

# 1) 파이썬과 알고리즘 트레이딩

여러분이 컴퓨터를 통해 자동으로 주식을 매수/매도하거나 주식 시장을 분석해서 유용한 정보를 제공하는 컴퓨터 프로그램을 만드는 방법에 대해서는 알려드릴 수 있습니다.

이를 통해 여러분은 여러분만의 투자 기법을 더 쉽게 개발할 수 있으며, 이를 잘 응용한다면 돈을 많이 벌게 될 수도 있습니다. 이 책을 통해 배운 내용을 바탕으로 뭘 할 수 있을지 잘 이해되지 않으신다고요? 다음과 같이 더 구체적인 시나리오를 생각해봅시다.

1) 내가 자는 새벽 동안 컴퓨터가 미국 증시의 패턴을 분석해 아침에 카카오톡으로 국내 주식 시장에 대한 투자 방향을 알려준다.

2) 최근 코스피/코스닥에서 거래량이 60 일 평균치에 비해 1,000%(10 배) 이상 급등한 종목들을 나에게 알려준다.

3) 내가 지정해둔 종목의 주식을 알아서 적정 매수가격에 도달하면 사고, 설정한 매도가격에 오면 다시 팔아준다.

4) 코스피/코스닥의 전 종목을 실시간으로 분석해 급등할 것 같은 종목들을 나에게 알려준다.

5) 주식 종목 추천 사이트의 무료 종목 추천을 매일매일 검색해 중복으로 추천된 종목을 나에게 미리 알려주고, 며칠 후에는 해당 종목들을 가상 투자했을 때의 수익률을 리포팅해준다.

이 책을 통해 알고리즘 트레이딩을 위한 프로그래밍 기초를 배우고 조금 더 공부하신다면 위의 시나리오뿐만 아니라 여러분이 생각하고 계신 아이디어를 그대로 컴퓨터가 실행할 수 있게 만들 수 있습니다. 그렇다면 이렇게 하기 위해서는 무엇을 공부해야 할까요?

# 1) 프로그래밍과 프로그래밍 언어

몇 년 동안 주식 투자를 해오신 분들은 나름대로 주식 투자를 위한 자신만의 노하우(Know-how)가 있을 겁니다. 예를 들어, 속칭 '상한가 따라잡기' 등과 같은 매수 기법도 일종의 노하우라고 할 수 있겠습니다. 컴퓨터와 관련된 학문에서는 노하우를 다른 말로 '알고리즘'이라고 부릅니다. 그리고 이러한 알고리즘을 컴퓨터가 알아들을 수 있도록 일련의 명령문으로 작성하는 것을 프로그래밍이라고 합니다.

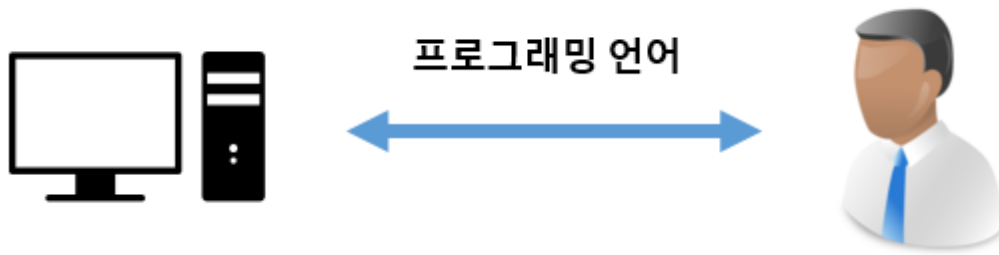
아직 잘 이해가 안 가시죠? 여러분이 누군가에게 여러분을 대신해 마트에서 물건을 사 오도록 지시하는 상황을 생각해 봅시다. 먼저 그 사람에게 가까운 마트에 어떻게 가는지(여러분만이 알고 있는 지름길이 있으시죠? 이게 바로 노하우, 즉 알고리즘입니다), 그리고 어떤 물건을 사서, 결제는 어떻게 하는지에 대해 알려줄 것입니다. 예를 들어 다음과 같이 말이죠.

1. 일단 마을버스 777 번을 타세요.
2. 3 개의 정거장을 지난 후 하차하세요.
3. 정거장 우측으로 100m 를 직진한 후 A 마트로 가세요.
4. 마트의 2 층 식품 판매대에서 B 를 사세요.
5. 1 층 결제 코너에서 C 카드를 통해 결제하세요.
6. 길을 건너서 맞은편 버스 정류장에서 777 버스를 타세요.
7. 세 개의 정거장을 지난 후 하차하세요.

컴퓨터에게 어떤 일을 시킬 때도 위와 같은 형태로 지시 내용을 나열해야 합니다. 이처럼 명령어가 나열된 것을 프로그램이라고 부르며, 프로그램을 작성하는 행위를 프로그래밍이라고 합니다. 어떤가요?

프로그래밍이라는 것이 여러분의 삶과 멀리 떨어진 이야기가 아니죠? 여러분은 이미 일상생활에서 프로그래밍을 해왔던 것입니다. 다만 이러한 지시 사항을 컴퓨터에게 이야기하는 것이 아니라 여러분의 가족이나 동료, 즉 사람에게 지시하거나 부탁해왔던 것이죠.

위와 같은 부탁이나 지시를 사람에게 할 때는 여러분과 지시를 받는 사람이 모두 알아들 수 있는 한국어를 이용하면 되겠죠? 프로그램을 작성할 때도 명령을 내리는 여러분과 지시를 받는 컴퓨터가 모두 알아들을 수 있는 언어가 필요한데, 이를 프로그래밍 언어라고 합니다. 위의 예처럼 마트에 가서 물건 사는 법을 여러분의 친구에게 말했다면 그것이 바로 프로그램이고, 여러분이 한국어로 말했다면 한국어가 바로 프로그래밍 언어가 되는 것입니다.



**그림 1.1 프로그래밍 언어의 역할**

그런데 사람이 아닌 컴퓨터에게 명령을 내리려면 어떻게 해야 할까요? 일단 앞에서 배운 것처럼 지시 사항을 순서대로 나열해야 합니다. 보통 이러한 작업을 명령어를 기술한다고 표현합니다. 이때 중요한 것은 명령어를 기술할 때 여러분과 컴퓨터 모두가 알아들을 수 있는 언어를 사용해야 한다는 점입니다.

우리가 사는 지구에 한국어, 영어, 중국어, 일본어 등 다양한 언어가 존재하는 것처럼 컴퓨터가 알아들을 수 있는 컴퓨터 프로그래밍 언어도 다양합니다. 이 책에서는 그중에서 파이썬(Python)이라는 언어를 배울 것입니다.

갑자기 프로그래밍 언어 이야기를 하니 벌써 걱정이 되고 머리가 아프죠? 걱정하지 마세요. 파이썬은 프로그래밍 언어 중 가장 배우기가 쉬워서 프로그래밍을 전공하지 않는 분들도 쉽게 배울 수 있습니다. 프로그래밍을 배우지 않은 분들이 시스템 트레이딩을 하고자 한다면 저는 다른 어떤 언어보다 파이썬을 추천해 드립니다. 파이썬은 엑셀을 기반으로 하는 VBA 보다 배우기가 쉽고, C/C++와 같은 언어의 난이도와는 비교할 수 없을 만큼 쉽습니다.

## 2) 주식투자

주식투자의 목적은 사람마다 다르겠지만 아마도 대부분의 사람은 그림 1.2와 같이 돈을 벌기 위해서일 것입니다. 쉽게 말하면 재테크의 한 수단이라는 것입니다.

부동산 투자를 통한 재테크는 집값 상승을 일으켜 사회적 약자에게 상실감을 주거나 경제를 심하게 왜곡할 수 있습니다. 이와 달리, 주식투자는 여러분이 주식투자를 통해 이익을 보거나 손해를 본다고 해서 주변 사람에게 피해가 가는 것은 거의 없습니다. 물론 가족들의 생활비를 주식투자로 모두 탕진해 버린다면 이야기가 달라집니다. 다시 말해, 주식투자는 사람들이 생각하는 것보다 상당히 건전하며, 주위 사람들에게 피해를 주지 않는 건전한 재테크의 한 방법이라고 할 수 있습니다.

또한 사람들이 잘 보지도 않던 경제 뉴스를 챙겨 읽게 하고, 사회에서 일어나는 여러 가지 일들에 대해서도 좀 더 관심 있게 지켜보게 하는 좋은 역할도 합니다. 그러나 문제는 개인 투자자들이 주식시장에서 돈을 벌기가 생각만큼 쉽지 않다는 것이죠. 왜 개인투자자들은 주식시장에서 돈을 벌기가 어려울까요?

