

# 빅데이터 분석 프로그래밍

## Week-01. 학습환경 설정

# 수업에 사용되는 툴/서비스

- open.yonsei: 공지사항 및 성적평가
- Github: 모든 강의 리소스가 공유되는 지점
- Shell: Windows는 Gitbash, macos는 terminal
- Slack: 커뮤니케이션 툴
- Kahoot: 실시간 퀴즈/설문 조사를 위한 툴
- Anaconda: 파이썬 통합 패키지 소프트웨어
- Jupyter notebook: 통합개발환경 (a.k.a, 코드에디터)



**GitHub**



# Shell

## 운영체제의 서비스에 접근하기 위한 유저 인터페이스

- Command Line Interface
  - 텍스트 명령어로 컴퓨터에 동작을 실행하는 인터페이스
  - 우리가 마우스를 클릭해서 동작을 실행할 수 있는 인터페이스는 Graphical User Interface
- 이러한 CLI를 제공하는 환경을 Shell 또는 Command Line Interpreter라고 불리움
- Windows와 다른 운영체제와의 CLI 명령어가 다르기 때문에, Windows에서는 **gitbash**라는 프로그램으로 Shell명령어를 사용한다.

\*모두가 그렇다는건 아니고 이번 수업에서만



최초의 매킨토시는 그래픽 사용자 인터페이스, 내장 스크린 및 마우스를 특징으로하는 최초의 대중 시장 개인용 컴퓨터입니다.

# Shell

## 예시화면

```
seojungwon@seojungwonui-MacBook-Air: ~  
Last login: Wed Sep 2 15:54:44 on console  
→ ~  
→ ~  
→ ~ ls  
Applications      ParlAI             nltk_data  
AspNetCoreOnDocker Pictures           openai  
Books              Postman            out  
Desktop            Public              release_key.keystore  
Development        PycharmProjects   release_key.keystore.jks  
Documents          anaconda3           robocode  
Downloads          arm-cs-tools        scikit_learn_data  
Dropbox            check.json          seaborn-data  
Library            dump.rdb             temp_video  
Movies             eclipse              tensorflow_datasets  
Music              eclipse-workspace    witsml-client  
OneDrive           get-pip.py  
→ ~ pwd  
/Users/seojungwon  
→ ~
```

# Shell

## 자주 사용되는 명령어

- ls : 현재 위치한 폴더에 존재하는 파일/폴더 출력
- cd <위치> : 특정 폴더로 이동
- pwd : root directory 기준으로 현재 위치 출력
- vi <파일이름>: vi editor를 이용하여, 텍스트 작업
- mkdir <폴더이름> : 폴더 생성
- touch <파일이름> : 파일 생성
- rm <파일이름> : 파일 삭제

# Github

## 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스

- Git:
  - 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템
  - 깃은 2005년에 리눅스 커널 개발을 위해 초기 개발에 기여한 다른 커널 개발자들과 함께 2005년에 **리누스 토르발스**가 처음 개발
  - 버전관리를 제대로 하지 않으면... *test1.py, test2.py, test\_final.py, test\_real\_final.py, test\_final\_final.py*
- Github:
  - Git을 웹상에서 호스팅해주는 서비스
  - 즉, Github 외에도 Git을 웹서비스로 제공하는 서비스가 더 있다. (Gitlab, Bitbucket ..)
  - 2018년 10월에 마이크로 소프트에 인수

# Github

## 주요 구조

- Repository (저장소):
  - 간단하게 말하면, 하나의 프로젝트 폴더
  - 이 폴더(저장소)를 기준으로, 폴더 내에 있는 파일 또는 폴더들이 버전관리가 가능
  - 즉, test.py, test\_v2.py 이런식으로 만들지 않아도, 같은 파일에 수정을 한뒤 git 명령어를 활용해 push를 하면, 자동으로 push 시점에 대한 버전이 생성됨
- Organization:
  - 기본적으로 개인용 Repository를 사용하지만, 팀 단위로 개인 Repository에 작업을 하는 것은 바람직하지 않음.
  - 또한 프로젝트당 Repository가 필요하기 때문에, 이들을 그룹지어 놓을 수 있는 단위가 필요함
  - 페이스북 그룹과 비슷한 느낌

