

# 1장

## 파이썬 프로그램

# 목차

1. 파이썬 소개
2. 파이썬 개발 환경 설정
3. 파이썬 개발 툴 소개

# 01. 파이썬 소개

## ■ 파이썬의 등장

- 파이썬(Python)은 귀도 반 로섬(Guido Van Rossum)이 1991년에 개발한 언어로, 처음에는 C 언어 기반으로 개발되었는데 이후 다양한 기능이 개발되어 추가되었다.



[ 귀도 반 로섬 ]

- 1991년에 첫 버전이 공개된 후 지금까지 많은 인기를 얻고 있는 스크립트 형태의 범용 프로그래밍언어
- 프로그래머가 원하는 모든 작업을 할 수 있도록 설계한 범용언어
- 명령형 언어이면서 스크립트 방식지원
- -프로그래밍적인 구현에 적합
- -데이터수집과처리에활용할수있는다양한라이브러리제공
- 웹서버프로그래밍, 데이터분석, 시스템자동화, IoT 프로그래밍까지 활용분야도 다양

# 01. 파이썬 소개













## ■ 파이썬의 특징

1. 플랫폼 독립적인 언어 : 어떤 운영체제든 상관없이 사용할 수 있는 언어.
2. 인터프리터 언어 : 컴파일러 언어와 달리, 소스코드 자체가 바로 실행되는 특징이 있는 언어. 속도는 느리지만, 굉장히 간편하게 사용할 수 있다.
3. 객체 지향 언어 : 해당 프로그램이 해결해야 할 문제의 구성요소를 요소별로 정의한 뒤, 각 요소의 기능(메서드)과 정보(속성)를 정의하여 요소들을 결합하고, 프로그램을 작성하는 방식.
4. 동적 타이핑 언어 : 프로그램의 실행 시점에서 각 프로그램 변수의 타입을 결정하는 언어.

구분	컴파일러	인터프리터
작동 방식	소스코드를 기계어로 먼저 번역하고, 해당 플랫폼에 최적화되어 프로그램을 실행함	별도의 번역 과정 없이 소스코드를 실행 시점에 해석하여 컴퓨터가 처리할 수 있도록 함
장점	실행 속도가 빠름	간단히 작성, 메모리가 적게 필요
단점	한 번에 많은 기억 장소가 필요함	실행 속도가 느림
주요 언어	C, 자바(Java), C++, C#	파이썬, 스칼라

# 01. 파이썬 소개

## ■ 프로그래밍언어 순위(TIOBE Index for October 2022)

May 2022	May 2021	Change	Programming Language		Ratings	Change
1	2	▲		Python	12.74%	+0.86%
2	1	▼		C	11.59%	-1.80%
3	3			Java	10.99%	-0.74%
4	4			C++	8.83%	+1.01%
5	5			C#	6.39%	+1.98%
6	6			Visual Basic	5.86%	+1.85%
7	7			JavaScript	2.12%	-0.33%
8	8			Assembly language	1.92%	-0.51%
9	10	▲		SQL	1.87%	+0.16%
10	9	▼		PHP	1.52%	-0.34%
11	17	▲▲		Delphi/Object Pascal	1.42%	+0.22%
12	18	▲▲		Swift	1.23%	+0.08%

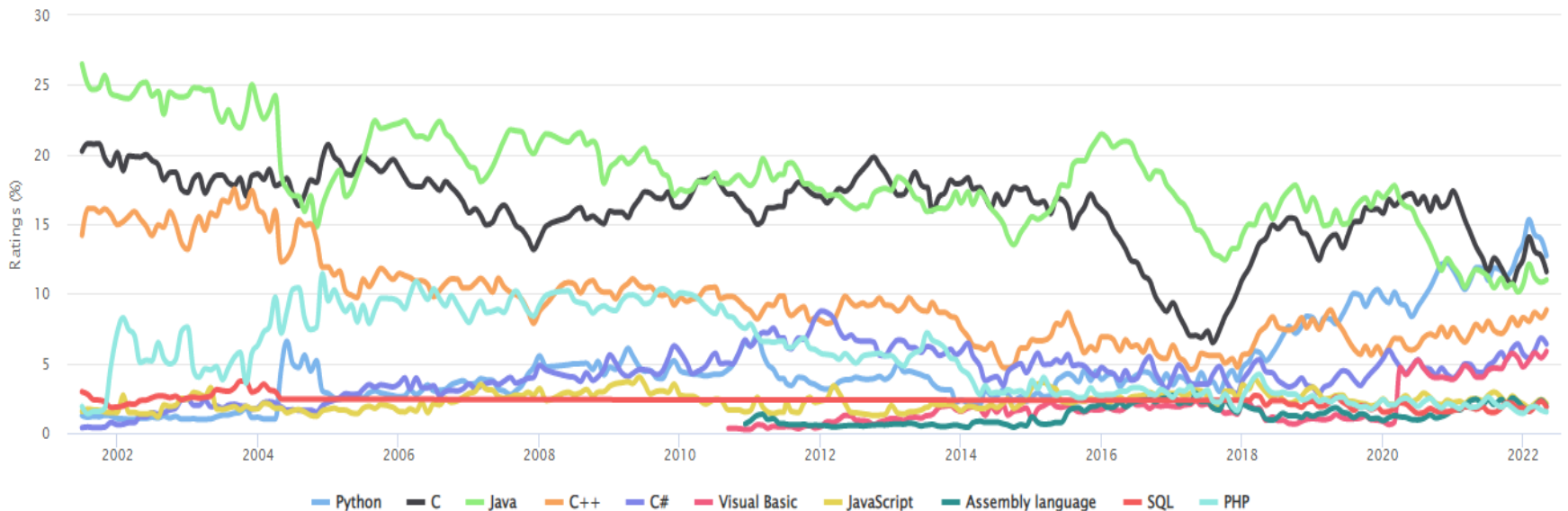
# 01. 파이썬 소개

## ■ 파이썬의 활용

- 파이썬은 2016년부터 사용량이 가파르게 높아짐  
⇒ 데이터분석분야에서파이썬이많이활용되고있기때문
- 파이썬은 '빠르게 성장하고 있는 언어'

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# 01. 파이썬 소개

## ■ 빅데이터처리 언어로서 파이썬의 장점

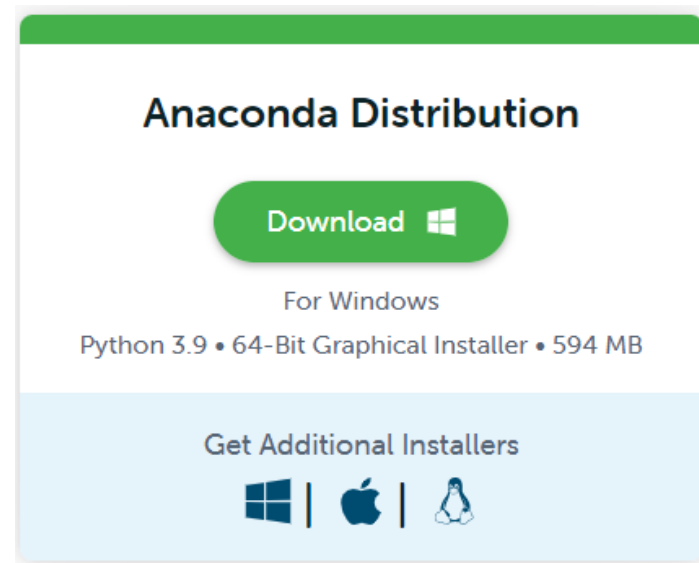
- 이해하기 쉽고 유연한 문법으로 좋은 접근성을 가짐
- 빅데이터처리 언어로서 많은 커뮤니티가 형성되어 있음
- 가독성이 좋고, 간결하며, 스탠다드 라이브러리가 잘 갖춰져 있음
- 데이터분석 관련 패키지가 최근 몇 년 사이 눈에 띄게 발전하여 NumPy, SciPy, Pandas, Matplotlib등 데이터 분석 관련 오픈 소스 라이브러리들을 무상으로 사용할 수 있음

## 02. 파이썬 개발 환경 설정

### ■ 파이썬 개발 환경 설정

#### 1. 다운로드 사이트

- 아래사이트에서 원하는 버전의 파이썬 설치 프로그램을 다운로드하여 설치 후 **필요한 라이브러리와 개발 툴 추가 설치**
  - <https://www.python.org/downloads/>
  - 일반적인 Python , 기본 모듈 포함
- 아래사이트에서 **Anaconda Distribution** 설치
  - <https://www.anaconda.com/products/individual>
  - 다양한 과학 계산을 모듈을 묶어 패키지로 제공





## 02. 파이썬 개발 환경과 설치

### ■ 파이썬 인터프리터 설치: Anaconda

1. 우선 Anaconda 다운로드 페이지(<https://www.anaconda.com/products/individual>)에서 Download 클릭 후 'Python3.x'로 시작하는 버전의 윈도우용 인스톨러를 다운로드

#### Anaconda Installers

##### Windows

Python 3.9

64-Bit Graphical Installer (594 MB)

32-Bit Graphical Installer (488 MB)

##### MacOS

Python 3.9

64-Bit Graphical Installer (591 MB)

64-Bit Command Line Installer (584 MB)

64-Bit (M1) Graphical Installer (428 MB)

64-Bit (M1) Command Line Installer (420 MB)

##### Linux

Python 3.9

64-Bit (x86) Installer (659 MB)

64-Bit (Power8 and Power9) Installer (367 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (568 MB)

64-bit (Linux on IBM Z & LinuxONE) Installer (280 MB)

## 02. 파이썬 개발 환경 설정

### ■ Anaconda

Anaconda

- 데이터 과학 및 머신 러닝을 위한 파이썬과 R 프로그래밍 언어의 'Freemium' 오픈 소스 배포판

기본적인 기능은 무료로 제공하고  
고급 기능은 돈을 받고 판매

- 세계에서 가장 유명한 파이썬 데이터 과학 플랫폼
  - 모든 데이터 과학패키지의 쉬운설치, 패키지·종속성 및 환경관리 가능
  - CONDA, 파이썬 및 150개가 넘는 과학 패키지와 그 종속성과 함께 제공되는파이썬 배포판
- CONDA : 패키지 및 환경관리자

## 02. 파이썬 개발 환경 설정

### ■ Anaconda

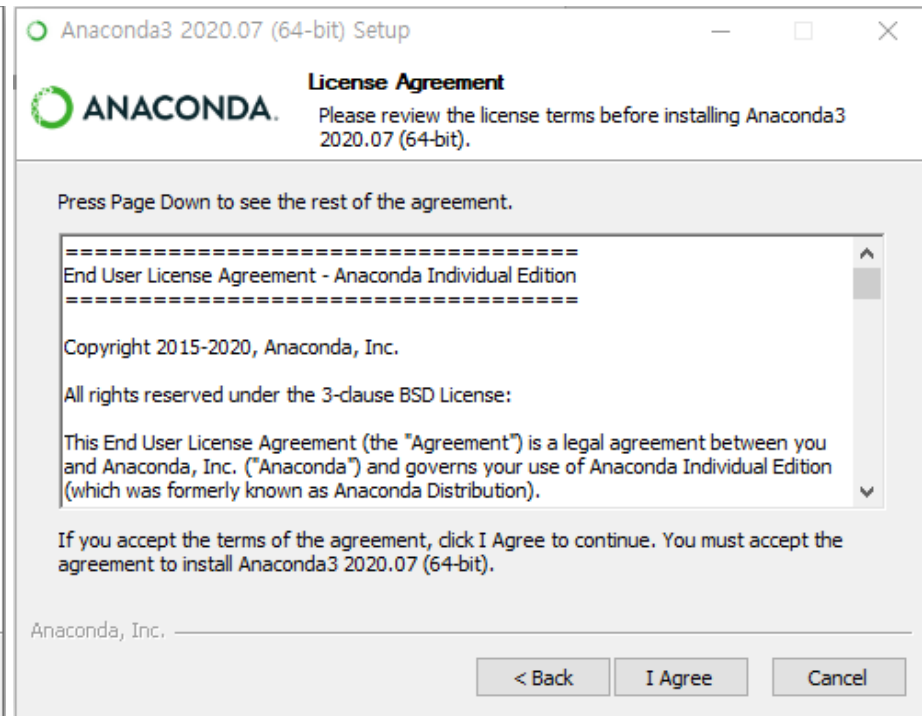
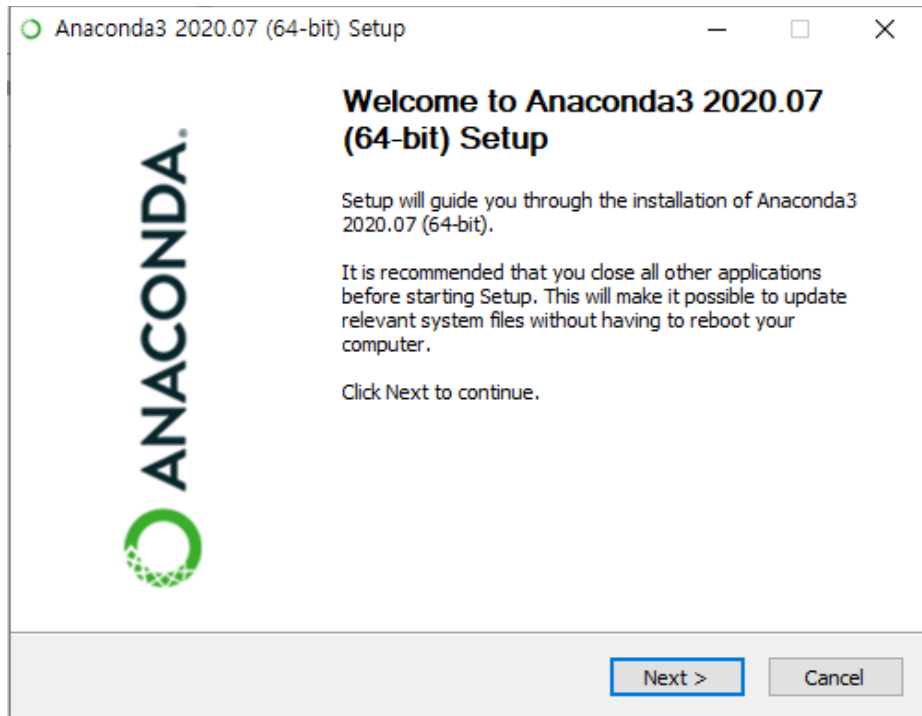
- 파이썬 개발환경과 수학·과학·데이터분석분야에서 필요한 거의 모든 패키지(NumPy, SciPy, Pandas, Matplotlib 등) 포함



