

1부 교육목표 체크리스트

1부의 내용을 공부한 전과 후에 표에 있는 항목들을 체크해 보세요. 체크표시가 늘어날수록 여러분의 실력도 늘어납니다

교육목표		✓
1	파이썬 개발 환경을 구축하고 코드를 작성해서 테스트할 수 있다.	
2	파이썬 언어의 특징을 3가지 이상 설명할 수 있다.	
3	변수를 선언하고 사용할 수 있다.	
4	파이썬 언어의 연산자를 사용하여 계산을 하고 그 결과를 변수에 대입할 수 있다.	
5	파이썬 언어의 자료형(정수, 실수, 논리, 문자)을 구분하고 설명할 수 있다.	
6	파이썬 언어의 명령문(if, for 등 제어문)의 흐름을 이해하고 기본 구문을 작성할 수 있다.	
7	매개변수 선언이 없는 함수를 정의하고 호출할 수 있다.	
8	매개변수, 지역변수, 전역변수를 구분할 수 있다.	
9	값에 의한 호출과 참조에 의한 호출을 구분할 수 있다.	
10	함수의 매개변수 종류를 이해하고 함수 정의 시 활용할 수 있다.	

학습 내용

The Python - 1부. 프로그래밍 언어 기본



1장. 파이썬 개요 및 개발환경 구성

2장. 자료형과 연산자

3장. 데이터 구조

4장. 제어문

5장. 함수

파이썬 개요

- 파이썬은?
- 파이썬 용도
- 파이썬 언어 특징
- 컴파일러 언어와 인터프리터 언어

파이썬 인터프리터

- 다운로드
- 설치 및 확인
- Hello World 출력하기

아나콘다를 이용한 개발환경

- 파이썬 배포판
- 아나콘다
- 아나콘다 네비게이터
- 주피터 노트북
- 주피터 노트북에서 소스코드 작성 및 실행
- 스파이더에서 코드 작성 및 실행

1.1. 파이썬

1절. 파이썬 개요

- 1989년 개발 시작, 1991년 발표
2000년 Python 2, 2008년 Python 3
- 언어의 기본 기능은 작게
부가 기능은 라이브러리 또는 패키지에 의해 제공
- 교육용 언어, 쉬운 코드 작성 및 실행
- 무료
- 공식 사이트 : <http://www.python.org/>

<https://www.tiobe.com/tiobe-index>

1.2. 파이썬 용도

1절. 파이썬 개요

파이썬으로 할 수 있는 것

- 알고리즘 코딩 공부
- 윈도우 응용프로그램 개발
- 웹 프로그래밍
- 통계 및 수치해석
- 데이터 탐색 및 시각화
- IoT 프로그래밍
- 기계학습을 이용한 데이터 분석
- 영상인식 프로그램
- 인공지능 딥러닝

파이썬으로 할 수 없는 것

- 시스템 프로그래밍
- 모바일 프로그래밍

1.3. 파이썬 언어 특징

1절. 파이썬 개요

- 가독성 - 문법이 간결하고 **들여쓰기를 기반으로** 가독성이 좋음
- 확장성 - **풍부한 라이브러리**를 바탕으로 무궁한 확장성이 있음
- 접착성 - **C 또는 C++로 구현된 모듈을 파이썬에서 사용할 수 있음.** 그 반대로 가능. 접착(glue) 언어
- 플랫폼 독립적 - 한번 작성한 코드는 **운영체제의 영향을 받지 않고 어디서든지 실행** 가능
- 객체지향 언어 - **객체지향**의 개념을 적용한 언어
- 동적 실행 - **인터프리터** 형 언어(Interpreted Language)이기 때문에 코드를 작성하면서 바로 실행시켜 볼 수 있음

1.4. 컴파일 언어와 인터프리터 언어

1절. 파이썬 개요

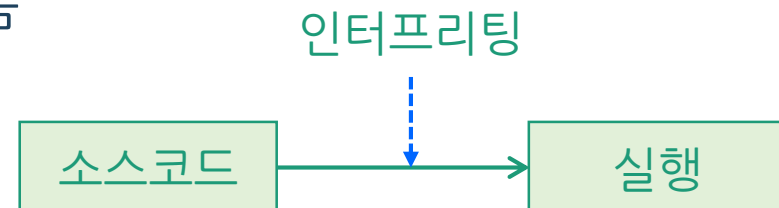
● 컴파일 언어

- 원시 소스코드를 컴파일이라는 과정을 통해 기계어로 번역한 파일을 만들고 이 파일을 통해 실행시키는 언어
- 컴파일 과정이 오래 걸리고 실행시키는 컴퓨팅 환경에 영향을 많이 받지만 실행 시 컴파일 된 실행 파일만 있으면 프로그램을 실행시킬 수 있으며 매우 빠른 속도로 실행되는 장점
- C, C++ 등



● 인터프리터 언어

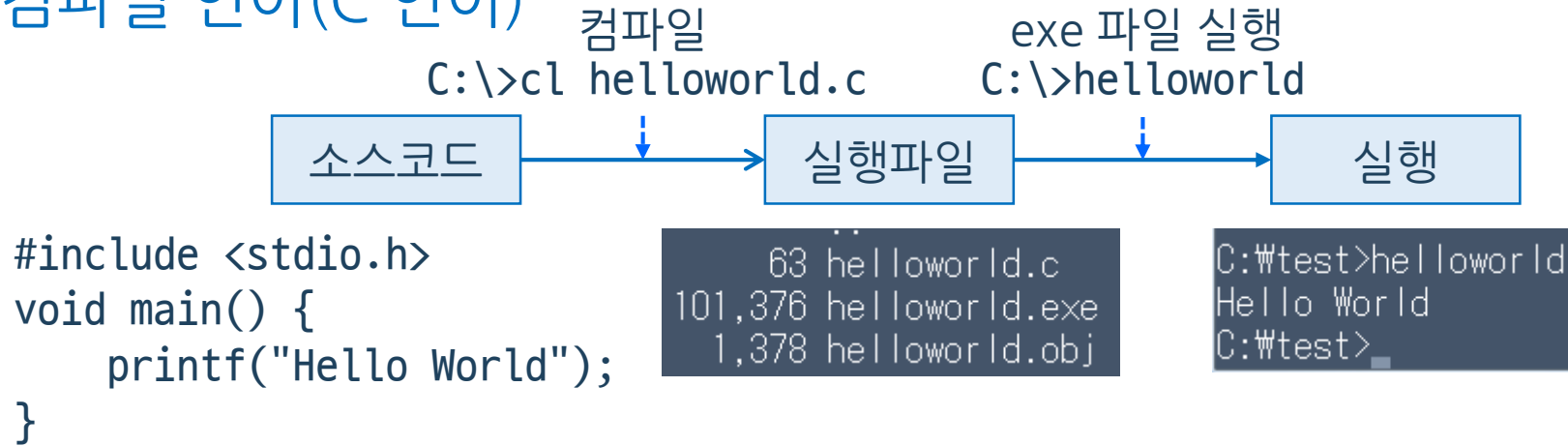
- 인터프리터(해석기)에 의해 원시 소스코드를 한 줄씩 읽어 실행하는 언어
- 컴파일 언어보다 더 느리게 실행
- 코드를 빠르게 테스트해 볼 수 있고, 프로그래밍을 대화식으로 할 수 있기 때문에 교육용으로 사용되는 경우가 많음
- Python, JavaScript, R 등



