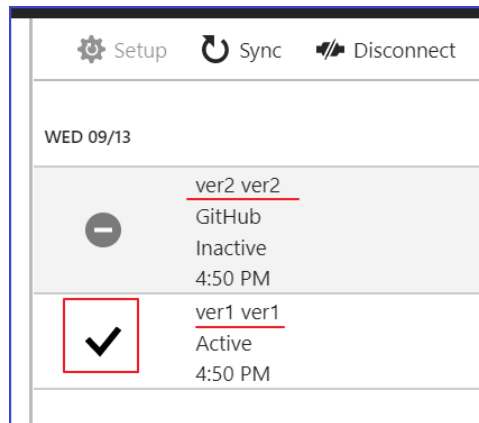


# 보충 자료

# WebApp을 Github에서 관리하기

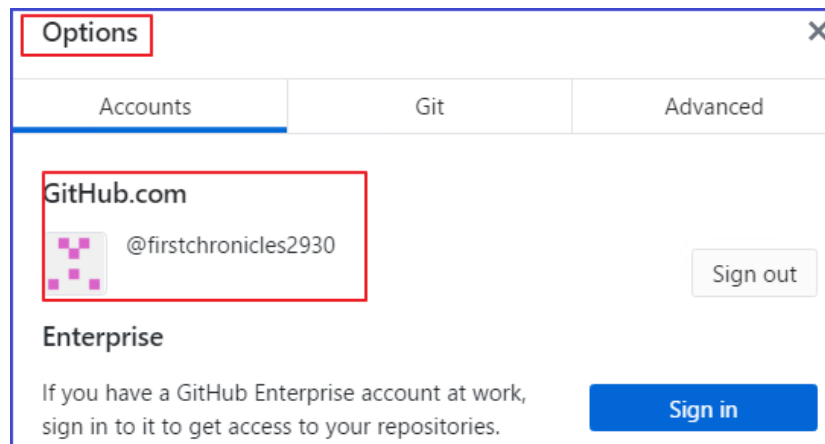
- WebApp을 GitHub에서 관리할 때의 장점
  - GitHub.com에 Contents를 올리면 Version 관리가 된다
    - 원하는 버전을 Redeploy를 하여 ver1, ver2를 번갈아 가면서 적용할 수 있다



- 자신이 개발한 것을 다른 사람과 공유할 수 있다
- 트래픽을 Github.com으로 보내므로 비용 절감이 된다(?)

# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - 먼저 WebApp을 Azure에서 생성한다
    - 접속을 하여 제대로 만들어졌는지 확인한다
  - GitHub.com에 회원 가입한다
  - <http://desktop.github.com>에서 로컬 컴퓨터에 GitHub Desktop 프로그램을 다운로드하여 설치한다
  - GitHub Desktop에서 GitHub에 로그인한다
    - File - Options



# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - GitHub.com에서 새로운 Repository를 생성한다
    - Kingdom 생성

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: firstchronicles2930 / Repository name: kingdom

Great repository names are short and memorable. Need inspiration? How about...

Description (optional)