# Github

원격저장소를 활용한 파일 교환

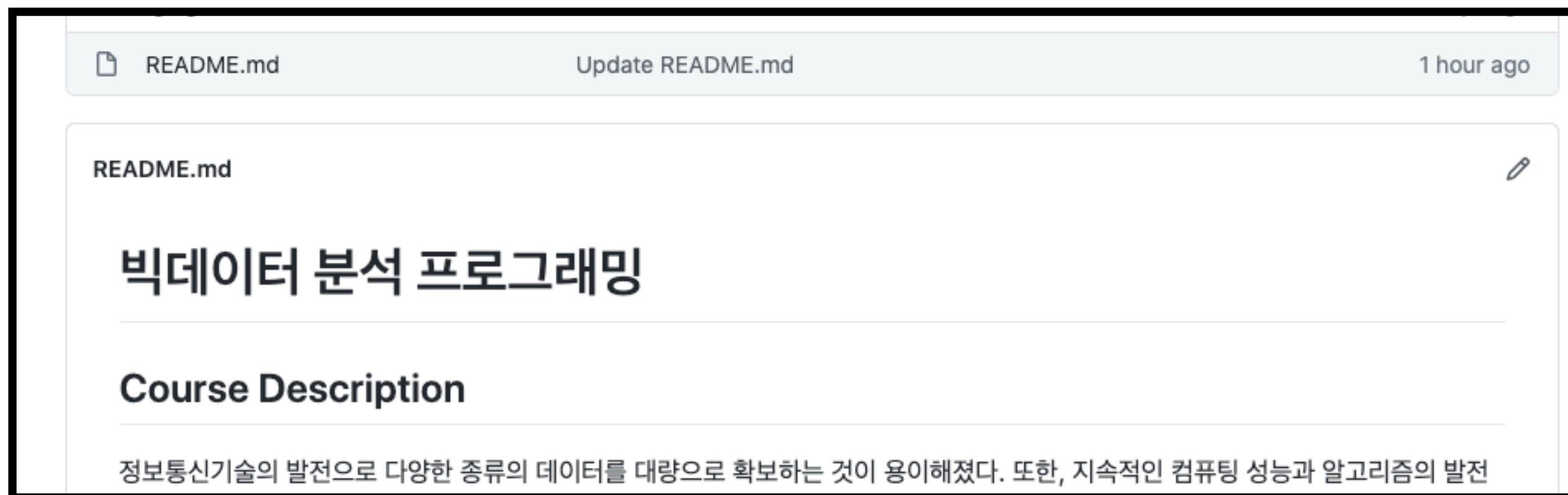




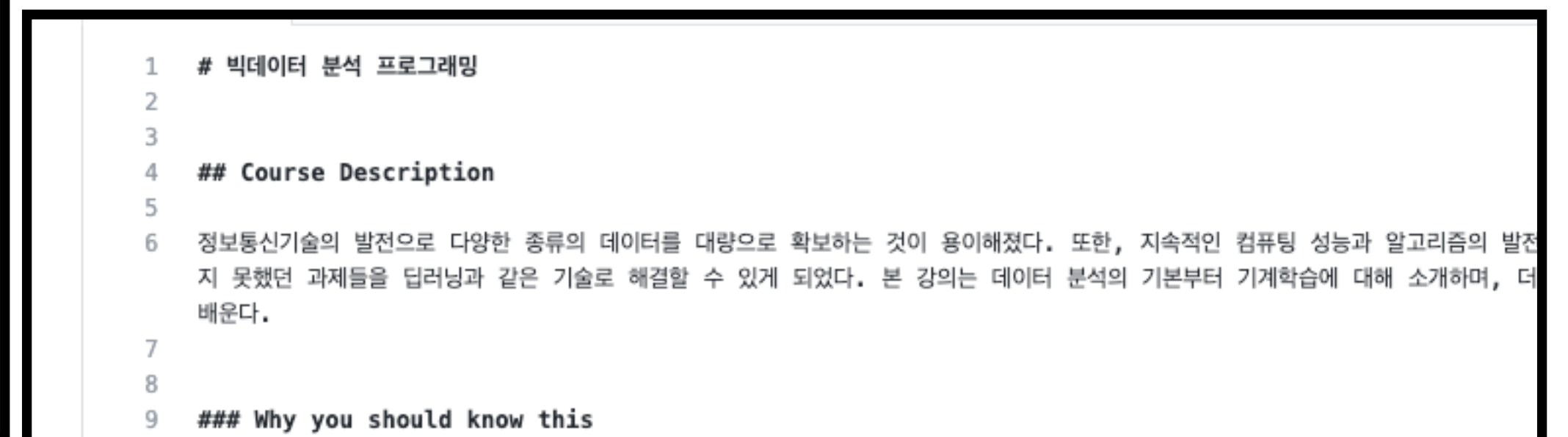
# Github

## README 파일

- README.md:
  - 간단하게 말해서, 해당 Repository 또는 폴더의 메인페이지와 같은 역할을 함.
  - README라는 예약어로, 깃허브 내에서는 자동으로 현재 디렉토리의 이 파일을 호출함
  - 만약 test.MD라는 형태로 저장을 하면, 이것은 일반 파일과 같게 취급
  - 작성 포맷은 Markdown이라는 markup-language를 사용