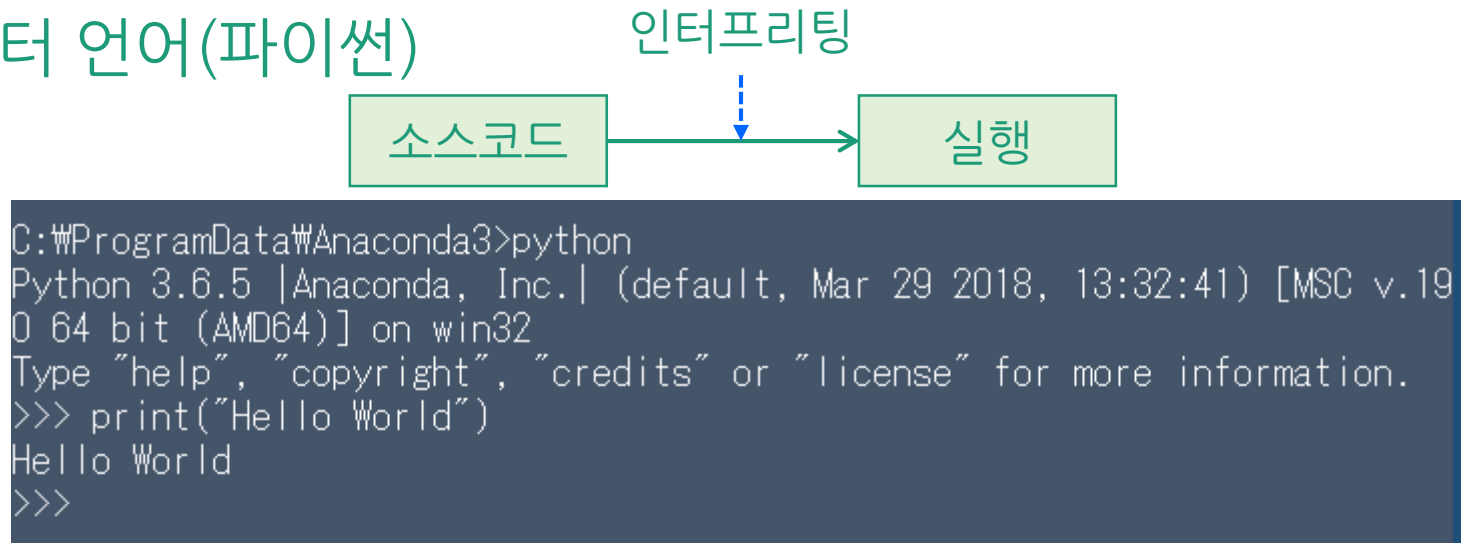
1.4. 컴파일 언어와 인터프리터 언어

1절. 파이썬 개요

● 컴파일 언어(C 언어)



● 인터프리터 언어(파이썬)



2절. 파이썬 인터프리터

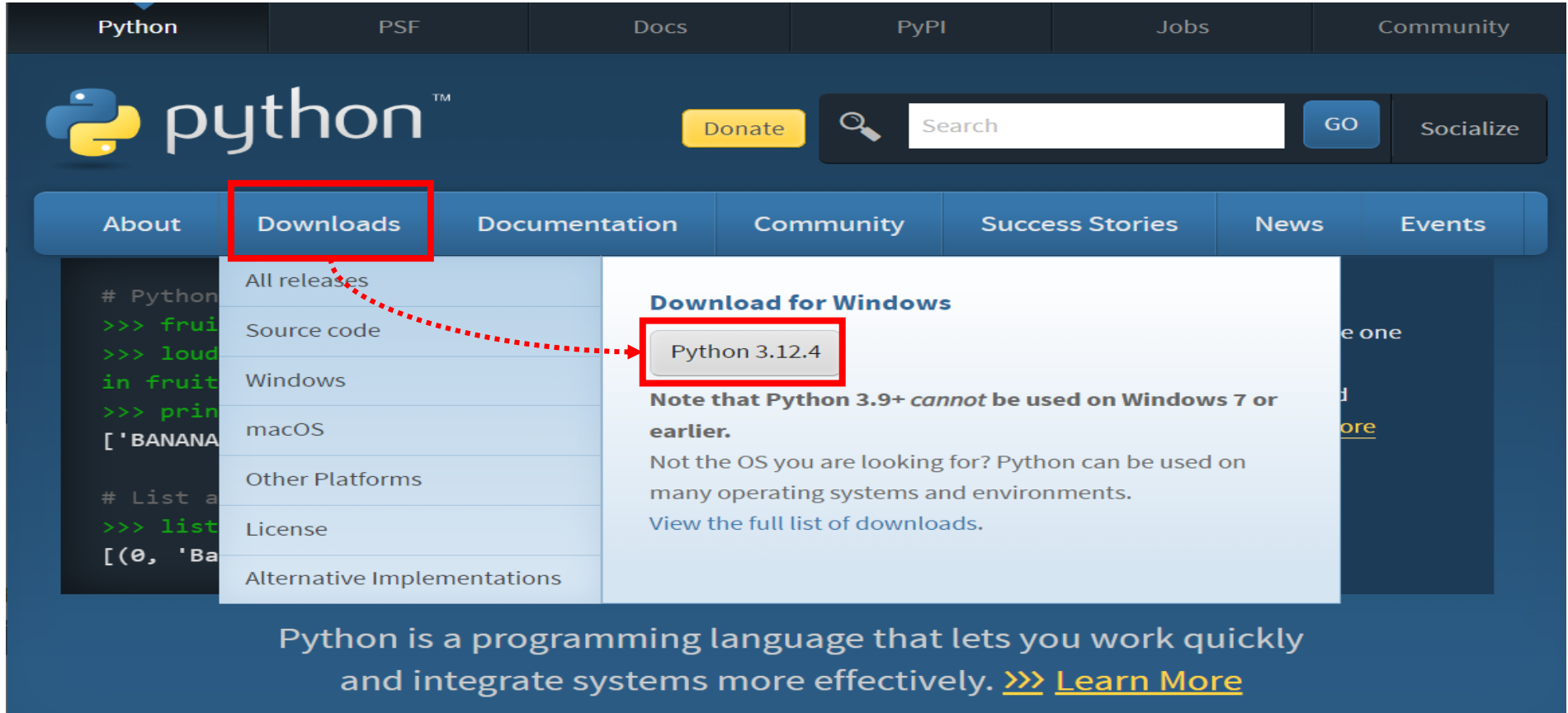
2절. 파이썬 인터프리터

- 파이썬 코드를 실행시키기 위한 프로그램
- 공식 사이트를 통해 다운로드 받을 수 있는 프로그램
- 파이썬 개발 도구
 - 파이썬 인터프리터(<http://www.python.org>)
 - Anaconda(<https://www.anaconda.com/>)의 Jupyter notebook 또는 Spyder
 - Visual studio code(<https://code.visualstudio.com>)
 - PyCharm(<https://www.jetbrains.com/pycharm/>)