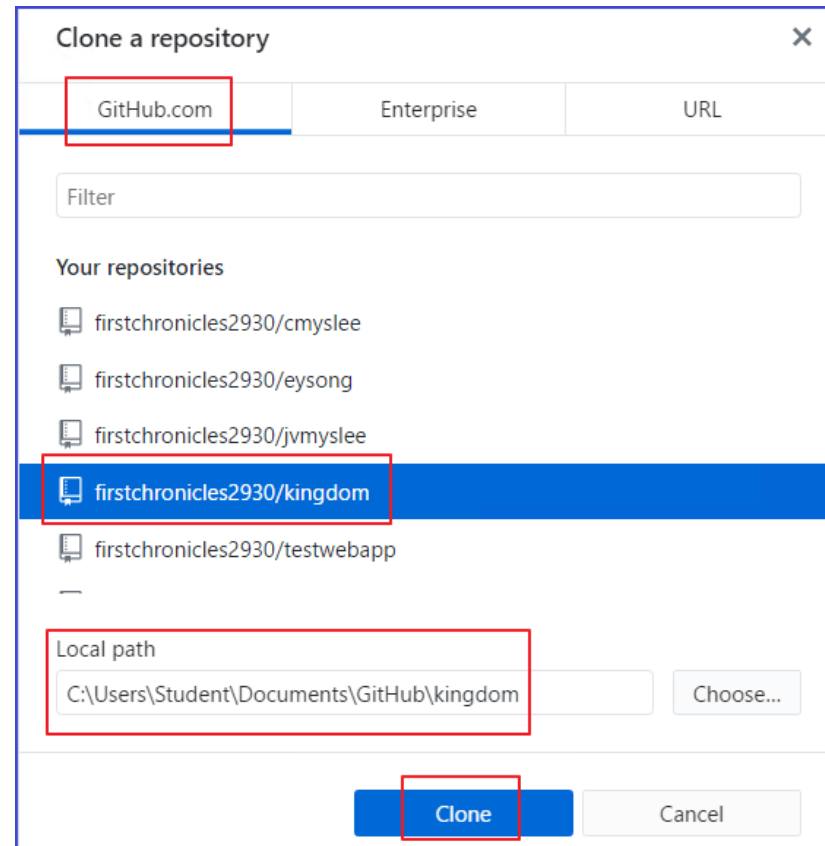
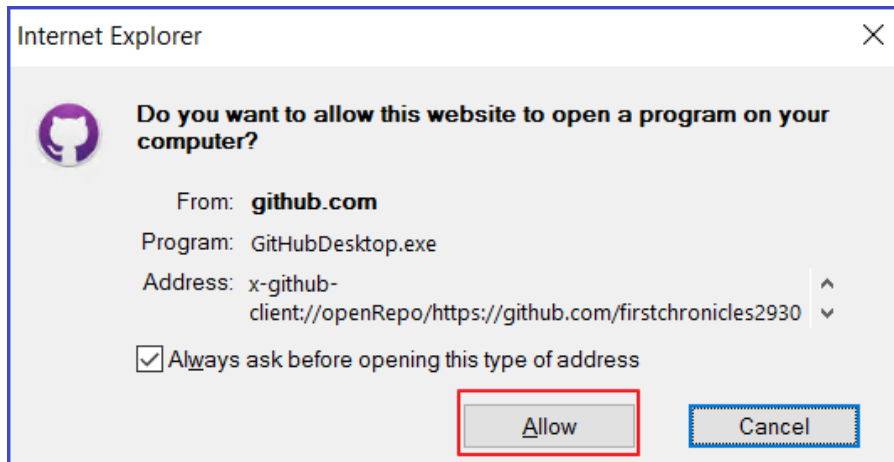
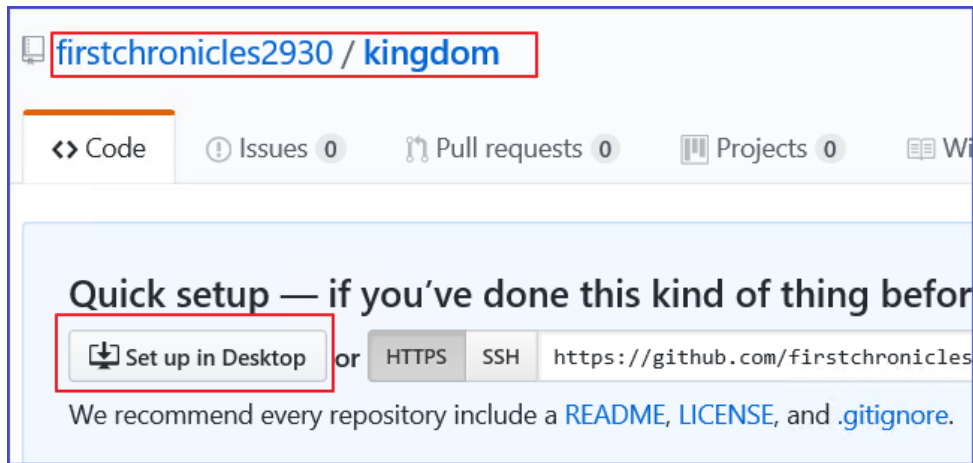
☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're...

