

## 1장

### 01장 파이썬이란 무엇인가?

필자는 파이썬의 프롬프트(>>>)를 처음 본 순간부터 지금까지 줄곧 파이썬과 함께 지내 온 듯하다. '프로그래밍은 어렵고 지루하다'라는 고정관념을 가지고 있던 필자에게 파이썬은 커다란 충격으로 다가왔다. 여러분도 이 책을 통해 파이썬의 매력에 흠뻑 빠져 보기를 바란다. 01장에서는 파이썬의 특징과 장단점을 알아보고 파이썬 프로그래밍을 위한 환경 구축 방법에 대해 배운다. 그리고 간단한 파이썬 프로그램도 작성해 본다.

#### 01-1 파이썬이란?

파이썬(Python)은 1990년 암스테르담의 귀도 반 로섬(Guido van rossum)이 개발한 인터프리터 언어이다. 귀도는 파이썬이라는 이름을 자신이 좋아하는 코미디 쇼인 '몬티 파이썬의 날아다니는 서커스(Monty python's flying circus)'에서 따왔다고 한다.

인터프리터 언어란 소스 코드를 한 줄씩 해석한 후 그때그때 실행해 결과를 바로 확인할 수 있는 언어를 말한다.

파이썬의 사전적 의미는 '고대 신화에 나오는 파르나소스 산의 동굴에 살던 큰 뱀'을 뜻하며, 아폴로 신이 델파이에서 파이썬을 퇴치했다는 이야기가 전해지고 있다. 대부분의 파이썬 책 표지와 아이콘이 뱀 모양으로 그려져 있는 이유는 바로 이 때문이다.

파이썬은 컴퓨터 프로그래밍을 교육할 때뿐만 아니라 기업에서 실무를 할 때도 많이 사용한다. 그 대표적인 예가 바로 구글이다. 구글에서 만든 소프트웨어의 50% 이상이 파이썬으로 작성되었다는 이야기도 있을 정도이다. 이 밖에도 인스타그램(Instagram), 넷플릭스(Netflix), 아마존(Amazon) 등 우리가 알고 있는 많은 IT 기업에서 파이썬을 사용한다.

또한 파이썬 프로그램은 공동 작업과 유지 보수가 매우 쉽고 편리하다. 이 때문에 이미 다른 언어로 작성된 많은 프로그램이 파이썬으로 재구성되고 있다. 국내에서도 그 가치를 인정받아 사용자 층이 더욱 넓어지고 있고 파이썬을 사용해 프로그램을 개발하는 업체 또한 늘어나고 있는 추세이다.

#### 01-2 파이썬의 특징

모든 프로그래밍 언어는 저마다 장점이 있지만 파이썬은 다른 언어에서 쉽게 찾아볼 수 없는 독특한 매력을 가지고 있다. 파이썬의 특징을 알면 왜 파이썬을 공부해야 하는지, 과연 시간을 투자할 만한 가치가 있는지 판단할 수 있을 것이다.

- 파이썬은 인간다운 언어이다
- 문법이 쉬워 빠르게 배울 수 있다
- 무료이지만 강력하다
- 간결하다
- 프로그래밍을 즐기게 해 준다
- 개발 속도가 빠르다

### **파이썬은 인간다운 언어이다**

프로그래밍이란 인간이 생각하는 것을 컴퓨터에 지시하는 행위를 말한다. 파이썬은 사람이 생각하는 방식을 그대로 표현할 수 있는 언어이다. 따라서 파이썬을 사용하는 프로그래머는 굳이 컴퓨터의 사고 체계에 맞추어 프로그래밍하려고 애쓸 필요가 없다. 이제 곧 어떤 프로그램을 구상하자마자 머릿속에서 생각한 대로 코드를 술술 써 내려가는 자신의 모습을 보고 놀라게 될 것이다.

다음 소스 코드를 보면 이 말이 쉽게 이해될 것이다.

```
if 4 in [1,2,3,4]: print("4가 있습니다")
```

이 소스 코드는 다음처럼 읽을 수 있다.

만약 4가 1, 2, 3, 4 중에 있으면 "4가 있습니다"를 출력한다.

프로그램을 모르더라도 직관적으로 무엇을 뜻하는지 알 수 있지 않는가? 마치 영어 문장을 읽는 듯한 착각에 빠져든다.