2.1. 파이썬 인터프리터 다운로드

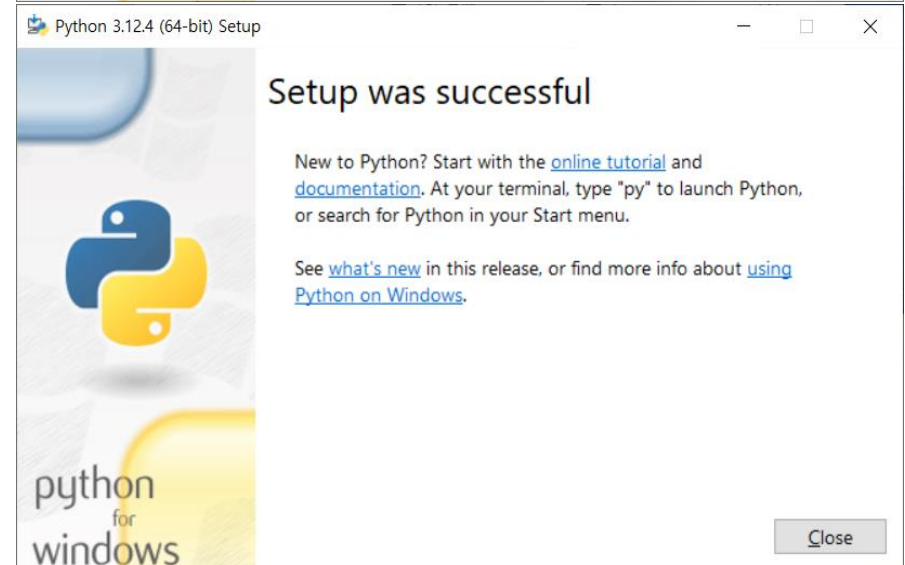
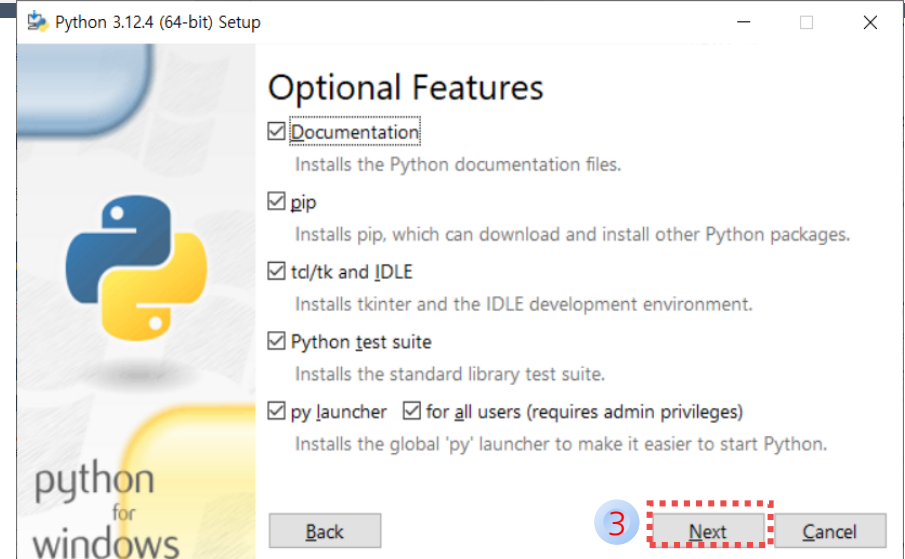
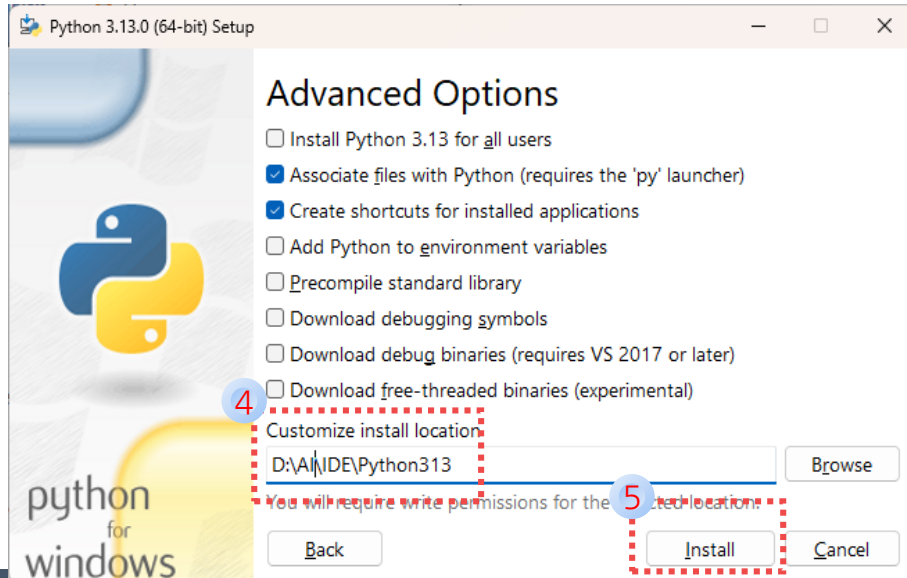
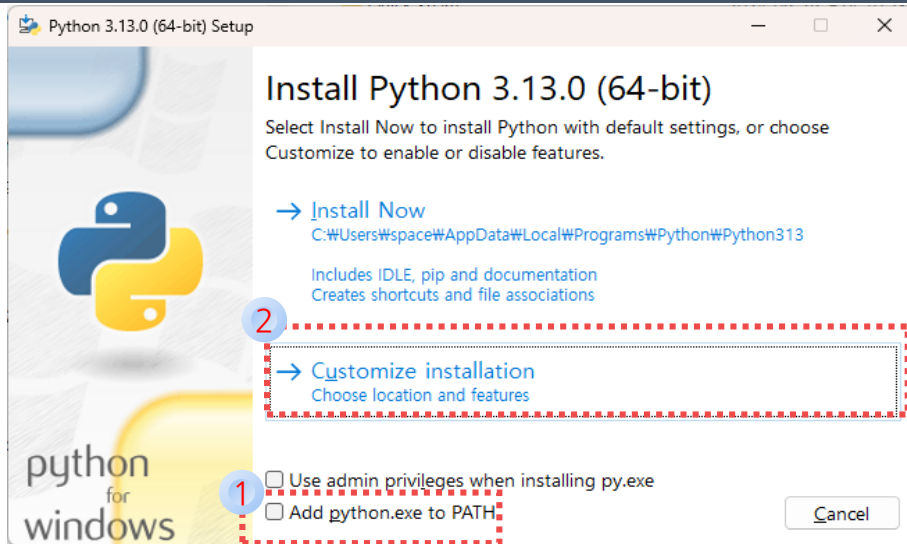
2절. 파이썬 인터프리터