렌더링 된 모습

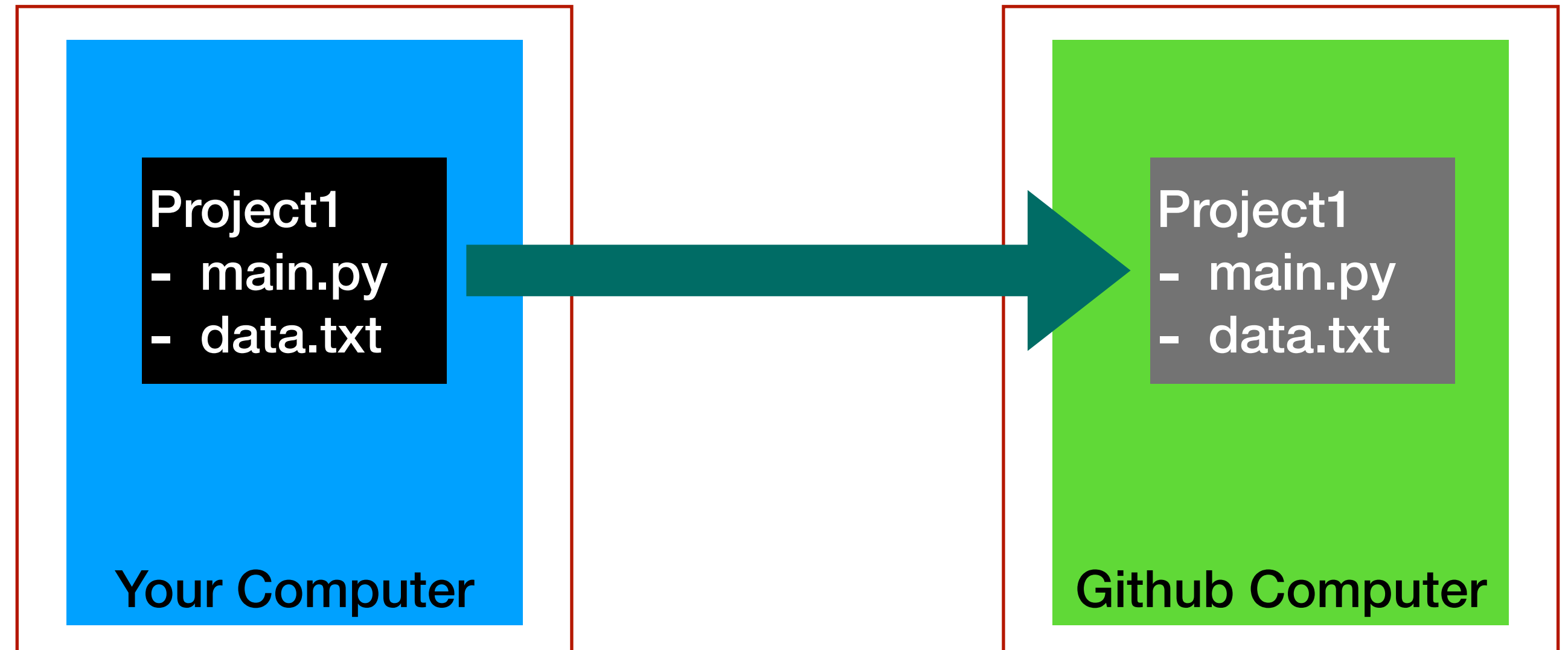


실제 작성된 내용

# Github

## 사용시나리오 #1

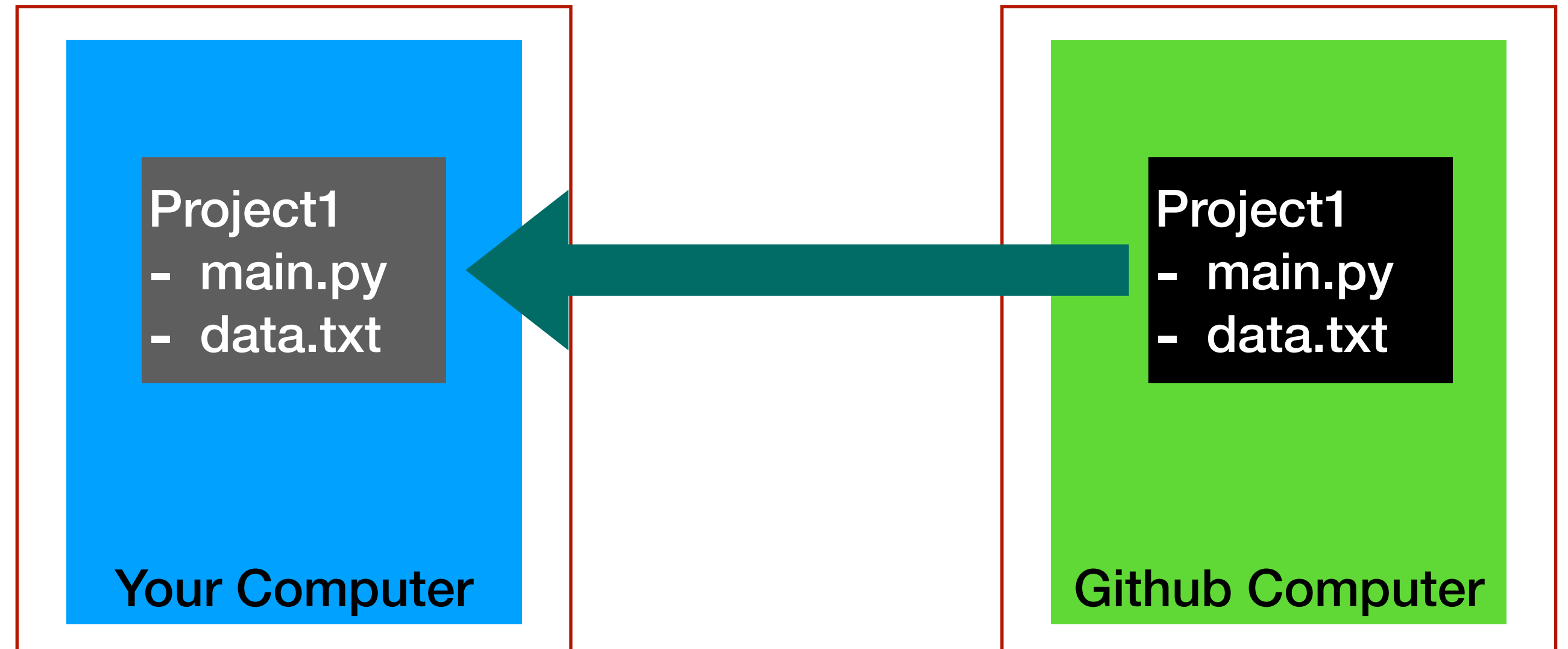
- Local에서 Remote로
- Shell 명령어
  - `cd Project1` (Project1 폴더로 이동)
  - `echo "# test" >> README.md` (README.md 파일 생성)
  - `git init` (폴더가 git 환경을 갖추수 있도록 초기화)
  - **`git add .`** (폴더 내에 있는 모든 파일을 staging area 에 추가)
  - **`git commit -m "my first commit"`** (변경된 내용을 Local Repository에 추가)
  - `git branch -M main` (최초에만 실행, 현재 브랜치를 main으로 설정)
  - `git remote add origin https://github.com/thejungwon/project1.git` (Remote 저장소 지정)
  - **`git push -u origin main`** (Remote 저장소에 배포)