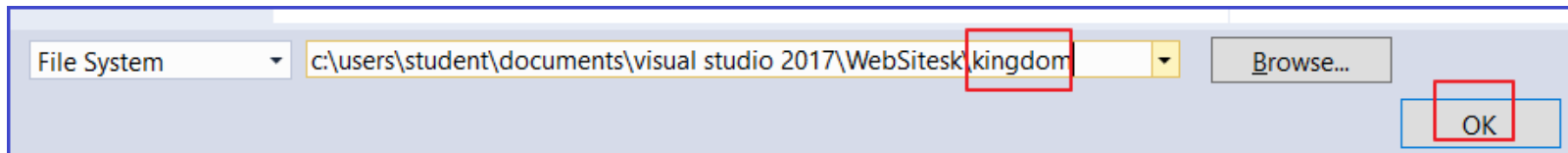
# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - GitHub Desktop 프로그램에서 작업 환경 설정하기

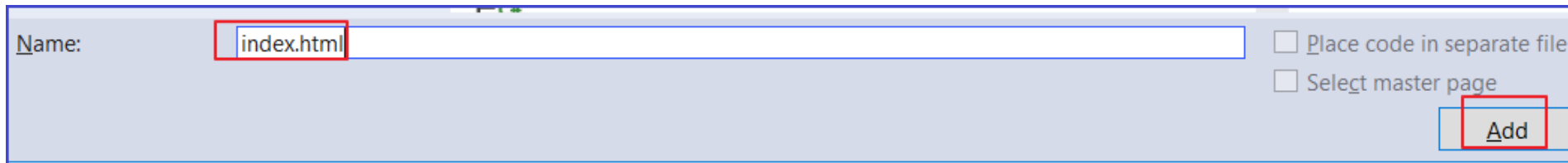


# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - Visual Studio 17에서 index.html 파일 생성하기
    - **File – New – Web site – ASP.NET Empty Web site**
      - Kingdom 경로 생성



- **Kingdom 우측 버튼 – Add – Add New Item – HTML Page**
  - **Index.html** 파일 생성 및 내용 작성 후 저장하기

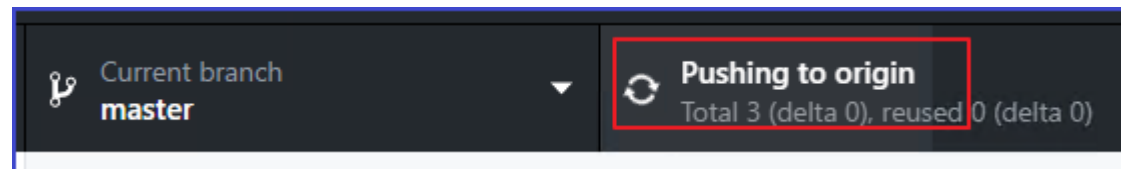
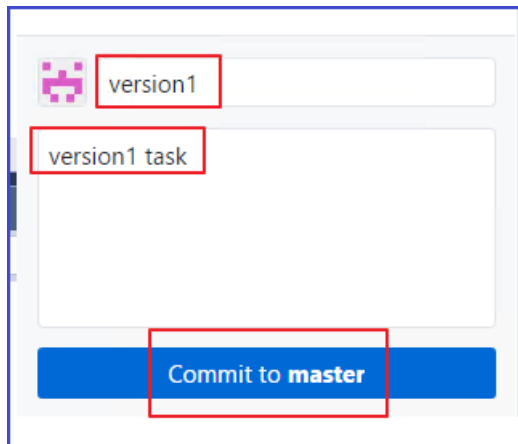
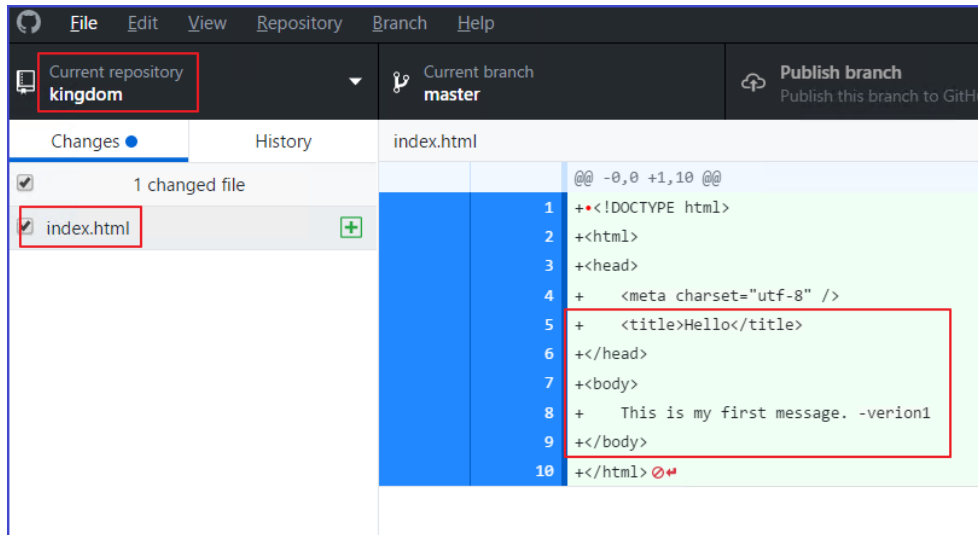