## 02. 파이썬 개발 환경 설정

### ■ Anaconda 설치

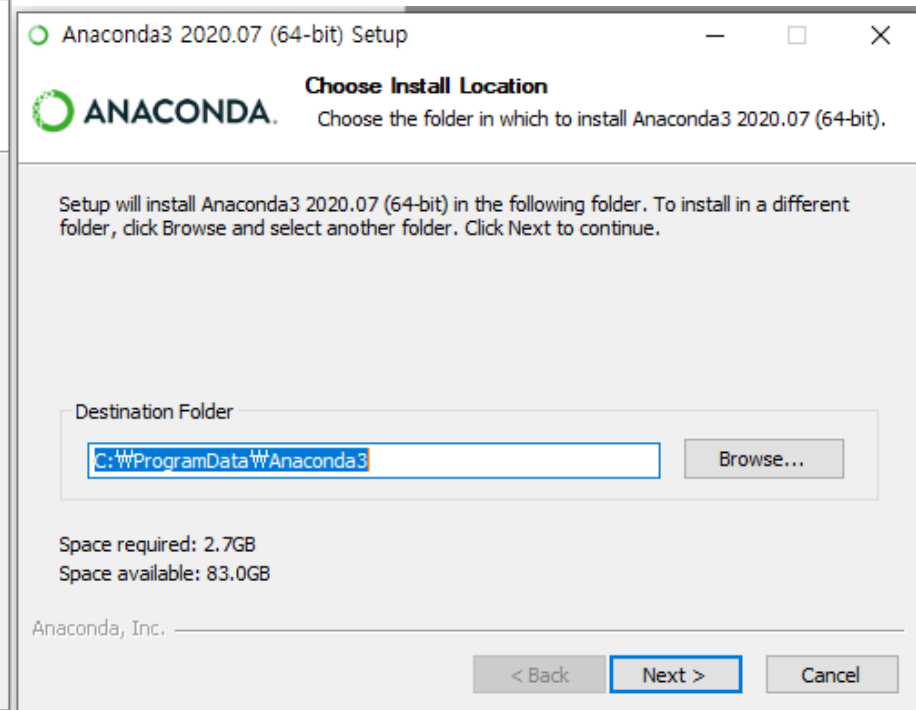
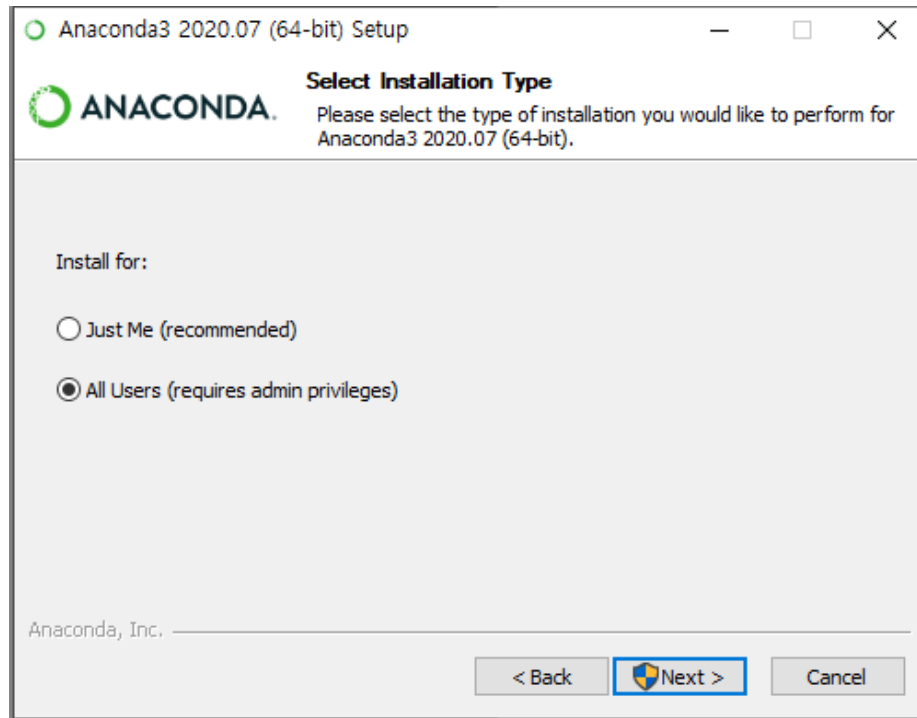
- 다운로드한 인스톨러를 실행하고 [Next]를 클릭한 후, 다음 화면에서 [I Agree]를 클릭한다.



## 02. 파이썬 개발 환경 설정

### ■ Anaconda 설치

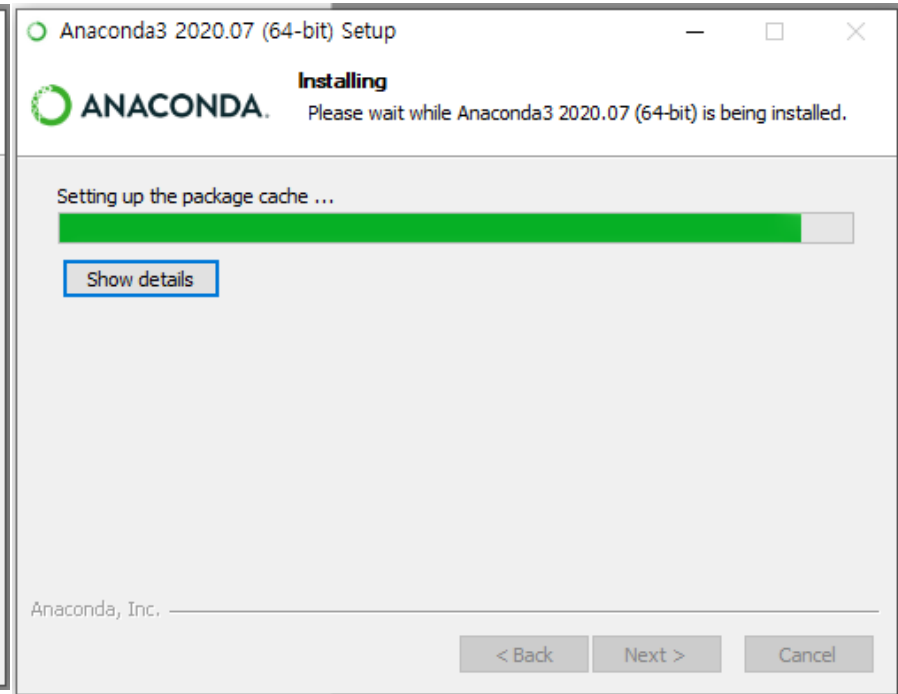
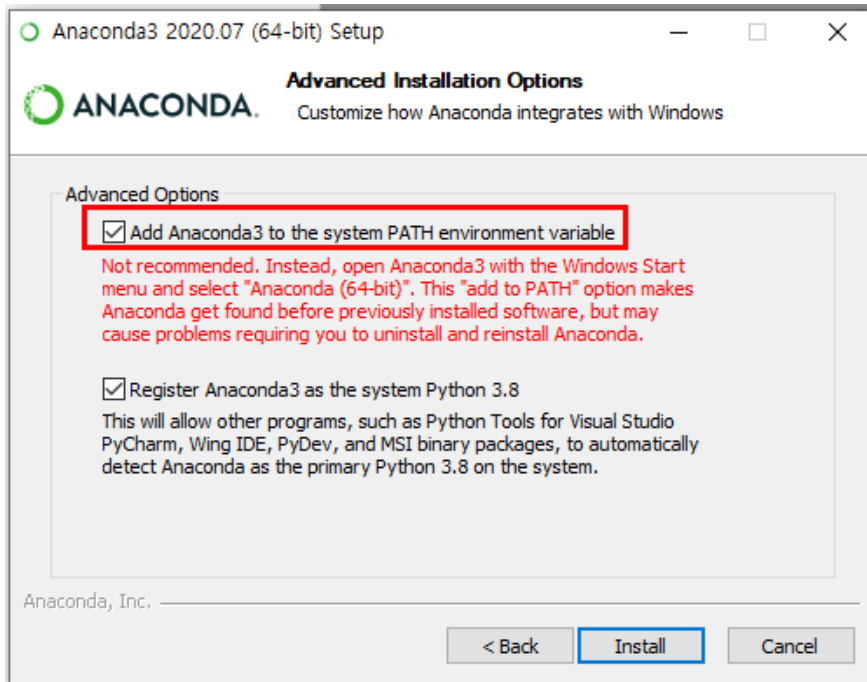
- 인스톨 타입 설정 창에서 'All Users'를 선택하고, [Next]를 클릭한다.
- 인스톨 경로 설정 창에서 [Browse]를 클릭하여 적절한 경로를 지정하고, [Next]를 클릭한다.



## 02. 파이썬 개발 환경 설정

### ■ Anaconda

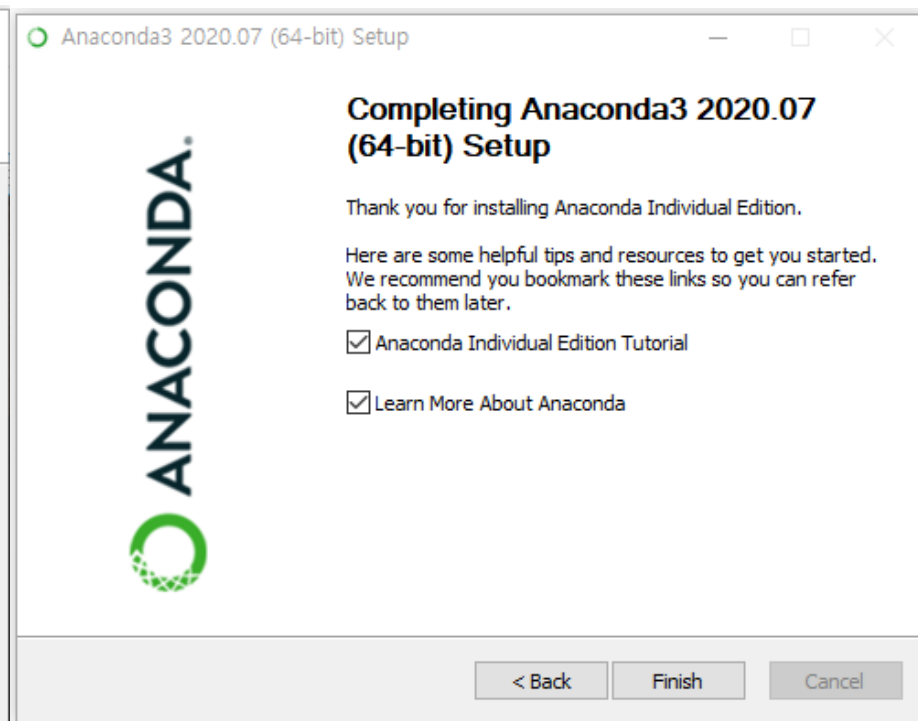
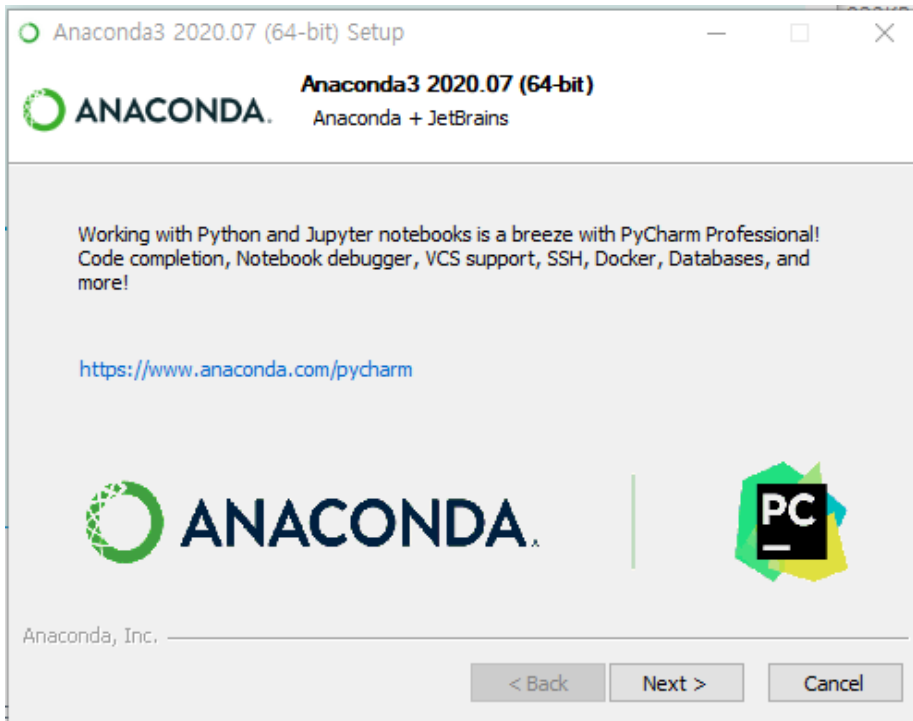
- 인스톨 옵션 설정 창에서 [Add Anaconda3 to the system PATH environment variable], [Register Anaconda3 as the system Python3.8] 체크박스를 모두 체크한 후, [Install]을 클릭한다.
- 설치가 완료되면 Next 버튼을 클릭한다.