- <https://www.python.org/>



2.2. 파이썬 인터프리터 설치

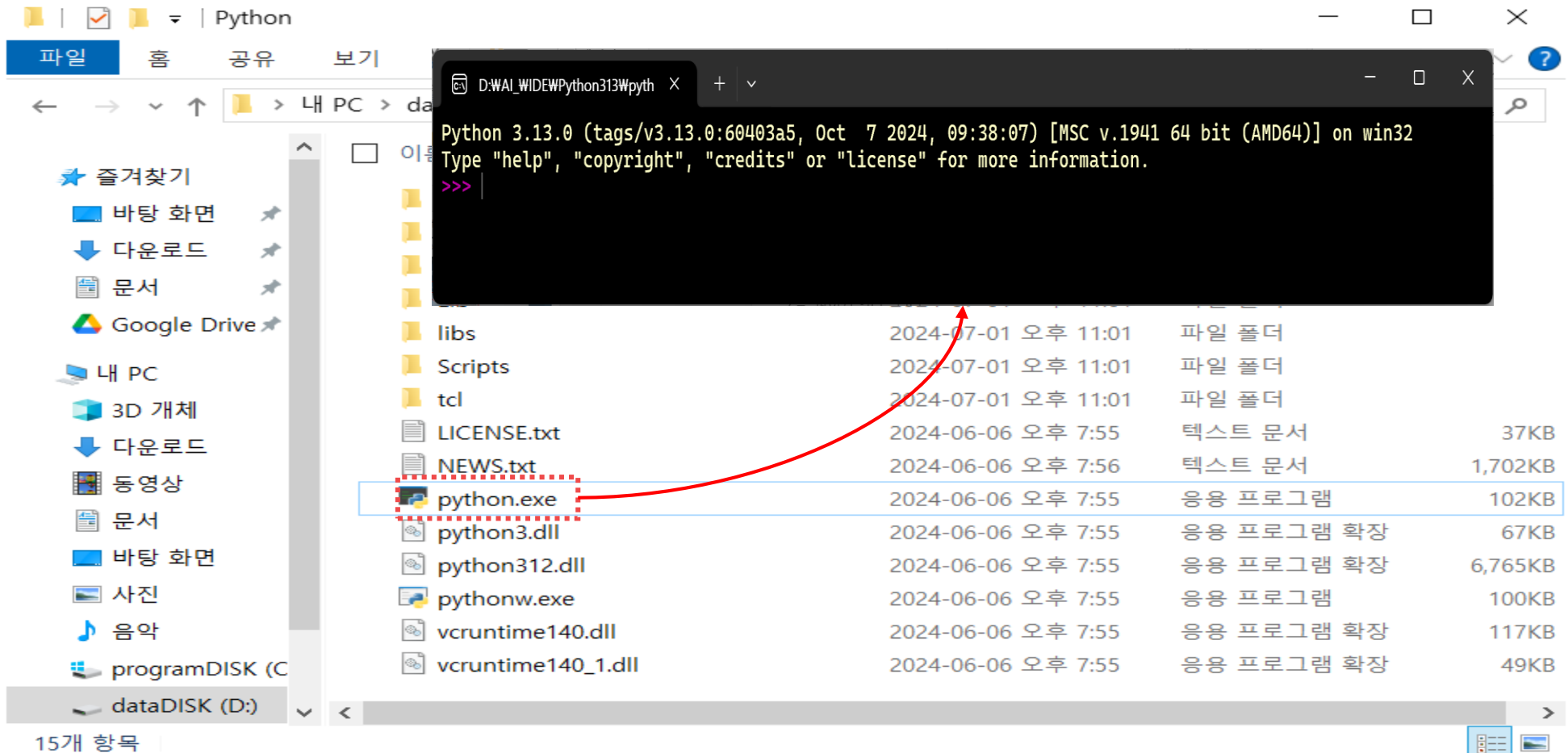
2절. 파이썬 인터프리터



2.2. 파이썬 인터프리터 설치 - 설치 확인

2절. 파이썬 인터프리터

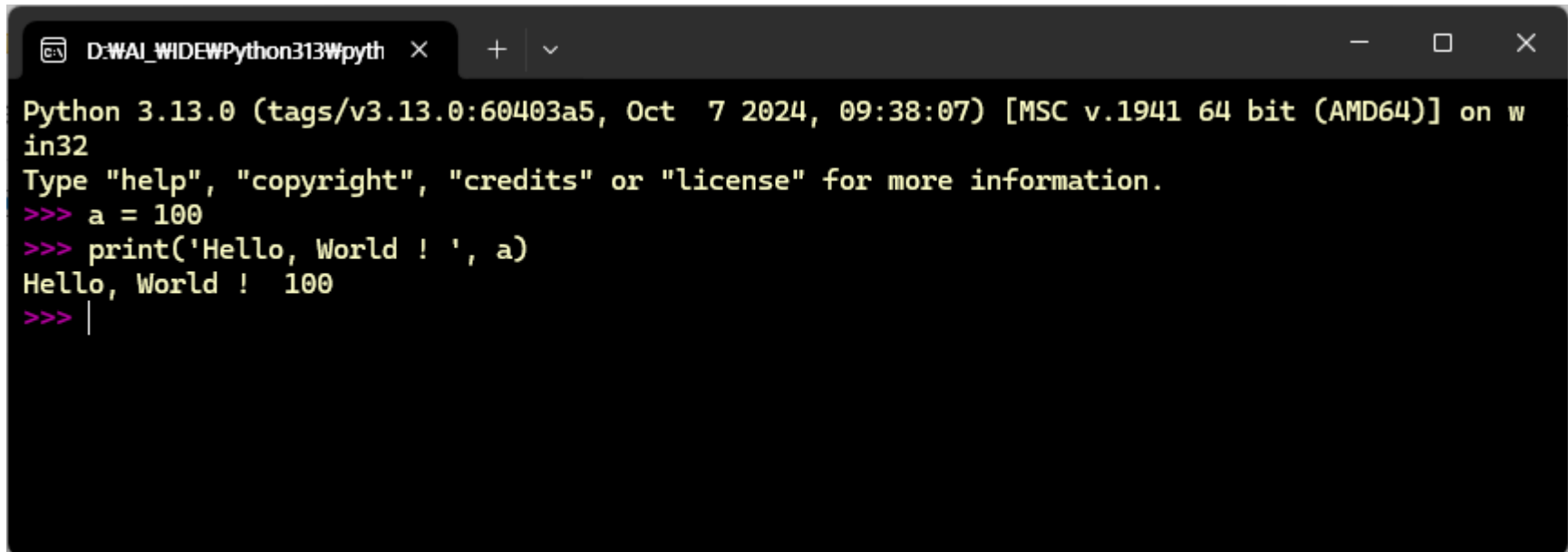
- D:\WAI\WIDE\Python\Python3 디렉토리에서 python.exe 실행파일 더블클릭



2.3. Hello World 출력하기

2절. 파이썬 인터프리터

- `print("Hello World")` 를 입력하고 엔터키를 누르면 실행 결과가 출력
- 코드 작성하고 엔터키를 누르면 코드의 문법 검사 및 실행이 동시에 이루어짐



```
D:\WAI_WIDEWPython313Wpyth x + v
Python 3.13.0 (tags/v3.13.0:60403a5, Oct 7 2024, 09:38:07) [MSC v.1941 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 100
>>> print('Hello, World ! ', a)
Hello, World ! 100
>>> |
```

3.1. 파이썬 배포판

3절. 아나콘다를 이용한 개발 환경

● 파이썬 개발 환경

- 파이썬 인터프리터 설치
- 개발 도구 설치
- pip 설치
- 패키지(Package) 설치

● 애플리케이션 개발 또는 데이터 분석 시

- 많은 패키지들을 설치하고 패키지들 사이의 의존성 관리 필요
- 패키지는 특정 기능을 수행하기 위하여 만들어 놓은 모듈 또는 라이브러리들의 모음

● 파이썬 배포판

- 개발자의 불편을 덜어주기 위해 주요 패키지와 개발 환경이 포함된 개발환경
- 데이터 처리 및 분석에 필요한 대부분의 패키지가 포함
- 파이썬을 이용한 개발 또는 데이터 분석 시 많이 사용되고 있음
- Anaconda, Winpython, python(x,y) 등이 있음

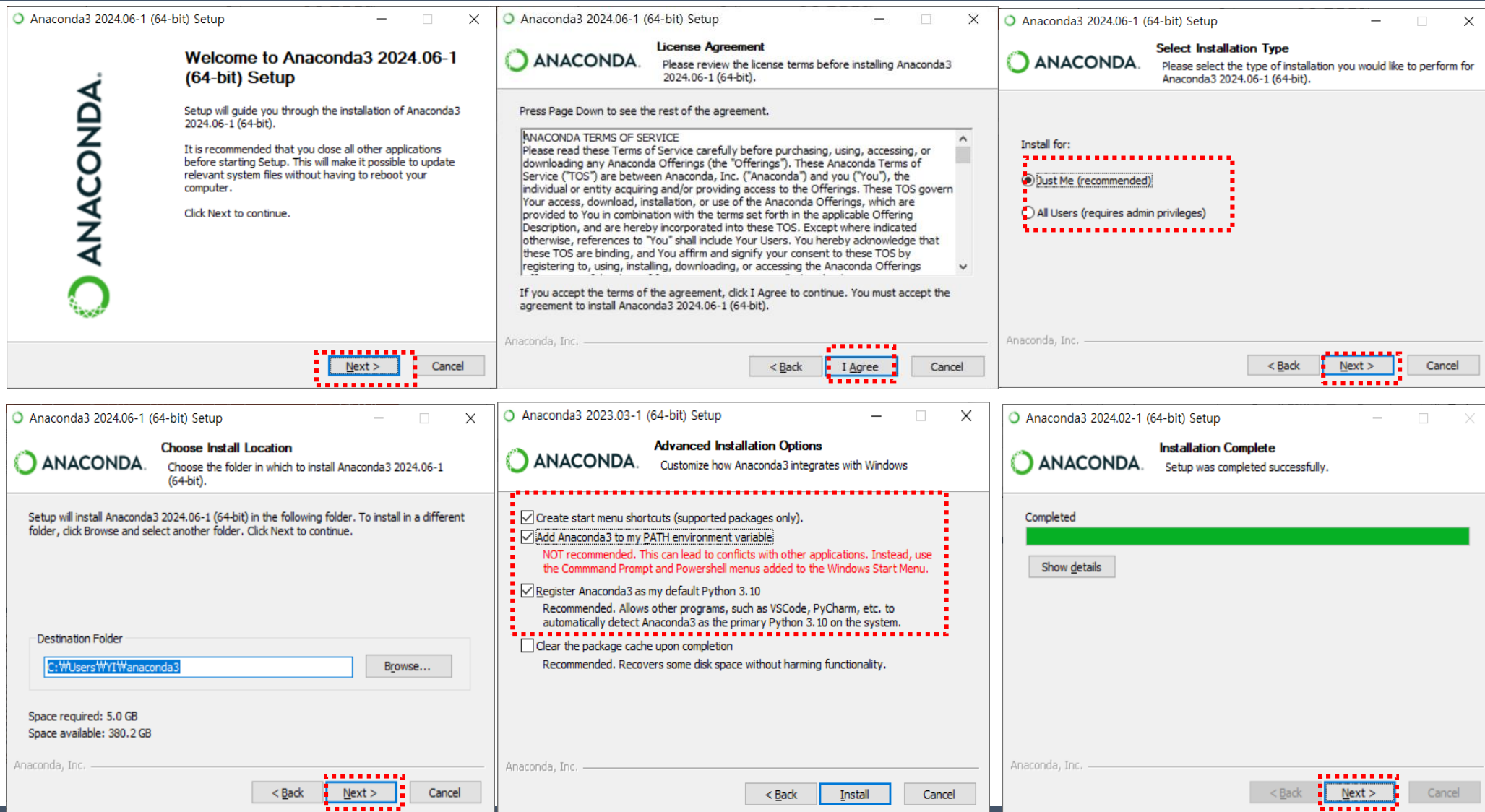
3.2 아나콘다

3절. 아나콘다를 이용한 개발 환경

- 파이썬을 이용한 데이터 과학 및 기계 학습을 수행할 수 있는 가장 쉬운 방법을 제공
- 아나콘다 배포판(Anaconda Distribution)
 - 7,500개 이상의 Python/R 데이터 과학 패키지를 빠르게 다운로드
 - conda 명령을 사용 하여 라이브러리, 종속성 및 환경 관리
 - Scikit-learn, TensorFlow 및 Theano로 머신러닝 및 딥러닝 모델 학습 및 개발 및 교육
 - Dask, NumPy, pandas 및 Numba를 사용하여 확장성과 성능을 갖는 데이터 분석
 - Matplotlib, Seaborn을 사용하여 결과를 시각화
 - GPU를 사용하려면, Anaconda3-2023.03-1-Windows-x86_64.exe
 - **다운로드 :** <https://repo.anaconda.com/archive>

