

3장

파이썬 프로그래밍 언어 소개

3장 파이썬 프로그래밍 언어 소개

3.1 파이썬 설치하기

3.2 작업 공간 설정하기

3.3 요약

3.1 파이썬 설치하기



3.1.1. 파이썬이란 무엇인가?

- » 파이썬(Python) : 네덜란드의 귀도 반 로섬(Guido van Rossum)이 만든 프로그래밍 언어
- » 파이썬 프로그래밍 언어에서는 모든 것이 특성(데이터)과 그 데이터가 작동하는 방법(동작)들로 이뤄져 있다.
- » 데이터와 그 데이터와 연관된 동작을 한데 묶은 존재를 객체(object)라고 한다.



3.1.2. 파이썬 버전 3.7.1 다운로드

» 파이썬 공식 웹 사이트(www.python.org)에서 내려받거나, 파이썬에 다른 패키지를 추가로 설치해 제공하는 서드파티 패키지3를 사용할 수도 있다.



3.1.3. 아나콘다 파이썬 배포판

- » 아나콘다 홈페이지(www.anaconda.com)에서 아나콘다 파이썬 배포판을 내려받을 수 있다.
- » 이 배포판은 무료로 다양한 파이썬 버전을 제공하며 과학, 수학, 엔지니어링, 데이터 분석 등의 분야 에서 유명한 400여 개의 패키지가 담겨 있다.
- » 추가 패키지가 없는 최소한의 버전도 제공하는데 이를 미니콘다(Miniconda)라고 한다.



3.1.4. 통합 개발 환경

- » 설치를 마치면 아나콘다에 속한 프로그램 중 하나인 스파이더(Spyder)를 열 수 있다.
- » 스파이더는 통합 개발 환경(Integrated Development Environment, IDE)으로 이 책에서는 스파이더를 사용해 프로그램을 작성하고 실행한다.



3.1.4.1. 스파이더 열기

» 윈도우 시작 메뉴에 있는 아나콘다 폴더에서 스파이더(Spyder)를 열 수 있다.

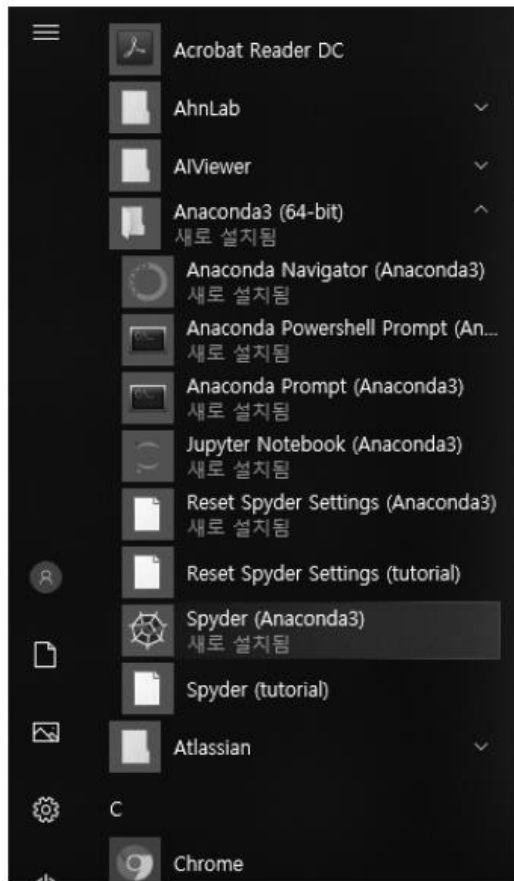


그림 3-1
시작 메뉴의 아나콘다 폴더



3.1.4.1. 스파이더 열기

» 스파이더 IDE가 제공하는 주요 특징

- 파이썬 프로그램을 작성할 수 있는 에디터
- 프로그램을 실행하기 전에 코드에 잠재적인 문제나 비효율적인 부분이 있을 경우 이를 표시하는 기능
- 프로그램이 사용자와 입력과 출력을 주고 받으며 상호작용하는 콘솔
- 프로그램에 있는 변수의 값을 살펴보는 방법
- 코드를 한 줄씩 실행하는 방법



