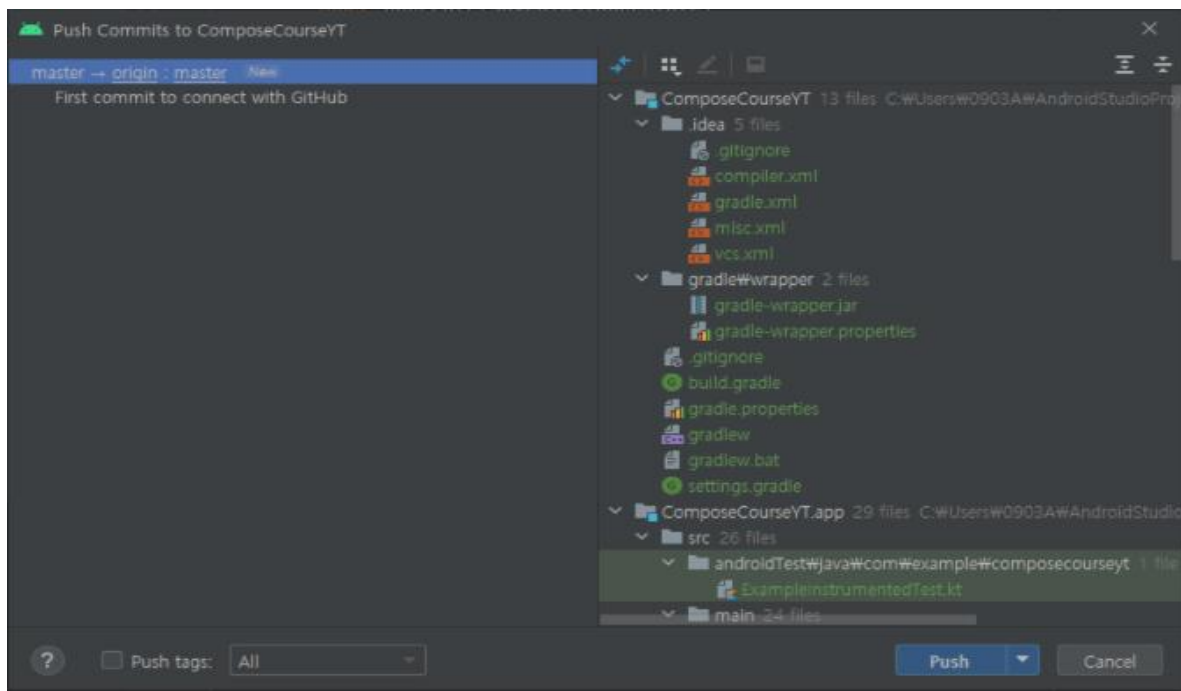
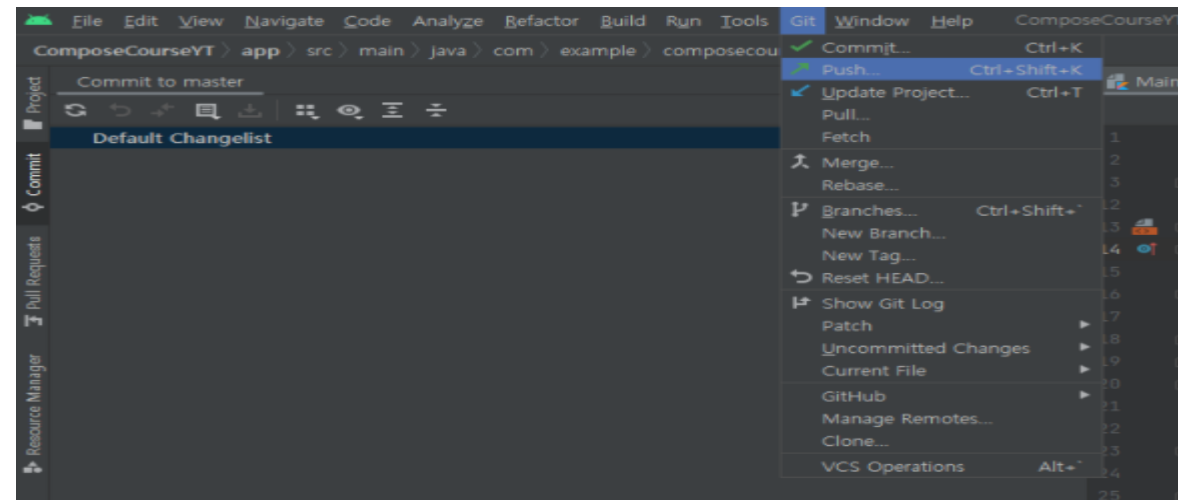
왼쪽 하단에 commit이 잘 되었다는 안내문이 뜬다.