- **File – Save index.html as**
  - C:\Users\Student\Documents\GitHub\kingdom

```
<head>
  <meta charset="utf-8" />
  <title>Hello</title>
</head>
<body>
  This is my first message. -veri
</body>
```

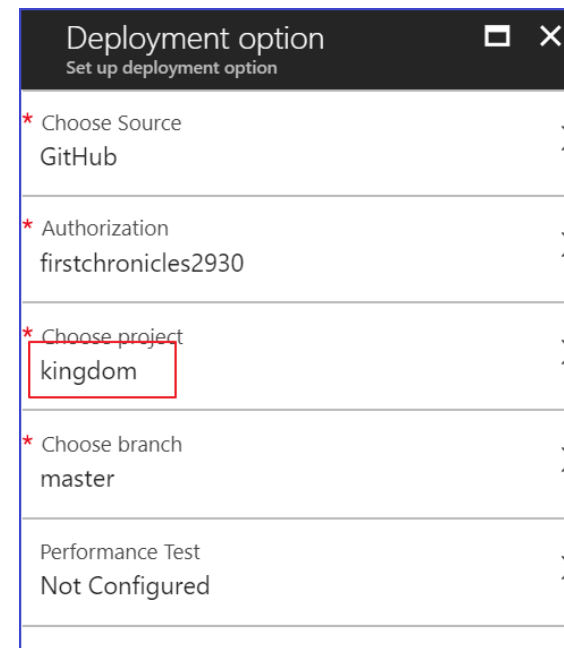
# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - GitHub Desktop 프로그램에서 작업하기



# WebApp을 Github에서 관리하기

- WebApp을 GitHub으로 변경하기
  - WebApp에서 GitHub으로 연결하기
    - **DEPLOYMENT – Deployment options – Choose sources – Git Hub**



- 웹사이트에 접속하여 본다
- VS2017에서 내용 수정하고 version2로 한 후 GitHub Desktop에서 commit하고 Publish한다. 그런 다음 다시 접속해 본다. (변경된 것 적용)
- 이전 것을 보려고 Version1을 **Redeploy**하여 다시 적용해 본다 (성공)



# Git 명령어 사용하기

- Git이란?

- Git(재수없고 멍청한 놈/자식)은 GitHub를 관리하는 명령어이다

- Git을 왜 사용하는가?