## 02. 파이썬 개발 환경 설정

### ■ Anaconda 설치

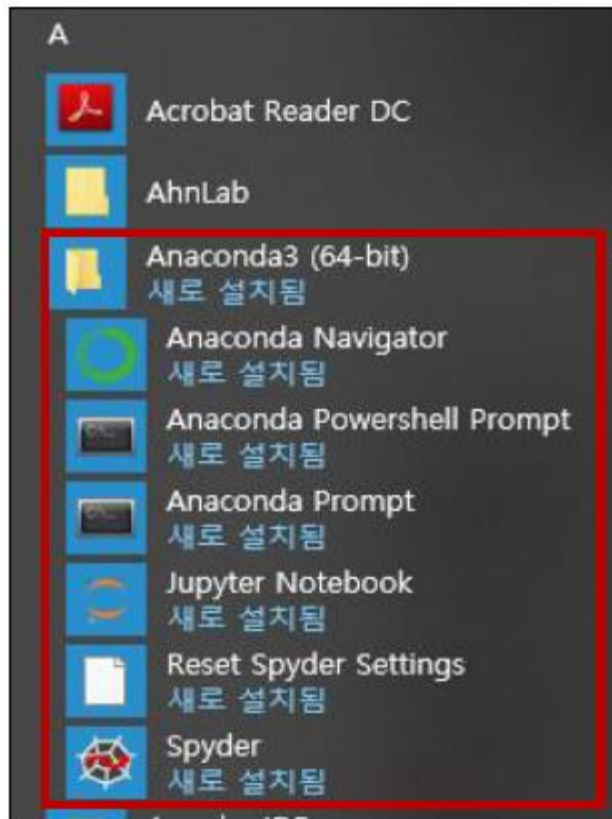
- 다음화면에서 각각 Next, Finish 버튼을 클릭하여 설치를 완료한다/



## 02. 파이썬 개발 환경 설정

### ■ Anaconda 설치

- Anaconda 설치를 모두 끝내면, 윈도우 운영체제의 '시작' 메뉴를 선택하여 Anaconda의 설치상태 확인

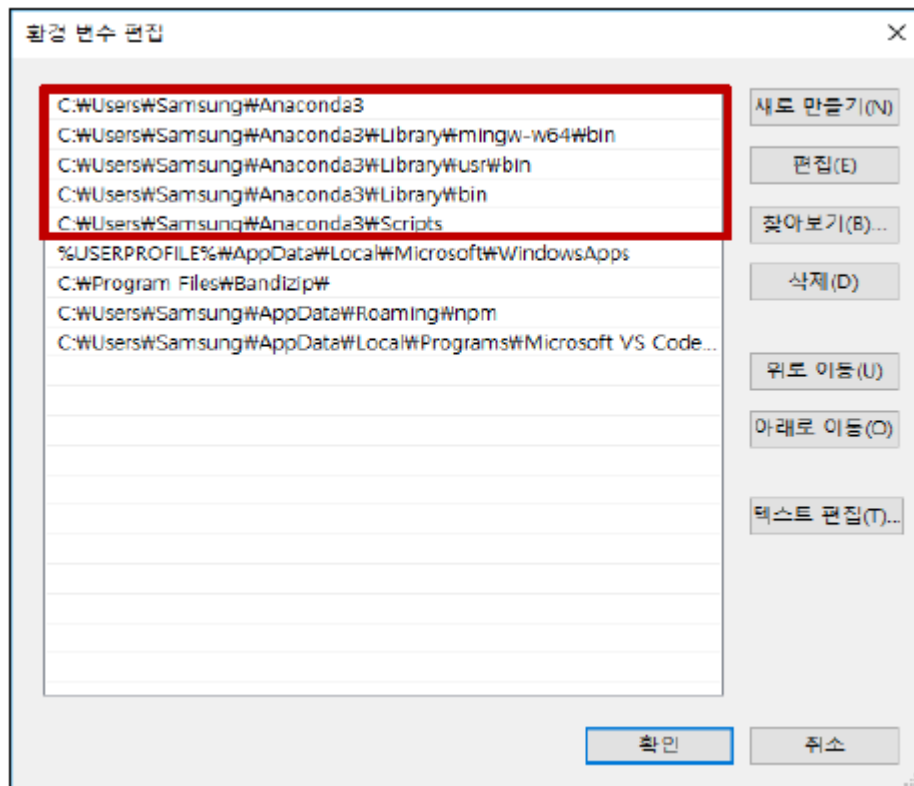




## 02. 파이썬 개발 환경 설정

### ■ Anaconda 설치

- 윈도우 운영체제의 사용자 환경변수 path 정보를 출력한 화면으로 Anaconda의 패스 설정 확인



### 03. 파이썬 개발 툴 소개

#### ■ IDLE

IDLE

- Integrated Development Environment의 약자로 '통합 개발 환경'을 뜻함

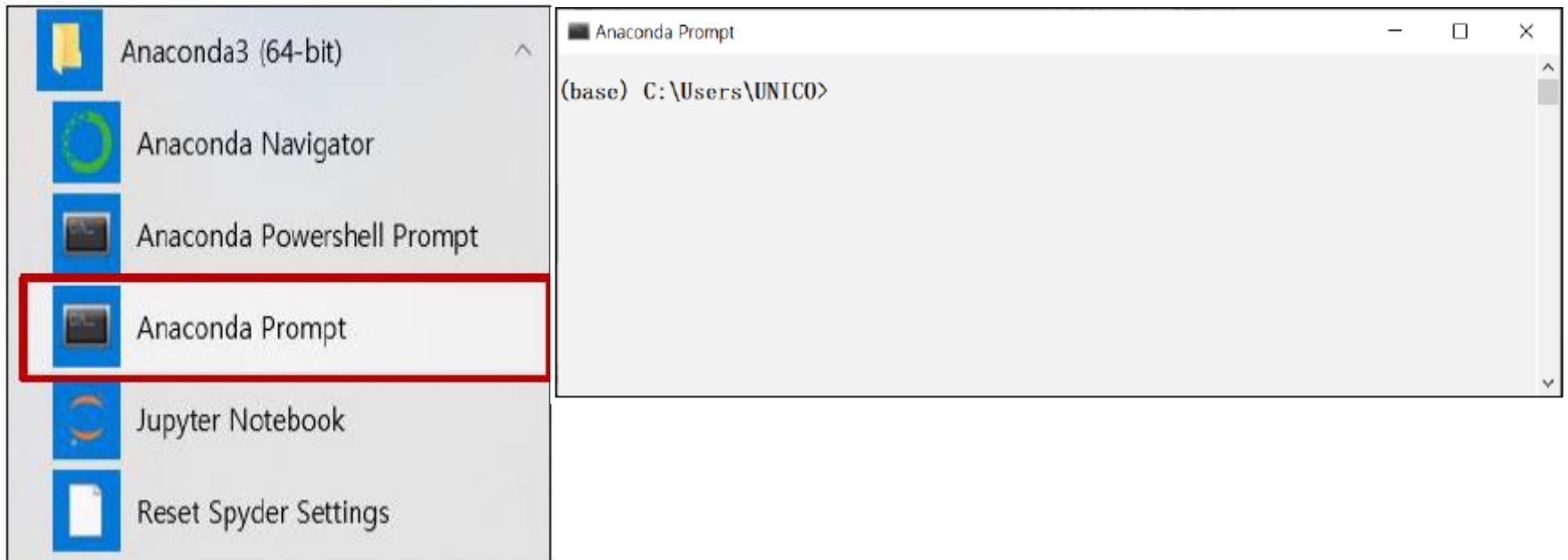
파이썬 셸  
(Python Shell)

코드 편집기

## 03. 파이썬 개발 툴 소개

### ■ IDLE

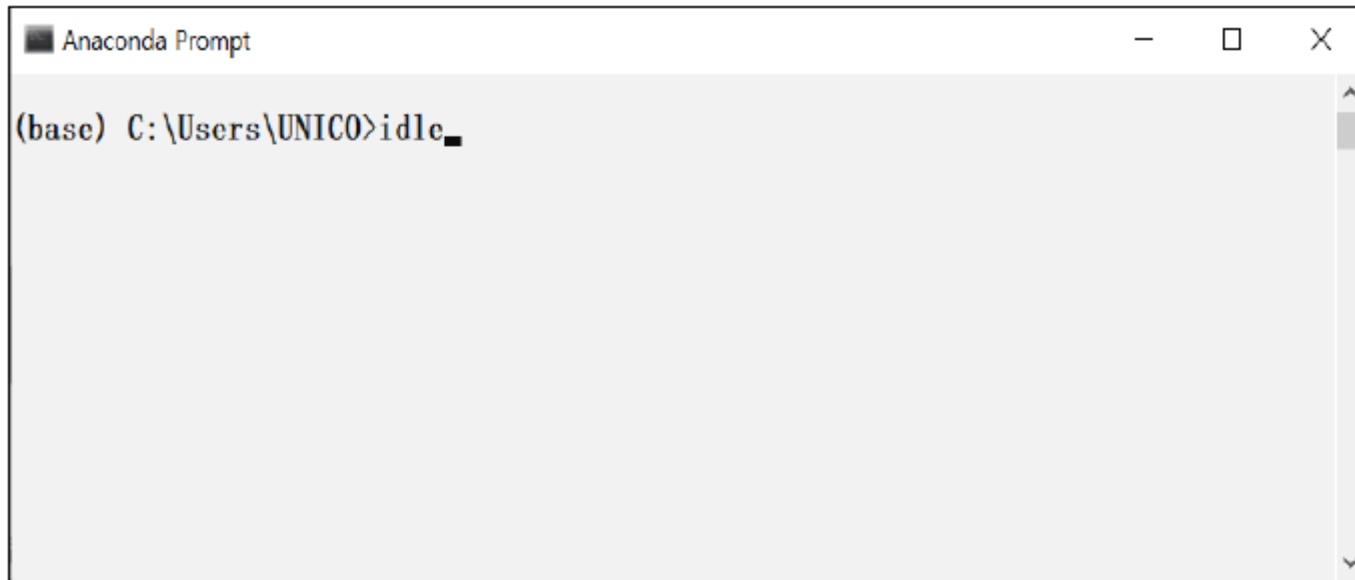
- Anaconda Prompt 창
  - 시작메뉴 -> Anaconda3 메뉴 -> Anaconda Prompt 서버메뉴 클릭



## 03. 파이썬 개발 툴 소개

### ■ IDLE

- Anaconda Prompt 창에서 'idle' 명령어 입력



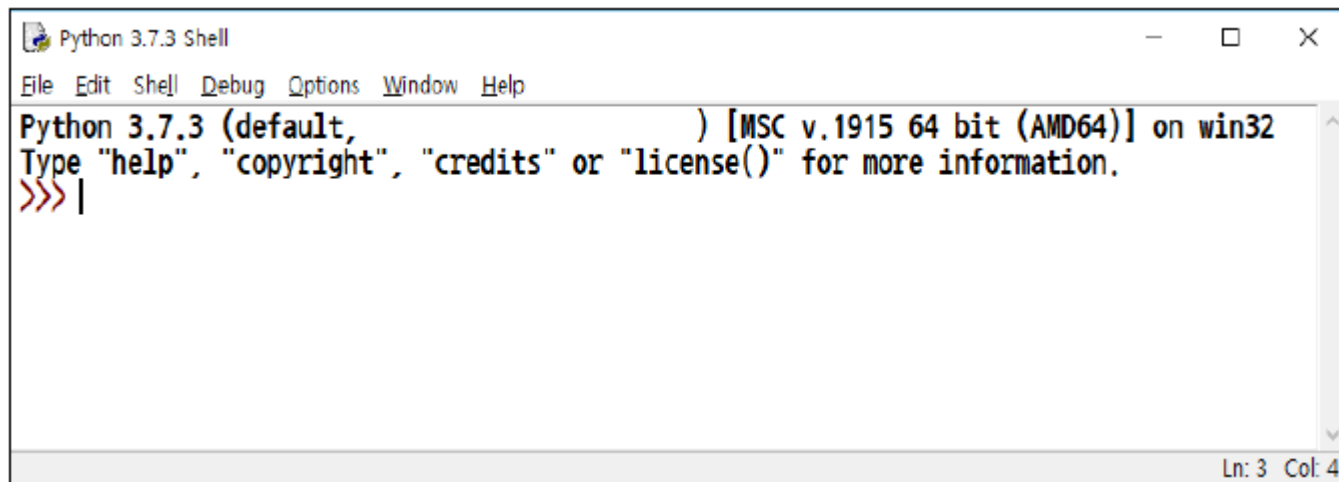
The image shows a screenshot of the Anaconda Prompt window. The title bar at the top reads "Anaconda Prompt". The command prompt shows the current directory as "C:\Users\UNICO" and the prompt character ">". The command "idle" has been entered, followed by a cursor. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Anaconda Prompt
(base) C:\Users\UNICO>idle
```

## 03. 파이썬 개발 툴 소개

### ■ IDLE

- IDLE 윈도우 창 출력
  - '>>>'는 프롬프트(Prompt)라고함
  - 프롬프트에 파이썬 문법에 맞는 코드를 작성하고 엔터키를 누르면 파이썬 인터프리터가 입력된 코드를 해석해 실제 동작 수행

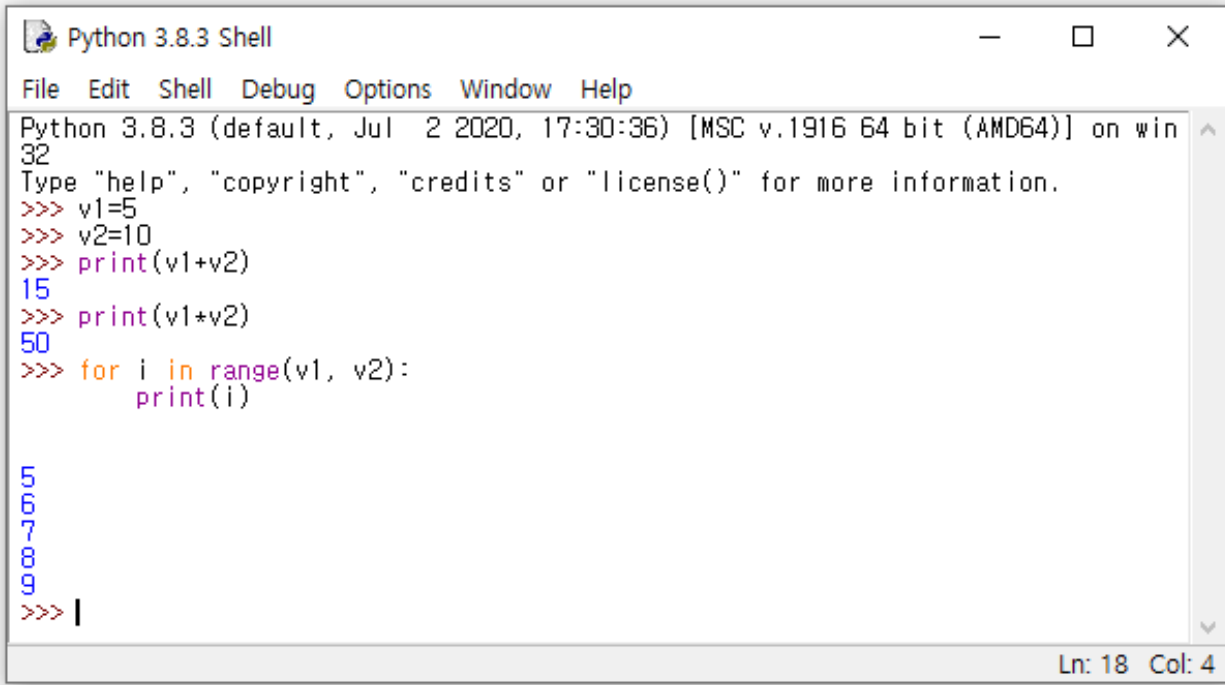