### **문법이 쉬워 빠르게 배울 수 있다**

어려운 문법과 수많은 규칙에 둘러싸인 언어에서 벗어나고 싶지 않은가? 파이썬은 문법이 매우 쉽고 간결하며 사람의 사고 체계와 매우 닮아 있다. 배우기 쉬운 언어, 활용하기 쉬운 언어가 가장 좋은 언어가 아닐까? 유명한 프로그래머인 에릭 레이먼드(Eric raymond)는 파이썬을 공부한 지 단 하루 만에 자신이 원하는 프로그램을 작성할 수 있었다고 한다.

프로그래밍 경험이 조금이라도 있다면 파이썬의 자료형, 함수, 클래스 만드는 법, 라이브러리 및 내장 함수 사용 방법 등을 익히는 데 일주일이면 충분하다고 생각한다.

## 무료이지만 강력하다

파이썬은 오픈 소스(open source)이며 무료 소프트웨어이다. 사용료 걱정 없이 언제 어디서든 파이썬을 내려받아 사용할 수 있다.

오픈 소스란 저작권자가 소스 코드를 공개하여 누구나 별다른 제한 없이 자유롭게 사용·복제·배포·수정할 수 있는 소프트웨어를 말한다.

또한 프로그래머는 만들고자 하는 프로그램의 대부분을 파이썬으로 만들 수 있다. 물론 시스템 프로그래밍이나 하드웨어 제어와 같은 매우 복잡하고 반복 연산이 많은 프로그램은 파이썬과 어울리지 않는다. 하지만 파이썬은 이러한 약점을 극복할 수 있도록 다른 언어로 만든 프로그램을 파이썬 프로그램에 포함시킬 수 있다.

‘파이썬과 C는 찰떡궁합’이라는 말이 있다. 프로그램의 전반적인 뼈대는 파이썬으로 만들고 빠른 실행 속도가 필요한 부분은 C로 만들어서 파이썬 프로그램 안에 포함시킬 수 있기 때문이다(정말 놀라울 만큼 영악한 언어가 아닌가?). 사실 파이썬 라이브러리 중에는 파이썬만으로 제작된 것도 많지만 C로 만든 것도 많다. C로 만든 것은 대부분 속도가 빠르다.

파이썬 라이브러리는 파이썬 프로그램을 작성할 때 불러와 사용할 수 있도록 미리 만들어 놓은 파이썬 파일의 모음을 말한다.

## 간결하다

귀도는 파이썬을 의도적으로 간결하게 만들었다. 만약 펄(Perl)과 같은 프로그래밍 언어가 100가지 방법으로 1가지 일을 처리할 수 있다면, 파이썬은 가장 좋은 방법 1가지만 사용하는 것을 선호한다. 이 간결함의 철학은 파이썬 문법에도 그대로 녹아 있어 파이썬 프로그래밍을 하는 사람들은 잘 정리되어 있는 소스 코드를 볼 수 있다. 그 덕분에 다른 사람이 작업한 소스 코드를 이해하기 쉽고 공동 작업과 유지 보수가 매우 편리하다.

프로그래밍 언어를 판별하여 그에 맞는 문장을 출력하는 파이썬 프로그램 예제이다. 다른 언어에서 늘 보게 되는 단락을 구분하는 괄호({})가 없고 줄을 잘 맞추는 코드라는 것을 알 수 있다. 파이썬 프로그램은 줄을 맞추지 않으면 실행되지 않는다. 코드를 예쁘게 작성하려고 줄을 맞추는 것이 아니라 프로그램이 실행되게 하려면 꼭 줄을 맞추어야 하는 것이다. 이렇듯 줄을 맞추어 코드를 작성하는 행위는 가독성에 많은 도움이 된다.

이렇게 코드의 줄을 맞추는 것을 ‘들여쓰기’라고 한다. 파이썬은 들여쓰기를 하지 않으면 프로그램이 실행되지 않는다

## 프로그래밍을 즐기게 해 준다

이 부분이 가장 강조하고 싶은 부분이다. 파이썬만큼 필자에게 프로그래밍을 즐기게 해 준 언어는 없었다. 파이썬은 프로그래머가 다른 부수적인 개념이나 제한 사항 등에 신경 쓸 필요 없이 만들고자 하는 기능에만 집중할 수 있게 해 준다. 파이썬을 배우고 나면 다른 언어로 프로그래밍하는 것이 지루하다고 느낄지도 모른다.

## 개발 속도가 빠르다

마지막으로 재미있는 문장으로 파이썬의 특징을 마무리하겠다.