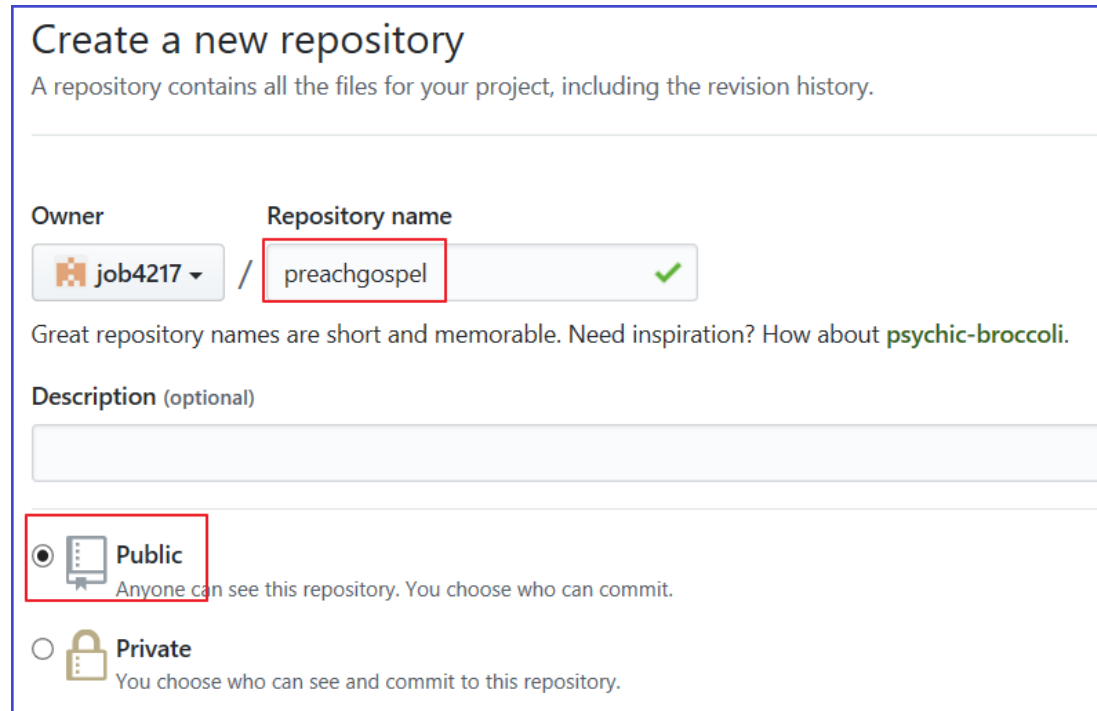
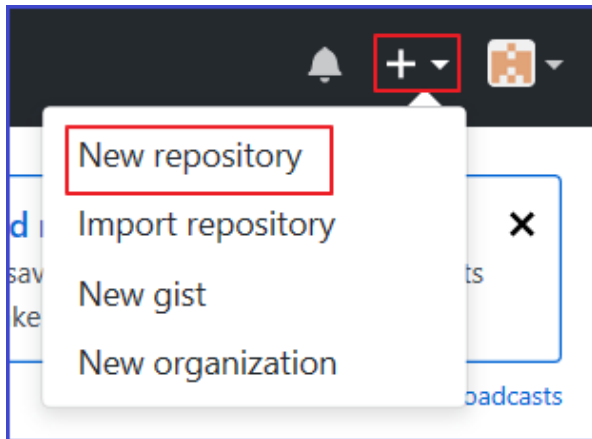
- 2명 이상의 개발자가 각각 다른 컴퓨터에서 coding을 하여 동일한 Web Site에 code를 upload하는 경우에 예상하지 못한 결과를 초래할 수 있다
  - Code를 잘못 수정한 경우에 이전 내용으로 돌아가고자 한다
  - 이럴 때 git을 사용하면 로컬 컴퓨터에서 commit을 하거나 GitHub.com으로 code를 push하면 항상 먼저 snapshot(checkpoint)을 수행하고 저장하도록 하는 것이 git이다
  - 그 결과 code의 충돌 방지 및 이전 내용으로 돌아갈 수 있어서 실수를 줄일 수 있다

- Git을 사용하려면?