# Github

## 사용시나리오 #2

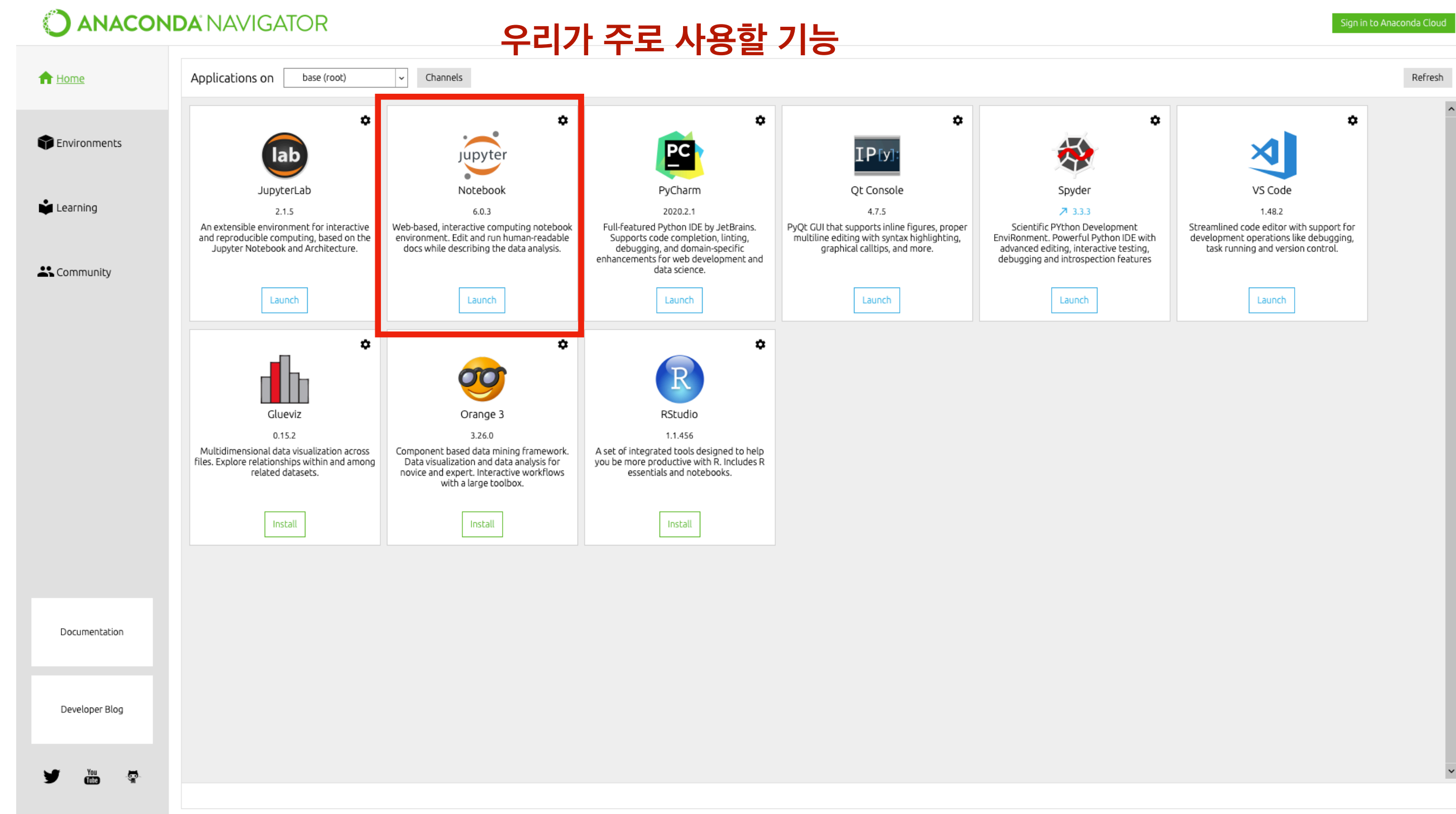
- Remote에서 Local로
- Shell 명령어
  - `cd Projects` (Project1을 받을 상위 폴더로 이동)
  - **`git clone https://github.com/thejungwon/project1.git`** (local로 복제)
  - `cd project1`
  - 무언가 변경을 한다!
  - **`git add .`** (폴더 내에 있는 모든 변경된 파일을 staging area 에 추가)
  - **`git commit -m "my first commit"`** (변경된 내용을 Local Repository에 추가)
  - **`git push -u origin main`** (Remote 저장소에 배포)