***"Life is too short, You need python." (인생은 너무 짧으니 파이썬이 필요해.)***

엄청나게 빠른 파이썬의 개발 속도 때문에 유행처럼 퍼진 말이다. 이 위트 있는 문장은 이 책에서 계속 예제로 사용할 것이다.

## 01-3 파이썬으로 무엇을 할 수 있을까?

프로그래밍 언어를 좋은 언어와 나쁜 언어로 구별할 수 있을까? 사실 현실에서 이런 구별은 무의미하다. 어떤 언어이든 강점과 약점이 존재하기 때문이다. 따라서 어떤 프로그래밍 언어가 어떤 일에 효율적인지를 안다는 것은 프로그래머의 생산성을 크게 높일 수 있는 힘이 된다.

그렇다면 파이썬으로 하기에 적당한 일과 적당하지 않은 일은 무엇일까? 이를 아는 것은 매우 가치 있는 일이므로 파이썬을 배우기 전에 간단히 알아보자.

- 파이썬으로 할 수 있는 일
  - 웹 프로그래밍
  - 인공지능과 머신러닝
  - 수치 연산 프로그래밍
  - 데이터 분석
  - 데이터베이스 프로그래밍
  - 시스템 유틸리티 제작하기
  - GUI 프로그래밍
  - C/C++와 결합하기
  - 사물 인터넷
- 파이썬으로 할 수 없는 일

- 시스템과 밀접한 프로그래밍 영역
- 모바일 프로그래밍

## **파이썬으로 할 수 있는 일**

파이썬으로 할 수 있는 일은 매우 많다. 파이썬은 대부분의 프로그래밍 언어가 하는 일을 쉽고 깔끔하게 처리한다. 파이썬으로 할 수 있는 일들을 나열하자면 끝도 없지만, 대표적인 몇 가지만 소개한다.

## **웹 프로그래밍**

누구나 한 번쯤 구글 크롬이나 마이크로소프트 엣지와 같은 웹 브라우저로 웹 서핑을 하면서 게시판이나 방명록에 글을 남겨 본 적이 있을 것이다. 이러한 게시판이나 방명록을 '웹 프로그램'이라고 한다. 파이썬은 웹 프로그램을 만들기에 매우 적합한 도구이며, 실제로 파이썬으로 제작한 웹 사이트는 셀 수 없을 정도로 많다.

## **인공지능과 머신러닝**

최근 인공지능(artificial Intelligence) 분야의 발전 덕분에 파이썬이 인기 언어로 떠올랐다. 인공지능은 기계가 인간처럼 생각하고 행동할 수 있는 능력을 제공하는 기술이다. 파이썬을 활용하면 자연어 처리, 음성 인식, 이미지 인식과 같은 인공지능 기술을 구현할 수 있다. 머신러닝(machine learning)은 인공지능의 한 분야로, 경험을 통해 자동으로 발전하는 컴퓨터 알고리즘을 연구하는 분야이다. 파이썬은 인공지능과 머신러닝 프로그래밍을 쉽게 할 수 있도록 사이킷런(scikit-learn), 텐서플로(TensorFlow), 파이토치(PyTorch), 케라스(Keras) 등과 같은 다양한 라이브러리를 제공한다.

## **수치 연산 프로그래밍**

사실 파이썬은 수치 연산 프로그래밍에 적합한 언어가 아니다. 수치가 복잡하고 연산이 많다면 C와 같은 언어로 프로그래밍하는 것이 더 빠르기 때문이다. 하지만 파이썬은 넘파이(Numpy)라는 수치 연산 모듈을 제공한다. 이 모듈은 C로 작성했기 때문에 파이썬에서도 수치 연산을 빠르게 수행할 수 있다.

## **데이터 분석**

파이썬은 데이터 분석에 이상적인 언어로, 넘파이(NumPy), 판다스(Pandas), 맷플롯립(Matplotlib) 등과 같은 라이브러리들을 활용하여 데이터 처리, 통계 분석, 시각화를 손쉽게 수행할 수 있다. 데이터 분석을 할 때 아직까지는 데이터 분석에 특화된 R이라는 언어를 많이 사용하고 있지만, 판다스가 등장한 이래 파이썬을 사용하는 경우가 점점 많아지고 있다.

## 데이터베이스 프로그래밍

파이썬은 사이베이스(Sybase), 인포믹스(Infomix), 오라클(Oracle), 마이에스큐엘(MySQL), 포스트그레스큐엘(PostgreSQL) 등의 데이터베이스에 접근하기 위한 도구를 제공한다.