Commit, Push, Pull 사용

Push 하기

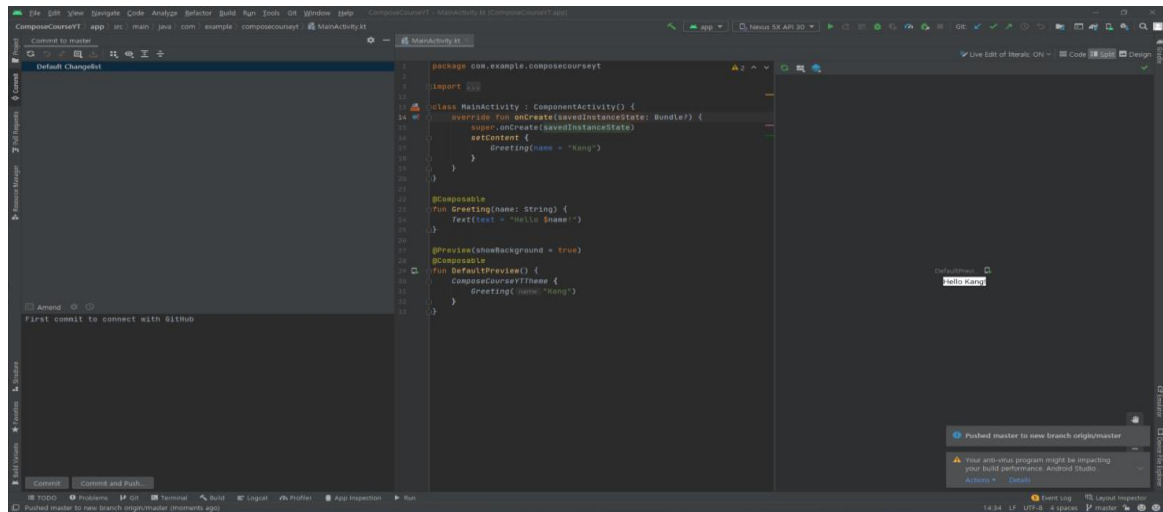
상단탭에 "Git"에서 "Push..."를 클릭한다.



push 하기 전에 commit 해 둔 파일들을 확인할 수 있다.
문제가 없다면 "Push" 버튼을 클릭한다.
(나의 로컬 저장소 master에서 remote하여 연결한
GitHub의 저장소인 origin으로 파일을 업로드한다고 생각
하면 된다.)