3.1.4.1. 스파이더 열기

The screenshot shows the Spyder Python IDE interface with the following components and callouts:

- Callout 1:** 파이썬 버전과 프로그램이 사용자에게 출력한 내용을 보여 주는 파이썬 콘솔 (Python console showing version and output).
- Callout 2:** 여러 파일을 열 수 있는 프로그램 에디터 창 (Editor window for opening multiple files).
- Callout 3:** 프로그램을 실행하기 전에 오류가 있을 수 있는 줄을 표시해 주는 공간 (Space for displaying error lines before execution).
- Callout 4:** 프로그램에 있는 여러 객체의 값을 보여 주는 변수 탐색기 (Variable explorer showing object values).
- Callout 5:** 프로그램을 한 단계씩 실행하는 방법을 다양하게 제공하는 디버그(Debug) 메뉴 (Debug menu for various execution methods).

The IDE interface includes a menu bar (File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help), a toolbar, and several panels: Editor (showing a Python script), Variable explorer (empty table), File explorer, IPython console (showing execution history), and a status bar at the bottom.

그림 3-2
코드 에디터, 콘솔,
여러 탐색창이 있는 스파이더 IDE

3.2 작업 공간 설정하기



3.2. 작업 공간 설정하기

» 스파이더를 열면 세 영역으로 창이 나뉘져 있다.

- 왼쪽 영역은 에디터다. 처음에는 아무 코드 없이 텍스트만 몇 줄 들어 있다. 텍스트는 녹색이다. 녹색이라는 건 그 텍스트가 실행할 코드가 아니라 여러 줄 주석이라는 뜻이다.
- 오른쪽 위 영역에는 객체 검사기(object inspector)나 변수 탐색기(variable explorer) 또는 파일탐색기(file explorer)가 들어 있다. 이 영역은 당분간 사용하지 않겠지만, 예를 들어 변수 탐색기의 경우 프로그램이 끝난 후 프로그램 안에 있는 각 변수의 값을 보여준다
- 오른쪽 아래 영역은 기본적으로 아이파이썬(IPython) 콘솔이다.