이런 굵직한 데이터베이스를 직접 사용하는 것 말고도 파이썬에는 재미있는 도구가 하나 더 있다. 바로 피클(pickle)이라는 모듈이다. 피클은 파이썬에서 사용하는 자료를 변형 없이 파일에 저장하거나 불러오는 일을 수행한다.

피클을 어떻게 사용하고 활용하는지는 '05-6 표준 라이브러리'를 참고하기 바란다.

## 시스템 유틸리티 제작하기

파이썬은 운영체제(윈도우, 리눅스 등)의 시스템 명령어를 사용할 수 있는 각종 도구를 갖추고 있기 때문에 이를 바탕으로 파일 관리, 로그 분석, 시스템 모니터링 등과 같은 여러 가지 시스템 유틸리티를 만드는 데 유리하다. 실제로 여러분은 시스템에서 사용 중인 서로 다른 유틸리티성 프로그램을 하나로 합쳐 보다 유용한 프로그램들을 무수히 만들어 낼 수 있다.

유틸리티란 컴퓨터를 사용하는 데 도움을 주는 여러 가지 소프트웨어를 말한다.

## GUI 프로그래밍

GUI(graphic user interface) 프로그래밍이란 쉽게 말해 화면에 윈도우 창을 만들고 그 창에 프로그램을 동작시킬 수 있는 메뉴나 버튼 등을 추가하는 것을 말한다. 파이썬은 GUI 프로그래밍을 위한 도구들이 잘 갖추어져 있기 때문에 GUI 프로그램을 쉽게 만들 수 있다. 대표적인 예로 파이썬 프로그램과 함께 설치되는 Tkinter(티케이인터)를 들 수 있다. Tkinter를 사용하면 단 5줄의 소스 코드만으로 윈도우 창을 띄울 수 있다.

## C/C++와 결합하기

파이썬은 접착(glue) 언어라고도 부르는데, 그 이유는 다른 언어와 결합해서 사용할 수 있기 때문이다. C나 C++로 만든 프로그램을 파이썬에서 사용할 수 있고 파이썬으로 만든 프로그램도 C나 C++에서 사용할 수 있다.

## 사물 인터넷

파이썬은 사물 인터넷(Internet of things) 분야에서도 활용도가 높다. 한 예로 라즈베리파이(Raspberry pi)는 리눅스 기반의 매우 작은 컴퓨터이다. 라즈베리파이를 사용하면 홈시어터나 매우 작은 게임기 등 여러 가지 재미있는 것을 만들 수 있는데, 라즈베리파이를 제어할 때 파이썬을 사용한다. 예를 들어 라즈베리파이에 연결된 모터를 작동시키거나 LED에 불이 들어오게 하는 일을 파이썬으로 할 수 있다.

## 파이썬으로 할 수 없는 일

### 시스템과 밀접한 프로그래밍 영역

파이썬으로 리눅스와 같은 운영체제 또는 엄청난 횟수의 반복과 연산이 필요한 프로그램을 만드는 것은 어렵다. 즉, 매우 빠른 속도를 요구하거나 하드웨어를 직접 건드려야 하는 프로그램에는 적합하지 않다.

### 모바일 프로그래밍

파이썬은 구글이 가장 많이 애용하는 언어이지만, 파이썬으로 안드로이드 네이티브 앱(android native app)을 개발하는 것은 아직 어렵다. 안드로이드에서 파이썬으로 만든 프로그램이 실행되도록 지원하기는 하지만, 이것만으로 앱을 만들기에는 아직 역부족이다. 이 밖에 아이폰 앱도 파이썬으로 개발할 수 없다.

## 01-4 파이썬 설치하기

이제 실습을 해 보기 위해 컴퓨터에 파이썬을 설치해 보자. 이 책에서는 윈도우와 맥에서의 설치 방법만 다룬다. 다른 시스템을 사용한다면 파이썬 홈페이지의 설명을 참고하기 바란다.

- 윈도우에서 파이썬 설치하기
- 맥에서 파이썬 설치하기

### 윈도우에서 파이썬 설치하기

1. 먼저 파이썬 공식 홈페이지의 다운로드 페이지에서 윈도우용 파이썬 언어 패키지를 내려받는다. 다음 화면에서 Python 3.x로 시작하는 버전 중 가장 최신의 윈도우 설치 파일을 내려받자(이 글을 작성하는 시점의 최신 버전은 3.11.2이다).

2. 설치 파일을 실행한 후 [Install Now]를 클릭하면 설치가 진행된다. 이때 파이썬이 어느 곳에서든지 실행될 수 있도록 [Add python.exe to PATH] 옵션을 반드시 선택해야 한다.