3.2. 아나콘다 (시간소요 다소 있음)

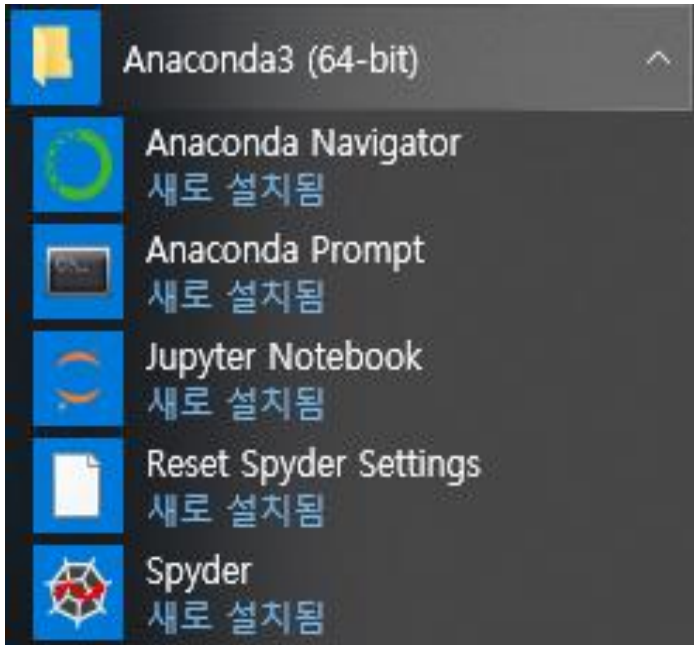
3절. 아나콘다를 이용한 개발 환경



3.2. 아나콘다

3절. 아나콘다를 이용한 개발 환경

● Anaconda3 설치 프로그램

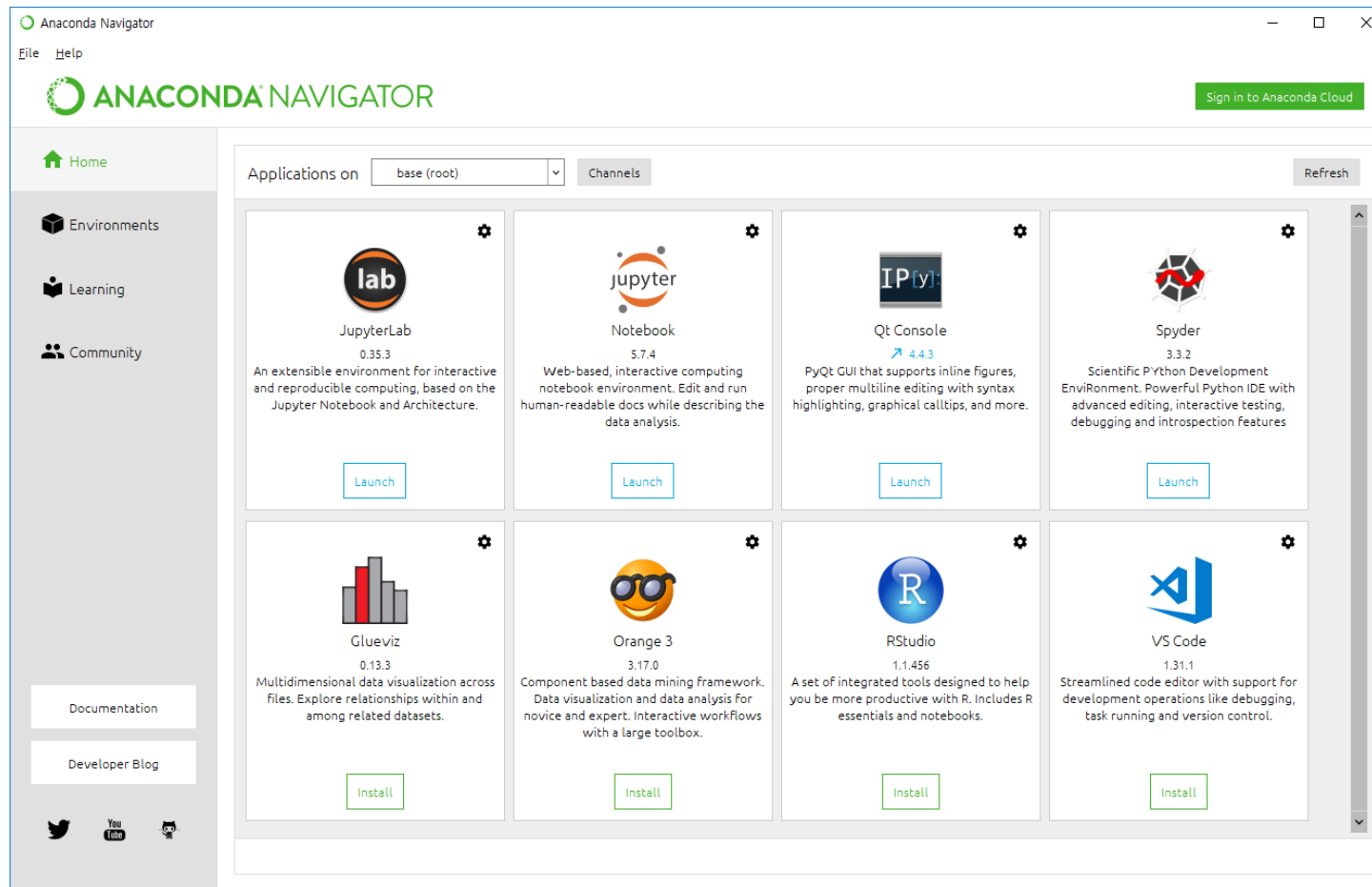


- ❖ Anaconda Navigator
 - 아나콘다의 환경과 프로젝트 및 설치 구성요소들을 관리할 수 있는 윈도우 애플리케이션
- ❖ Anaconda Prompt
 - 아나콘다 명령을 직접 실행시킬 수 있는 명령행 프롬프트
- ❖ Jupyter Notebook
 - ❖ 주피터 노트북 실행
- ❖ Spyder
 - ❖ 파이썬 애플리케이션을 개발하기 위한 통합개발 환경(IDE)

3.3. 아나콘다 네비게이터

3절. 아나콘다를 이용한 개발 환경

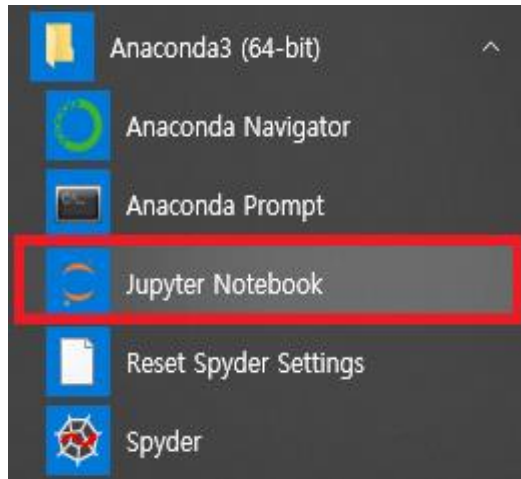
- 아나콘다를 설치할 때 함께 설치되는 구성요소들을 확인할 수 있고 원하는 구성 요소를 실행시킬 수 있음



