

데이터 과학을 위한 파이썬 프로그래밍



01. 프로그래밍 언어와 파이썬

목차

1. 프로그래밍 언어의 이해
2. 파이썬 소개
3. 파이썬 개발 환경과 설치

01

프로그래밍 언어의 이해

01. 프로그래밍 언어의 이해

■ 프로그래밍 언어의 개념

- 프로그래밍 언어(programming language) : '인간이 원하는 것을 컴퓨터로 실행시키기 위해 사용하는, 컴퓨터가 이해할 수 있는 언어'이다.

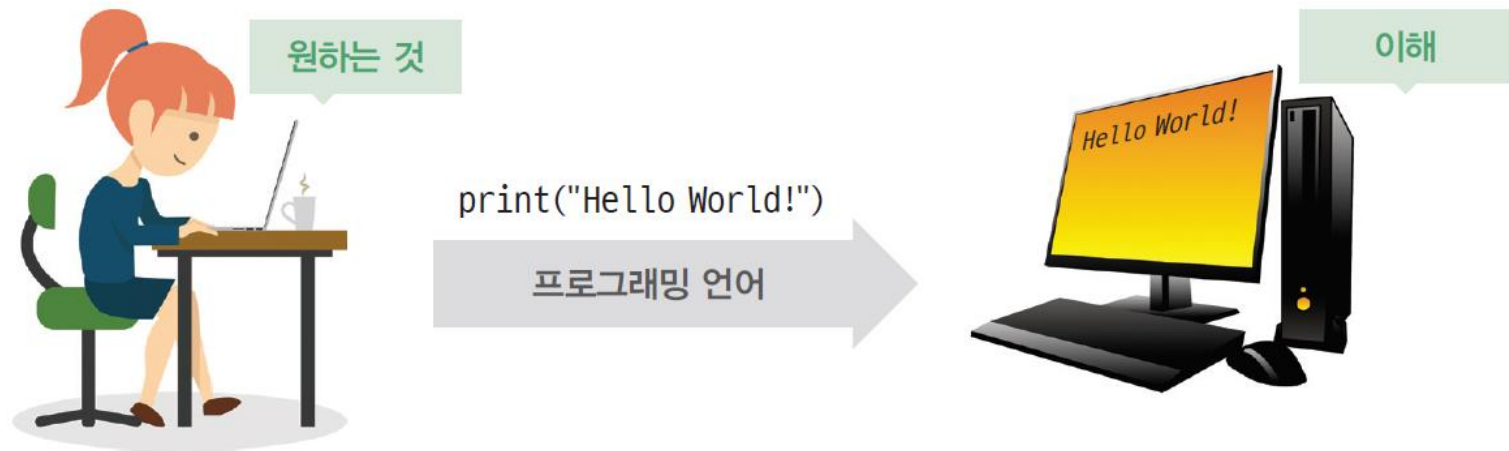


[알파고와 이세돌의 바둑 대결(출처: Becoming Human)]

01. 프로그래밍 언어의 이해


■ 프로그래밍 언어의 개념

- 프로그래밍(programming) : 프로그래밍 언어를 사용하여 프로그램을 개발하는 것



[프로그래밍 언어의 사용]

01. 프로그래밍 언어의 이해

여기서  잠깐! 일상생활 속 프로그래밍의 결과물



(a) 아마존 고



(b) 아마존 키바



(c) 무인지하철



(d) 무인자동차

[일상생활 속 프로그래밍의 결과물]

01. 프로그래밍 언어의 이해

■ 프로그래밍 언어를 배우는 이유

- 하나의 기술이나 학문으로서 프로그래밍 언어를 배우는 것이 아니라, 논리적인 사고를 훈련하기 위해 프로그래밍 언어를 배워야 한다.

"이 나라의 모든 사람이 프로그램을 배워야 한다. 왜냐하면 프로그램이라는 것은 생각을 어떻게 하는지 가르쳐 주기 때문이다."

- 스티브 잡스

01. 프로그래밍 언어의 이해

여기서 잠깐! 프로그래밍적 사고

- 점수별로 학점을 매기는 문제를 풀 때 어떤 절차를 걸쳐야 할까? 첫 번째로 할 일은 기준을 정하는 것이다. 즉, 몇 점 이상을 A로 줄 것인지, 몇 점 이하를 F로 줄 것인지를 결정한다. 다음으로 각 점수를 기준에 따라 차례대로 성적을 매긴다. 마지막으로 성적 매기기를 종료하는 시점을 결정하는데, 점수가 더는 존재하지 않으면 종료하거나 30개 이상을 하면 종료하거나 등을 결정한다.
- 이러한 과정이 흔히 말하는 프로그래밍적 사고에서 절차적 프로그래밍 과정이다.

점수	학점	점수	학점	점수	학점
38		4		77	
37		57		43	
7		47		82	
16		96		22	
95		16		92	
71		89		53	
63		14		27	
48		29		20	
49		6		22	
66		94		38	
37		2		76	

[점수별 학점 계산]

02

파이썬 소개

02. 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]



[파이썬 로고]

02. 파이썬 소개

■ 파이썬의 특징

1. 플랫폼 독립적인 언어 : 어떤 운영체제든 상관없이 사용할 수 있는 언어를 말한다.
2. 인터프리터 언어 : 컴파일러 언어와 달리, 소스코드 자체가 바로 실행되는 특징이 있는 언어이다. 이로 인해 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 방식이다.
4. 동적 타이핑 언어 : 프로그램의 실행 시점에서 각 프로그램 변수의 타입을 결정하는 언어이다.

02. 파이썬 소개

여기서 잠깐! 컴파일러와 인터프리터

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

[컴파일러와 인터프리터 비교]

02. 파이썬 소개

■ 파이썬을 배우는 이유

1. 쉽고 간단한 프로그래밍 언어

- 화면에 'Hello World!'라는 텍스트를 출력하는 프로그램을 작성한다고 가정하자. 파이썬과 자바로 코드를 작성하면 아래와 같다.

파이썬

```
print("Hello World!")
```

자바(Java)

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

02. 파이썬 소개

■ 파이썬을 배우는 이유

1. 쉽고 간단한 프로그래밍 언어

- 파이썬과 자바로 화면에 1부터 10까지 출력하는 프로그램을 코드로 작성하면 다음처럼 두 언어를 비교할 수 있다.

파이썬

```
for i in range(1, 11):  
    print (i)
```

자바(Java)

```
for (i = 1; i <11; i++){  
    System.out.println (i)  
}
```

- 이처럼 파이썬은 프로그래밍을 처음 배우는 초보자도 다른 프로그래밍 언어보다 훨씬 간단하고 이해하기 쉽다는 장점이 있다.