[Add python.exe to PATH] 옵션을 선택하지 않으면 실습 시 오류가 발생할 수 있다.

3. 설치가 완료되면 [close]를 클릭하여 종료한다. 파이썬이 정상적으로 설치되었다면 [시작] 메뉴의 검색 창에서 python을 검색하여 파이썬 실행 프로그램을 찾을 수 있다.

### 맥에서 파이썬 설치하기

파이썬 공식 홈페이지([www.python.org](http://www.python.org))에서 [Downloads] 메뉴를 클릭하여 맥(Mac)용 파이썬 설치 파일을 내려받은 후 다음 화면에서 [Download Python 3.11.x]를 클릭하면 된다.

그런 다음 내려받은 python-3.11.x-macos1.pkg 파일을 실행하여 설치한다.

설치가 완료되면 파이썬이 제대로 설치됐는지 확인하기 위해 터미널에서 다음과 같이 명령을 입력해 자신의 맥에 설치된 파이썬 버전을 확인해 보자. 구 버전의 맥에는 파이썬 2.7 버전이 기본으로 설치되어 있기 때문에 python을 입력하면 파이썬 2.7이 실행된다. 맥에서는 항상 'python' 대신 'python3' 명령을 사용하자. 파이썬 버전이 제대로 출력되면 성공적으로 설치한 것이다.

## 01-5 파이썬 둘러보기

파이썬이라는 언어는 어떻게 생겼는지 간단한 소스 코드를 작성하면서 알아보자. 파이썬에 대해 자세히 알아보기 전에 전체 모습을 훑어보는 것은 파이썬을 이해하는 데 많은 도움이 될 것이다.

‘백문이 불여일견, 백견이 불여일타’라고 했다. 직접 따라 해 보자.

- 파이썬 기초 실습 준비하기
- 파이썬 기초 문법 따라 해 보기
  - 사칙 연산
  - 변수에 숫자 대입하고 계산하기
  - 변수에 문자 대입하고 출력하기
  - 조건문 if
  - 반복문 for
  - 반복문 while
  - 함수

### 파이썬 기초 실습 준비하기

파이썬 프로그래밍 실습을 시작하기 전에 기초적인 것을 준비해 보자.

먼저 [시작] 메뉴에서 'Python 3.11(64-bit)'을 찾아 실행한다.

맥 사용자는 터미널에서 'python3'을 입력하자.



위와 같은 창을 '파이썬 대화형 인터프리터'라고 부른다. 여기서 인터프리터란 사용자가 입력한 소스 코드를 실행하는 환경을 뜻한다.

파이썬 대화형 인터프리터를 간단히 '파이썬 셸(python shell)'이라고도 부른다. 파이썬 셸 창에서 3개의 꺾은 괄호(>>>)를 프롬프트(prompt)라고 부른다. 프롬프트(>>>)가 있는 곳이 입력하는 부분이다.

그렇다면 '대화형'은 무슨 의미일까?

입력에 따른 결과값이 바로 출력된다. 이렇게 입출력이 번갈아 이어지는 것이 마치 대화하는 것처럼 느껴지기 때문에 파이썬 대화형 인터프리터라고 부른다. 대화형 인터프리터를 사용하면 간단한 예제를 풀 때 결과를 바로 확인할 수 있어 학습을 하는 데 효과적이다. 앞으로 파이썬의 기본적인 내용은 이 창을 사용해 실습한다.

대화형 인터프리터를 종료할 때는 [Ctrl+Z]를 누른 후 [Enter]를 누른다(유닉스 계열에서는 [Ctrl+D]). 또는 다음과 같이 내장 함수인 quit()이나 exit()를 사용하여 종료한다.

다음 예와 같이 sys 모듈을 사용하여 종료할 수도 있다.

```
import sys
```

```
sys.exit()
```

모듈을 사용하는 방법은 05장을 참고하기 바란다

## 파이썬 기초 문법 따라 해 보기

이제 파이썬 대화형 인터프리터(파이썬 셸)를 실행하여 파이썬 기초 문법을 따라 해 보자. 여기에서 소개하는 내용은 뒤에서 자세하게 다룰 것이므로 지금 당장 이해되지 않는다고 해서 절망하거나 고심하지는 말자.