3.2. 작업 공간 설정하기

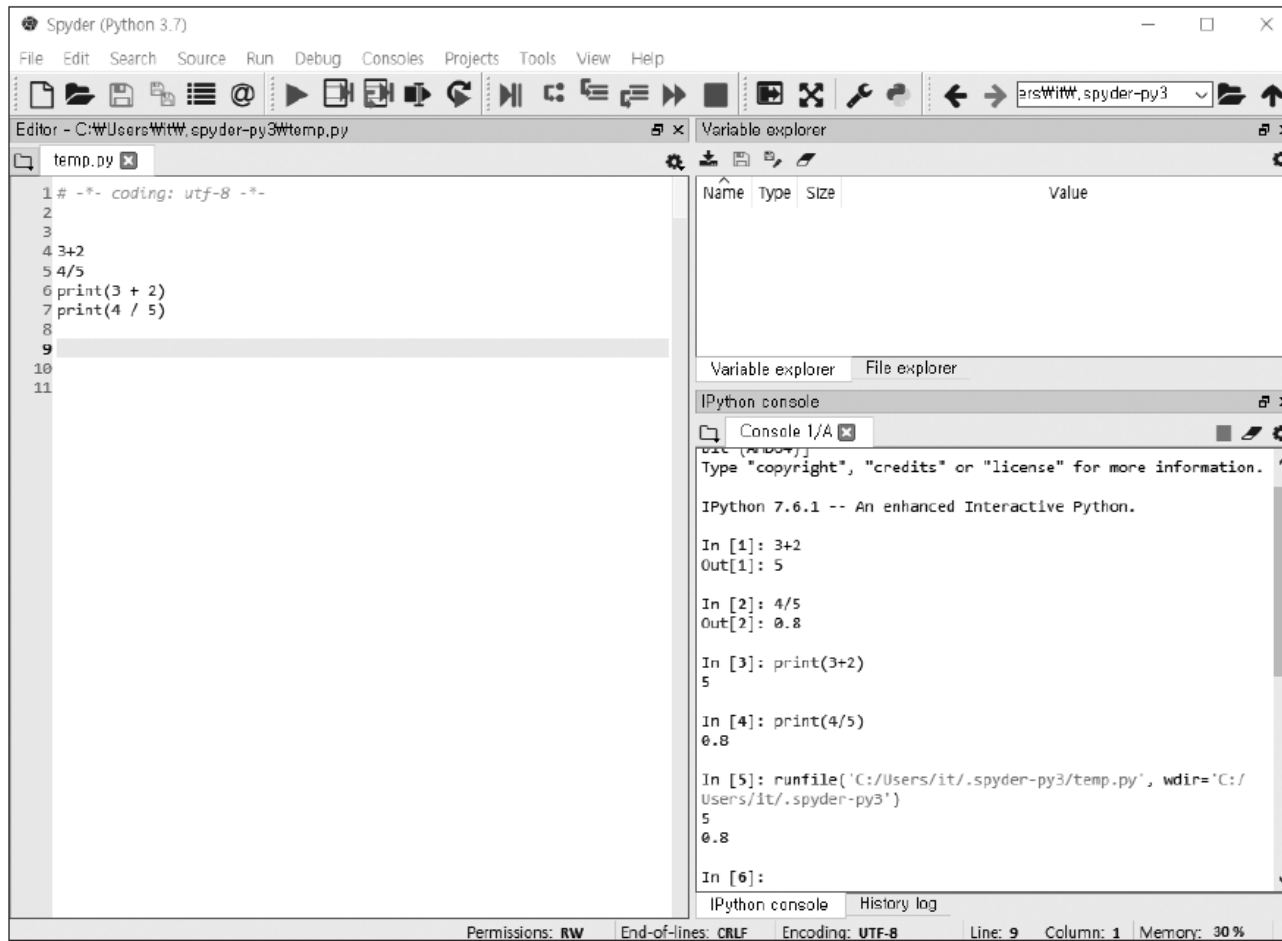


그림 3-3
아이파이썬과 코드 에디터에
식을 입력한 뒤의 스파이더 모습



3.2.1. 아이파이썬 콘솔

» 아이파이썬 콘솔

- 명령을 입력하고 그 결과를 빠르게 살펴보고 싶을 때 우선적으로 사용할 수 있는 기능
- 사용자 입장에서 콘솔을 통해 프로그램과 상호작용할 수 있다는 점이 중요하다.
- 아이파이썬 콘솔은 자동 완성, 이전 명령 히스토리, 특별한 단어를 여러 가지 색으로 하이라이트하는 기능등을 제공하는 고급 콘솔이다.



3.2.1.1. 콘솔에 직접 명령 입력하기

» 아이파이썬 콘솔에 직접 명령을 쓰면 파이썬의 여러 요소가 하는 일을 볼 수 있다

- 콘솔 In[] 부분에 $3 + 2$ 를 입력하고 **Enter** (또는 리턴) 키 누르기 → 5라는 결과와 그 앞에 Out[]이 나타남.
- 콘솔에 $4 / 5$ 를 입력해서 나눗셈을 하기 → Out[] 다음에 0.8이라는 결과가 표시됨

» 결과를 사용자에게 보여 주려면 명시적으로 콘솔 화면에 값을 출력하게 만들어야 한다.

- 콘솔에 `print(3 + 2)`를 입력 → 이번에도 5가 출력될 것이다. 하지만 이번에는 앞에 Out[]이 없다.

» 일반적으로 파이썬에서 값으로 평가될 수 있는 모든 문장을 식이라고 한다

- $3 + 2$ 와 $4 / 5$ 는 모두 파이썬 식(expression, 표현식)이다.