02. 파이썬 소개

■ 파이썬을 배우는 이유

2. 다양한 라이브러리 제공

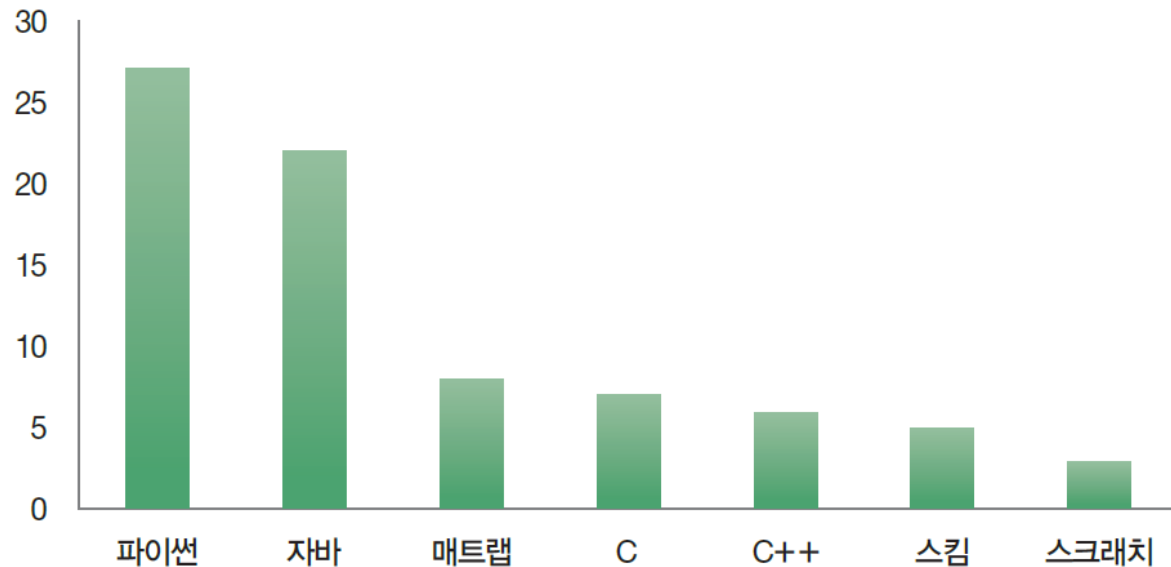
- 파이썬은 다양한 라이브러리를 제공하여 활용 범위가 넓다.
- 개발자들이 만들어 놓은 많은 모듈과 패키지를 쉽게 사용할 수 있다.

02. 파이썬 소개

■ 파이썬을 배우는 이유

3. 대중적인 프로그래밍 언어

- 대표적인 컴퓨터 공학 학회지인 「ACMAssociation for Computing Machinery」은 2014년 조사를 통해 미국 39개 대학 중 파이썬을 기초 프로그래밍 언어로 선택한 대학이 가장 많다고 밝혔다.



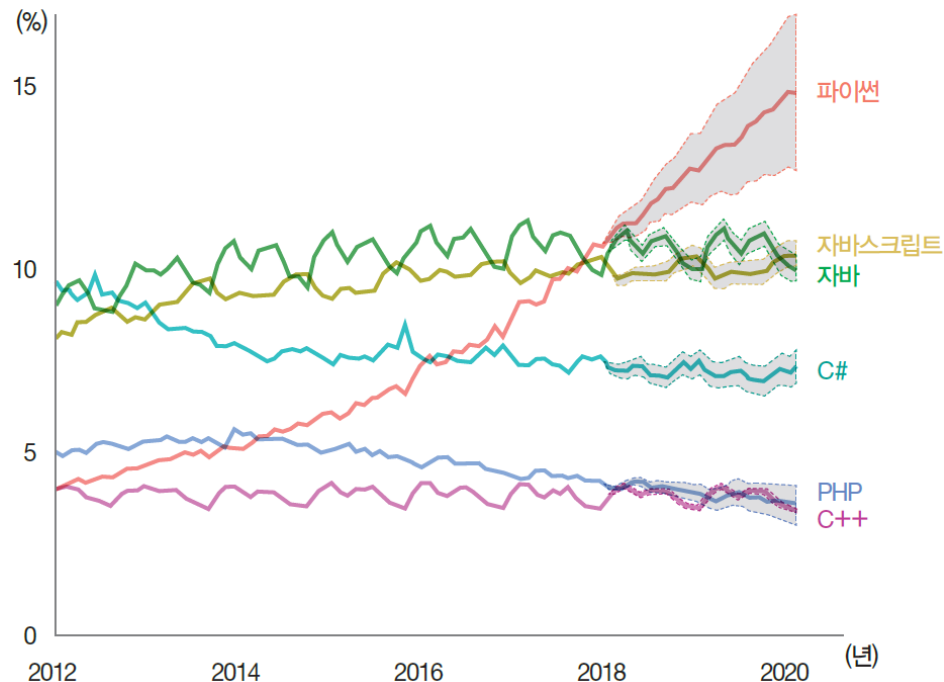
[미국 39개 대학의 기초 프로그래밍 언어 강의 선택 현황(출처: ACM)]

02. 파이썬 소개

■ 파이썬을 배우는 이유

3. 대중적인 프로그래밍 언어

- 프로그래밍계의 '네이버 지식인' 같은 역할을 하는 Stack Overflow의 조사 결과, 현재 가장 많은 질의응답을 받는 프로그래밍 언어는 파이썬인 것으로 나타났다.



[파이썬의 성장(출처: Stack Overflow)]

03

파이썬 개발 환경과 설치

03. 파이썬 개발 환경과 설치

■ 파이썬 개발 환경 설정

1. 운영체제 선정

운영체제	장점	단점
윈도(Windows)	국내 사용자가 쓰기에 가장 쉬운 운영체제로, 프로그래밍을 시작하는 사람에게 상당히 좋은 선택임	다른 사용자가 개발한 다양한 모듈을 설치하기 어렵고, 참고 문서가 부족함
리눅스(Linux)	모듈 설치가 매우 쉽고, 다양한 레퍼런스를 인터넷에서 검색하기 쉬움	초기 사용자가 사용하기에는 운영체제 자체가 매우 어려움
맥 OS(Mac OS)	기본적으로 리눅스의 장점이 있으면서, 윈도처럼 안정적임	다른 운영체제보다 상대적으로 가격이 비쌈

[운영체제별 특징]

03. 파이썬 개발 환경과 설치

■ 파이썬 개발 환경 설정

2. 파이썬 인터프리터 선정

종류	설명
Python	일반적인 파이썬, 기본적인 모듈 포함
Anaconda	다양한 과학 계산용 모듈을 묶어 패키지로 제공, 이 책에서 사용하는 툴
Canopy	다양한 과학 계산용 모듈을 묶어 패키지로 제공