- 로컬 컴퓨터에 git 프로그램을 설치해야 한다
    - <https://git-scm.com/download/win>에 접속하여 설치한다
  - Windows에 git을 설치하여 Linux 명령어를 사용하게 된다

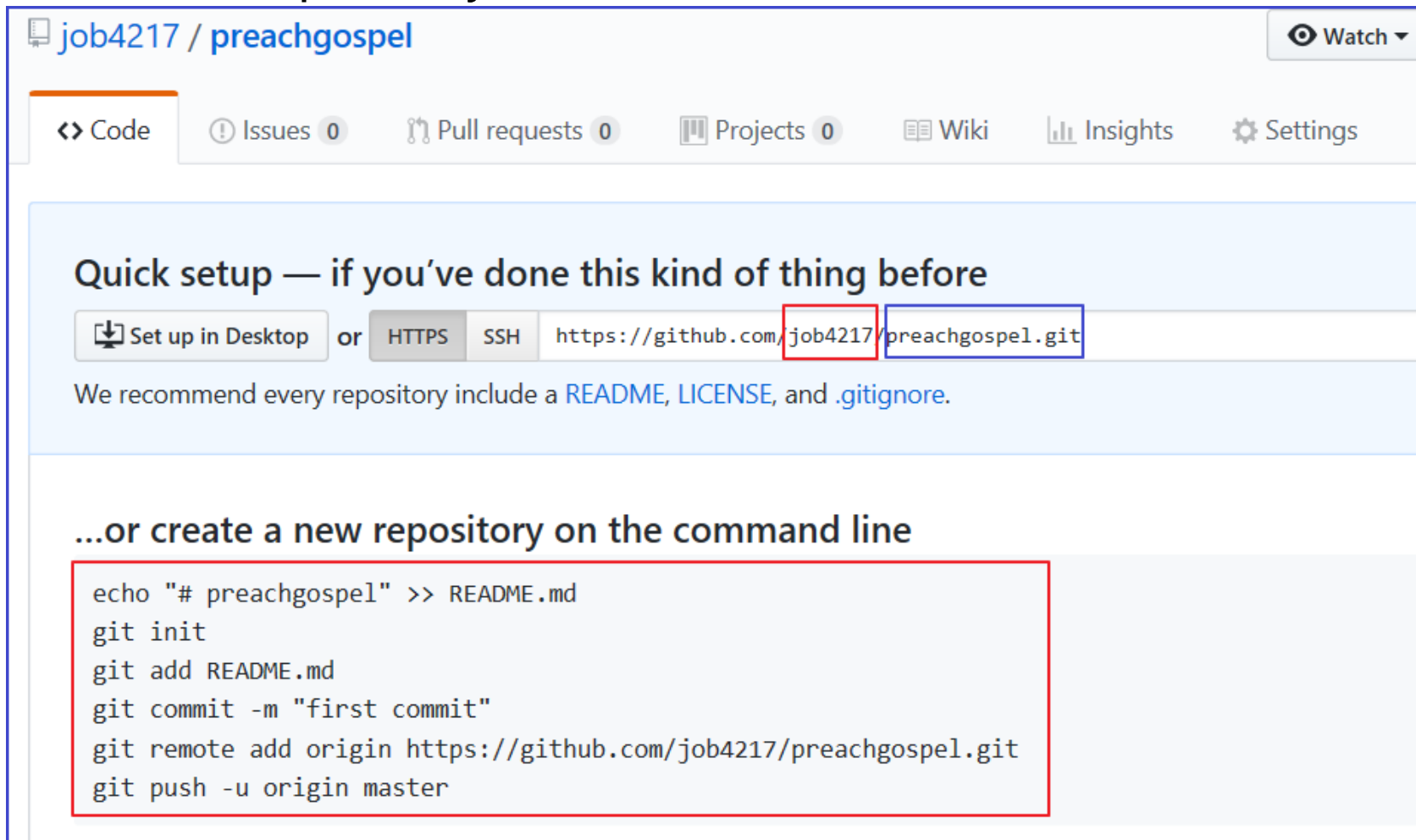
# Git 명령어 사용하기

- GitHub.com에 회원 가입하기
  - github.com에 접속하여 회원 가입한다
  - 이 때 Email 주소와 ID를 꼭 기억해 둔다
- GitHub에 Repository 생성하기

A screenshot of the 'Create a new repository' form on GitHub. The form title is 'Create a new repository' with a subtitle 'A repository contains all the files for your project, including the revision history.' Below this, there are two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'job4217' with a dropdown arrow. The 'Repository name' is 'preachgospel' with a green checkmark. Below these, there is a text input field for 'Description (optional)'. At the bottom, there are two radio button options for visibility: 'Public' (selected, with a red box around it) and 'Private'. The 'Public' option has a subtext: 'Anyone can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.'