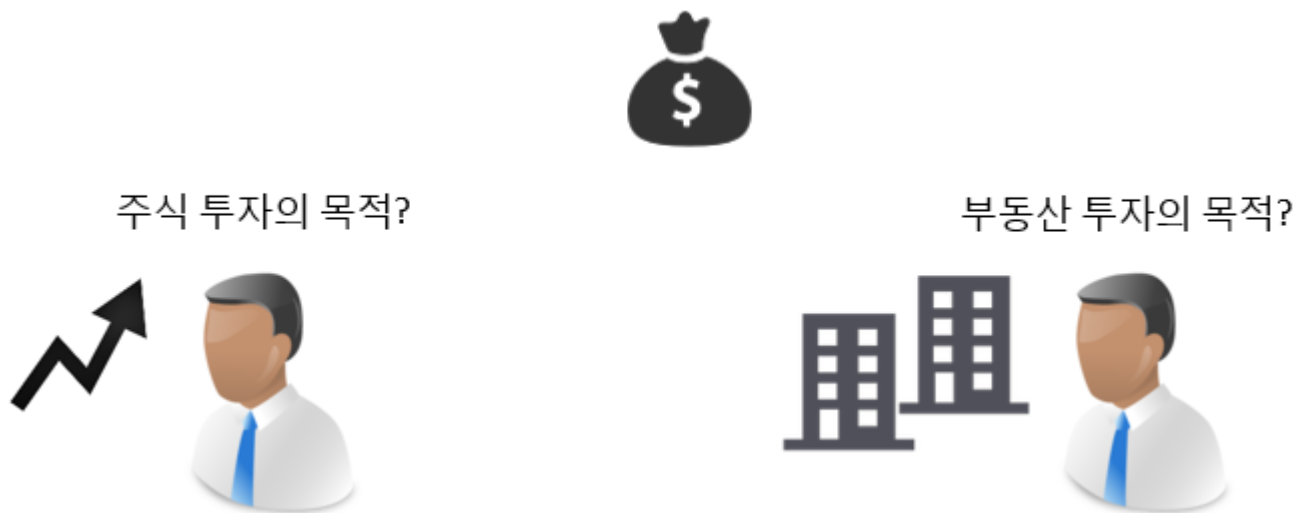


그림 1.2 주식과 부동산 투자의 목적

여기에는 여러 가지 요인이 있겠지만 우선 개인 투자자들은 기관 투자자나 외국인보다 얻을 수 있는 정보가 한정적이라는 것이 첫 번째 이유입니다. 아시는 분도 있겠지만 일부 회사들은 회사의 정보를 미리 기관 투자자나 외국인에게 불법적으로 제공하기도 합니다. 가끔은 증권방송에서 일부 애널리스트들이 특정 주식 종목을 추천하지만 정작 자신들은 이미 매수를 해놓고 기다리다가 일반 투자자들이 매수를 시작하면 바로 해당 주식을 매도해서 수익을 올리기도 합니다. 이러한 요인은 모두 개인 투자자의 역량과는 상관없이 환경적으로 발생하는 외적 요인이라 할 수 있습니다.

개인 투자자들이 주식투자에 실패하는 또 다른 이유로는 개인 투자자들은 주식을 생업으로 삼고 투자하는 경우가 많지 않아 실제로 주식 투자에 대해 공부하기보다는 무작정 '특정 종목이 오를 것 같다', '이 종목은 충분히 저점이다'라고 판단하거나, 또는 주변의 누군가로부터 추천받은 종목을 아무런 사전 조사 없이 사는 경향이 강하다는 점입니다. 이러한 것들은 모두 개인과 관련된 내적 요인이라 할 수 있습니다.

이 같은 내적, 외적 요인을 극복하고 '용돈 벌이', '재테크'를 목적으로 주식 투자를 하는 개인 투자자들은 어떻게 해야 거대 기관 투자자나 외국인 사이에서 주식투자에 성공할 수 있을까요?

이 질문의 답은 여러 가지가 있겠지만 저자는 이 책에서 다룰 '알고리즘 트레이딩'이 하나의 답이 될 수 있다고 생각합니다. 개미들이 시장에서 성공할 수 없는 이유를 다음과 같은 두 가지로 요약했을 때 알고리즘 트레이딩을 이용하면 이러한 문제를 어느 정도 해결할 수 있기 때문입니다.

- 정보의 부족
- 시간의 부족

### 3) 알고리즘 트레이딩

---

그렇다면 '알고리즘 트레이딩'이란 무엇일까요? 일단 트레이딩은 한국어로 '거래'를 의미하고, 알고리즘은 주로 컴퓨터 분야에서 많이 사용하는 단어로 '어떤 문제를 해결하기 위한 절차나 방법'을 의미합니다. 즉, 알고리즘 트레이딩이란 '컴퓨터를 이용해 어떤 일정한 알고리즘에 따라 자동으로 거래하는 매매 방식'을 의미합니다.

앞에서도 간단히 예를 들어 설명했지만 알고리즘 트레이딩의 구체적인 시나리오에 대해 한 번 더 함께 생각해 봅시다. 먼저 컴퓨터가 주식 시장 개장 전에 전날의 거래 패턴(거래량, 증가, 시가, 주식 가격의 변동 폭)을 분석하고 이를 통해 여러분에게 투자 유망 종목을 자동으로 추천해 줄 수 있습니다. 또는 더 나아가 여러분이 낮에 일하는 동안 컴퓨터가 자동으로 주식을 매수하고, 매도하면서 수익을 낼 수도 있습니다.

최근 정기적금의 은행이자가 연 3% 이하임을 고려할 때, 여러분이 잘 만든 시스템이 하루에 15% 이상의 수익을 낼 수도 있는 것입니다. 물론 온종일 사고팔고를 잘한다면 그 이상의 수익률도 이론적으로 불가능하지는 않습니다.

알고리즘 트레이딩의 개념에 대해 알아보았으니, 그럼 이번에는 어떻게 해야 컴퓨터로 저런 것들을 만들 수 있을지 알아보시다. 효율적인 '프로그램 매매'를 위해서는 무엇보다도 여러분만의 알고리즘(투자 노하우)이 있어야 합니다.

예를 들어, 여러 주식 책을 보면 거래량 분석을 통해 다수의 세력이 장기간에 걸쳐서 매집하는 종목을 찾는 알고리즘이 있습니다. 이 알고리즘은 몇 개월 동안 거래량을 분석하고 있다가 특정 종목의 일일 거래량이 최근의 평균 거래량보다 1,000% 이상 급등하는 종목을 찾습니다. 이처럼 단기간의 거래량이 평균 거래량보다 폭발적으로 늘어나는 종목은 단기간에 급등할 수 있다는 것이 주요 아이디어입니다.

문제는 국내 주식 시장(코스피, 코스닥)의 주식 종목이 2,000 개 정도 되기 때문에 매일 퇴근한 후 이런 종목들을 자세히 조사하고 투자 계획을 세우기가 어렵다는 것입니다. 회사에서 이런 일을 한다면 아마 여러분은 당장 해고될 것입니다.

이 문제에 대한 간단한 해결책은 그런 일들은 컴퓨터가 대신 하게끔 시키면 된다는 것이죠. 사실 컴퓨터는 스스로 생각하거나 판단하는 것은 사람보다 못하지만(물론 아직은 그렇다는 애깁니다) 다수의 데이터를 반복적으로 처리하는 것은 사람보다 월등히 잘합니다. 그리고 여러분의 인건비에 비하면 컴퓨터를 사용하는 데 드는 비용은 구매 비용과 전기세 정도밖에 안 됩니다.

알고리즘이 준비되면 다음 단계는 프로그래밍 언어를 이용해 여러분의 '알고리즘'을 컴퓨터가 알아들을 수 있게 프로그래밍하면 됩니다.

## 4) 파이썬이란?

---

파이썬은 프로그래밍 언어의 일종이라고 앞에서 간단히 살펴봤습니다. 파이썬이 프로그래밍 언어라는 사실을 아는 걸로도 충분하겠지만 다른 누군가에게 파이썬에 대해 좀 더 자세히 설명해줄 수 있다면 좋겠죠? 이를 위해 조금 지루할 수 있겠지만 파이썬이라는 프로그래밍 언어의 역사와 특징에 대해 좀 더 알아보겠습니다.