[주요 파이썬 인터프리터]

03. 파이썬 개발 환경과 설치

■ 파이썬 개발 환경 설정

3. 코드 편집기 선정

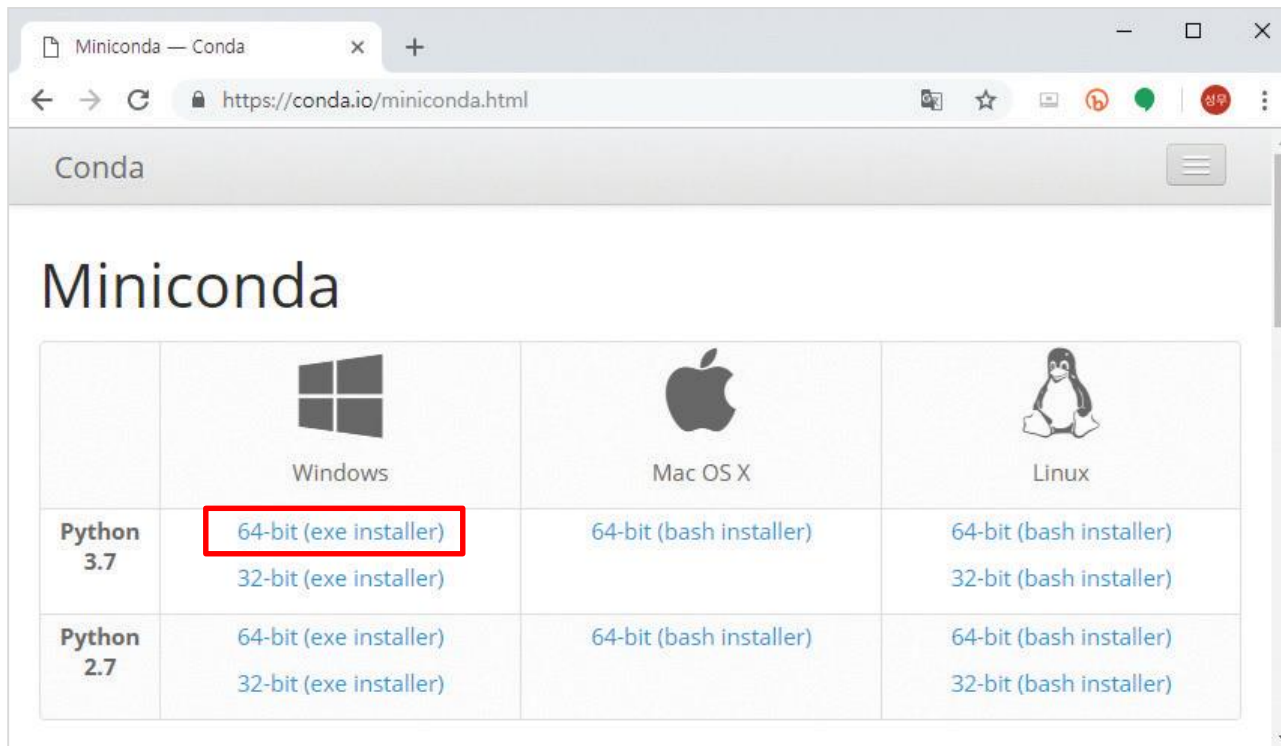
종류	설명
메모장	윈도의 기본 문서 편집 도구
VI editor	리눅스의 기본 문서 편집 도구
Sublime Text	프로그래밍에 특화된 문서 편집 도구
Atom	프로그래밍에 특화된 문서 편집 도구(Github에서 개발), 이 책에서 사용하는 툴
PyCharm	다양한 기능을 갖춘 파이썬 전용 개발 도구(IDE)
Visual Studio Code	MS에서 제공하는 코드 전용 에디터, 다양한 기능으로 최근 Atom과 함께 가장 많은 관심을 받고 있음

[주요 코드 편집기와 IDE]

03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

1. 우선 Miniconda 다운로드 페이지(<https://conda.io/miniconda.html>)에서 'Python3.x'로 시작하는 버전의 윈도우용 인스톨러를 다운로드한다.

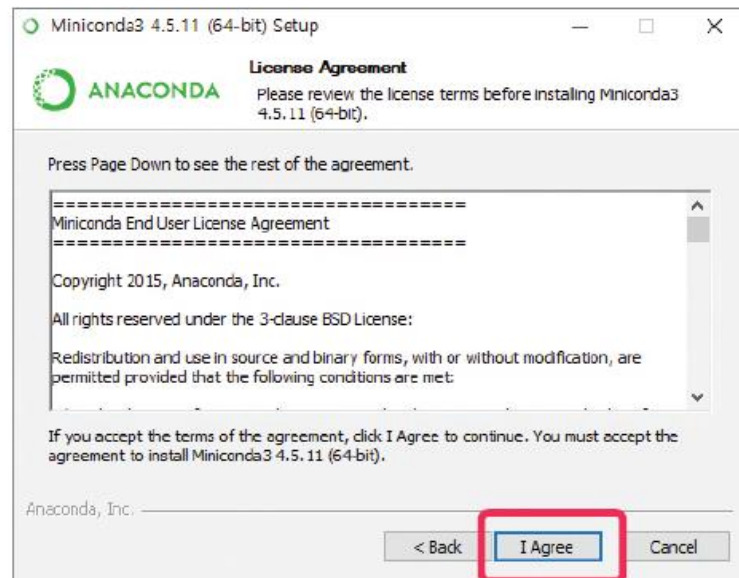


[Miniconda 다운로드 페이지]