```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, ) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

## 03. 파이썬 개발 툴 소개

### ■ IDLE

- IDLE 창에서 파이썬 코드 실행
  - 코드를 작성하고 엔터키를 누를 때마다 해당 코드를 수행 시키는 방식으로 대화식 프로그래밍 가능



The screenshot shows a window titled "Python 3.8.3 Shell" with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The text area contains the following text:

```
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>> v1=5
>>> v2=10
>>> print(v1+v2)
15
>>> print(v1+v2)
50
>>> for i in range(v1, v2):
    print(i)

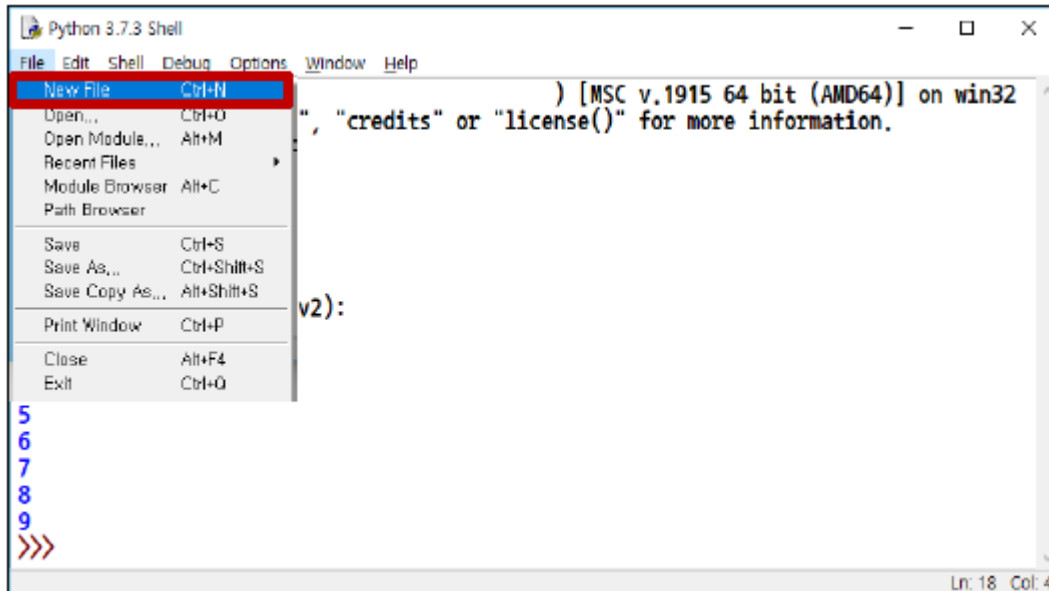
5
6
7
8
9
>>> |
```

The status bar at the bottom right indicates "Ln: 18 Col: 4".

## 03. 파이썬 개발 툴 소개

### ■ IDLE

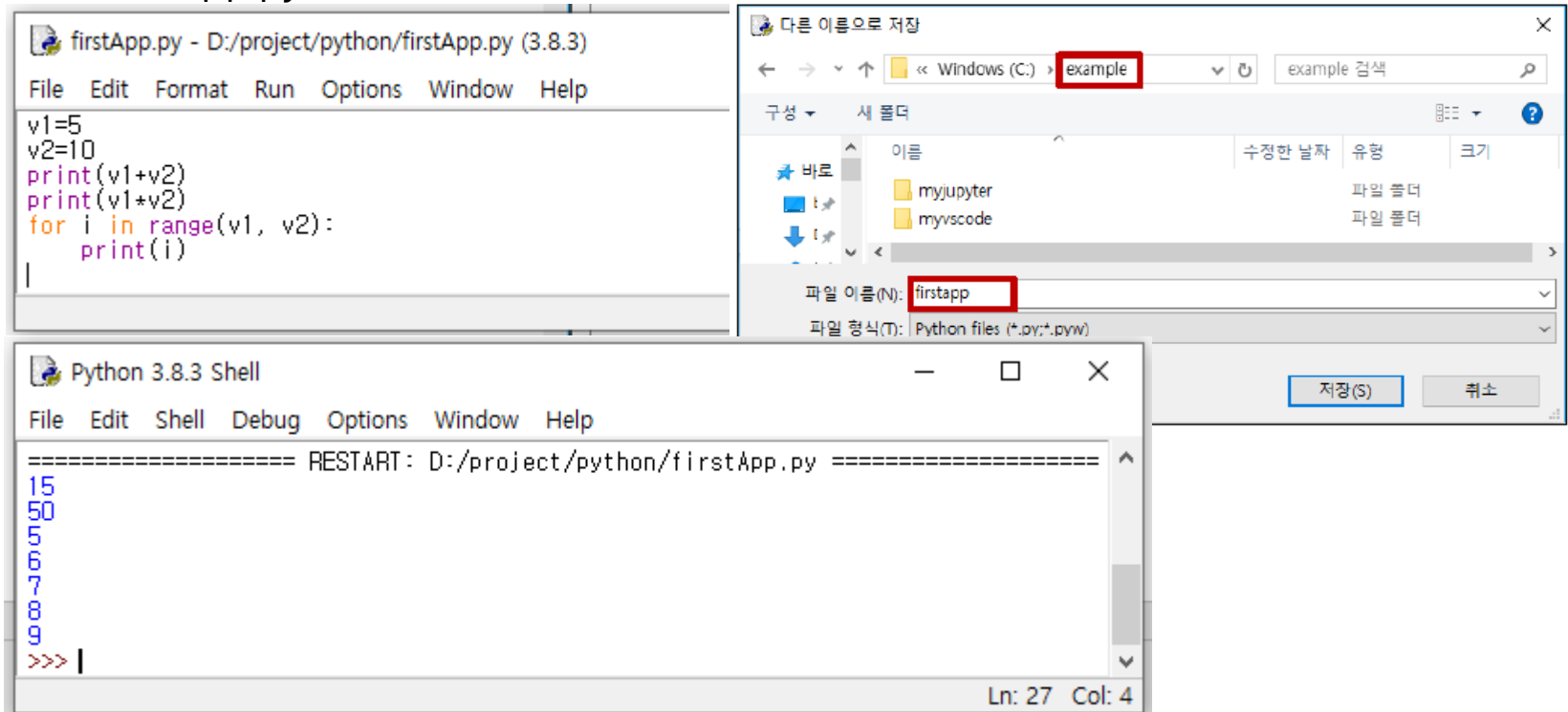
- IDLE 창에서 코드 수행
  - 파이썬 코드를 입력하는 코드 단위 수행, 파이썬 파일 생성 후 수행도 가능
  - File 메뉴의 New File 메뉴선택



## 03. 파이썬 개발 툴 소개

### ■ IDLE

- IDLE 창에서 파이썬 코드 실행
  - 소스 편집기 창이 독립적으로 출력되며, 하나의 모듈로 구성하려는 파이썬 코드 작성
  - example 폴더에 firstapp 명칭으로 저장
  - firstapp.py 라는 파일명으로 파일이 생성됨





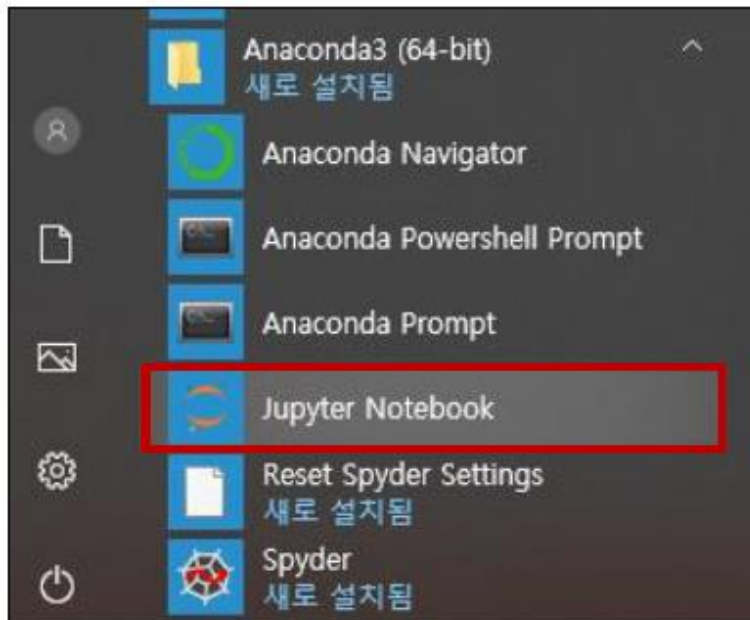
## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

- 크롬과 같은 웹브라우저에서 코드를 작성하고 실행이 가능한 툴

### ■ Jupyter Notebook 실행

- Anaconda에는 Jupyter Notebook이 포함되어 있으므로 '시작' 메뉴를 클릭하여 수행 시작
- Jupyter Notebook 실행을 시작하는 윈도우 출력



```
Jupyter Notebook
[I 04:20:36.008 NotebookApp] JupyterLab application directory is C:\Users\Samsung\Anaconda3\share\jupyter\lab
[I 04:20:36.011 NotebookApp] Serving notebooks from local directory: C:\Users\Samsung
[I 04:20:36.011 NotebookApp] The Jupyter Notebook is running at:
[I 04:20:36.011 NotebookApp] http://localhost:8888/?token=eae5c62cf28c90d84d70bda78d8eae467b66e81e2761f715
[I 04:20:36.012 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 04:20:36.130 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/Samsung/AppData/Roaming/jupyter/runtime/nbserver-16064-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=eae5c62cf28c90d84d70bda78d8eae467b66e81e2761f715
```

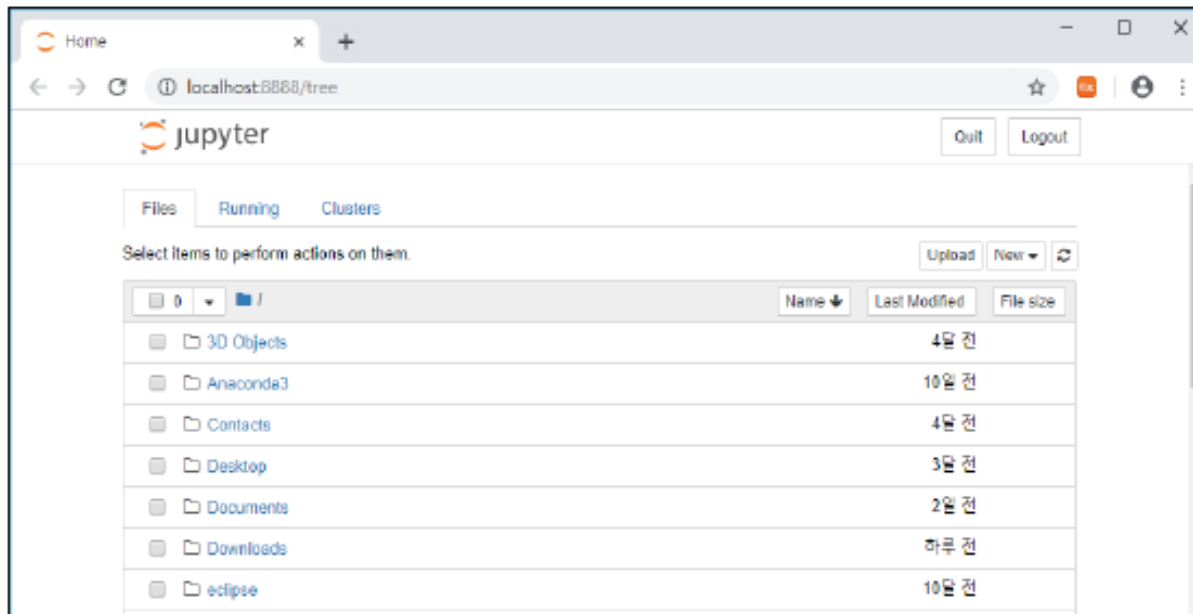
## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ Jupyter Notebook의 작업 페이지