### 사칙 연산

1 더하기(+) 2는 3이라는 값을 출력해 보자. 보통 계산기를 사용하듯이 +만 넣어 주면 된다.

```
>>> 1 + 2
```

```
3
```

나눗셈(/)과 곱셈(\*) 역시 예상한 대로 결과값을 보여 준다.

```
>>> 3 / 2.4
```

```
1.25
```

```
>>> 3 * 9
```

```
27
```

우리가 일반적으로 알고 있는 ÷나 ×가 아니라는 것에 주의하자.

### **변수에 숫자 대입하고 계산하기**

```
>>> a = 1
```

```
>>> b = 2
```

```
>>> a + b
```

```
3
```

a에 1, b에 2를 대입한 후 a와 b를 더하면 3이라는 결과값을 보여 준다.

### **변수에 문자 대입하고 출력하기**

```
>>> a = "Python"
```

```
>>> print(a)
```

```
Python
```

a 변수에 "Python"이라는 값을 대입한 후 print(a)라고 작성하면 a 값을 출력한다.

파이썬은 대소문자를 구별한다. print를 PRINT로 쓰면 정의되지 않았다는 오류 메시지가 나온다.

또는 다음과 같이 print 문을 생략하고 변수 이름 a만 입력하여 a의 값을 확인할 수도 있다.

```
>>> a = "Python"
```

```
>>> a
```

```
'Python'
```

## 조건문 if

다음은 간단한 조건문 if를 사용한 예제이다.

예제는 a가 1보다 크면 "a는 1보다 큽니다."라는 문장을 출력(print)하라는 뜻이다. a는 3이므로 1보다 크다. 따라서 'a는 1보다 큽니다.'라는 문장이 출력된다.

인터프리터 창에서 프롬프트(>>>) 자리에 대신 나타나는 '...'은 아직 입력 중인 문장이 끝나지 않았다는 것을 의미한다.

if a > 1: 다음 문장은 [Spacebar] 4개 또는 [Tab]을 이용해 반드시 들여쓰기 한 후에 print("a는 1보다 큽니다.")라고 작성해야 한다. 들여쓰기 규칙은 '03장 제어문'에서 자세하게 알아본다. 바로 뒤에 이어지는 반복문 for, while 예제도 이와 마찬가지로 들여쓰기가 필요하다.

## 반복문 for

for 문을 사용하면 실행해야 할 문장을 여러 번 반복해서 실행할 수 있다. 위 예는 대괄호([]) 사이에 있는 값을 하나씩 출력한다. 이 코드의 의미는 '[1, 2, 3] 리스트의 앞에서부터 하나씩 꺼내 a 변수에 대입한 후 print(a)를 수행하라'이다. 이를 실행하면 a에 차례대로 1, 2, 3이라는 값을 대입하고 print(a)에 의해서 그 값을 차례대로 출력한다.

## 반복문 while

while이라는 영어 단어의 뜻인 '~인 동안'에서 알 수 있듯이 while 문은 일정한 조건이 만족하는 동안 문장을 반복해서 수행한다. 위 예제는 i 값이 3보다 작은 동안 i = i + 1과 print(i)를 수행하라는 말이다. i = i + 1이라는 문장은 'i의 값을 1씩 더한다'라는 의미이다. i 값이 3보다 커지면 while 문을 빠져나간다.

## 함수

def는 파이썬에서 함수를 정의할 때 사용하는 예약어이다. 위 예제는 add 함수를 만들고 그 함수를 어떻게 사용하는지를 보여 준다. add(a, b)에서 a, b는 입력값, a + b는 결과값이다. 즉 3, 4가 입력으로 들어오면 3 + 4를 수행하고 그 결과값인 7을 리턴한다.

예약어란 프로그래밍 언어에서 이미 문법적인 용도로 사용하고 있는 단어를 말한다. 리턴(return)은 함수에서 값을 반환할 때 사용한다.

지금까지 기초적인 파이썬 문법에 대해서 간략하게 알아보았다.

## 01-6 파이썬과 에디터

파이썬 대화형 인터프리터는 간단한 예제를 풀 때는 편리하지만 여러 줄의 복잡한 소스 코드를 가진 프로그램을 만들 때는 불편하다. 또한 인터프리터를 종료하면 작성한 프로그램이 사라져 다시 사용하지 못한다는 단점이 있다. 따라서 여러 번 사용하기 위한 프로그램을 만들 때는 에디터를 사용해야 한다.

에디터(editor)란 소스 코드를 편집할 수 있는 프로그래밍 도구를 말한다. 에디터에는 여러 가지 종류가 있다. 이 책에서는 그중 파이썬을 설치할 때 기본으로 설치되는 파이썬 IDLE(Integrated development and learning environment)의 에디터를 사용한다.