03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

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

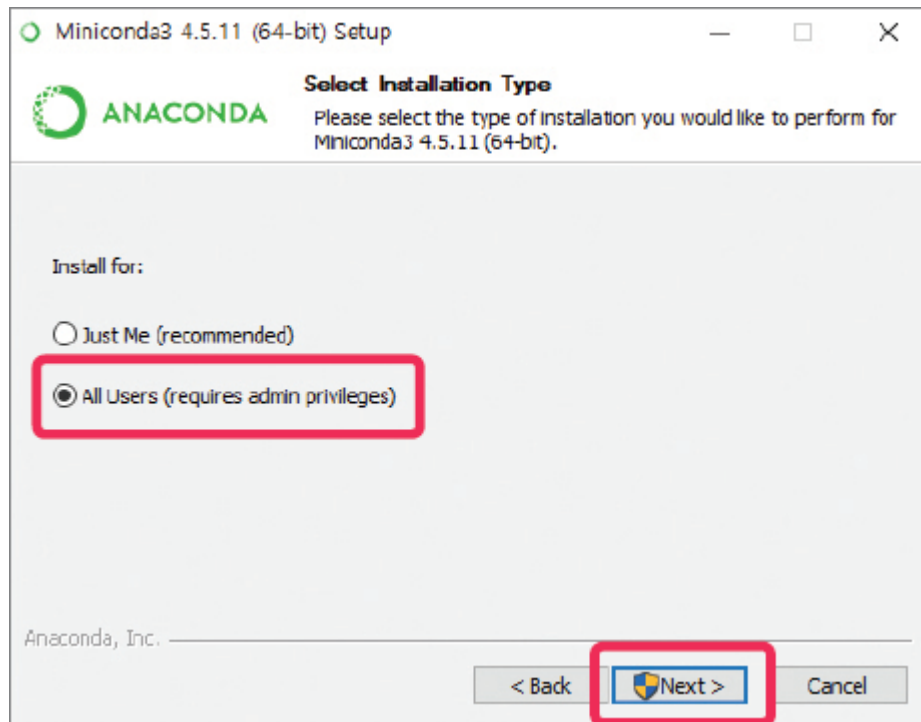


[Miniconda 설치 진행]

03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

3. 인스톨 타입 설정 창에서 'All Users'를 선택하고, [Next]를 클릭한다.

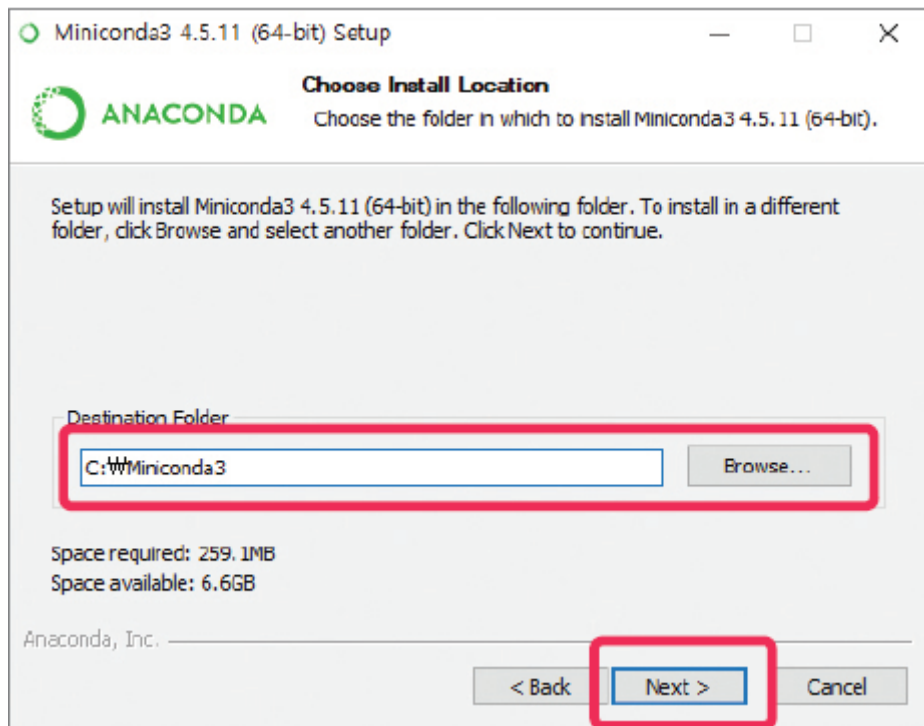


[Miniconda 인스톨 타입 설정]

03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

4. 인스톨 경로 설정 창에서 [Browse]를 클릭하여 적절한 경로를 지정하고, [Next]를 클릭한다.

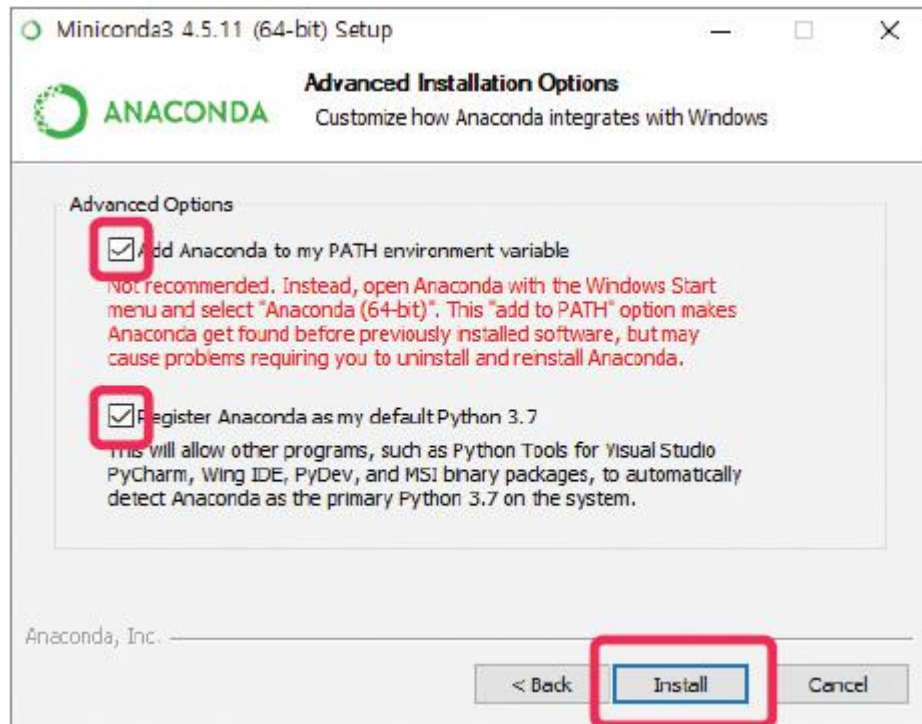


[Miniconda 인스톨 경로 설정]

03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

5. 인스톨 옵션 설정 창에서 체크박스를 모두 체크한 후, [Install]을 클릭한다.