# Git 명령어 사용하기

- GitHub에 Repository 생성하기(계속)



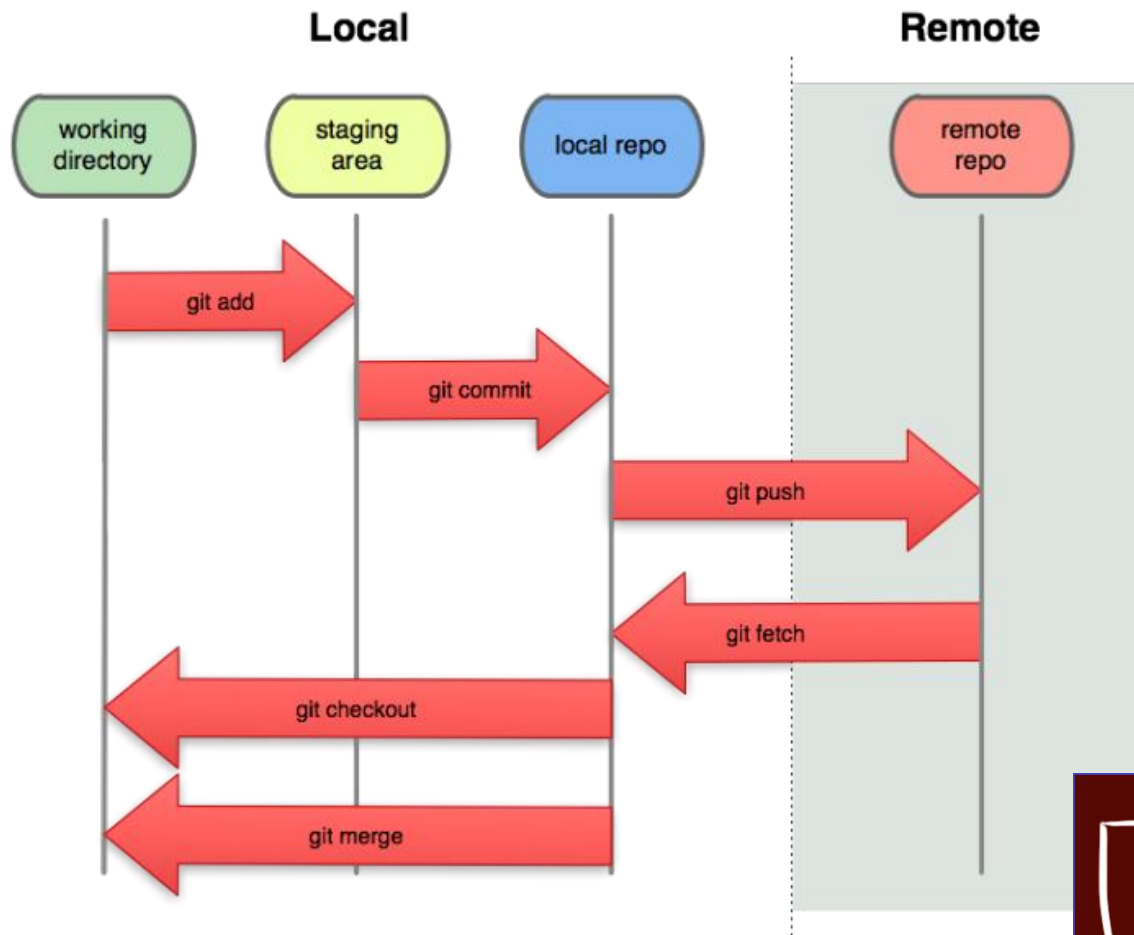
The screenshot shows the GitHub interface for a new repository named 'preachgospel' by user 'job4217'. The 'Code' tab is selected. Under 'Quick setup', the 'HTTPS' option is chosen, and the URL 'https://github.com/job4217/preachgospel.git' is displayed, with 'job4217' highlighted in a red box and 'preachgospel.git' in a blue box. Below this, a list of commands for creating a new repository is shown, enclosed in a red box:

```
echo "# preachgospel" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/job4217/preachgospel.git
git push -u origin master
```