파이썬은 1991년 귀도 반 로섬(Guido van Rossum)이 만든 배우기 쉽고 강력한 프로그래밍 언어입니다. 곧 알게 되겠지만 파이썬은 C, C++, 자바(Java) 같은 프로그래밍 언어보다 훨씬 배우기 쉽고 문법도 간단합니다.

또한 파이썬은 C, C++, 자바보다 약 1/3 또는 1/5 정도의 코드만으로도 같은 기능을 수행할 수 있어 개발 생산성이 매우 높습니다. 쉽게 말해서 컴퓨터에게 대충 명령을 내려도 컴퓨터가 알아서 일을 잘 처리하기 때문에 C, C++, 자바보다 프로그래밍하는 시간이 줄어든다는 것입니다. 같은 알고리즘을 C, C++ 프로그래머가 구현해서 트레이딩 시스템을 구축하는 것보다 파이썬을 이용했을 때 개발 시간을 크게 단축할 수 있다는 뜻입니다.

파이썬이라는 이름은 귀도 반 로섬이 좋아하는 코미디 프로그램이던 "Monty Python's Flying Circus"에서 따온 것이라고 합니다. 하지만 파이썬이라는 이름의 뱀도 있어서 실제로 파이썬 프로그래밍 책을 보면 표지에 뱀이 그려진 것이 많습니다. 그리고 파이썬과 관련된 프로그램의 아이콘도 뱀으로 돼 있는 것을 종종 볼 수 있습니다. 그렇다면 파이썬을 개발한 귀도 반 로섬은 요즘 무슨 일을 하고 있을까요?

<https://www.python.org/~guido/>

위의 URL로 들어가 보면 파이썬 공식 홈페이지에 귀도를 위한 페이지가 있는 것을 볼 수 있습니다. 뭔가 창시자에 대한 예우처럼 보이지 않습니까? 귀도는 파이썬 언어의 발전에 큰 역할을 해오고 있는데, 2013년까지는 구글(Google)에서 일했고 최근에는 드롭박스(Dropbox)로 회사를 옮겼습니다. 중요한 점은 구글과 드롭박스에서 약 50%의 시간에는 회사를 위해 일하고, 나머지 50%는 파이썬을 개발하는 데 시간을 보내고 있다고 합니다. 그만큼 세계 최고 수준의 프로그래머에 대한 회사 차원의 배려가 아닌가 합니다. 물론 구글과 드롭박스의 많은 부분이 파이썬으로 개발되고 있어서 파이썬 자체의 발전을 위한 투자로도 생각해볼 수 있을 것 같습니다.

## 2) 파이썬 설치

본격적으로 파이썬이라는 프로그래밍 언어로 프로그래밍을 공부하기에 앞서 이번 절에서는 파이썬을 설치해보겠습니다. 이 책에서는 마이크로소프트 윈도우 7 환경에서 파이썬을 설치하는 경우만 다루겠지만 파이썬은 윈도우 버전에 크게 상관없이 설치할 수 있습니다. 또한 리눅스나 맥 OS X 과 같은 시스템에도 쉽게 설치할 수 있습니다.

파이썬을 설치하려면 먼저 설치 프로그램을 내려받아야 하는데, 파이썬 설치 파일은 크게 두 가지로 구분됩니다.

- 파이썬 공식 홈페이지에서 제공하는 설치 파일
- 배포판이라고 불리는 설치 파일

첫 번째는 파이썬 공식 홈페이지(<http://www.python.org>)에서 제공하는 설치 파일로, 이를 CPython 이라고 합니다. 해당 파일을 내려받아 설치하면 파이썬 인터프리터와 몇 가지 프로그램이 설치됩니다. 이 설치 파일은 파이썬으로 프로그래밍할 수 있도록 최소한의 프로그램을 컴퓨터에 설치합니다. 그러므로 설치 파일의 용량도 23MB 정도로 작은 편입니다.

두 번째는 배포판이라고 불리는 설치 파일입니다. 뒤에서 설명하겠지만 파이썬에는 파이썬 패키지라고 하는 것들이 있는데, 여기서 패키지라는 것은 특정 분야에 최적화된 프로그램의 묶음이라고 보면 됩니다. 예를 들어, 주식 프로그램을 작성하는데 처음부터 모든 것을 다 작성하는 것이 아니라 주식 패키지를 설치하면 기본적인 것들이 이미 구현돼 있는 것이죠. 어떨까요? 말로만 들어도 편하겠지요? 좀 더 쉽게 설명하면 어떤 보고서를 작성하는데, 이미 누군가가 그 보고서의 초안을 작성한 후 제공하는 것과 비슷합니다.

그런데 문제는 파이썬에는 너무나 많은 패키지가 있어서 필요한 패키지를 그때마다 내려받아서 직접 설치해야 한다는 것이죠. 그리고 요즘은 상황이 좀 더 나아졌지만 파이썬 버전에 따라 일부 패키지는 초보자가 설치하기에는 좀 어려운 경우가 있습니다.

이러한 문제를 깔끔하게 해결하고자 배포판이라고 하는 설치 파일들이 등장했습니다. 쉽게 말해서 배포판은 '파이썬 인터프리터 + 파이썬 패키지'라고 생각하면 됩니다. 배포판은 파이썬 인터프리터와 자주 사용하는 백여 개의 패키지를 한 번에 컴퓨터에 설치해줍니다. 초보자 입장에서 보면 배포판은 매우 편리한 설치 파일의 한 종류입니다. 다만 설치 파일의 크기가 200MB~1GB 정도로 좀 크다는 단점이 있습니다. 물론 둘 중 어떤 설치 파일로 설치하든 파이썬 프로그램을 작성하는 데는 문제가 없지만 이 책에서는 여러 유용한 패키지까지 한 번에 설치할 수 있는 '아나콘다'라는 이름의 배포판을 이용할 것입니다. 현재 많이들 사용하는 배포판을 표 1.1 에 정리했습니다.

표 1.1 파이썬 배포판

배포판 이름	주소
Anaconda	<a href="https://store.continuum.io/cshop/anaconda/">https://store.continuum.io/cshop/anaconda/</a>
Python(x,y)	<a href="https://code.google.com/p/pythonxy/">https://code.google.com/p/pythonxy/</a>



배포판 이름	주소
Enthought Canopy	<a href="https://www.enthought.com/products/epd/">https://www.enthought.com/products/epd/</a>
WinPython	<a href="http://winpython.github.io/">http://winpython.github.io/</a>

# 1) 아나콘다 설치 파일 다운로드

파이썬의 아나콘다는 Continuum Analytics 라는 곳에서 만든 파이썬 배포판으로, 445 개 정도의 파이썬 패키지를 포함하고 있습니다. 또한 상업용으로도 무료로 사용할 수 있으므로 회사 내에서도 사용할 수 있다는 장점이 있습니다.

먼저 아나콘다 배포판을 설치하기 위해 다음 URL 로 이동합니다. 파이썬은 현재 2.7 버전과 3.5 버전이 있는데, 이 책에서는 파이썬 3.5 를 사용합니다.

<https://www.continuum.io/downloads>

파이썬 3.5 버전에도 다시 윈도우 64 비트용 설치 파일과 32 비트용 설치 파일이 있습니다. 보통은 사용 중인 PC 에 64 비트 윈도우가 설치돼 있으면 64 비트용 설치 파일을 내려받고, 32 비트 윈도우가 설치돼 있다면 32 비트용 설치 파일을 내려받으면 됩니다.

그러나 이 책의 후반부에서 다룰 국내 증권사 API 를 정상적으로 사용하려면 반드시 그림 1.3 과 같이 32 비트용 설치 파일을 클릭해서 내려받아야 합니다. 특별한 경우가 아니라면 파이썬 3.5 32 비트용 설치 파일을 내려받기 바랍니다.