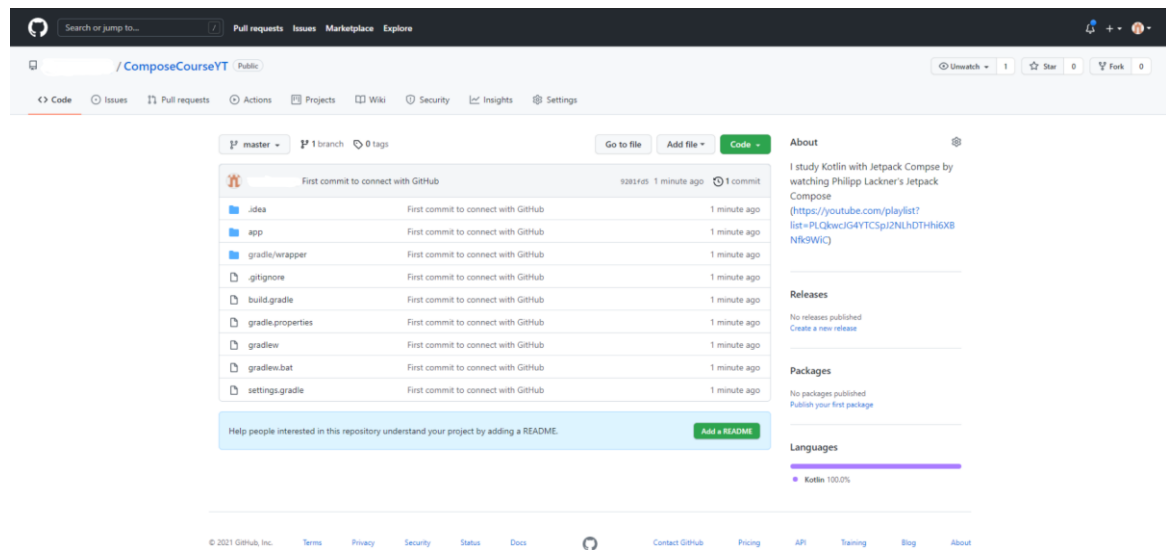
Part 4

Commit, Push, Pull 사용

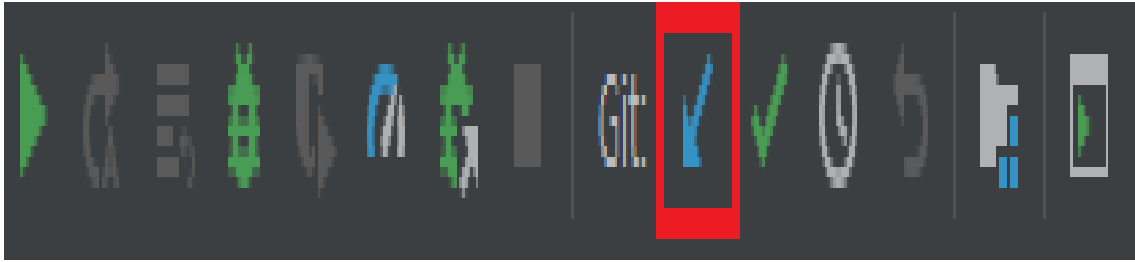


push가 잘 완료되면 오른쪽 하단에 Pushed master to new branch origin/master라는 문구가 뜬다.

GitHub에 가서 전에 생성했던 Repository에 로컬에서 생성하고 수정했던 프로젝트가 잘 업로드된 것을 확인할 수 있다.