IDLE은 '아이들'이라고 읽는다.

- IDLE 에디터로 파이썬 프로그램 작성하기
- 명령 프롬프트 창에서 파이썬 프로그램 실행하기
- 추천하는 에디터
  - 비주얼 스튜디오 코드
  - 파이참

## IDLE 에디터로 파이썬 프로그램 작성하기

파이썬 IDLE은 파이썬 프로그램 작성을 도와주는 통합 개발 환경이다. [시작] 메뉴에서 'idle'을 검색한 후 파이썬 IDLE을 찾아 실행해 보자.

맥 사용자는 터미널에서 idle3 명령을 입력하자.

그러면 다음과 같은 IDLE 셸(shell) 창이 나타난다.

IDLE 셸은 IDLE 에디터에서 실행한 프로그램의 결과가 표시되는 창으로, 앞에서 살펴본 파이썬 셸과 기능 면에서 동일하다. 단, IDLE 셸에서 들여쓰기를 표시하는 방법이 기본 파이썬 셸과 다르므로 이 책에서는 IDLE 에디터로 작성한 프로그램의 실행 결과를 표시하는 용도로만 사용한다. 즉, 앞으로 프롬프트(>>>)로 시작하는 예제는 IDLE 셸이 아닌 파이썬 셸로 실행해야 한다.

이제 IDLE 에디터를 실행해 보자.

IDLE 셸 창 메뉴에서 [File → New File]을 선택한다.

그러면 다음과 같이 빈 창이 나타나는데, 이 창이 IDLE 에디터이다.

이제 IDLE 에디터에서 다음과 같이 파이썬 프로그램을 작성해 보자.

```
# hello.py  
  
print("Hello World")
```

첫 줄에 작성한 # hello.py는 주석이다. 파이썬에서 #으로 시작하는 문장은 주석으로 처리하므로 프로그램 수행에 전혀 영향을 미치지 않는다. 주석은 이 프로그램의 소스 코드를 읽을 미래의 나 또는 다른 프로그래머를 위해 소스 코드에 설명을 달 때 사용한다.

### 점프 투 파이썬 여러 줄짜리 주석을 작성하는 방법

주석을 여러 줄 달고 싶다면 다음과 같이 작성하자.

```
""" Author: EungYong Park Date: 2023-05-01 이 프로그램은 Hello World를 출력하는 프로그램이다. """
```

여러 줄로 이루어진 주석은 큰따옴표 3개를 연속으로 사용한 """ 사이에 작성하면 된다. 큰따옴표 대신 작은따옴표 3개('')를 사용해도 된다.

이제 작성한 프로그램을 실행해 보자.

IDLE 에디터 창 메뉴에서 [Run → Run Module]을 선택한다(단축키: F5)

프로그램을 실행하면 파일을 먼저 저장하라는 대화 상자가 나타난다.

[확인]을 선택하면 [다른 이름으로 저장] 창이 나타난다. 로컬 디스크(C:)로 이동한 후 'doit'이라는 이름으로 디렉터리(폴더)를 생성하고 그 안에 hello.py라는 이름으로 파일을 저장하자. 에디터로 파이썬 프로그램을 작성한 후 저장할 때는 파일 이름의 확장자를 함

상 .py로 해야 한다. .py는 파이썬 파일이라는 것을 알려 주는 확장자이다.

실습 경로를 간단히 하기 위해 C:\wdoit 디렉터리를 생성한 후 파일을 저장했다. C:\w에 파일을 바로 저장하면 오류가 발생할 수 있으므로 꼭 C:\w 하위에 디렉터리를 생성한 후 그 안에 파일을 저장하자.

파일을 저장하면 hello.py 프로그램이 실행된다. 실행 결과는 다음과 같이 IDLE 셸 창에 표시된다.

앞으로 간단한 예제는 파이썬 셸에 입력해 결과를 바로 확인하고 여러 줄을 한꺼번에 작성하거나 저장했다가 다시 사용할 프로그램은 IDLE 에디터에서 작성한다.

### 점프 투 파이썬파이썬 셸과 IDLE 에디터의 차이점

앞으로 이 책은 파이썬 셸과 IDLE 에디터를 번갈아 가면서 예제를 작성한다. 이때 주의할 점이 1가지 있다.

파이썬 셸의 경우, 변수 a와 표현식 1+1의 값을 확인하기 위해서 print 함수를 사용해도 되지만 다음처럼 print 함수 없이도 그 값을 확인할 수 있다.