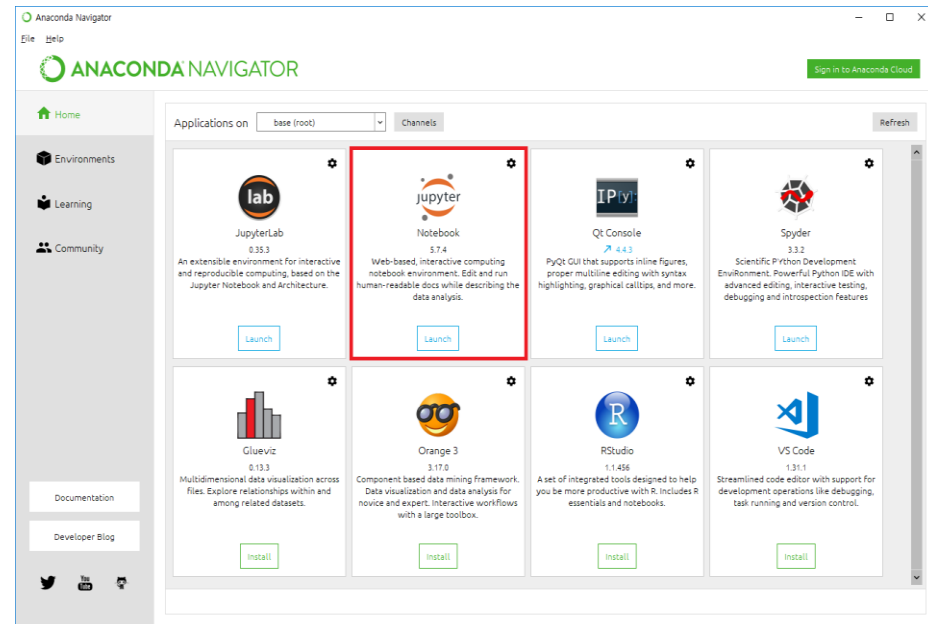
3.4. 주피터 노트북

3절. 아나콘다를 이용한 개발 환경

- 라이브 코드(live code), 방정식(equation), 시각화(visualization), 설명문(explanatory text) 등을 작성 할 수 있는 웹 애플리케이션
- 데이터 정제(data cleaning), 변환(transformation), 수치 시뮬레이션(numerical simulation), 통계 모델링(statistical modeling), 기계 학습(machine learning) 등 많은 용도로 사용이 가능한 도구



메뉴에서 실행



아나콘다 네비게이터에서 실행

3.5. 주피터 노트북에서 코드 작성 및 실행(1)

3절. 아나콘다를 이용한 개발 환경

- Jupyter Notebook 아이콘을 바탕화면에 끌어 놓고 오른쪽 마우스 클릭 후, 속성정보에 들어가 수정



"%USERPROFILE%/" 대신
" D:\\ai\\source\\07_python"

3.5. 주피터 노트북에서 코드 작성 및 실행(2)

3절. 아나콘다를 이용한 개발 환경

- Jupyter Notebook 파일의 속성 정보에 들어가 대상 정보 수정

대상 형식:	응용 프로그램
대상 위치:	Anaconda3
대상(T):	.\jupyter-notebook-script.py "%USERPROFILE%/"

"%USERPROFILE%/" 삭제

대상 형식:	응용 프로그램
대상 위치:	Anaconda3
대상(T):	.\Anaconda3\Scripts\jupyter-notebook-script.py

- cmd 창에서 다음 명령을 수행시켜 Jupyter Notebook의 설정

jupyter notebook --generate-config

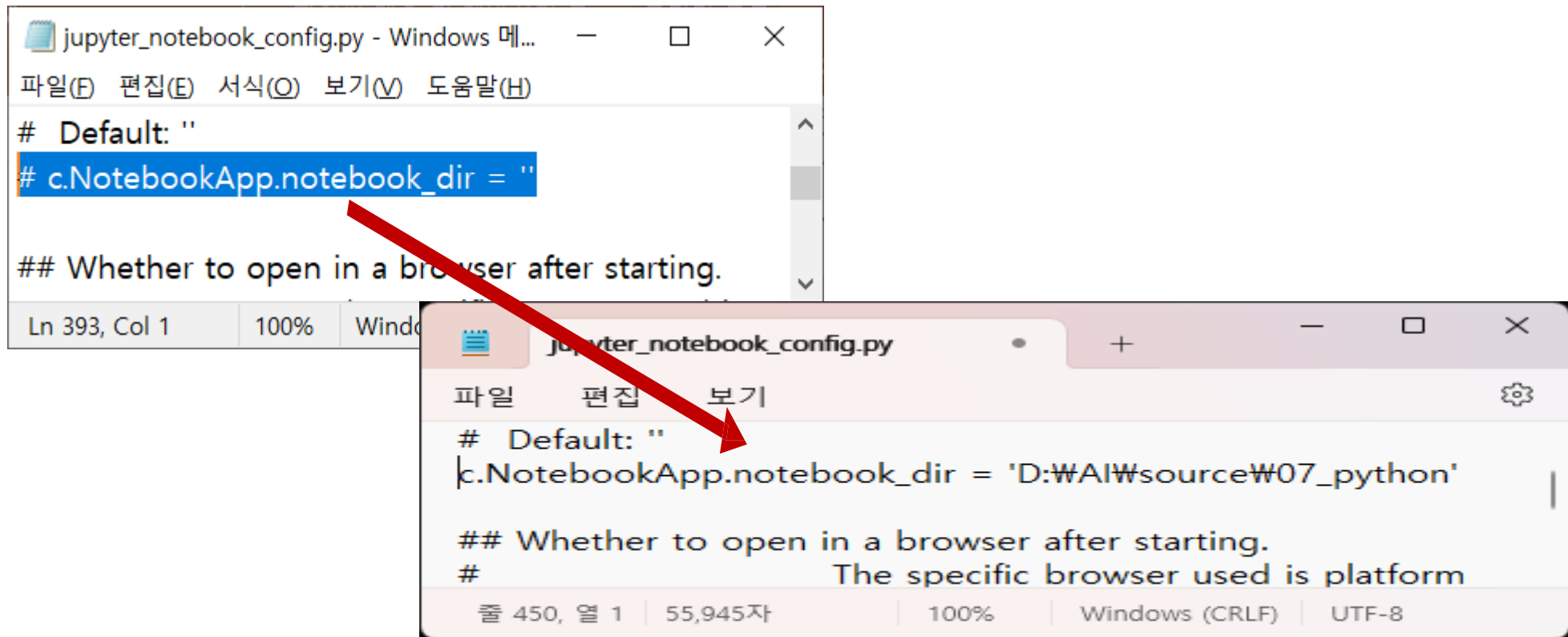
```

명령 프롬프트
C:\Users\YI>jupyter notebook --generate-config
Writing default config to: C:\Users\YI\jupyter\jupyter_notebook_config.py
C:\Users\YI>
  
```

3.5. 주피터 노트북에서 코드 작성 및 실행(2)