- 브라우저가 실행되고 Jupyter Notebook의 작업 페이지 출력

- Jupyter Notebook 디폴트로 인식하는 시작 디렉토리가 시스템의 사용자 디렉토리이므로 `c:/example/myjupyter`로 변경 필요



## 03. 파이썬 개발 툴 소개

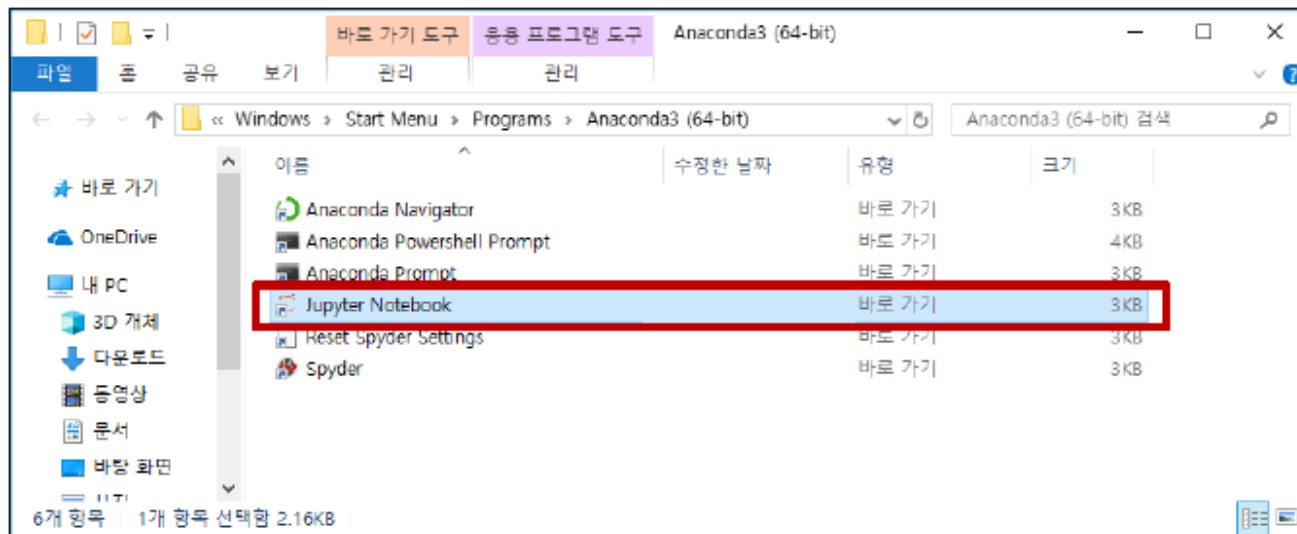
### ■ Jupyter Notebook

#### ■ Jupyter Notebook의 작업 페이지

- C:\Users\사용자 NAME

#### ■ Jupyter Notebook의 시작 디렉토리 변경

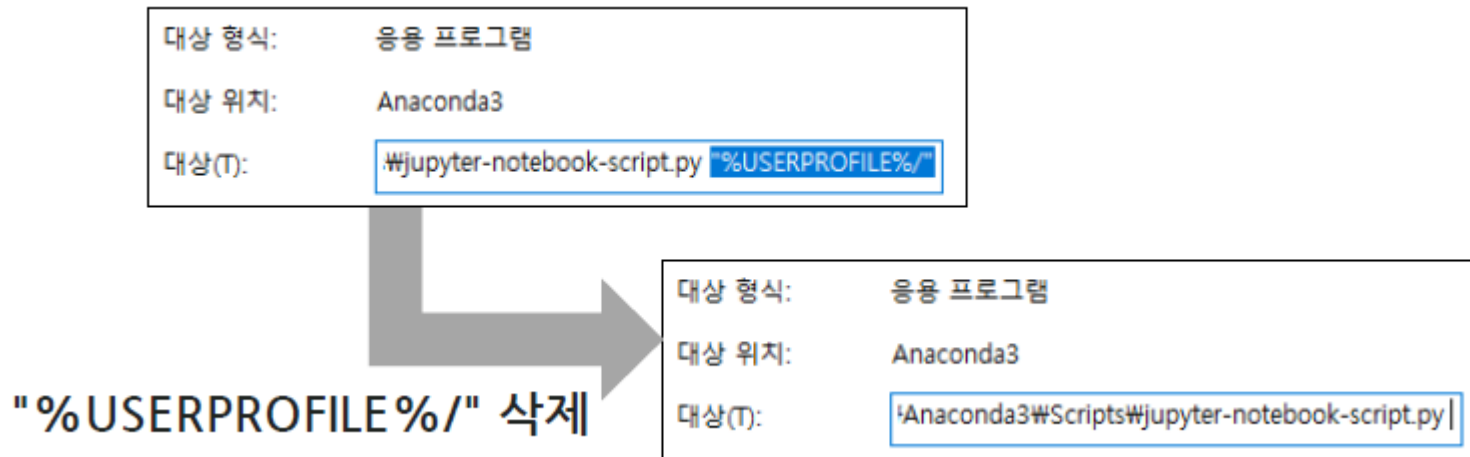
- 시작 메뉴에서 Jupyter Notebook 선택
- Jupyter Notebook의 설치 디렉토리 파악 후 해당 디렉토리로 이동



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

- Jupyter Notebook의 시작 디렉토리 변경
  - Jupyter Notebook 파일의 속성 정보에 들어가 대상 정보 수정



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

- Jupyter Notebook의 시작 디렉토리 변경
  - cmd 창에서 다음 명령을 수행시켜 Jupyter Notebook의 설정파일 생성

jupyter notebook --generate-config

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.706]
(c)  Microsoft Corporation. All rights reserved.

C:\Users\Samsung>jupyter notebook --generate-config
Writing default config to: C:\Users\Samsung\.jupyter\jupyter_notebook_config.py

C:\Users\Samsung>cd .jupyter

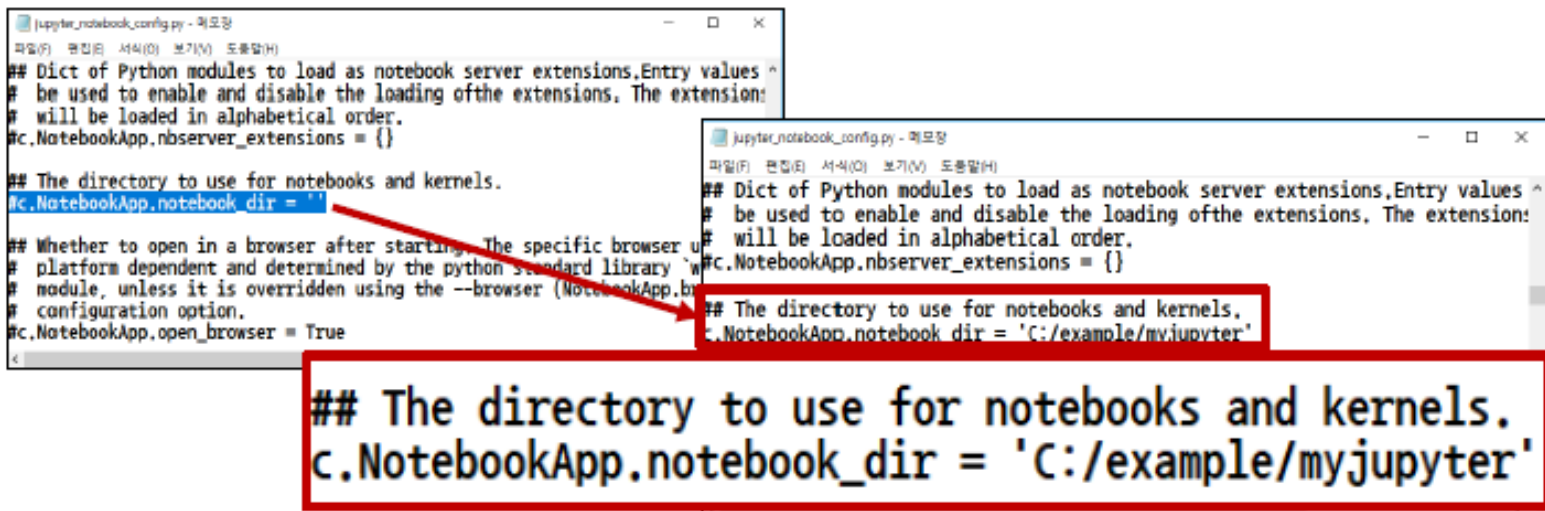
C:\Users\Samsung\.jupyter>
```

## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ Jupyter Notebook의 시작 디렉토리 변경

- jupyter\_notebook\_config.py 파일에서 notebook\_dir 항목을 찾아서 주석을 해제 하고 'C:/example/myjupyter'작성



```
## Dict of Python modules to load as notebook server extensions. Entry values ^  
# be used to enable and disable the loading of the extensions. The extensions  
# will be loaded in alphabetical order.  
#c.NotebookApp.nbserver_extensions = {}  
  
## The directory to use for notebooks and kernels.  
#c.NotebookApp.notebook_dir = ''  
  
## Whether to open in a browser after starting. The specific browser u  
# platform dependent and determined by the python standard library 'w  
# module, unless it is overridden using the --browser (NotebookApp.br  
# configuration option.  
#c.NotebookApp.open_browser = True
```

```
## Dict of Python modules to load as notebook server extensions. Entry values ^  
# be used to enable and disable the loading of the extensions. The extension:  
# will be loaded in alphabetical order.  
#c.NotebookApp.nbserver_extensions = {}  
  