- `https://github.com/아이디/프로젝트이름.git` 주소를 기억해 둔다

# Git 명령어 사용하기

- Git 작업 흐름 이해하기



# Git 명령어 사용하기

- Git 명령어 사용하기

- Git은 항상 git으로 시작한다

- **git** status
    - **git** init
    - Git 명령어 도움말 사용하기
    - **git help** init
    - **git help** status

- Git 사용자 정보 등록하기

- **git config --global user.name** "Yongshik Lee"
    - **git config --global user.email** "job4217@outlook.com"
    - Email은 github.com에 등록할 때 사용한 것을 입력해야 한다

- Local Repository로 사용할 디렉터리 생성하기

- cd
    - mkdir preachgospel
    - cd !\$
      - i\$: 이전 명령어의 마지막 인자(preachgospel)를 대신하여 사용한다
    - pwd

# Git 명령어 사용하기

- Git 명령어 사용하기

- 현재 directory를 git이 관리하는 디렉터리로 초기화하기(Local Repo 생성)
  - ls -la
  - **git init**
  - ls -la
  - 프로젝트 파일이 저장될 Local Directory를 생성한 것이다
- 파일 생성하기
  - **echo** "This is my web site in Azure." > **index.html**
  - **cat** index.html
- 파일 상태 확인하기
  - **git status**
    - 아무 정보가 없다
- Staging Area로 파일을 이동하기
  - **git add** index.html
  - **git status**

# Git 명령어 사용하기

- Git 명령어 사용하기

- Local Repo로 파일 이동하기

- **git commit -m** "1-Adding index.html file to Local Repo" **index.html**
    - 이 때 내부적으로 snapshot을 찍어 둔다

- **echo** "God is Love." > **gospel.html**
    - **git add** gospel.html
    - **git commit -m** "2-Adding gospel.html"
    - **echo** "Good Job!" >> index.html
    - **echo** "Good news." >> gospel.html
    - **git status**
    - **git commit -a -m** "3-Adding all files to Local repo"
      - working directory에 변경된 모든 파일을 commit하기
    - **git status**

# Git 명령어 사용하기

- Git 명령어 사용하기

- Commit하여 Local Repo에 기록된 내용 확인하기(log 확인)

- **git log**
- **git log --oneline**

- 특정한 파일이 어떤 commit을 했는지 확인하기

- **git log -- index.html**
- **git log -- gospel.html**



- GitHub의 원격 저장소 지정하기

- 로컬 저장소에 저장된 파일을 GitHub에 올리기 전에 먼저 원격 저장소가 어떤 것인지 지정해두어야 한다
- **git remote add origin <https://github.com/job4217/preachgospel.git>**
- **git remote -v**
  - 연결이 잘되었는지 확인한 것임



# Git 명령어 사용하기

- Git 명령어 사용하기
  - GitHub에 파일 업로드하기
    - **git push origin master**
      - GitHub의 ID와 암호를 입력한다. 그러면 2개의 파일이 업로드된다
  - GitHub에서 추가된 파일 확인하기

job4217 4-Adding updated files to Local Repo		Latest commit c812bd1 13 minutes ago
 <a href="#">gospel.html</a>	4-Adding updated files to Local Repo	13 minutes ago
 <a href="#">index.html</a>	4-Adding updated files to Local Repo	13 minutes ago

# Custom Azure DNS Domain 위임하기

- myazurekr.wordpress.com 접속하기
  - Azure DNS에 centos7한