이것은 표현식의 실행 결과를 출력하는 파이썬 셸 본래의 기능이기 때문이다. 하지만 IDLE 에디터를 사용할 경우에는 다음처럼 print 문을 반드시 사용해야 한다.

```
# editor.py a = "python" print(a) print(1 + 1) # print 문을 사용해야 결과값을 출력할 수 있다.
```

### 명령 프롬프트 창에서 파이썬 프로그램 실행하기

이번에는 앞에서 IDLE 에디터로 작성한 hello.py 파일을 명령 프롬프트(command prompt) 창에서 실행해 보자. IDLE 에디터에서 F5를 눌러 실행할 수도 있지만, 실제 업무에서는 일반적으로 명령 프롬프트 창에서 파이썬 프로그램을 실행한다.

우리는 이미 다음과 같은 프로그램을 C:\wdoit 디렉터리에 'hello.py'라는 이름으로 저장했다.

```
# hello.py  
  
print("Hello world")
```

hello.py 프로그램을 실행하기 위해 먼저 명령 프롬프트 창을 열어 보자. [윈도우 키 + R]을 누르면 다음과 같은 실행 창이 나타난다. 빈칸에 'cmd'를 입력하고 [확인]을 클릭하면

명령 프롬프트 창이 열린다.

첫 번째 줄에는 다음과 같은 프롬프트가 나타나 있을 것이다. C:\Users\박응용에서 박응용은 필자의 컴퓨터 이름이다(물론 여러분의 명령 프롬프트는 여러분의 컴퓨터 이름으로 표시될 것이다).

맥 사용자의 경우, 명령 프롬프트 창 대신 터미널을 사용하자.

이제 다음과 같이 프롬프트에서 `cd c:\wdoit`이라고 입력한다. `cd`(change directory) 명령어는 프롬프트를 바로 다음에 나오는 디렉터리로 옮긴다.

`c:\wdoit`에서 역슬래시(`\`)는 [Enter] 바로 위에 있는 키(원화 표시 문자)로 입력한다.

프롬프트가 `c:\wdoit>`으로 바뀐 것을 확인할 수 있다. 이제 명령어 `python`과 실행할 프로그램 이름을 입력하면 파이썬 프로그램을 실행할 수 있다. 다음과 같이 입력해 `hello.py` 프로그램을 실행해 보자.

위와 같이 `hello.py` 프로그램이 실행되는 것을 확인할 수 있다. 만약 결괏값이 위와 같지 않다면 `hello.py` 파일이 `C:\wdoit` 디렉터리에 존재하는지 다시 한번 살펴보자.

명령 프롬프트 창에서 예제를 실행한 화면은 다음과 같다.

## 추천하는 에디터

파이썬으로 실무 프로그램을 작성할 때는 앞에서 소개한 IDLE보다 많은 기능을 갖춘 에디터를 쓰는 것이 좋다. 추천할 만한 에디터를 소개한다.

이 책의 예제를 풀다가 에디터를 사용하여 프로그램을 작성해야 할 때는 파이썬 공부에 최적화된 IDLE을 사용하자. 여기서 소개하는 에디터는 파이썬 공부를 한 번 끝낸 후에 사용해 보기를 권장한다.

## 비주얼 스튜디오 코드

비주얼 스튜디오 코드(Visual studio code)는 파이참(Pycharm)과 더불어 프로그래머에게 가장 많은 사랑을 받는 파이썬의 대표적인 에디터이다. 비주얼 스튜디오 코드는 공식 다

운로드 사이트에서 내려받을 수 있다.

비주얼 스튜디오 코드는 파이썬 전용 에디터가 아니기 때문에 파이썬 외에도 여러 가지 언어를 지원한다. 따라서 비주얼 스튜디오 코드를 설치한 후 파이썬을 사용하기 위해 가장 먼저 해야 할 일은 파이썬 extension을 설치하는 것이다. 파이썬 extension은 비주얼 스튜디오 코드의 [Extensions] 메뉴에서 설치할 수 있다.

## **파이참**

파이썬에 어느 정도 익숙해졌다면 파이참을 사용해 보기를 적극 추천한다. 파이참은 가장 유명한 파이썬 에디터 중 하나로, 코드를 작성할 때 자동 완성, 문법 체크 등 편리한 기능을 많이 제공한다. 파이참 공식 다운로드 사이트에서 내려받을 수 있다.

파이참은 파이썬 전용 에디터이므로 별도의 extension이나 플러그인을 설치할 필요가 없다.