Commit, Push, Pull 사용

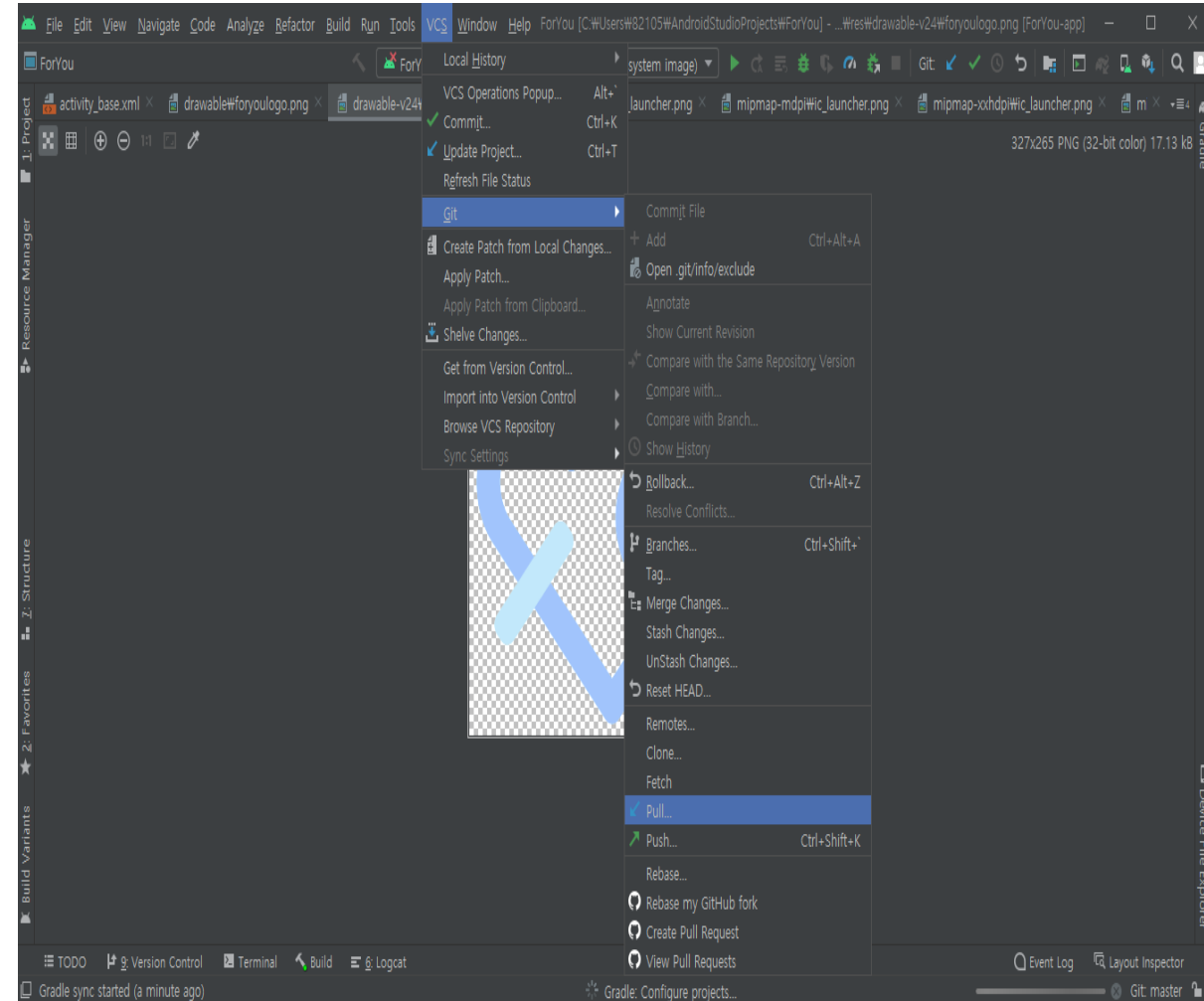


수정내용(변경사항)을 협업 프로젝트에 최신으로 업데이트

상단 Git - Update Project 를 통해 빠르게 업데이트

또는

상단 메뉴 VCS - Git - Pull 을 통해 업데이트 가능



포트폴리오를 마치며



”

개발자의 꿈을 안고 학교에 입학한 지 어느덧 4년이라는 시간이 흐르고 말았다. 이제는 빠르지만 학교를 졸업하고 사회에 진출하는 시기가 다가오고 있음을 점점 느끼게 된다.

2학년 2학기 졸업작품 교과목에 포트폴리오를 작성하면서 느낀 것은 그동안의 뿌듯함도 있겠지만 나를 반성하는 시간도 가지게 되어서 의미 있었다.

아직 훌륭한 개발자가 되기에는 부족하기에 하루하루 발전하는 훌륭한 개발자가 되기 위해 열심히 노력하겠다.

감사합니다.