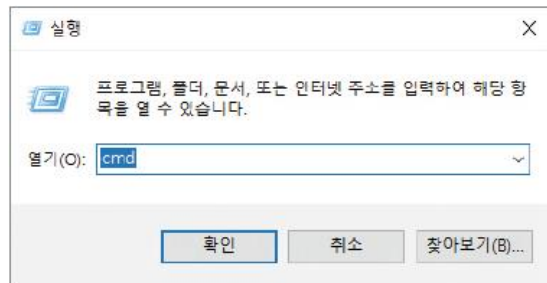


[Miniconda 인스톨 옵션 설정]

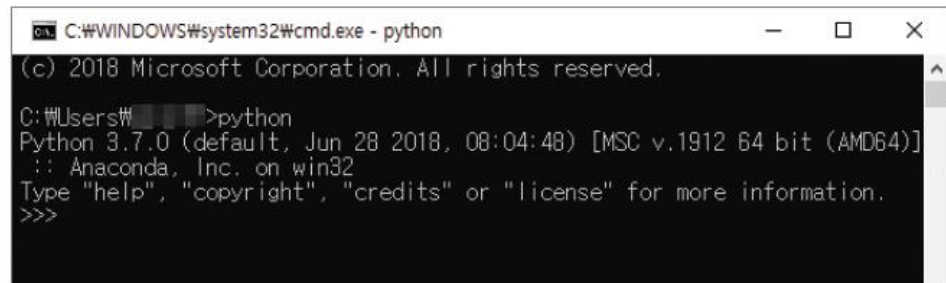
03. 파이썬 개발 환경과 설치

■ 파이썬 인터프리터 설치: Miniconda

6. 설치가 완료되면 <윈도> + <R> 키를 누른 후, 실행 창에서 'cmd'를 입력하고 <Enter> 키를 누른다. cmd 창이 열리면 'python'을 입력한 후 <Enter> 키를 눌러 파이썬이 작동하는지 확인한다.