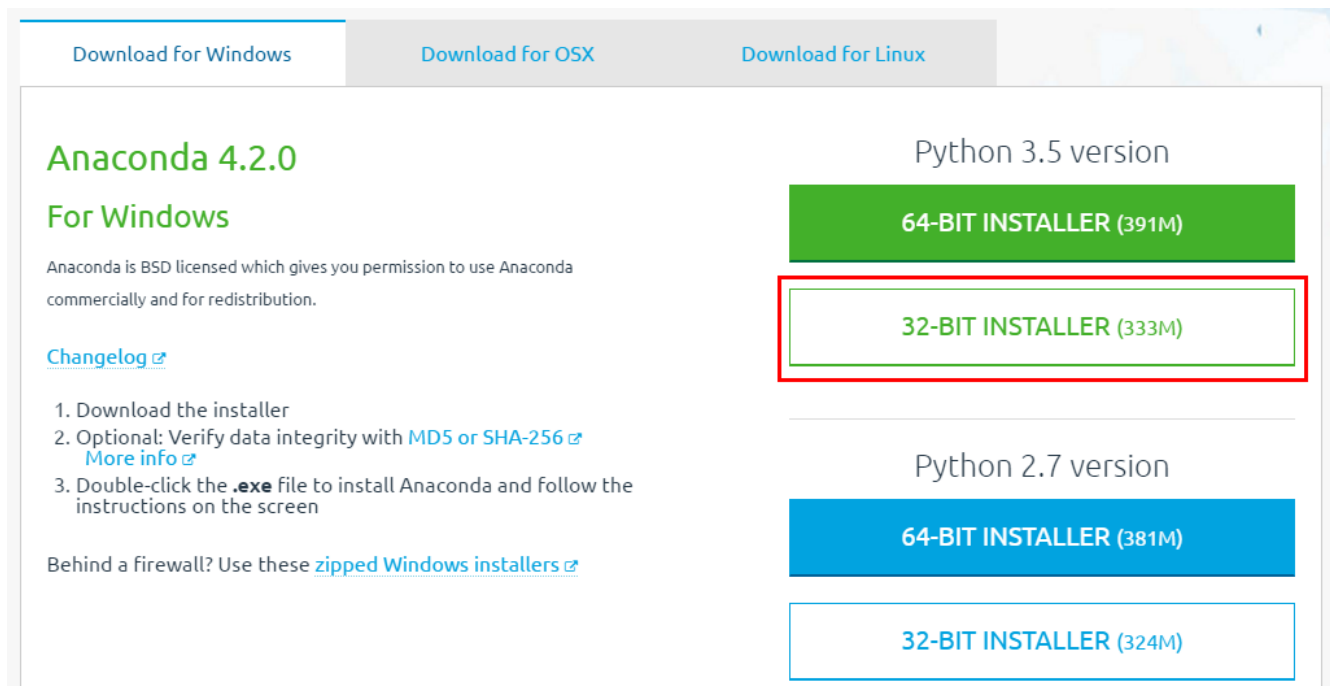


그림 1.3 아나콘다 32 비트 배포판 설치 파일 선택

## 2) 아나콘다 설치

1.2.1 절을 통해 정상적으로 아나콘다 32 비트 배포판 설치 파일을 내려받았다면 해당 파일을 더블 클릭해서 설치를 시작합니다. 그림 1.4 와 같이 첫 번째 설치 화면이 나오면 'Next' 버튼을 클릭해 다음 화면으로 진행합니다.

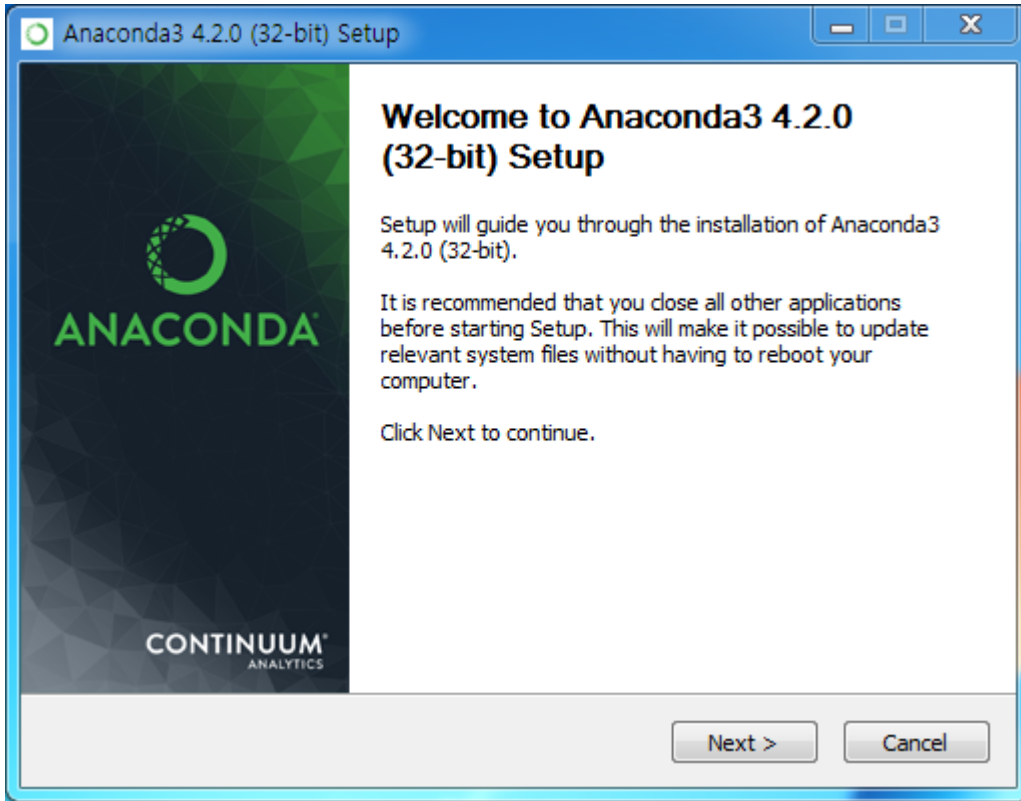


그림 1.4 아나콘다 설치(1/7)

그림 1.5 는 라이선스 동의에 관한 내용입니다. 'I Agree' 버튼을 클릭해 다음 설치 단계로 이동합니다.

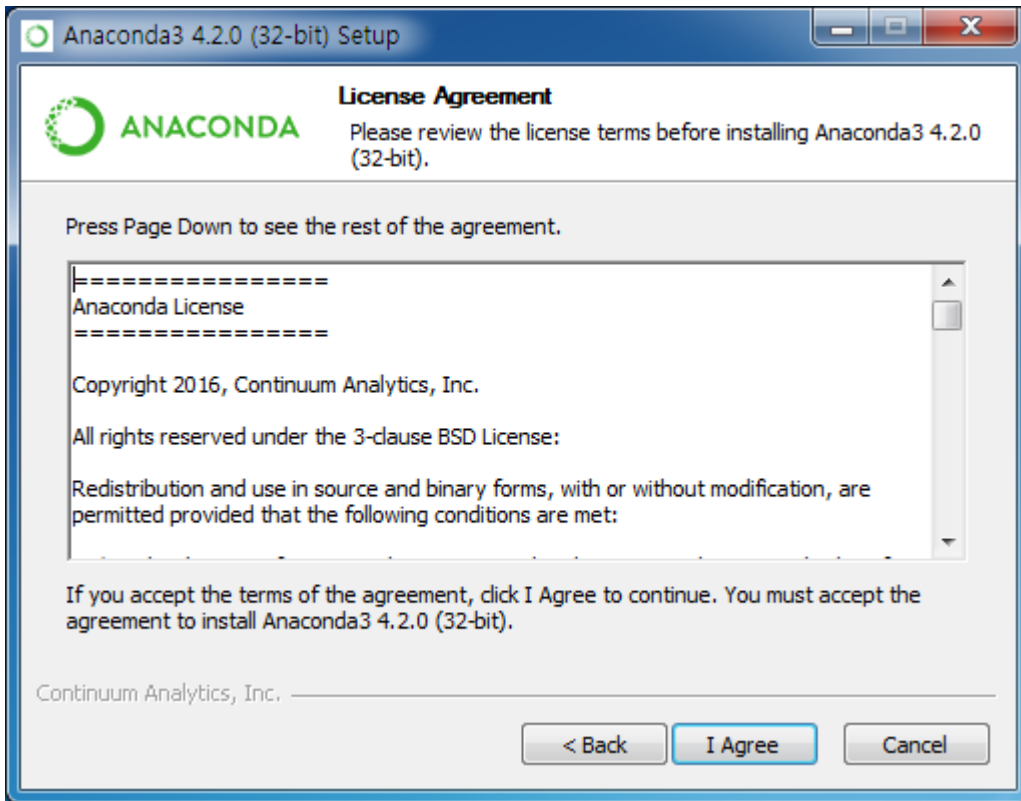


그림 1.5 아나콘다 설치(2/7)

설치 유형을 선택하는 단계에서는 'All Users'를 선택한 후 'Next' 버튼을 클릭합니다.