## The directory to use for notebooks and kernels.  
#c.NotebookApp.notebook_dir = 'C:/example/myjupyter'
```

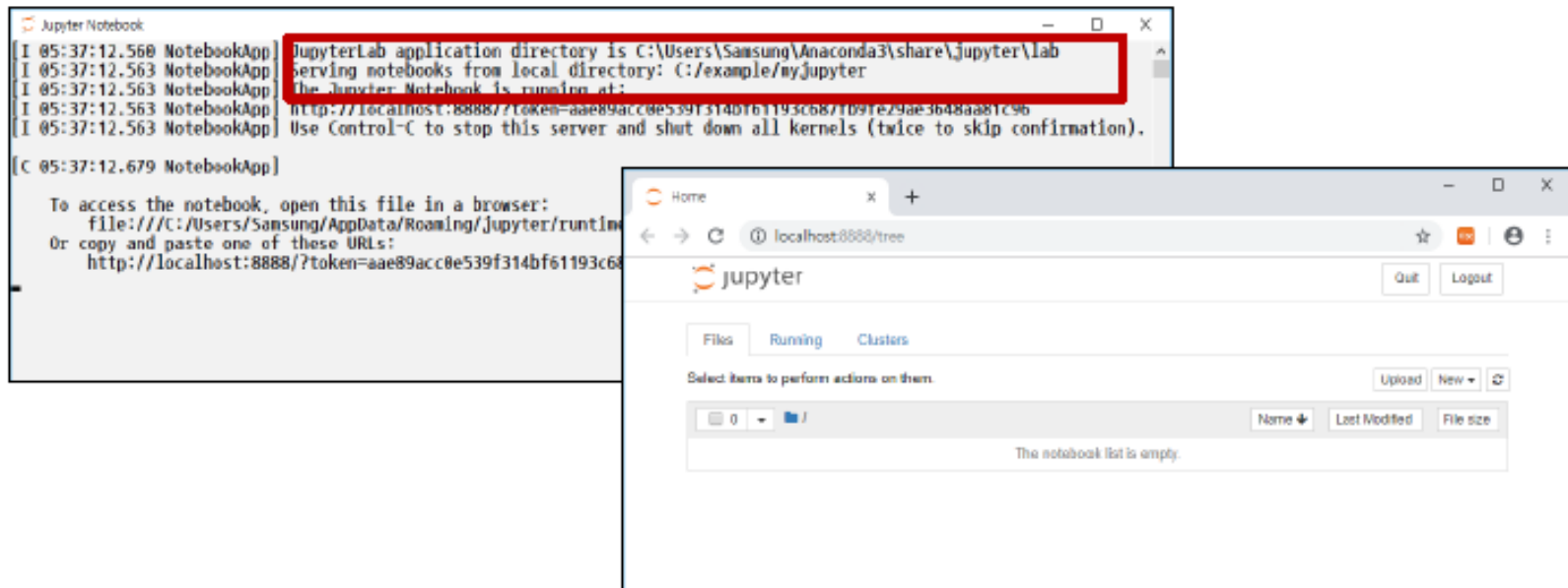
```
## The directory to use for notebooks and kernels.  
c.NotebookApp.notebook_dir = 'C:/example/myjupyter'
```

## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ Jupyter Notebook의 작업 페이지

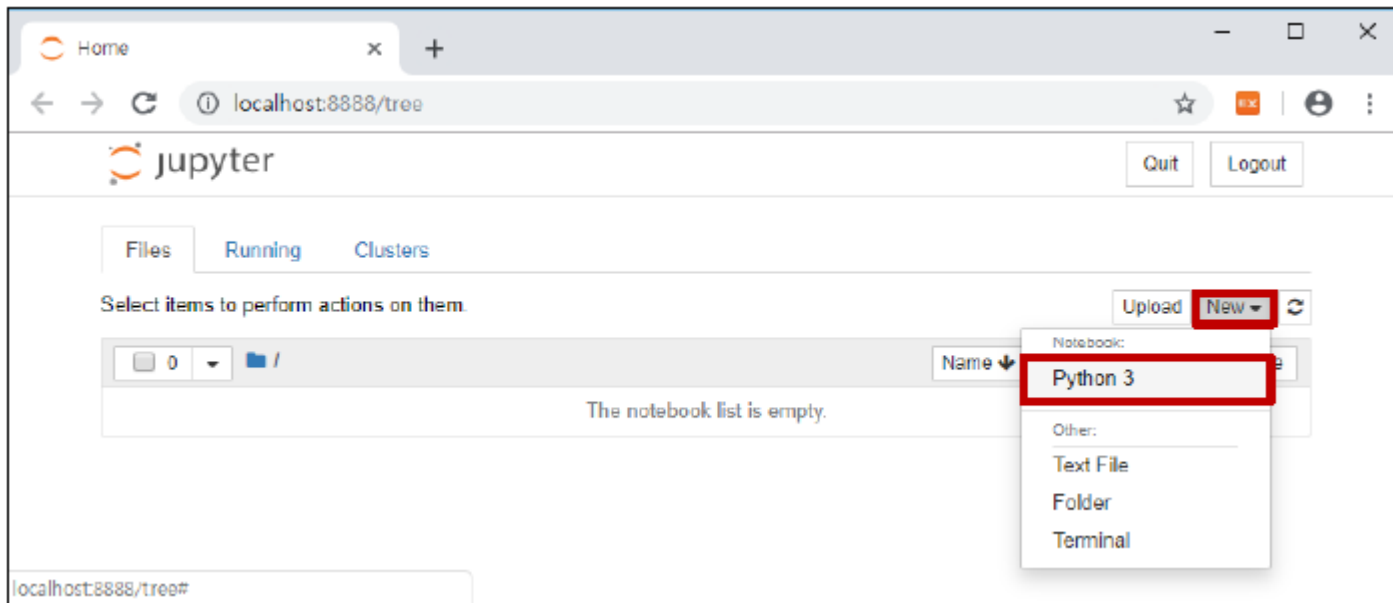
- Jupyter Notebook을 재기동하면 시작 디렉토리가 변경된 것을 볼 수 있음



# 파이썬 개발 툴 소개

## ■ Jupyter Notebook

- 파이썬 코드 테스트
  - 우측에 있는 'New'를 누르고 Python 3 클릭

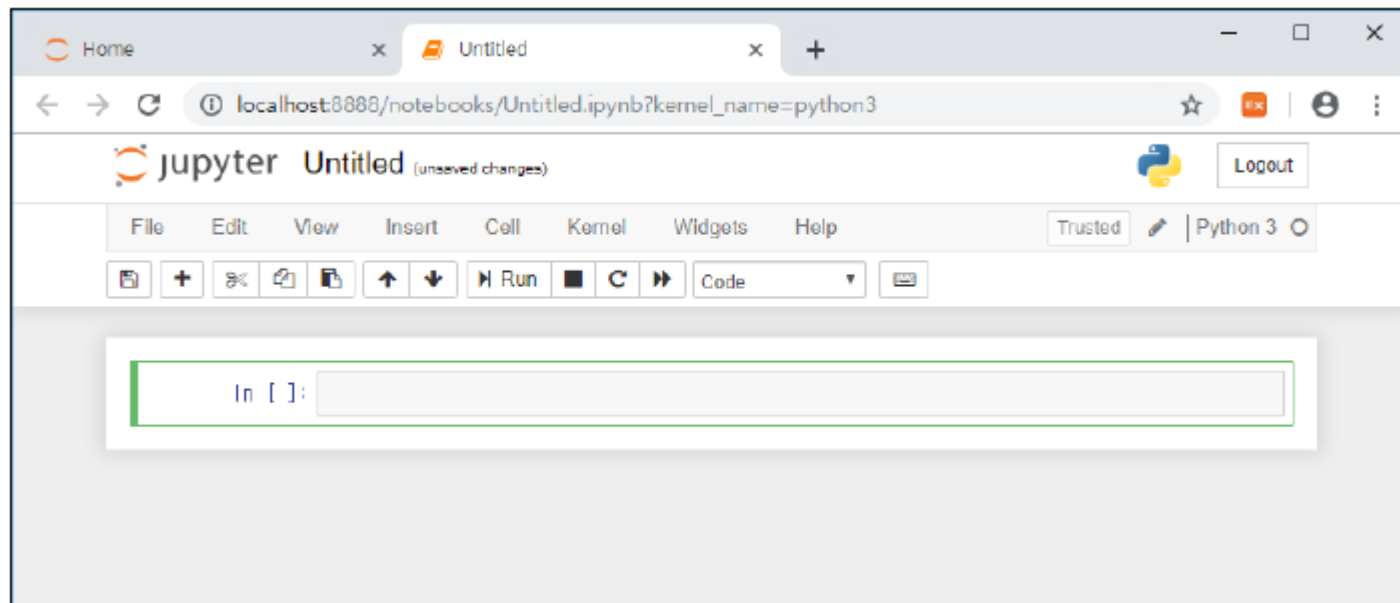




## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

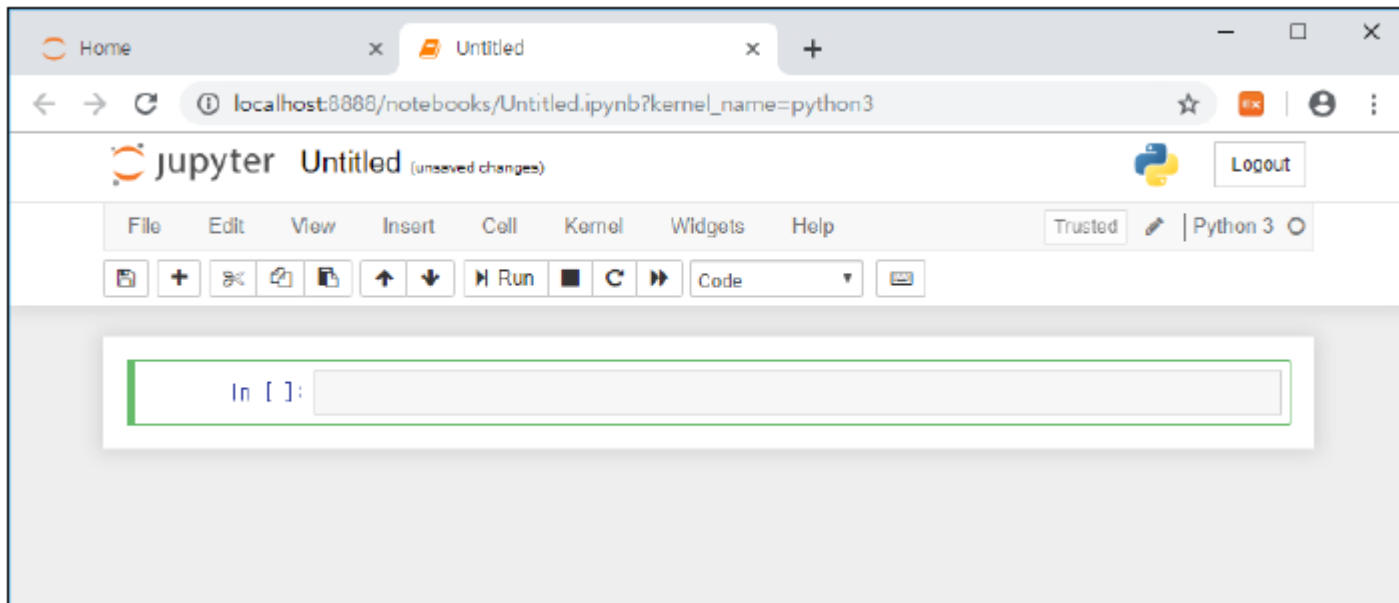
- 파이썬 코드 테스트
  - **Untitled**라는 이름의 새로운 창이 출력되고 파이썬 코드를 입력할 수 있는 박스 출력



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

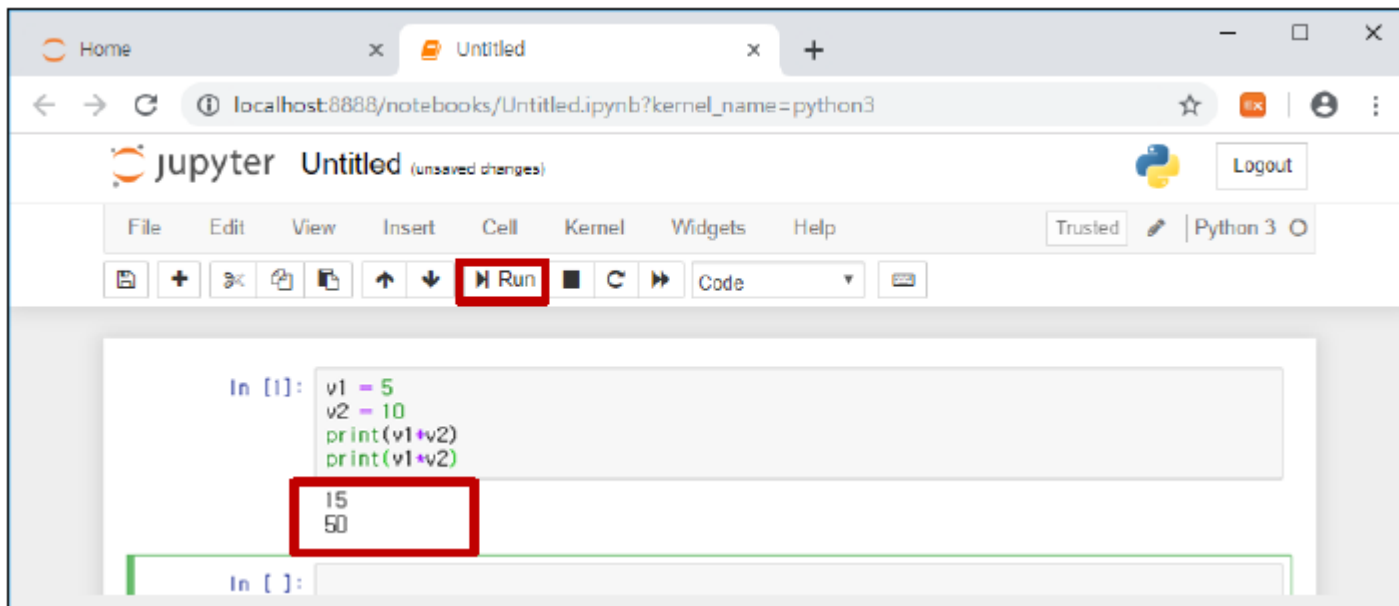
- 파이썬 코드 테스트
  - **Untitled**라는 이름의 새로운 창이 출력되고 파이썬 코드를 입력할 수 있는 박스 출력



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

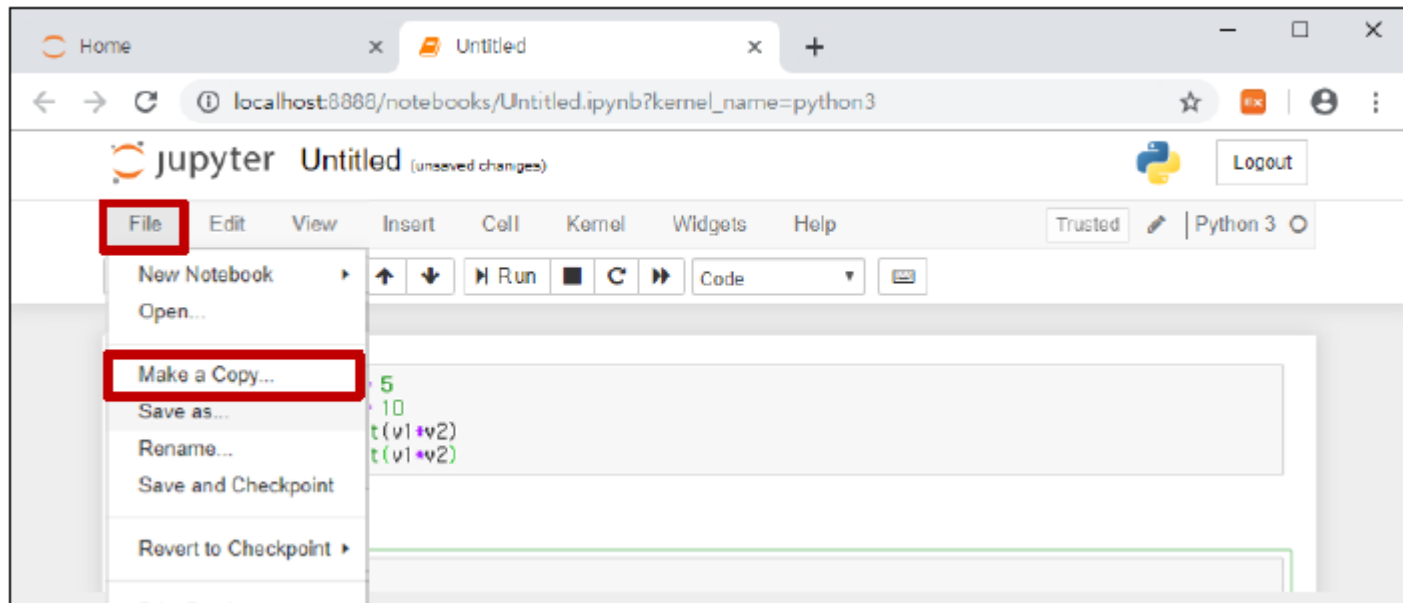
- 파이썬 코드 테스트
  - 박스안에 적당한 파이썬 코드를 작성하고 위에 있는 'Run' 버튼을 클릭하면 박스 아래에 수행 결과 출력



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

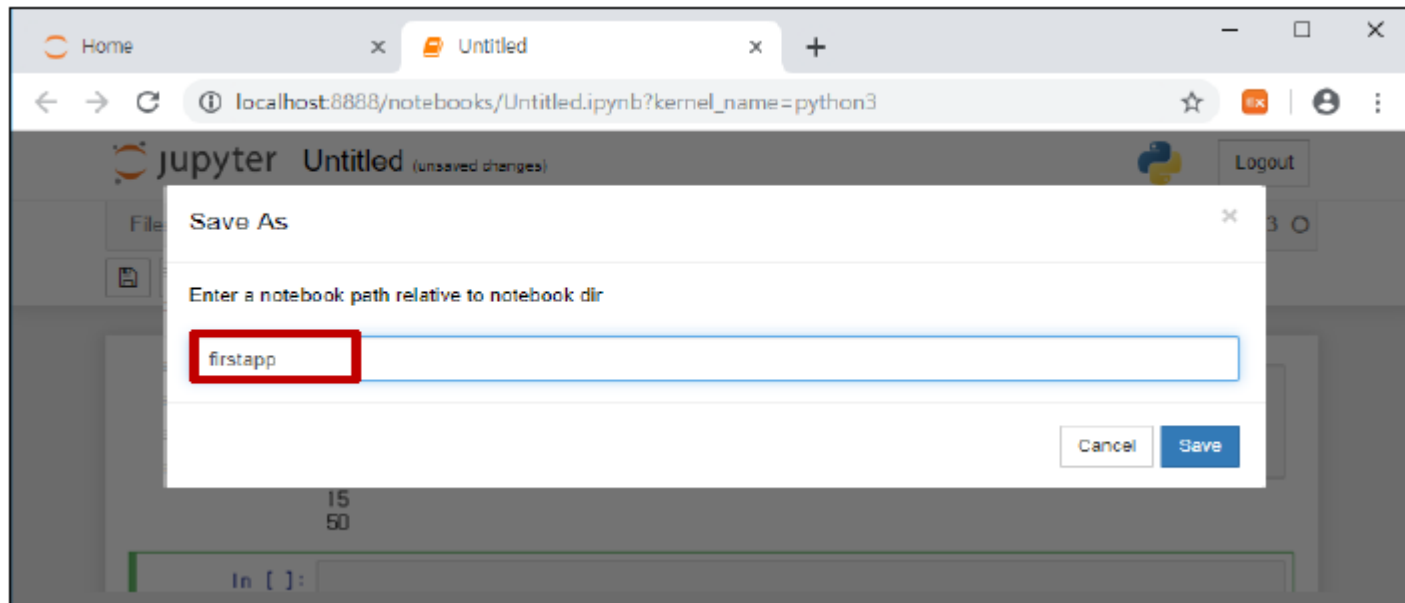
- 파이썬 코드 테스트
  - 작성한 파이썬 코드를 파일로 저장하려면 File 메뉴를 선택하고 'Save as...' 메뉴 클릭



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

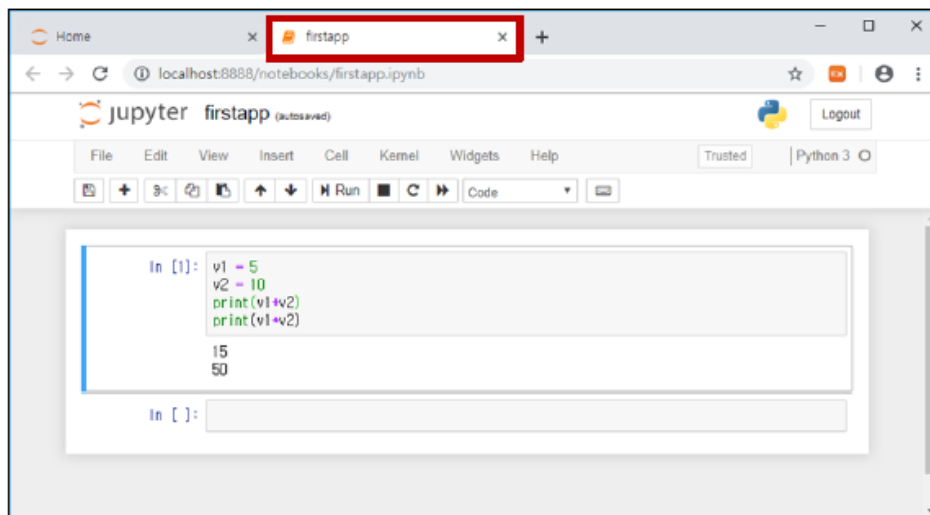
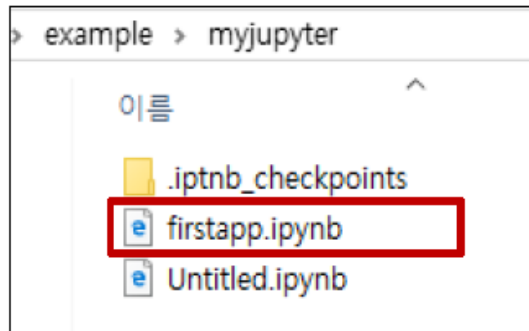
- 파이썬 코드 테스트
  - 파일명입력 : firstapp 입력



## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

- 파이썬 코드 테스트
  - **firstapp.ipynb**라는 파일 생성, 창의 이름도 **firstapp** 으로 변경됨

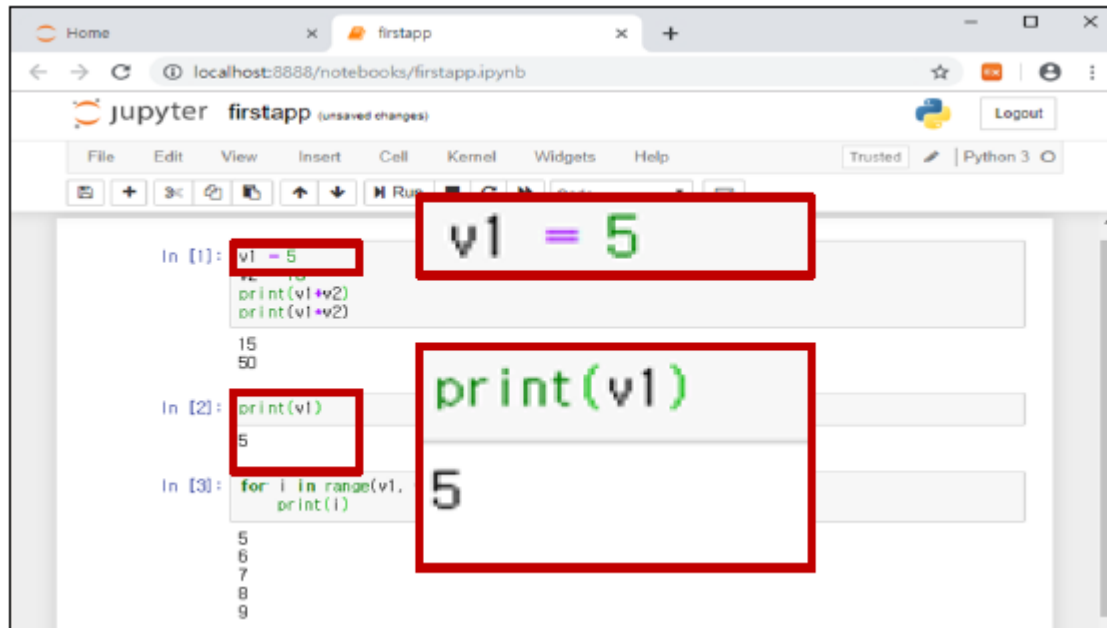


## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ 파이썬 소스 코드 출력

- 파이썬 소스 코드를 입력하는 각각의 박스들은 하나의 파일내에서 계속 이어짐
- 위의 박스에서 선언한 v1 변수를 아래 박스들에서도 사용 가능