(a) 실행 창



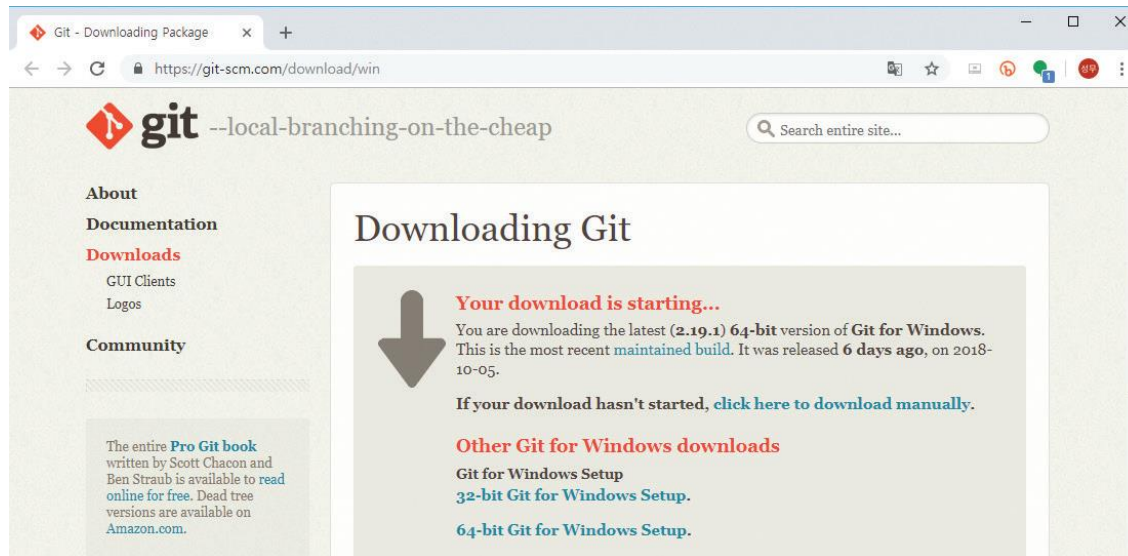
(b) cmd 창

[파이썬 인터프리터 실행]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

1. Atom은 플러그인 등의 업데이트를 위해 Git을 사용한다. Git의 다운로드 페이지(<https://git-scm.com/download/win>)에서 Git을 다운로드한 후, 인스톨 파일을 설치한다.

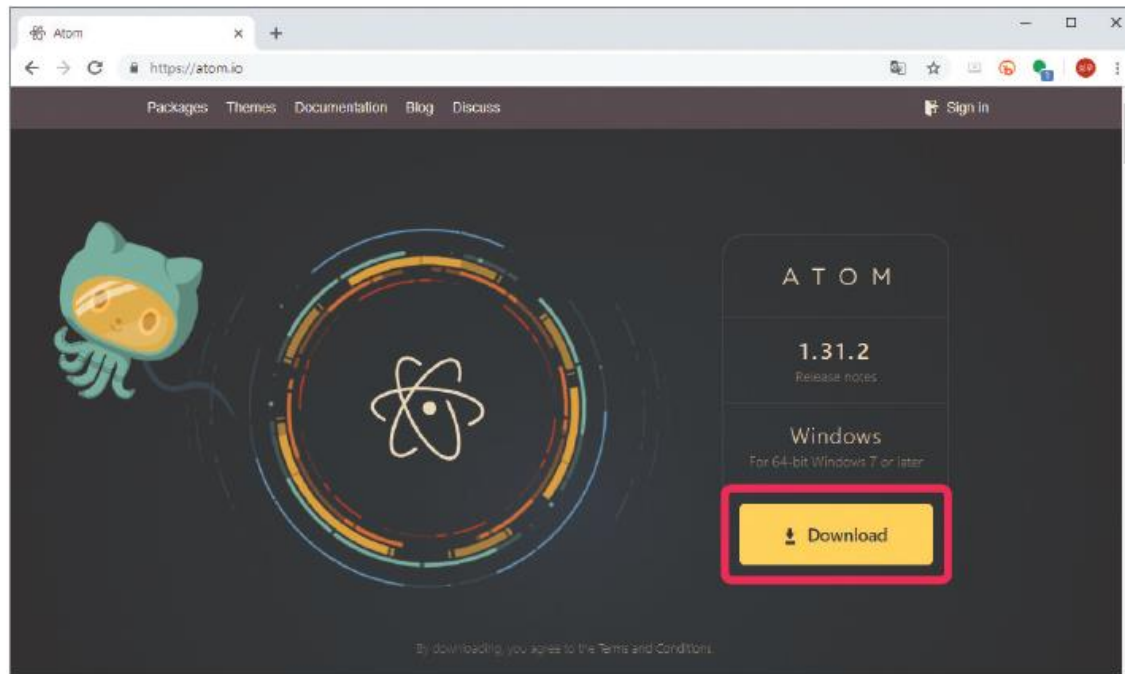


[Git 다운로드 페이지]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

2. Atom을 다운로드하기 위해 Atom 공식 홈페이지(<https://atom.io>)에서 [Download] 버튼을 클릭하여 설치 파일을 다운로드한다.

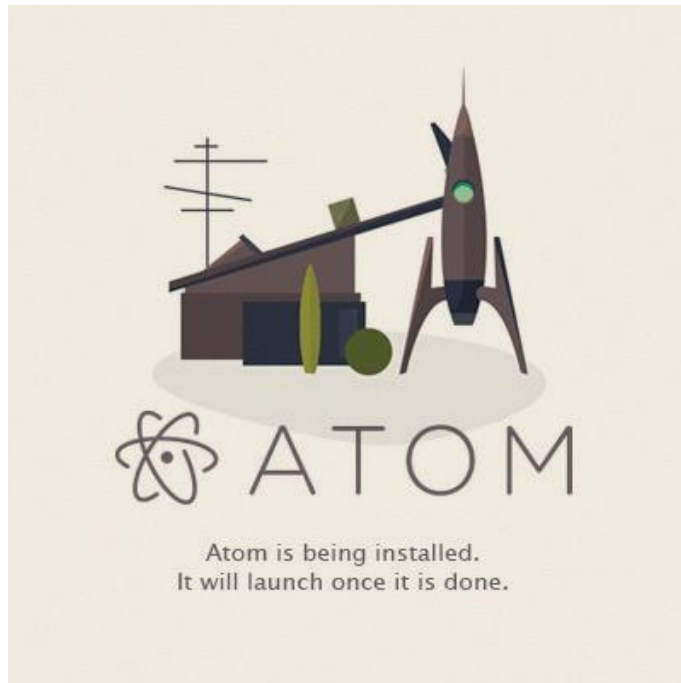


[Atom 홈페이지]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

3. 다운로드한 Atom 설치 파일을 실행하고, 설치를 진행한다. 설치 파일을 실행하면 다음 이미지가 나타나면서 설치가 시작된다.