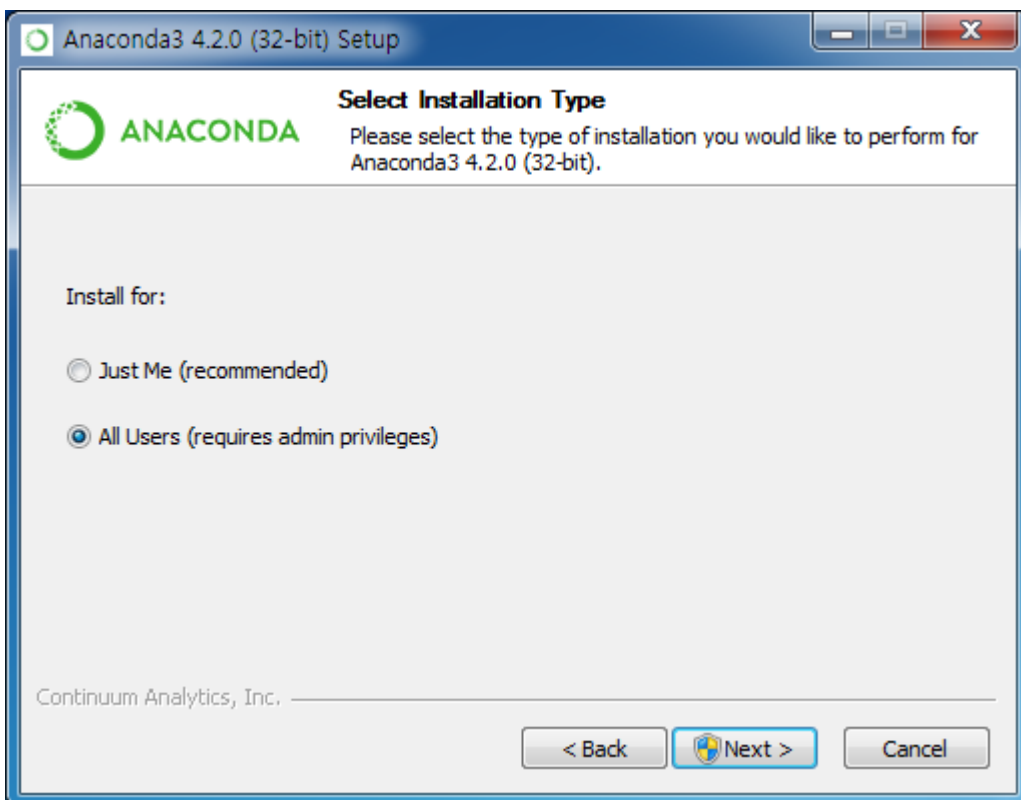


그림 1.6 아나콘다 설치(3/7)

아나콘다 배포판이 설치될 디렉터리를 선택하는 화면에서는 Destination Folder 를 'C:\Anaconda3'으로 변경한 후 'Next' 버튼을 클릭합니다.

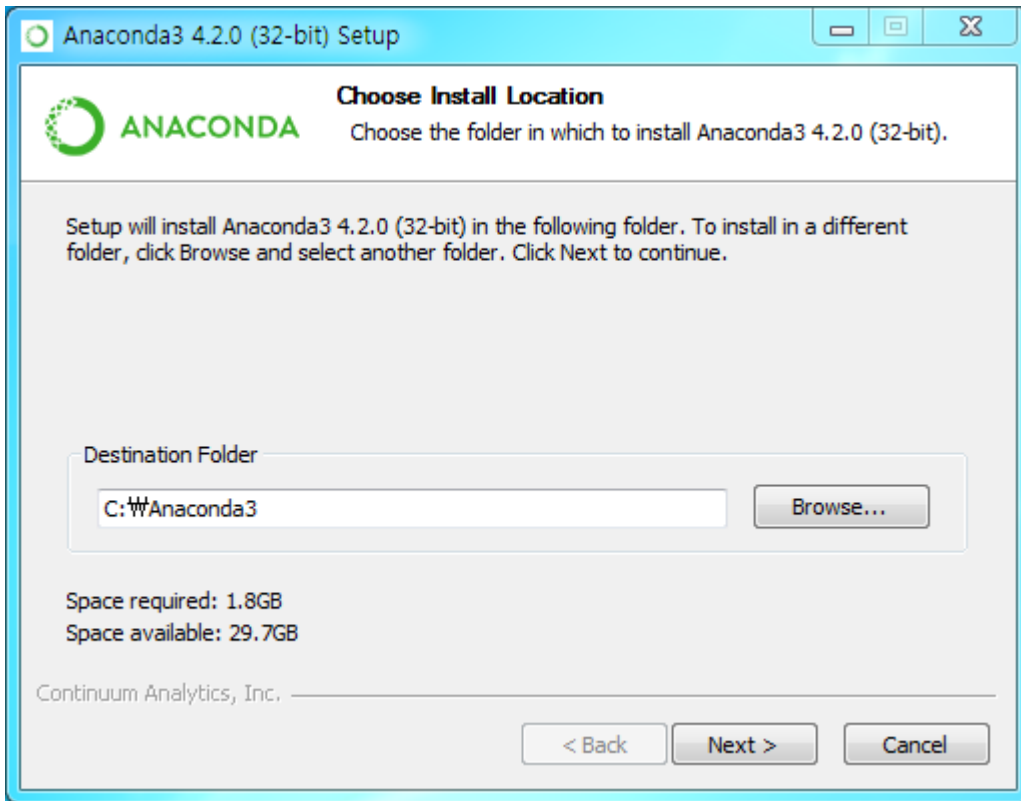


그림 1.7 아나콘다 설치(4/7)

추가 옵션 선택 화면에서 그림 1.8 과 같이 두 가지 옵션을 모두 선택한 후 'Install' 버튼을 클릭해 다음 단계로 이동합니다.

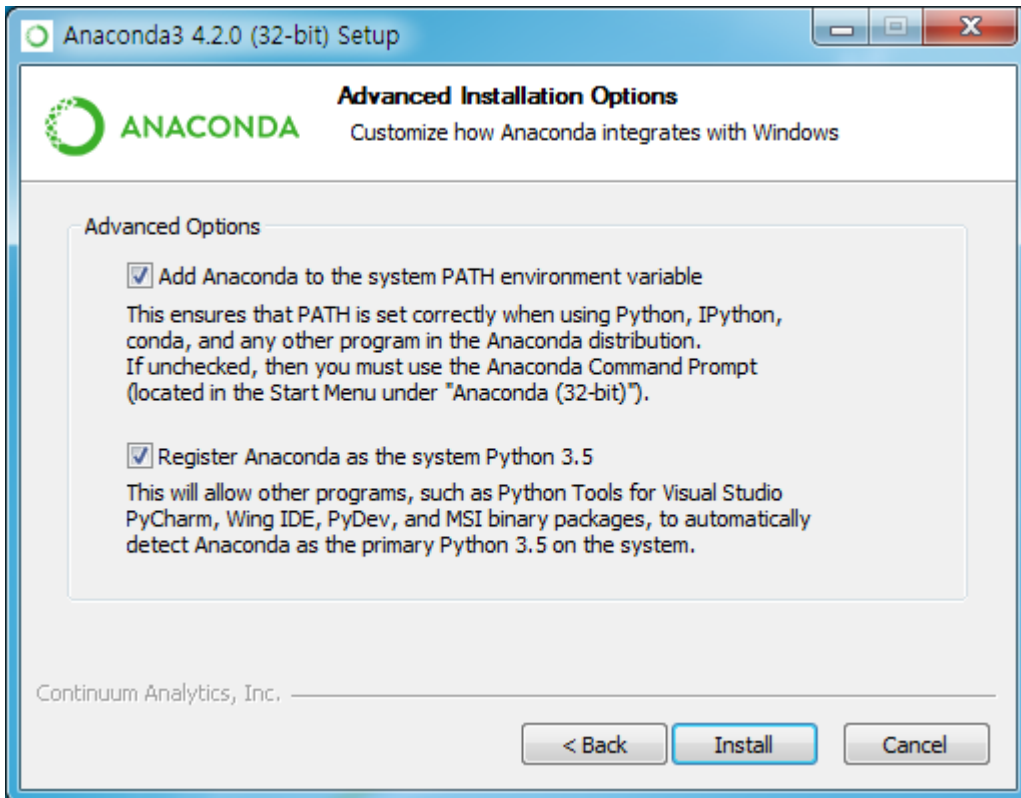


그림 1.8 아나콘다 설치(5/7)

모든 설치가 완료되면 화면에 'Completed'라는 메시지가 출력됩니다. 이때 'Next' 버튼을 눌러 다음 단계로 이동합니다.