```
In [1]: v1 = 5
        v2 = 10
        print(v1+v2)
        print(v1*v2)
15
50

In [2]: print(v1)
5

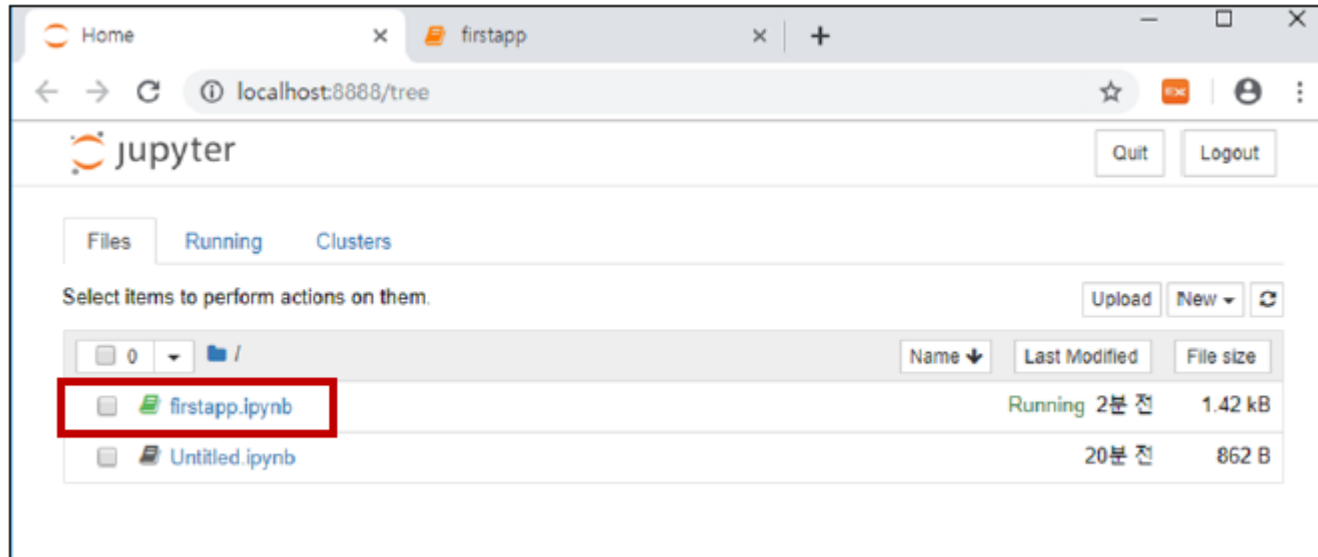
In [3]: for i in range(v1,
        print(i)
5
6
7
8
9
```

## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ 파이썬 소스 코드 출력

- Jupyter Notebook를 재가동하고 <http://localhost:8888/tree>으로 요청
- 시작 디렉토리에 저장된 firstapp.ipynb 파일이리스트로 출력되는 것을 볼 수 있음



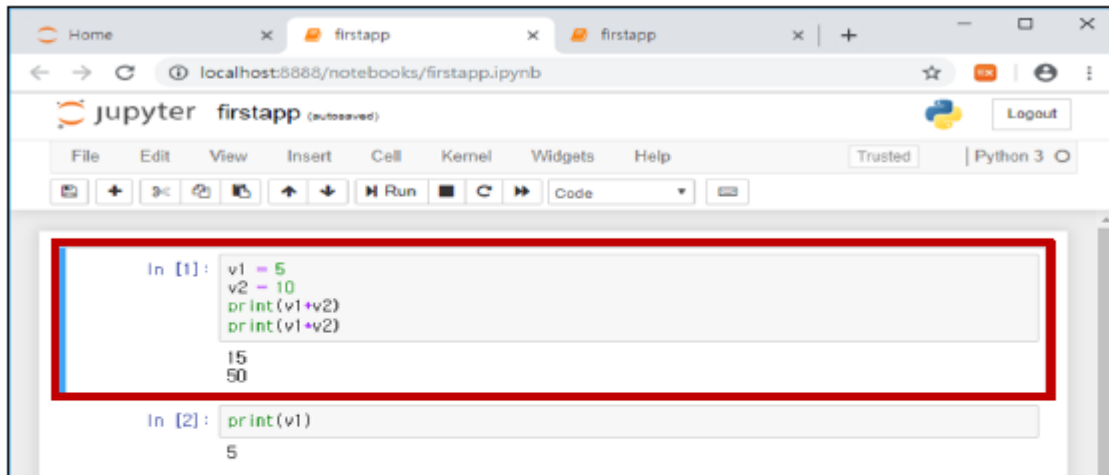


## 03. 파이썬 개발 툴 소개

### ■ Jupyter Notebook

#### ■ 파이썬 소스 코드 출력

- 리스트에서 firstapp.ipynb 파일을 클릭하면 새로운 창이 출력
- 마지막으로 저장된 소스 코드의 내용이 출력되는 것을 볼 수 있음



```
In [1]: v1 = 5
        v2 = 10
        print(v1+v2)
        print(v1*v2)

15
50

In [2]: print(v1)

5
```

```
In [1]: v1 = 5
        v2 = 10
        print(v1+v2)
        print(v1*v2)