3절. 아나콘다를 이용한 개발 환경

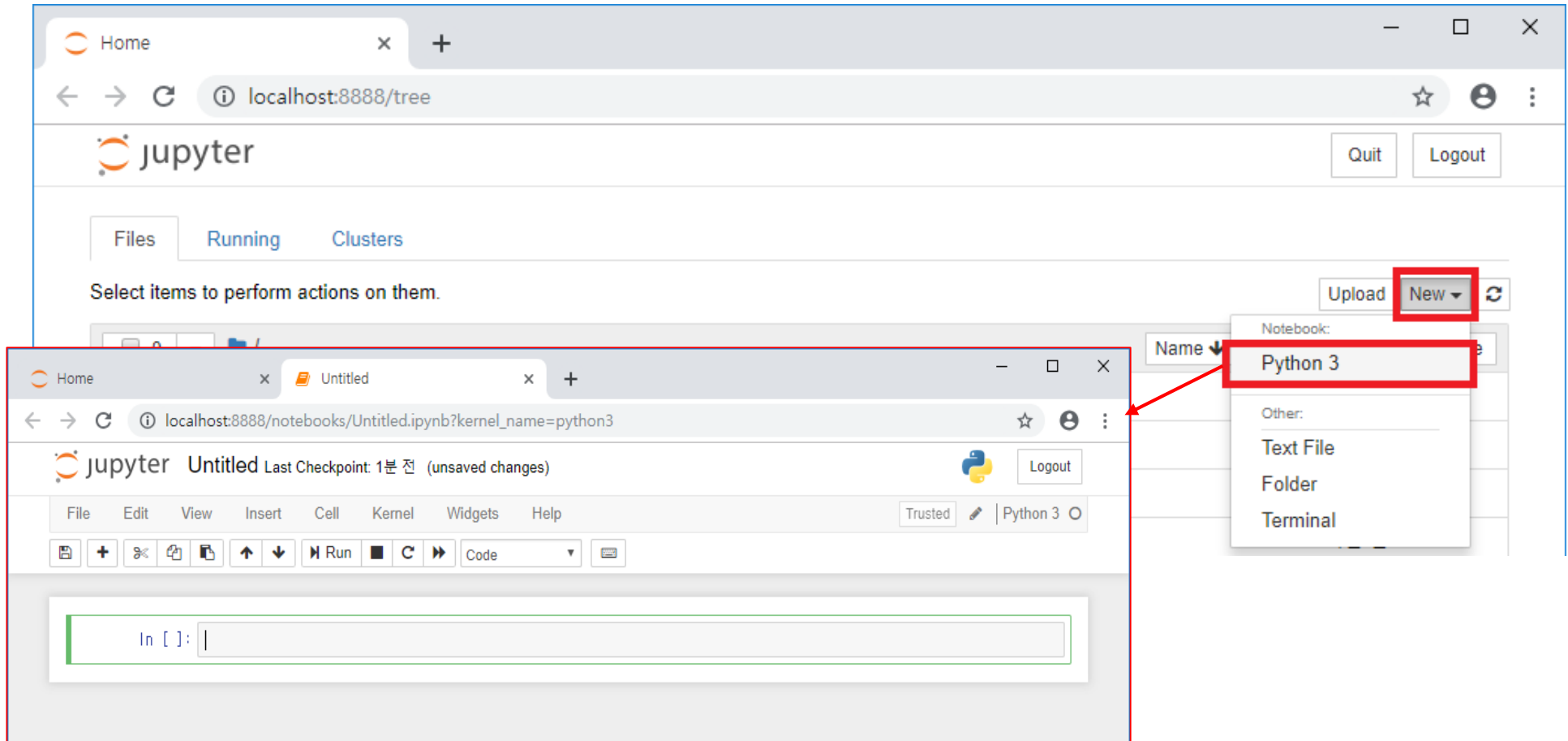
- jupyter_notebook_config.py 파일에서 notebook_dir항목을 찾아 주석을 해제하고 'd:/src/python'을 작성