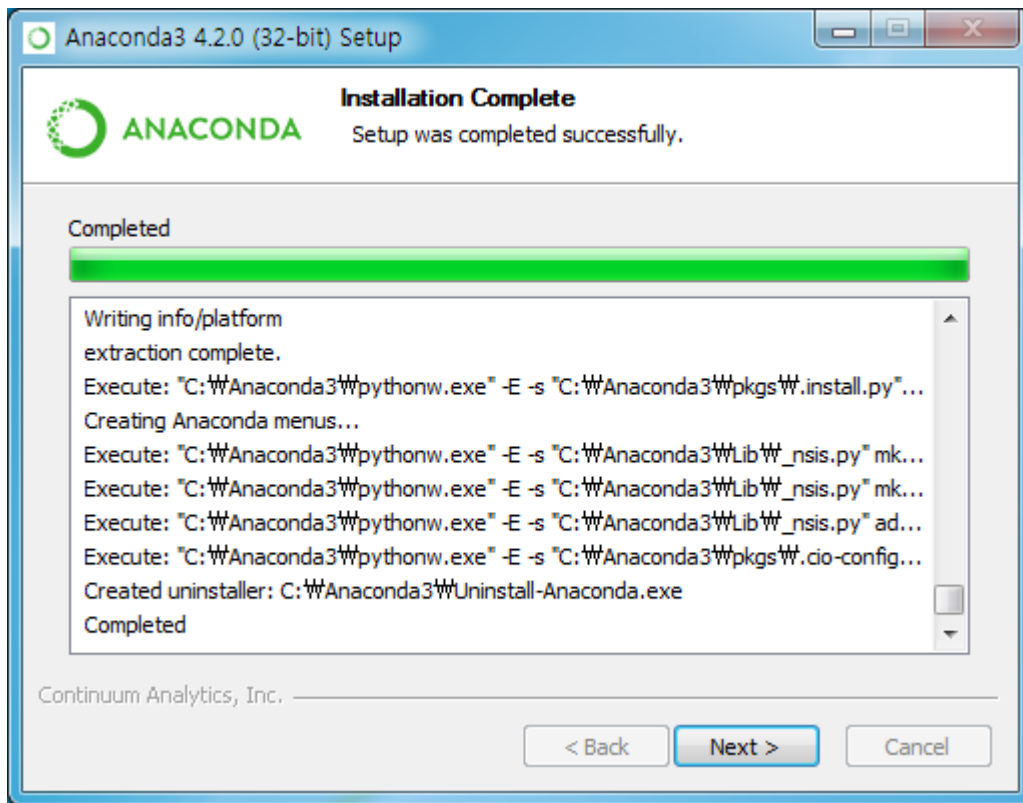


그림 1.9 아나콘다 설치(6/7)

아나콘다 배포판의 모든 설치 과정이 완료됐습니다. 그림 1.10 과 같이 'Finish' 버튼을 클릭해 설치 과정을 종료합니다.



그림 1.10 아나콘다 설치 (7/7)

### 3) 파이썬 인터프리터 실행

1.2 절을 통해 PC 에 정상적으로 '아나콘다'라는 이름의 파이썬 배포판을 설치했습니다. 아마도 프로그래밍이란 것을 한 번도 해보시지 않는 분들은 배우기도 전에 뭔가 복잡한 설치 과정이 있어 조금 당황스러우셨을 겁니다. 그렇지만 옛말에 '시작이 반'이라고 했습니다. 조금만 참고 따라오시면 재미있는 프로그래밍의 세계로 입문하실 수 있을 것입니다.

여러분은 아나콘다 배포판을 'C:\Anaconda3' 경로에 설치했습니다. 먼저 윈도우의 탐색기 등을 사용해 해당 경로로 이동해 봅시다. 그림 1.11 과 같이 해당 경로에 여러 파일과 디렉터리가 설치돼 있음을 확인할 수 있습니다. 이 경로에서 Lib 디렉터리로 이동한 후 다시 idlelib 디렉터리로 이동합니다. 최종 경로는 다음과 같습니다.

C:\Anaconda3\Lib\idlelib

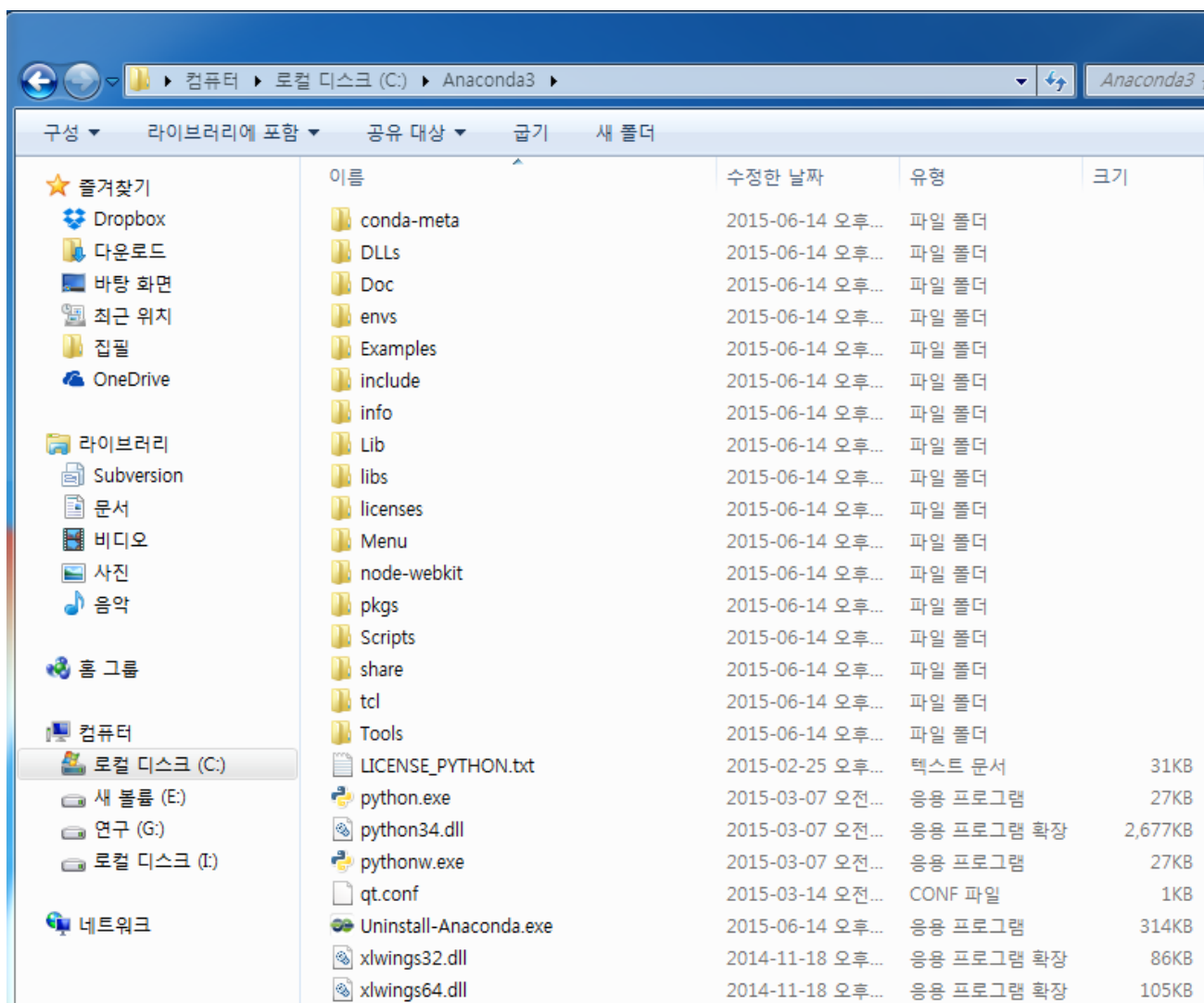


그림 1.11 C:\Anaconda3 디렉터리 구성

위의 경로에도 여러 디렉터리와 파일이 있는데 그림 1.12 와 같이 idle.bat 파일에 마우스 오른쪽 버튼을 클릭한 후 '보내기(N)'->'바탕 화면에 바로 가기 만들기'를 선택합니다. 파이썬으로 프로그래밍을 하려면 파이썬 IDLE(Integrated Development Environment)가 필요한데, 해당 파일을 더 쉽게 실행하기 위해

바탕화면에 바로 가기 아이콘을 생성한 것입니다. 이 과정을 모두 잘 따라 했다면 그림 1.13 과 같이 윈도우의 바탕화면에 아이콘이 생성된 것을 확인할 수 있습니다.