3.2.1.2. 콘솔의 주 사용법

» 프로그램에 버그가 있으면 크기와 관계 없이 버그를 수정하려고 노력해야 한다.

- 버그를 해결하는 디버깅(debugging) 과정에서 많은 것을 배울 수 있다.

» 콘솔을 사용하는 것을 두 가지 관점에서 생각할 수 있다.

- 프로그램을 작성하는 프로그래머의 관점
- 두 번째는 프로그램과 상호작용하는 사람(사용자)의 관점

» 프로그래머는 콘솔을 사용해 주로 명령을 테스트하고 프로그램을 디버깅한다.

» 사용자는 콘솔에서 키보드를 타이핑해 입력을 보내고, 프로그램이 출력한 내용을 보는 방식으로 실행 중인 프로그램과 상호작용한다.



3.2.1.2. 콘솔의 주 사용법

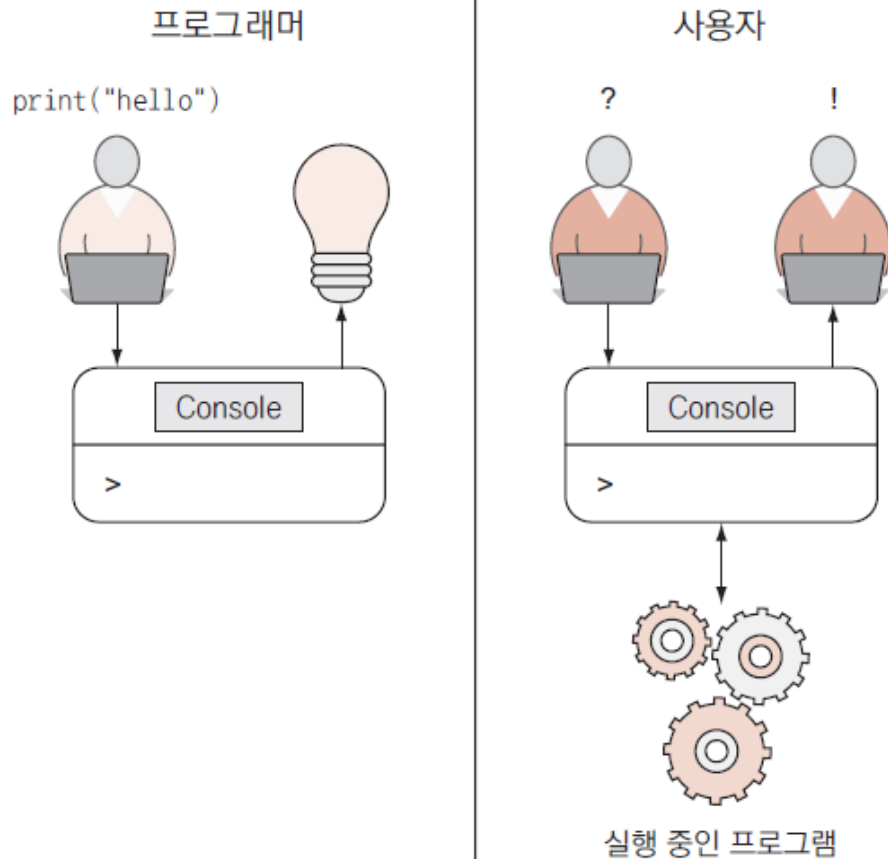


그림 3-4
프로그래머는 콘솔을 테스트와 디버깅에
사용하며, 콘솔에 명령을 입력해 인터프리터와
직접 대화하고 명령의 결과를 살펴본다.

사용자는 콘솔을 통해 프로그램과
상호작용한다. 프로그램이 입력을 요청하면
콘솔에서 키보드로 프로그램에게 입력을
제공하고, 프로그램이 출력한 결과를 본다

3.2.1.2. 콘솔의 주 사용법

» 사용자는 콘솔에서 프로그램의 요청인 프롬프트(prompt)에 따라 텍스트, 숫자, 기호를 입력한다. 그리고 프로그램은 결과를 콘솔에 표시한다.