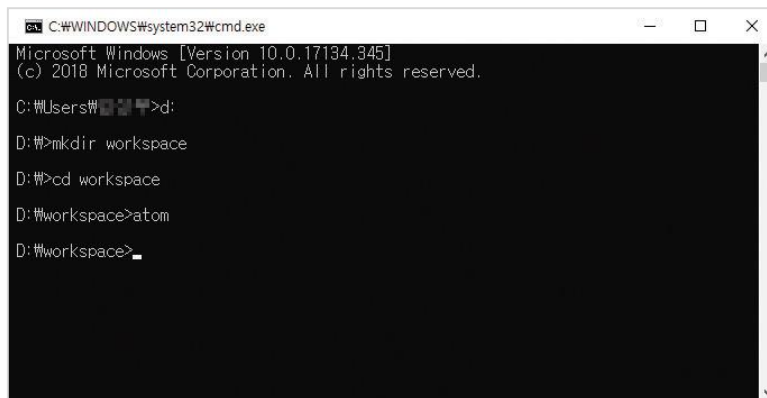


[Atom 설치 이미지]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

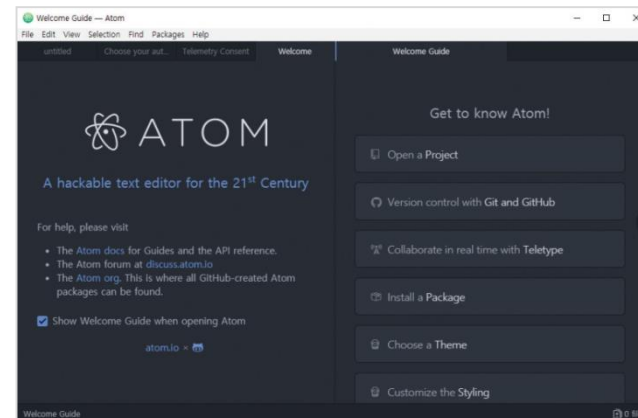
4. Atom으로 파이썬을 실행하는 방법은 몇 가지가 있는데, 이 책에서는 cmd 창에서 Atom을 실행하는 방법을 사용한다. 먼저 cmd 창을 실행한다. 일반적으로 프로그래밍 작업 환경을 만들기 위해 'workspace'라는 폴더를 만드는데, 이를 위해 cmd 창에서 'mkdir workspace'를 입력하고 <Enter> 키를 누르면, 새로운 'workspace' 폴더가 생성된다. 이후 'cd workspace' 명령어로 해당 폴더에 이동하여 'atom'을 입력하고 <Enter> 키를 누르면 Atom이 실행된다.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user>cd
D:\>mkdir workspace
D:\>cd workspace
D:\workspace>atom
D:\workspace>
```

[cmd 창에서 workspace 폴더 생성하고, Atom 실행]



[Atom 화면]

03. 파이썬 개발 환경과 설치

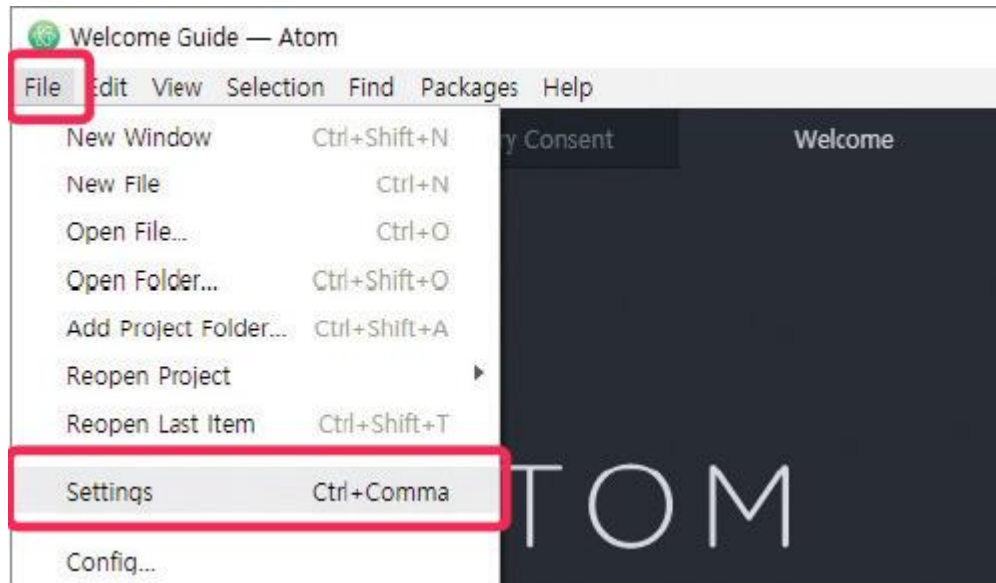
여기서  잠깐! 바탕화면의 아이콘을 클릭하여 Atom을 실행하는 방법

- 바탕화면의 단축 아이콘을 사용하여 Atom을 실행하는 방법이 가장 간단하지만, Atom의 특성상 특정 디렉터리(폴더)에서 실행하는 경우가 많아 단순히 단축 아이콘의 더블클릭으로 실행할 경우 어려움이 있을 수 있다. 따라서 cmd 창에서 실행하는 방법을 추천한다.

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

5. Atom에서 파이썬을 원활하게 사용하기 위해 두 가지 패키지를 설치하는데, 먼저 Atom을 실행하고 메뉴 바에서 [File]-[Settings]를 선택한다.



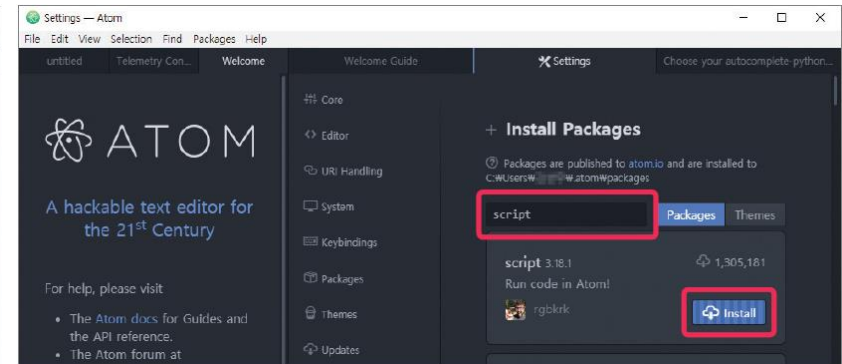
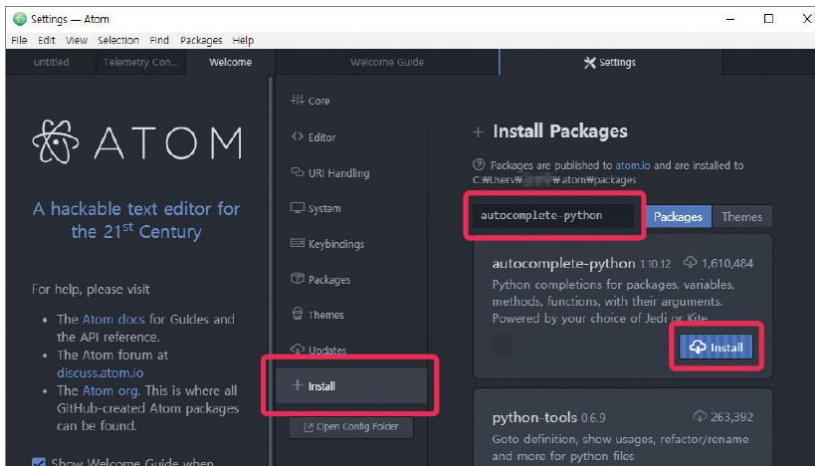
[[Git 다운로드 페이지](#)]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

6. [Install] 메뉴를 클릭하고, 검색 창에 다음 두 가지 패키지를 검색하여 설치한다

- autocomplete-python: 파이썬 코드를 자동 완성하는 코드
- script: 파이썬 등 코드를 실행하는 프로그램

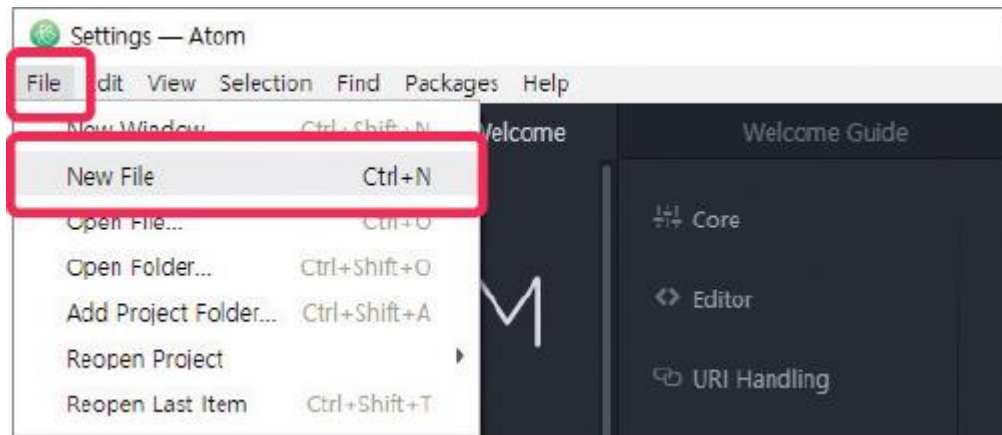


[패키지 설치: autocomplete-python, script]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

7. 두 가지 패키지를 설치한 후, 메뉴 바에서 [File]-[New File]을 선택하여 파이썬 파일을 생성한다.



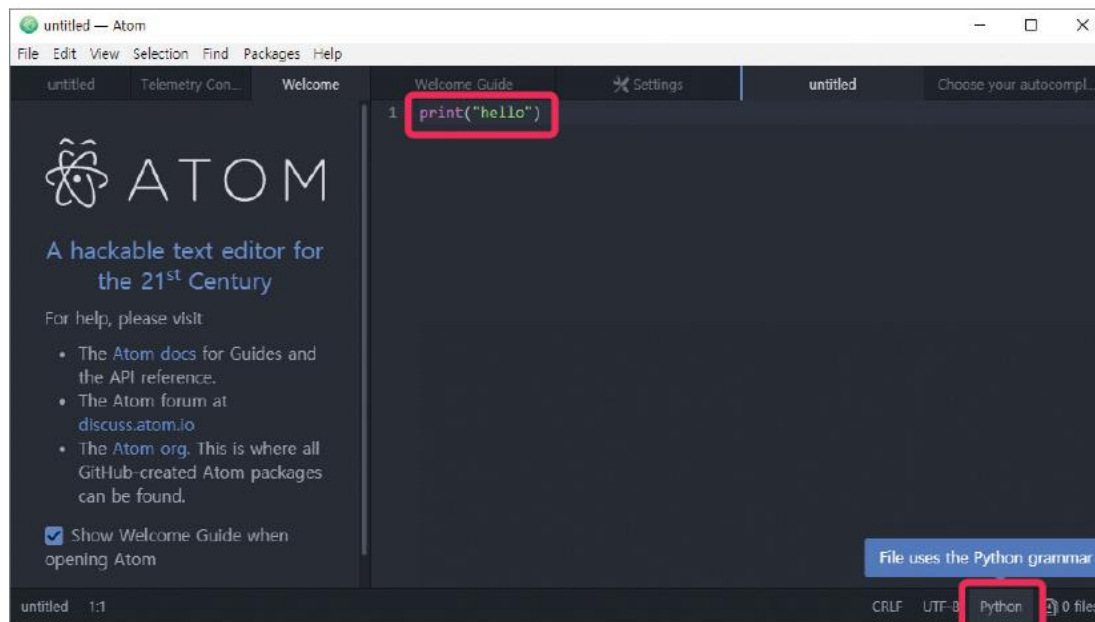
[새로운 파이썬 파일 생성]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

8. 오른쪽 하단에서 사용 언어를 'Python'으로 변경하고, 다음과 같은 간단한 코드를 작성한다.