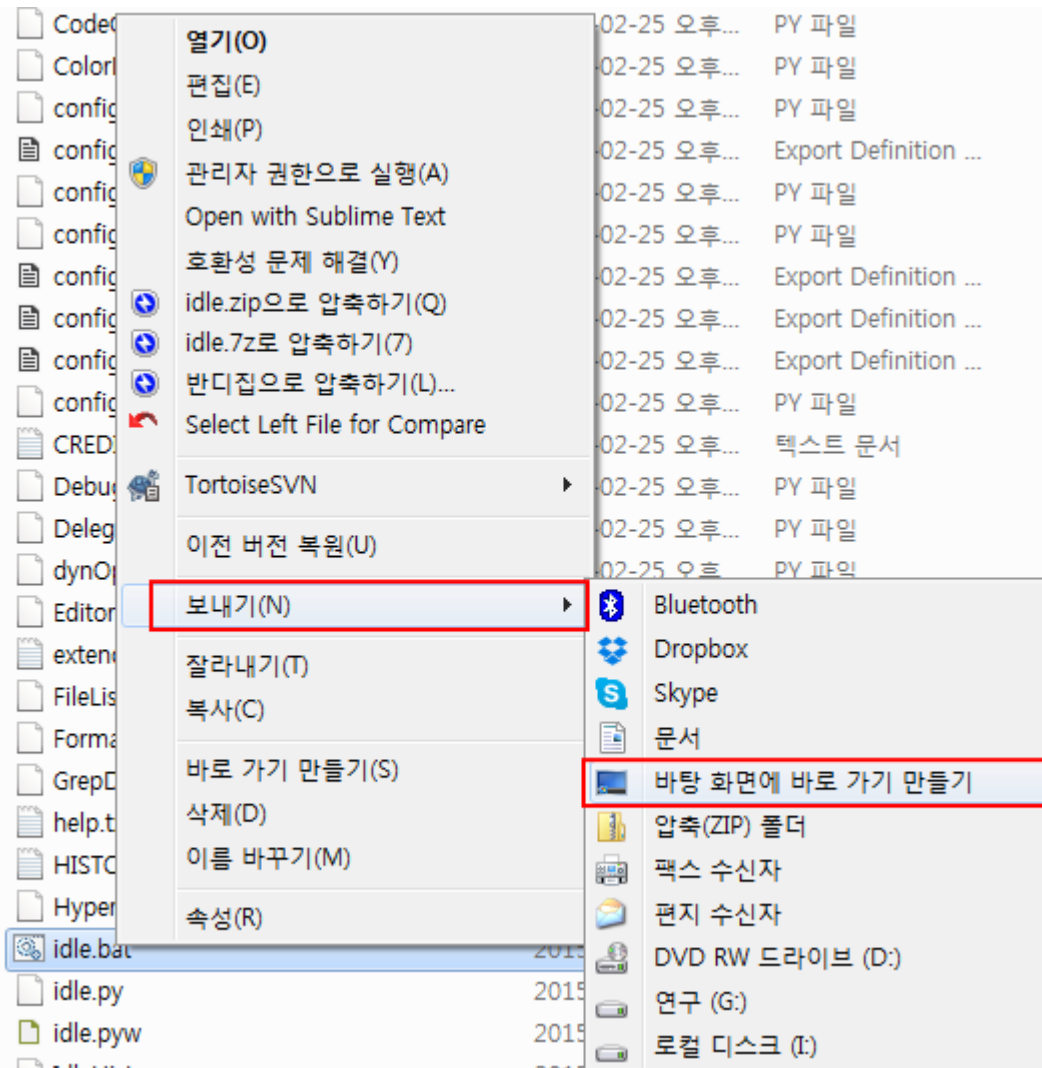


그림 1.12 idle.bat 파일에 대한 바로 가기 아이콘 만들기



그림 1.13 idle.bat 바로 가기 생성



# 1) 파이썬 IDLE 이용하기

바탕화면에 생성된 'idle.bat – 바로 가기' 아이콘을 클릭해 파이썬 IDLE 를 실행해 보기 바랍니다. 지금까지 설명한 과정을 문제 없이 따라 했다면 그림 1.14 와 같이 파이썬 IDLE 가 실행됩니다. .

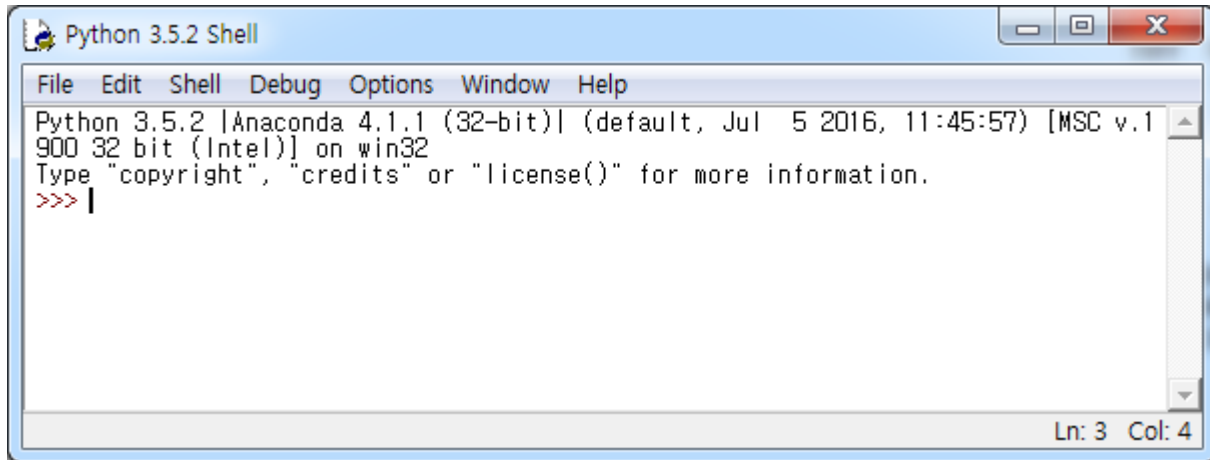


그림 1.14 Python IDLE 실행 화면

컴퓨터로 보고서를 쓸 때 마이크로소프트 워드나 한글과 컴퓨터의 한글이라는 프로그램을 실행한 후 글을 쓰듯이 파이썬 프로그램을 작성하기 위해서는 파이썬 IDLE 를 사용하는 것입니다. 여기서 IDLE(Integrated DeveLopment Environment)라는 영어 단어를 해석해보면 "통합 개발 환경"이라는 의미입니다. 아마도 해당 프로그램이 파이썬으로 개발을 통합적으로 지원하는 개발 환경이라는 뜻에서 IDLE 라는 이름을 붙인 것 같습니다.

한 번도 프로그래밍을 해보지 않았던 분들은 무슨 뜻인지 모르겠지만, 앞에서 워드와 한글이 글쓰기를 위한 여러 가지 기능을 제공하는 것처럼 파이썬 IDLE 도 여러분들이 좀 더 쉽게 프로그래밍할 수 있게 몇 가지 기능을 제공합니다. 파이썬 문법을 배우는 1 부에서는 주로 IDLE 를 이용해 프로그래밍하는 법을 배우겠습니다.

옛말에 '백문이 불여일견'이라는 말이 있는데, 프로그래머들 사이에는 '백문이 불여일타'라는 말이 있습니다. 백 번 듣는 것보다 한번 보는 것이 나은 것처럼, 프로그래밍을 공부할 때 백 번 듣는 것보다 한 번 타이핑(typing) 해보는 것이 낫다는 의미입니다.

사람들이 영어 공부할 때 보면 선생님이나 강사님이 아무리 강의를 잘해도, 자기가 직접 말해보고 하지 않는다면 실력이 빨리 늘지 않는 것과 같은 이치입니다. 프로그래밍 언어도 하나의 "언어"임을 잊지 말고 이 책에서 나오는 각 예제에 대해 똑같이 실행하다 보면 금방 실력이 향상되는 것을 확인할 수 있을 것입니다.

그림 1.14 에서 '>>>'는 프롬프트(prompt)라고 하는데, 앞으로 이 프롬프트에 파이썬 문법에 맞는 명령을 내리면 컴퓨터가 여러분의 명령을 해석해 실제 동작을 수행하게 됩니다. 여기서 중요한 점은 여러분이 내리는 명령이 파이썬의 문법을 따라야 한다는 점입니다. IDLE 프롬프트에 다음과 같이 print 명령어를 입력한 후 키보드의 엔터(ENTER) 키를 눌러 봅시다.