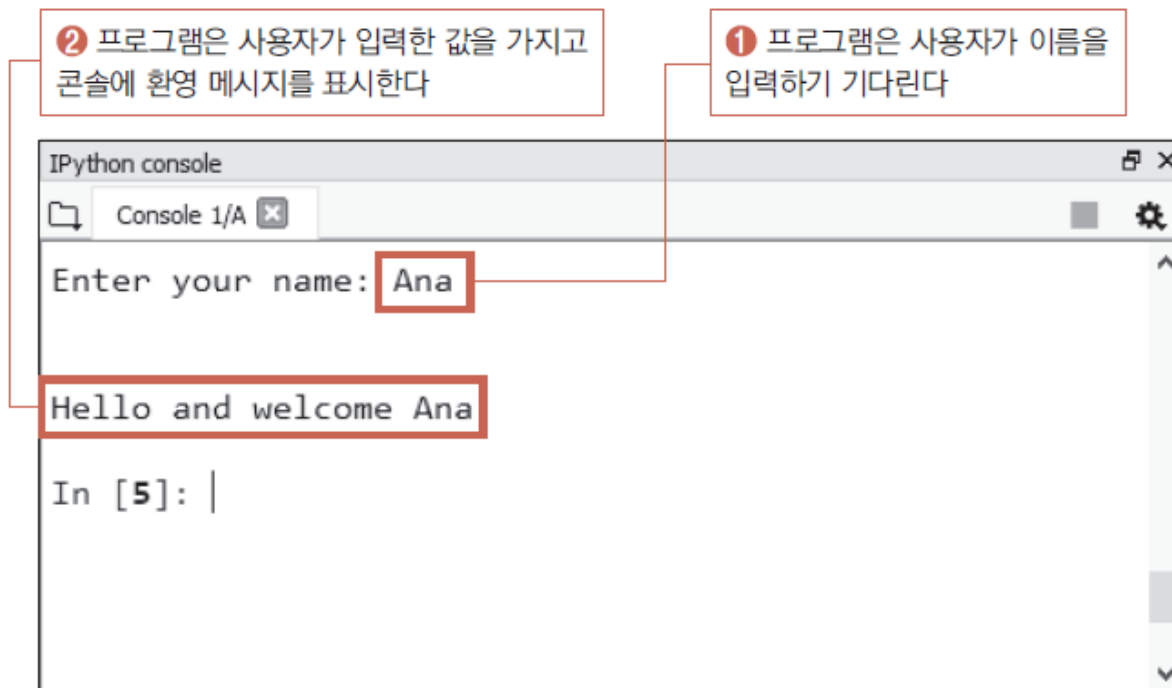


그림 3-5
프로그램과 사용자의 상호작용 예



3.2.1.2. 콘솔의 주 사용법

» 파일 에디터를 사용해더 복잡한 프로그램을 작성할 때도 콘솔에서 최종 값뿐 아니라 프로그램의 중간 객체나 계산 결과 값을 출력해 보면 유용한 경우가 많다.

- 이렇게 해서 프로그램의 중간값을 이해하고, 디버깅을 더 잘 할 수 있다.

» 식을 하나 입력하고 그 결과를 보고 싶을 때도 콘솔이 유용하다.

- 식을 다시 실행하고 싶으면 콘솔에 식을 재입력하거나, ↑ 를 눌러서 예전에 입력한 식 중 원하는 식을 찾아 **Enter** 를 누른다.
- 한 줄보다 더 긴 프로그램을 작성해야 할 경우 파일에 저장하면 시간을 많이 절약할 수 있다.



3.2.2. 파일 에디터

- » 한두 줄짜리 코드가 아닌 더 복잡한 파이썬 프로그램을 작성할 때는 파일 에디터 영역을 사용해야한다.
- » 파일 에디터 영역에서는 명령을 한 줄에 하나씩 입력할 수 있다.
- » 일련의 명령을 입력한 뒤에는 스파이더 상단 도구 모음에 있는 녹색 화살표를 클릭해서 프로그램을 실행할 수 있다.
- » 파일을 편집하고 실행하는 과정은 아나콘다가 지원하는 모든 시스템(PC, 리눅스, macOS)에서 동일하다.