# Anaconda

패키지 관리와 배포를 단순케 할 목적으로 파이썬과 R 프로그래밍 언어의 무료-오픈 소스 배포판

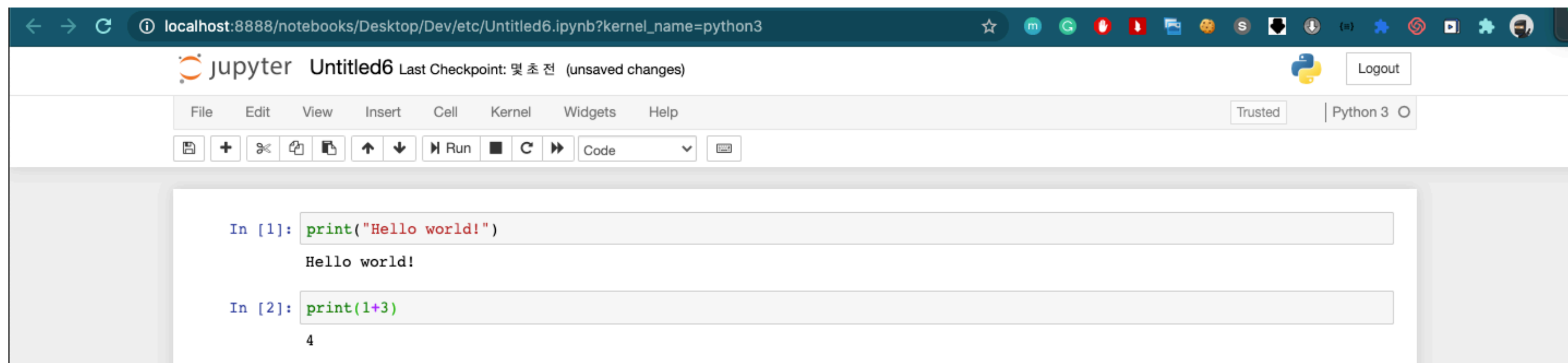
- 파이썬에 대한 패키지 관리 및 데이터 과학을 위한 각종 툴을 종합적으로 모아서 제공
  - 각각의 패키지, 언어, 소프트웨어를 따로 설치하고 환경 설정을 하는 수고를 덜 수 있음



# Jupyter Notebook

라이브 코드, 방정식, 시각화 및 설명 텍스트가 포함 된 문서를 만들고 공유 할 수있는 오픈 소스 웹 애플리케이션

- 데이터 사이언스에 적합한 통합개발환경(IDE)
  - 일반 IDE (e.g., visual studio code, atom, pycharm 등)은 코드 작성에만 치중되어있고, 저장후 파일 전체를 실행시켜야만 함.
  - 각각의 함수 또는 로직이 **오래걸리는** 데이터 사이언스의 코딩 방식에는 적합하지 않음
  - **Interactive** 하게 셀(cell) 단위로 코드를 실행할 수 있음
  - 일반적으로 배포 전, 개발 및 모델 작성 과정에서는 주피터 노트북을 주로 활용



**E.O.D**