15
50
```

## 03. 파이썬 개발 툴 소개

### ■ Jupyter lab

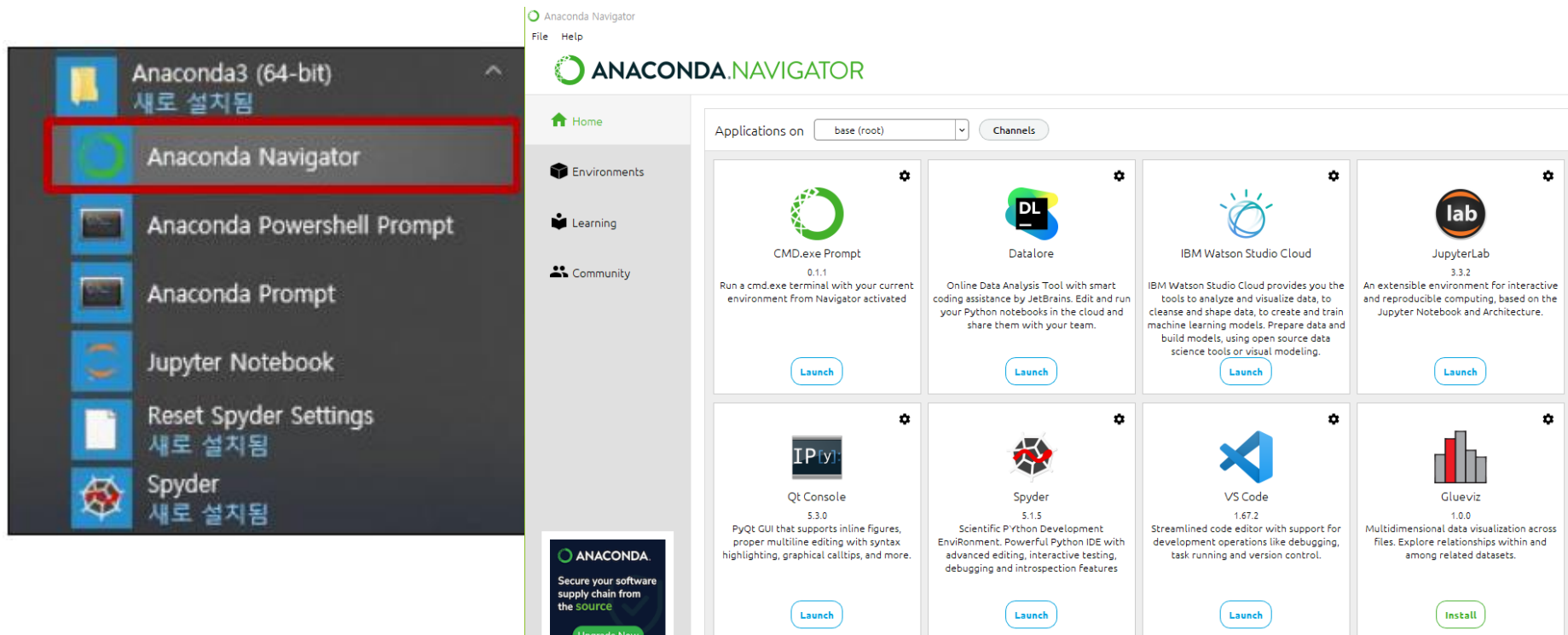
- 주피터 노트북을 개선한 것으로 기존 주피터 노트북에 다중 탭 지원, 수려한 UI, 다양한 확장 기능 등을 추가해 만든 웹 어플리케이션.
- 장점
  - 셀(Cell) 단위 코드 실행
  - 대화형(interactive) 개발 환경
  - 문서화의 용이성
  - 웹 기반의 개발 환경

## 03. 파이썬 개발 툴 소개

### ■ Jupyter lab

#### ■ Jupyter lab 실행

- 시작 메뉴에서 Anaconda Navigator를 선택하면 ANACONDA NAVIGATOR 창 출력

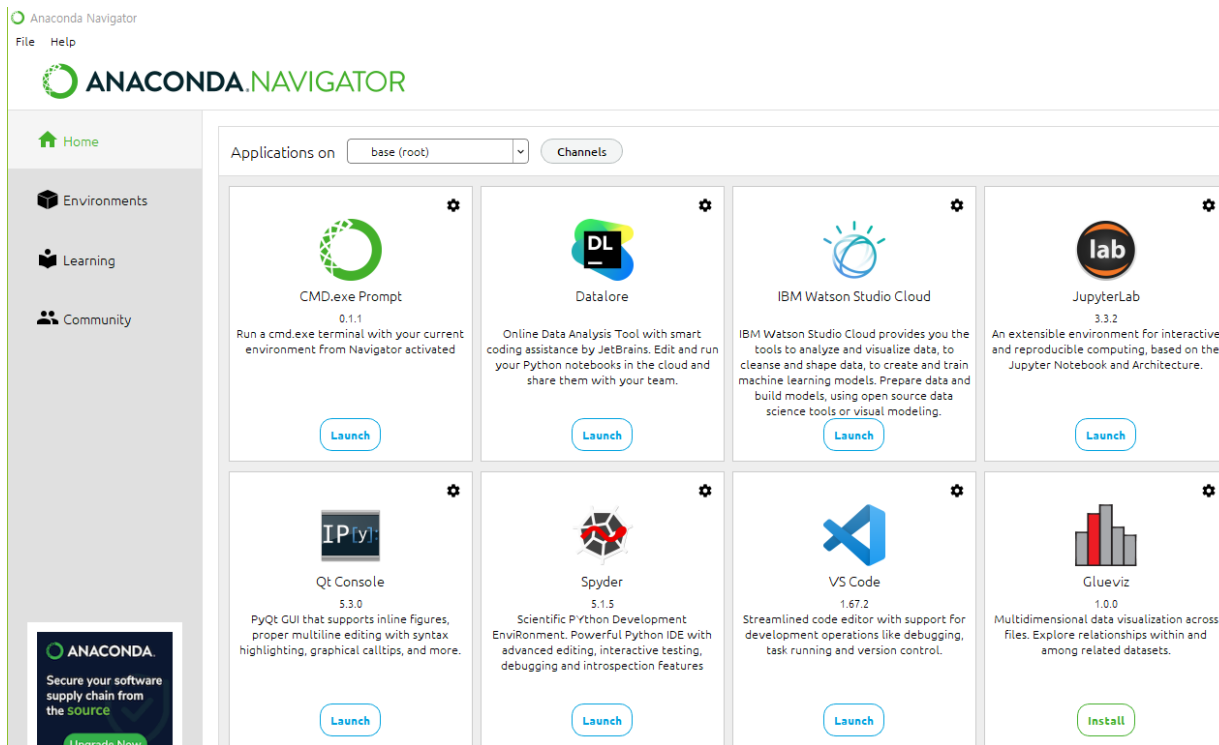


## 03. 파이썬 개발 툴 소개

### ■ Jupyter lab

#### ■ Jupyter lab 실행

- Anaconda Navigator 창의 Jupyter lab의 Launch 버튼을 클릭
- 자동적으로 브라우저가 오픈되면서 주피터랩이 실행됨
- 또는 consol에서 [jupyter lab]을 입력하면 실행됨

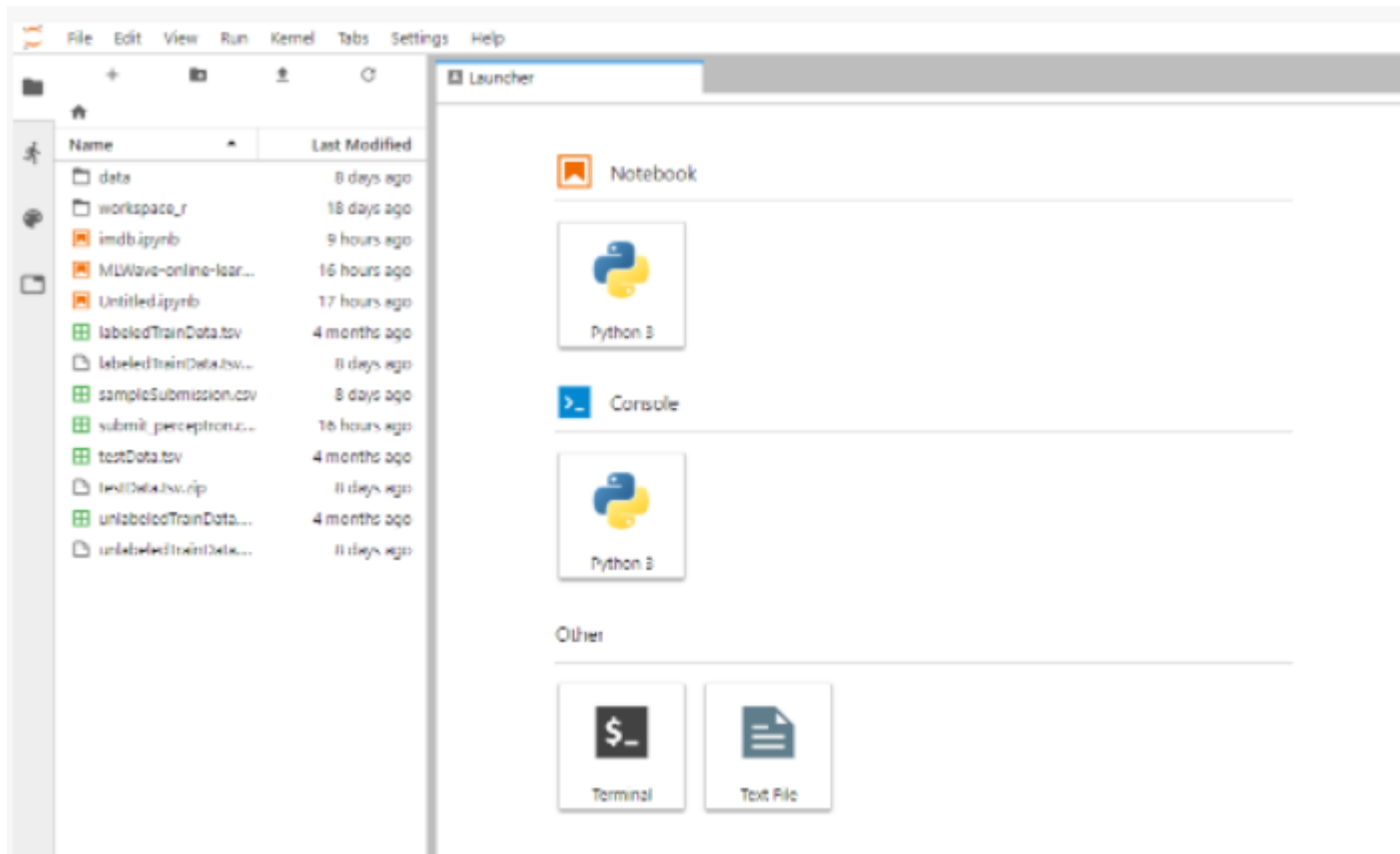


## 03. 파이썬 개발 툴 소개

### ■ Jupyter lab

#### ▪ Jupyter lab 실행

- 기본적으로 8888포트로 오픈되지만 8888이 사용되고 있으면 포트번호는 하나씩 자동 증가(기본주소 :<http://localhost:8888/lab>)
- Notebook, Console 중 노트북 선택

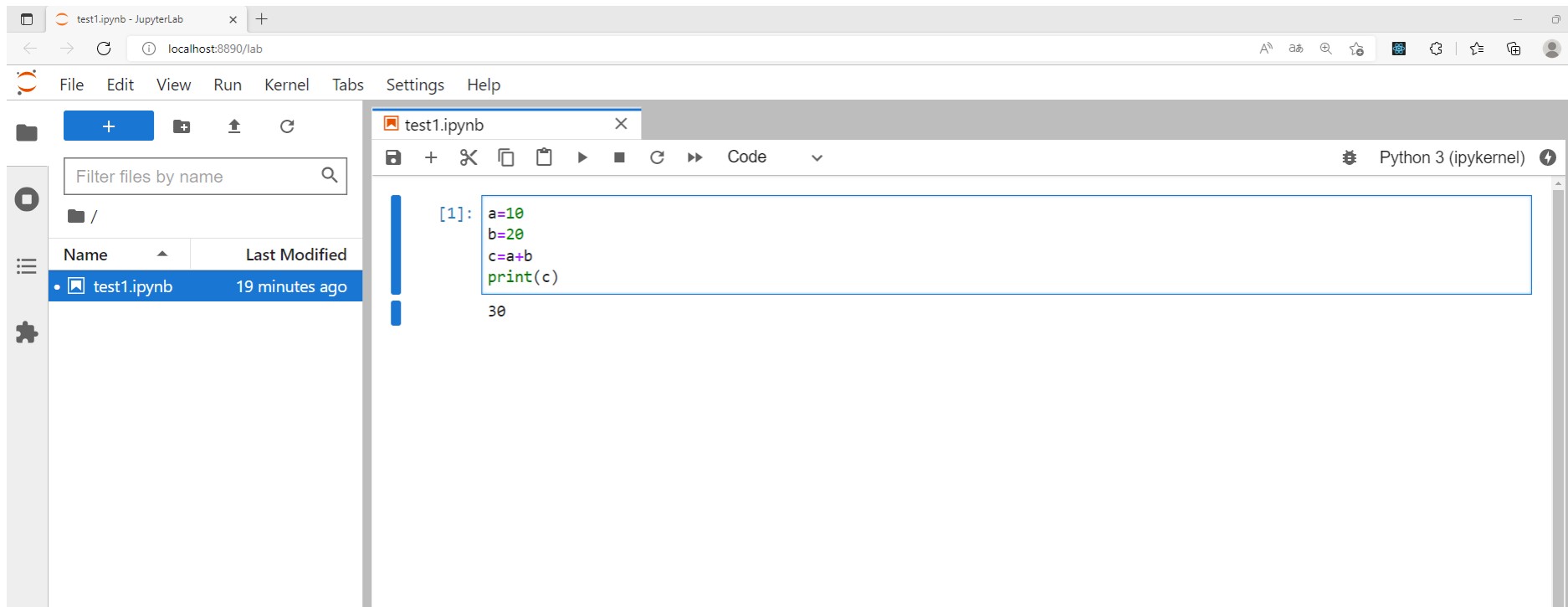


## 03. 파이썬 개발 툴 소개

### ■ VS Code

#### ■ VS Code 실행

- 다음의 코드를 입력



## 03. 파이썬 개발 툴 소개

### ▪ Jupyter lab 단축키

- **ESC** 를 눌러 커맨드 모드로 진입하여 **ENTER**를 통해 cell 을 수정할 수 있습니다. 아래 커맨드는 커맨드 모드에서 동작합니다.
- **A** 는 현재 cell 위에 새로운 cell 을 추가합니다.
- **B** 는 현재 cell 밑에 새로운 cell 을 추가합니다.
- **D + D** D를 연속해서 두 번 누르면 현재 cell 을 삭제합니다.
- **M** 은 Markdown 셀로 변환하며, **Y** 는 Code 셀로 변환하고 **R** 은 Raw Cell 로 변환합니다.
- **CTRL + B** 화면을 더 크게 사용할 수 있습니다. 왼쪽 파일 탐색기가 사라집니다.
- **SHIFT + M** 두 개의 셀을 한개의 셀로 Merge 합니다.
- **CTRL + SHIFT + -** 현재 커서 위치를 기준으로 두 개의 셀로 구분합니다.
- **SHIFT+J** or **SHIFT + DOWN** 현재 셀에서 아래쪽 위치로 새로운 셀을 같이 선택합니다.
- **SHIFT + K** or **SHIFT + UP** 현재 셀에서 위쪽 위치로 새로운 셀을 같이 선택합니다.
- **CTRL + /** 선택한 코드를 주석처리합니다.