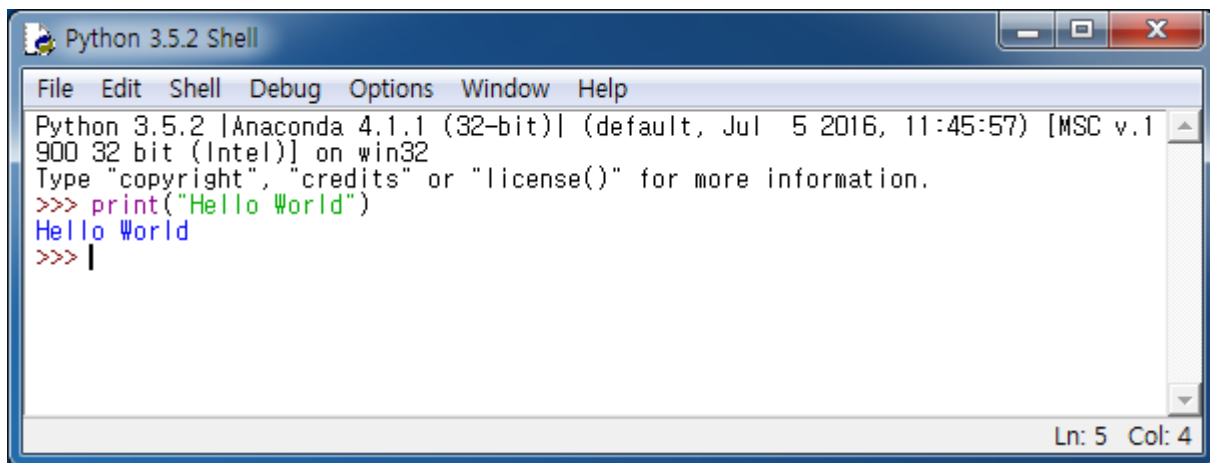
```
>>> print("Hello World")
```

```
Hello World
```

파이썬 IDLE 를 보면 그림 1.15 와 같이 화면에 Hello World 라는 글자가 출력(print)된 것을 볼 수 있습니다. 조금 전 파이썬 프롬프트(>>>) 상에서 print 라는 명령어를 사용해 Hello World 라는 문장을 화면에 출력하라는 명령을 내렸기 때문에 컴퓨터가 이를 알아듣고 해당 명령어를 수행한 것입니다.

파이썬을 공부하다 보면 앞으로 기본적으로 외워야 하는 것들이 여럿 나오는데, 대표적으로 print 라는 함수가 있습니다. 함수는 중학교 수학책에 나오는데, 보통 어떤 입력을 주면 어떤 정해진 기능을 수행하는 것을 의미합니다.

파이썬에서도 비슷하게 print 라는 함수는 괄호 안으로 입력되는 함수의 인자(argument)를 화면에 출력하는 기능을 제공합니다. 그래서 print("Hello World")의 수행 결과로 화면에 Hello World 가 출력되는 것입니다. 참고로 print 함수로 문자를 출력할 때는 출력하고자 하는 문자를 큰따옴표로 양쪽을 둘러싸거나 작은따옴표로 양쪽을 둘러싸면 됩니다.



### 그림 1.15 print 명령어의 수행 결과

그렇다면 꼭 영어만 출력되는 것일까요? 물론, 정답은 "아니오"입니다. 다음과 같이 프롬프트 상에서 print 이라는 함수를 호출해도 함수의 인자 값인 '안녕하세요'가 정상적으로 화면에 출력되는 것을 확인할 수 있습니다.

```
>>> print("안녕하세요.")
```

```
안녕하세요.
```

막상 처음으로 프로그래밍을 해보니 어떨까요? 뭔가 부족함이 느껴지지 않습니까? 프로그래밍을 하게 되면 뭔가 주식 가격을 알아맞히고, 대규모 매수 세력의 존재를 확인하고 이를 분석하고 종목을 추천하는 그런 대단한 프로그램을 만들 것을 기대했는데, 고작 화면에 문장을 출력하고 있단지요.

그러나 너무 조급해 하지 마시기 바랍니다. 영어 단기 완성이나 1주일 완성이라고 내건 곳으로 몇 년 동안 다녀도 영어를 제대로 하는 분들을 찾아보기 힘든 것처럼 프로그래밍을 아무것도 모르던 분들이 하루 이틀

공부한다고 프로그래밍을 잘 하게 되는 것은 아닙니다. 물론 이 책을 보고 일주일 정도 열심히 공부한다면 잘할 수 있을 것입니다. 희망을 품고 꾸준히 공부하기 바랍니다.

## 2) 파이썬 들여쓰기

파이썬은 다른 프로그래밍 언어와 달리 들여쓰기(indent)에 매우 민감한 언어입니다. 보통 C 나 C++ 같은 다른 프로그래밍 언어는 '{'와 '}' 같은 기호를 사용해 코드가 수행되는 영역을 표시합니다. 그렇지만 '{'와 '}' 기호는 초보자가 보기에는 조금 어려워 보입니다. 그리고 C 나 C++ 같은 프로그래밍 언어는 사실 띄어쓰기가 없이도 코딩할 수 있으므로 코드 자체는 수행되지만 사람들이 코드를 보고 이해하기가 어려울 수 있습니다.

"나는아침을먹고학교에갔는데오늘은학교개교기념일이었다."

우리의 모국어인 한국어도 위와 같이 띄어쓰기 없이 붙여서 쓰면 정말 읽기가 어렵죠? 프로그램도 마찬가지입니다. 프로그램을 작성할 때는 작성한 코드에 적당한 띄어쓰기를 해야 가독성이 좋아진답니다(코드들 읽기 쉽다는 뜻입니다).

파이썬은 {와 } 기호 없이 들여쓰기로만 영역을 나타내기 때문에 항상 조심해서 들여쓰기해야 합니다. 예를 들어, 그림 1.16 과 같이 파이썬 IDLE 에서 한 칸 공백을 주고(키보드의 스페이스 바를 누르면 됩니다) 명령어 코드를 작성하면 정상적으로 수행되지 않습니다.

아직은 잘 모르겠지만 '파이썬에서는 띄어쓰기가 정말 중요하다'라는 점을 꼭 기억하기 바랍니다. 사실 저도 파이썬을 처음 배울 때는 언제 띄어쓰기를 하고 언제는 하지 말아야 할지 판단하기가 어려웠는데, 이는 파이썬 문법에 좀 더 익숙해지면 저절로 해결되는 문제입니다. 앞으로 파이썬 IDLE 에서 프로그래밍할 때 'Indent'라는 단어가 포함된 오류 메시지가 나타나면 띄어쓰기에 문제가 있는지 코드를 확인해보기 바랍니다.

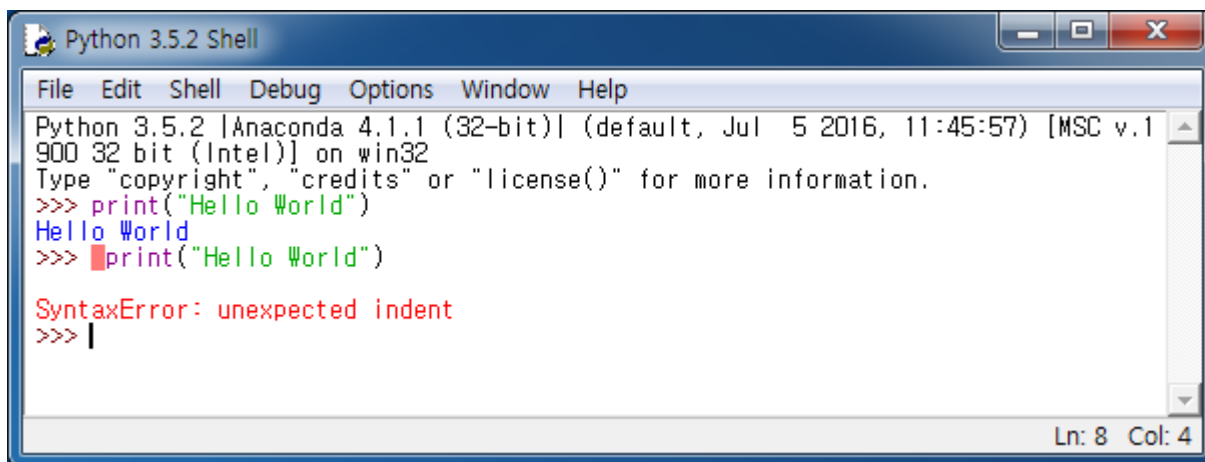


그림 1.16 파이썬 들여쓰기 오류의 예