3.2.2. 파일 에디터

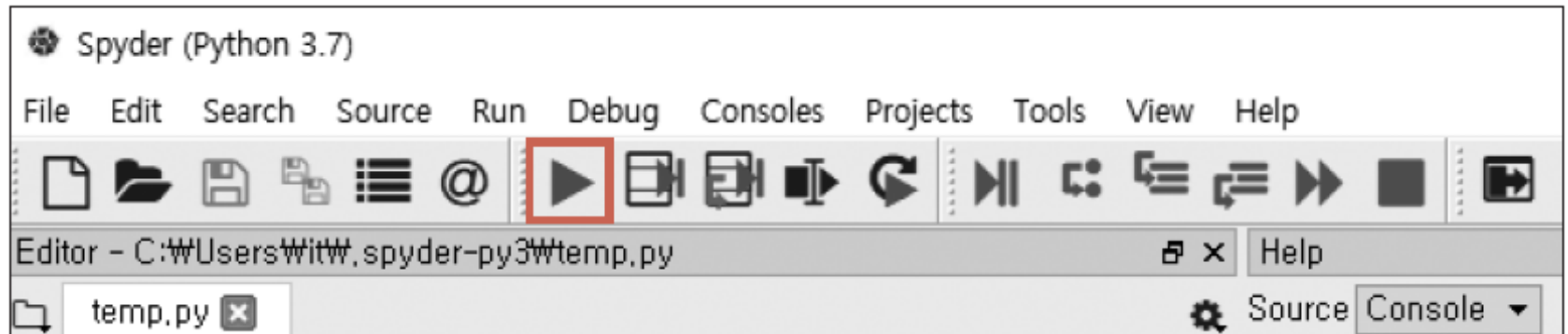


그림 3-6

프로그램을 실행하려면 녹색 화살표 버튼을 클릭한다



3.2.2.1. 코드 각 줄의 실행 결과를 항상 사용자가 볼 수 있는 것은 아니다

» 빈 파일의 8번째 줄에 `3 + 2`라고 입력해 보자. 그 다음 줄에는 `4 / 5`를 입력하자.

» 녹색 화살표 버튼을 클릭해 프로그램을 실행하자.

```
runfile('C:/Users/Justi/.spyder2-py3/temp.py',
wdir='C:/Users/Justi/.spyder2-py3')
```

- 프로그램이 실행됐지만 사용자에게 보여줄 내용이 없다

» 10번째 줄에 `print(3 + 2)`를 입력하고, 다음 줄에 `print(4 / 5)`를 넣자. 그리고 프로그램을 다시 실행하자.

- 파이썬 인터프리터는 파일을 한줄한줄 실행한다. 그래서 맨 처음에 `3 + 2`라는 명령을 보고 내부적으로 이 식의 결과를 계산한다.
- 다음으로 `4 / 5`를 계산한다.
- 두 명령문 모두 파이썬에게 계산 결과를 출력하라고 요청하지는 않았다.
- 따라서 인터프리터는 계산 결과를 콘솔에 표시하지 않는다.



3.2.2.2. 파일 저장하기

» 작성한 프로그램은 각각 별도의 파일로 저장하여 관리해야 한다.

- 스파이더 메뉴 바에서 새파일을 열어보자. 그리고 앞에서 본 두 가지 print 명령문을 새 파일에 다시 입력하라.
- 이제 원하는 디렉터리에 파일을 저장하라. 저장할 때는 .py라는 확장자를 붙여야 한다.