3.5. 주피터 노트북에서 코드 작성 및 실행

3절. 아나콘다를 이용한 개발 환경


- 주피터 노트북을 실행시키면 나타나는 브라우저 화면에서 [New] 버튼의 하위 메뉴에서 [Python3] 메뉴를 선택

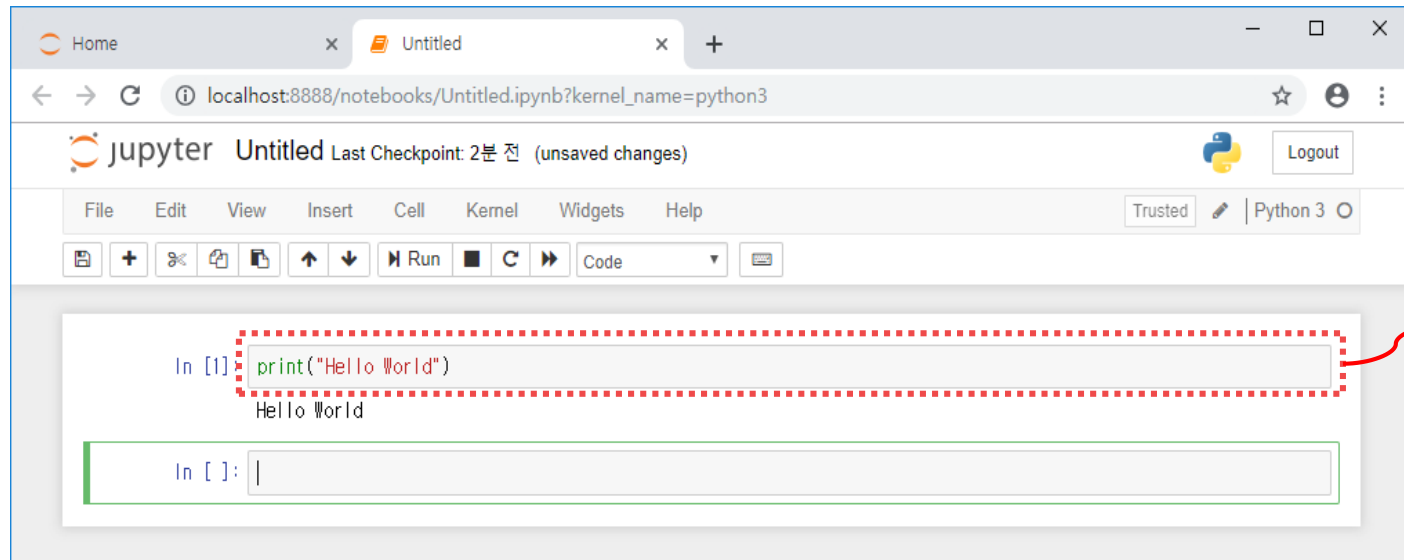


3.5. 주피터 노트북에서 코드 작성 및 실행

3절. 아나콘다를 이용한 개발 환경

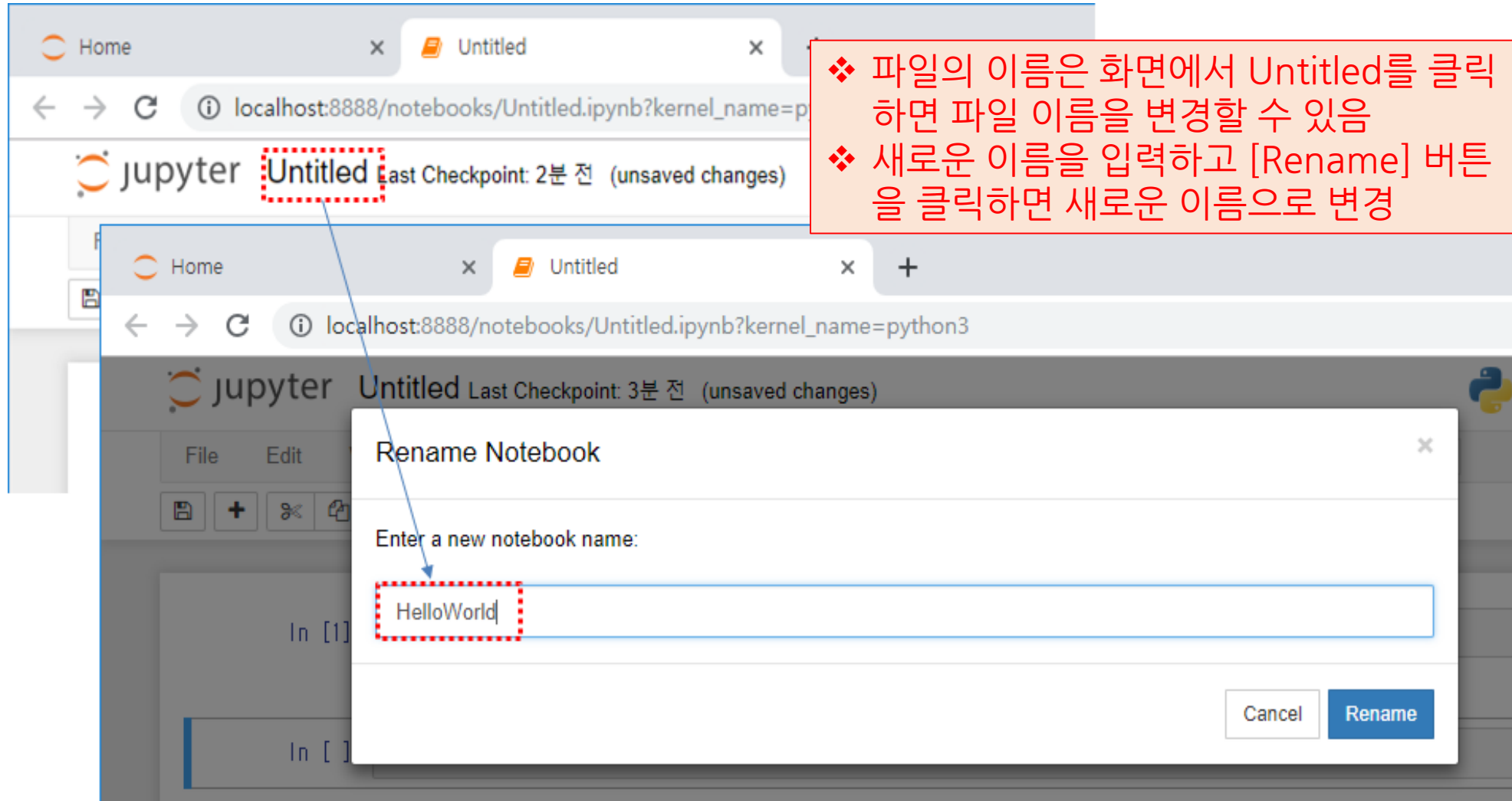
● 실행 방법

- **Shift+Enter** 또는  버튼 클릭
 - 선택한 셀을 실행시키고 포커스를 다음 셀로 이동. 다음 셀이 없을 경우 셀을 추가
- **Ctrl+Enter**
 - 선택한 셀을 실행 시킴. 포커스는 다음 셀로 이동하지 않음
- **Alt+Enter**
 - 선택한 셀을 실행 시킴. 항상 아래에 새로운 셀을 생성하고 포커스를 이동시킴



3.5. 주피터 노트북에서 코드 작성 및 실행

3절. 아나콘다를 이용한 개발 환경



❖ 파일의 이름은 화면에서 Untitled를 클릭 하면 파일 이름을 변경할 수 있음

❖ 새로운 이름을 입력하고 [Rename] 버튼을 클릭하면 새로운 이름으로 변경

Rename Notebook

Enter a new notebook name:

HelloWorld

Cancel Rename

3.6. 스파이더에서 코드 작성 및 실행

3절. 아나콘다를 이용한 개발 환경

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.7)". The interface is divided into several panes:

- Editor (1):** The central pane for writing code. It shows a file named "temp.py" with the following content:


```
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7 print("Hello World")
8
9 #%%
10
11 a = 10
12 b = 20
13 print(a+b)
14 #%%
```
- Variable Explorer (2):** A pane on the right showing the current state of variables. It contains a table:

Name	Type	Size	Value
a	int	1	10
b	int	1	20
- IPython console (3):** A pane at the bottom right for executing code. It shows the output of the code in the editor:


```
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.2.0 -- An enhanced Interactive Python.

In [1]: """
...: Spyder Editor
...:
...: This is a temporary script file.
...: """
...: print("Hello World")
Hello World

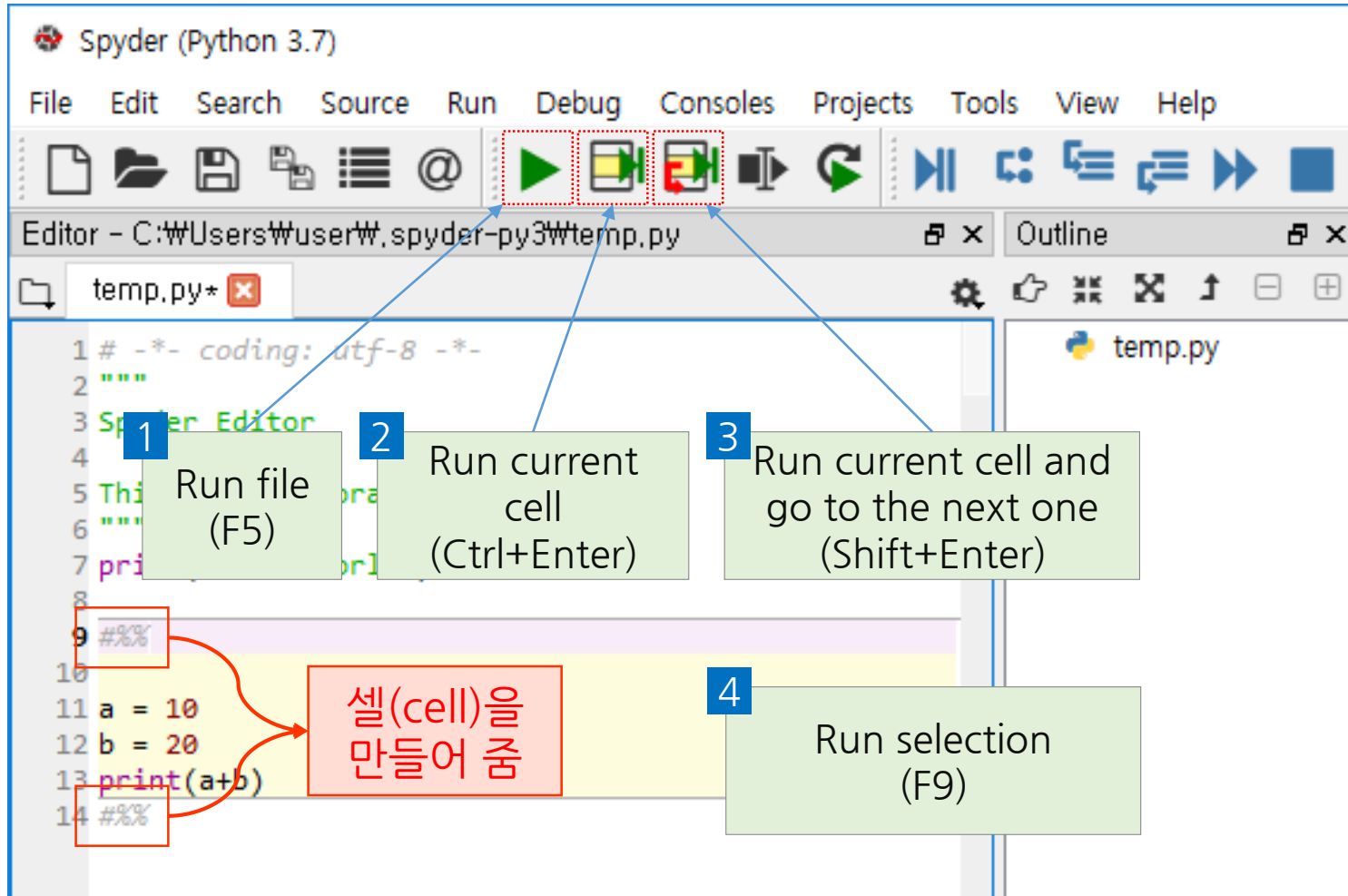
In [2]: a = 10
...: b = 20
...: print(a+b)
30

In [3]:
```

The bottom status bar shows: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 14, Column: 1, Memory: 49%.

3.6. 스파이더에서 코드 작성 및 실행

3절. 아나콘다를 이용한 개발 환경



정리

1장. 파이썬 개요 및 개발환경 구성

- ✓ 파이썬은 객체지향 언어이며 인터프리터 언어
- ✓ 파이썬 인터프리터 설치 및 실행
- ✓ 아나콘다를 이용한 개발환경
 - 파이썬 배포판
 - 아나콘다
 - <https://www.anaconda.com/distribution>
 - 아나콘다 네비게이터
 - 주피터 노트북
 - 데이터 분석 시 주로 사용함
 - 주피터 노트북에서 소스코드 작성 및 실행
 - 셀 단위로 실행(Shift+Enter, Ctrl+Enter, Alt+Enter)
 - 스파이더에서 코드 작성 및 실행
 - #%%는 셀 구분자
 - 전체 실행(F5), 셀 단위로 실행(Shift+Enter, Ctrl+Enter), 선택영역 실행(F9)

편리한 소스 코딩을 위해 첫 셀에 항상 추가할 내용

```
1 from IPython.display import display, HTML
2 display(HTML("""
3 <style>
4 div.container{width:99% !important;}
5 div.cell.code_cell.rendered{width:85%;}
6 div.CodeMirror {font-family:Consolas; font-size:12pt;}
7 div.output {font-size:12pt; font-weight:bold;}
8 div.input {font-family:Consolas; font-size:12pt;}
9 div.prompt {min-width:70px;}
10 div#toc-wrapper{padding-top:150px;}
11 span.toc-item-num{display:none;}
12 div.text_cell_render ul li{font-size:14pt;padding-top:5px;}
13 </style>
14 """)
15
16
```