```
print("hello")
```

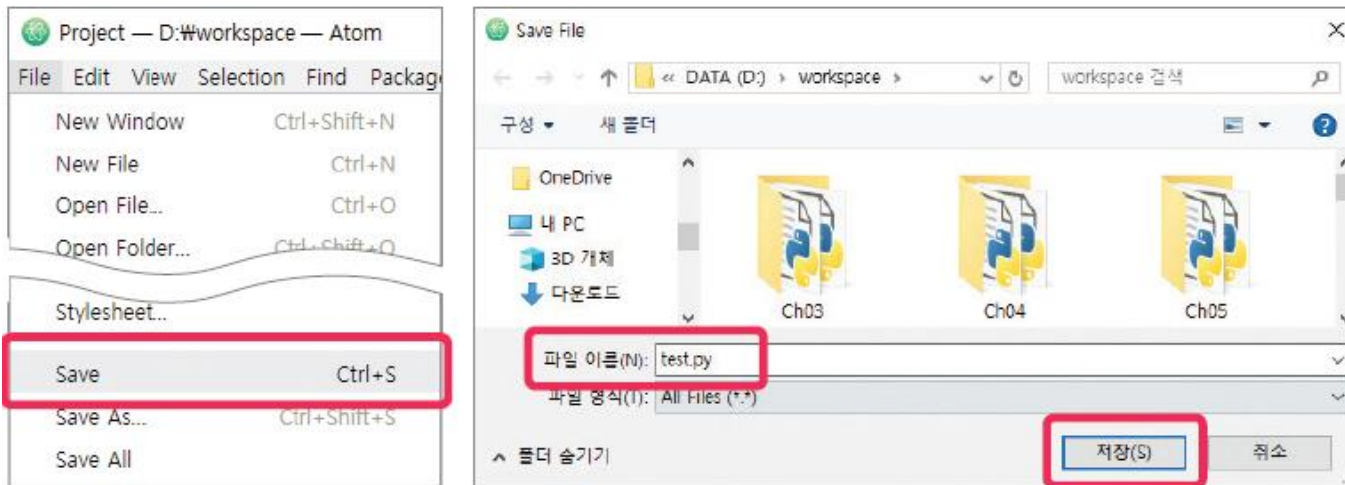


['Python'으로 언어 변경하고, 코드 작성]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

9. 메뉴바에서 [File]-[Save]를 선택한다. '파일 이름'을 'test.py'로 입력하고 저장한다.



[파일 저장]

03. 파이썬 개발 환경과 설치

■ 파이썬 코드 편집기 설치: Atom

10. cmd 창에서 파일을 저장한 위치로 이동한 후, 'python test.py'를 입력하여 코드를 실행한다.



```
D:\workspace>python test.py
hello
D:\workspace>
```

[저장한 'test.py' 코드 실행]

03. 파이썬 개발 환경과 설치

■ 파이썬 셸에서 코드 작성

- 파이썬 셸은 파이썬을 사용하기 위한 가장 기초적인 환경으로, 앞에서 언급한 것처럼 cmd 창에서 'python'이라는 명령어를 입력하면 실행된다.

① `윈도` + `R`

② 실행 창에서 'cmd'를 입력한 후, `Enter`

③ cmd 창에서 'python'을 입력한 후, `Enter`

03. 파이썬 개발 환경과 설치

여기서 잠깐! 파이썬 셸

- 일반적으로 프로그래밍 또는 컴퓨터 환경에서 '셸(shell)'이라고 하면, 인터프리터에 입력할 수 있는 인터페이스(interface)라고 생각하면 된다. 즉, 파이썬 셸은 파이썬 인터프리터에 명령어를 입력할 수 있는 환경을 말한다. 참고로, 리눅스 계열에서 '셸'은 리눅스 커널(Linux kernel)에 명령을 입력할 수 있는 환경을 말한다.

03. 파이썬 개발 환경과 설치

■ 파이썬 셸에서 코드 작성

- 파이썬 셸이 실행되면 현재 인터프리터에 대한 간단한 설명과 함께 '>>>'라는 입력 커맨드가 화면에 나타나는데, 여기에 다음과 같이 입력한다.

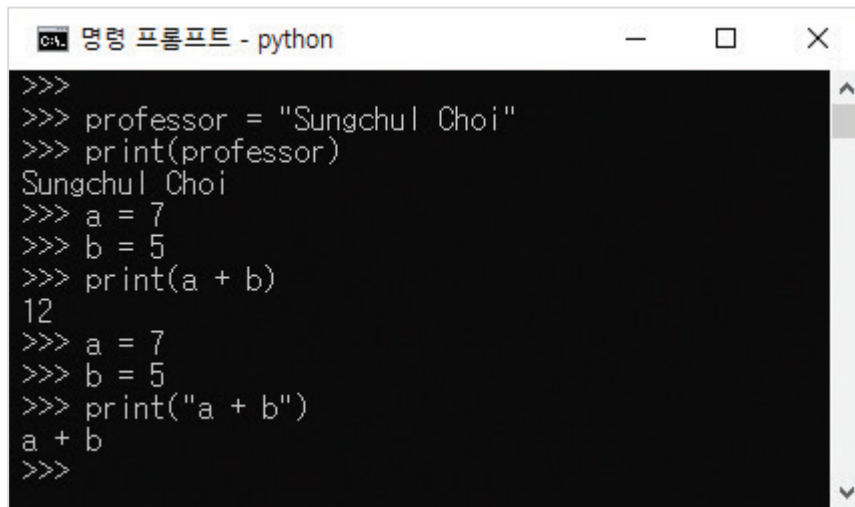
```
>>> professor = "Sungchul Choi"
>>> print(professor)
Sungchul Choi

>>> a = 7
>>> b = 5
>>> print(a + b)
12

>>> a = 7
>>> b = 5
>>> print("a + b")
a + b
```

03. 파이썬 개발 환경과 설치

■ 파이썬 셀에서 코드 작성

A screenshot of a Windows command prompt window titled "명령 프롬프트 - python". The window has a black background with white text. It shows a series of Python commands and their outputs. The commands are: >>>, >>> professor = "Sungchul Choi", >>> print(professor), >>> a = 7, >>> b = 5, >>> print(a + b), >>> a = 7, >>> b = 5, >>> print("a + b"), and >>>. The outputs are: Sungchul Choi, 12, and a + b. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
>>>
>>> professor = "Sungchul Choi"
>>> print(professor)
Sungchul Choi
>>> a = 7
>>> b = 5
>>> print(a + b)
12
>>> a = 7
>>> b = 5
>>> print("a + b")
a + b
>>>
```

[파이썬 셀에서 코드 작성 후, 실행]

Thank You !