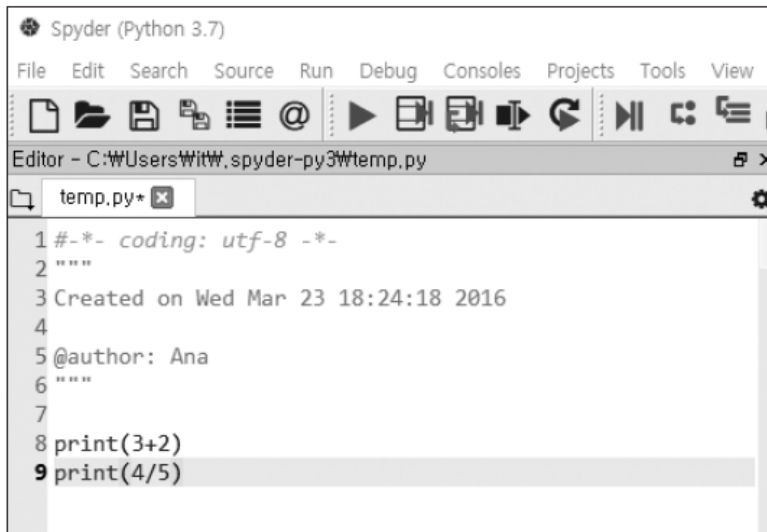


그림 3-7

스파이더에서 여러 파일을 연 모습.

각 파일은 에디터 창에서 서로 다른 탭에 표시된다

» 방금 저장한 파일은 창을 닫아도 프로그램이 사라지지 않는다. File 메뉴에서 그 파일을 다시 열면 된다.



3.2.2.2. 파일 저장하기

» 작성한 프로그램은 각각 별도의 파일로 저장하여 관리해야 한다.

- 스파이더 메뉴 바에서 새파일을 열어보자. 그리고 앞에서 본 두 가지 print 명령문을 새 파일에 다시 입력하라.
- 이제 원하는 디렉터리에 파일을 저장하라. 저장할 때는 .py라는 확장자를 붙여야 한다.

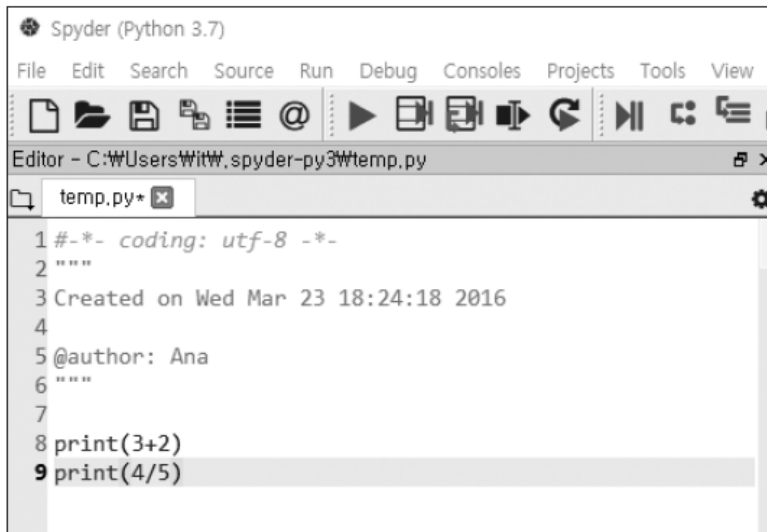


그림 3-7

스파이더에서 여러 파일을 연 모습.

각 파일은 에디터 창에서 서로 다른 탭에 표시된다

» 방금 저장한 파일은 창을 닫아도 프로그램이 사라지지 않는다. File 메뉴에서 그 파일을 다시 열면 된다.

3.3 요약



3.3 요약

- » 아나콘다라는 파이썬 배포판을 설치하는 방법과 파이썬 3.7.1, 스파이더라는 IDE를 사용하는 방법
- » 새 파일을 만들고 간단한 프로그램을 파일에 작성하고 저장한 다음 실행하는 방법
- » 파일 에디터에서 코드를 작성하는 방법과 에디터 창에서 여러 파일을 여는 방법
- » 콘솔을 사용해 변수의 값을 살펴보거나 사용자에게 출력을 보여 주는 방법
- » print 명령을 사용해 식의 값을 콘솔에 출력하는 방법