

01

CHAPTER

파이썬 들여다보기

학습목표

- 프로그래밍 언어를 이해한다.
- 파이썬 프로그래밍 언어를 이해한다.
- 파이썬을 사용할 수 있는 프로그램을 설치한다.
- 간단한 파이썬 프로그램을 만들어 본다.

SECTION 01 프로그래밍 언어의 개념과 종류

SECTION 02 파이썬 소개

SECTION 03 파이썬 설치와 실행

요약

연습문제





참고문헌



1. 파이썬 for Beginner/ 우재남/ 한빛아카데미/2017
2. 두근두근 파이썬/ 천인국/생능출판사/2017

Section 01 프로그래밍 언어의 개념과 종류

■ 프로그래밍 언어의 개념

- 프로그래밍 언어 : 컴퓨터가 이해하는 말로 컴퓨터에서 작동하는 소프트웨어(엑셀, 한글, 인터넷 익스플로러 등)를 만드는 도구
- 프로그래머 : 프로그래밍 언어를 사용해 소프트웨어나 앱을 만드는 사람



그림 1-1 프로그래머, 프로그래밍 언어, 소프트웨어

Section 01 프로그래밍 언어의 개념과 종류

■ 프로그래밍 언어의 종류

- 수백 가지가 넘는 종류 중 많이 사용되는 프로그래밍 언어는 C/C++, 자바(Java), HTML, PHP, 파이썬 등



다양한 스포츠



다양한 프로그래밍 언어

그림 1-2 스포츠와 프로그래밍 언어 비교

Section 02 파이썬 소개

■ 파이썬 역사

- 배우기도 쉽고 결과도 바로 확인할 수 있어 초보자에게 적합한 프로그래밍 언어
- 귀도 반 로섬(1956년~)이라는 프로그래머가 C 언어로 제작해 1991년에 공식으로 발표
- 사전적인 의미는 비단뱀으로 로고도 파란색과 노란색 비단뱀 두 마리가 서로 얹혀 있는 형태



그림 1-3 파이썬 로고(출처 : <https://www.python.org>)



그림 1-4 파이썬의 창시자 귀도 반 로섬(출처 : 위키피디아)

여기서 잠깐



파이썬의
다양한 분류

귀도 반 로섬이 만든 파이썬은 C로 만들어져 CPython이라고도 하며 일반적으로 사용하는 파이썬은 CPython을 의미한다. 그 외에 다양한 개발자 또는 프로젝트가 파이썬의 분기된 언어를 개발했다. 대표적인 예로 자바로 구현된 Jython, C#으로 구현된 IronPython, CPython으로 작성한 PyPy, CPython의 C 스택 문제를 없앴 Stackless Python, CPython에 강력한 기능이 추가된 IPython, 웹 브라우저에서 실행되는 Brython 등이 있다.

■ 파이썬 특징

❶ 강력한 기능을 무료로 사용할 수 있다

- 파이썬은 오픈 소스이며, 비용을 지불하지 않고 무료로 사용 가능. 다양한 추가 라이브러리도 무료

❷ 읽기 쉽고 사용하기 쉽다

- 직관적인 코드를 사용해 C나 자바 같은 언어보다 읽기 쉬워 프로그램을 빨리 제작할 수 있어 비용 절감 효과 제공

❸ 사물인터넷과 잘 연동된다

- 라즈베리파이 기반의 사물인터넷이 파이썬을 잘 지원하므로 사물인터넷 개발 및 운영에 적극 활용

❹ 다양하고 강력한 외부 라이브러리들이 풍부하다

- 파이썬에서 제공하는 라이브러리뿐 아니라, 외부에서 제공하는 다양한 서드 파티(Third Party) 라이브러리까지 사용 가능

❺ 강력한 웹 프레임워크를 사용할 수 있다

- 파이썬의 웹 프레임워크를 사용해 강력하고 빠른 웹 환경을 구축 가능

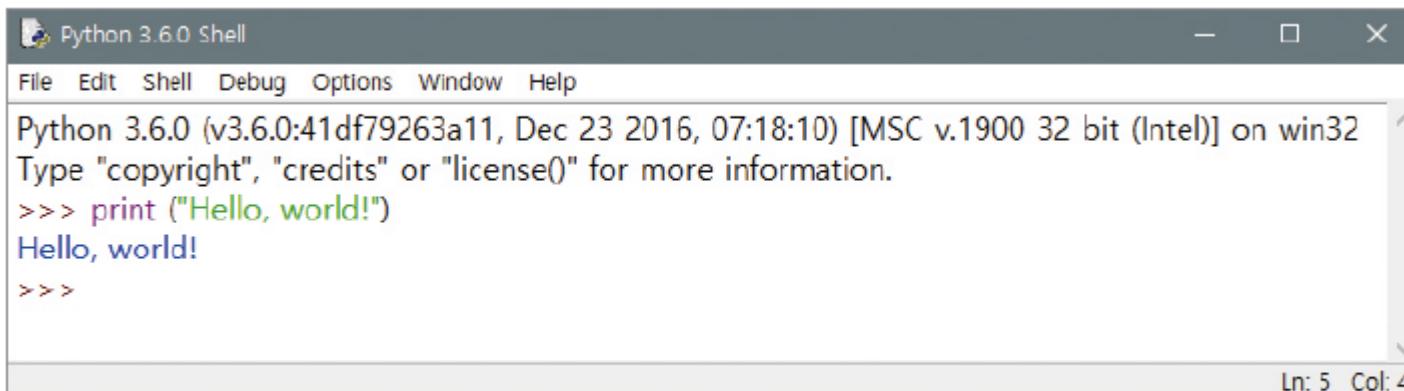
Section 02 파이썬 소개

■ 파이썬의 단점

- 느린 속도
 - 파이썬은 컴파일러 언어가 아닌 스크립트 언어이기 때문에 컴파일러 언어보다 느림
→ 이를 보완하려고 많은 파이썬 패키지를 최적화시키고 있음
- 모바일 컴퓨팅 분야에 지원이 약하고 하드웨어 제어 등과 관련된 부분 사용이 어려움

■ 파이썬의 실행 화면

- `print("Hello, world!")`를 입력한 후 [Enter]를 눌러 Hello, world!를 출력한 화면
- `print`는 무언가를 프린트하라는 의미이므로 `print()`에서 괄호 안에 있는 것을 화면에 출력



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print ("Hello, world!")
Hello, world!
>>>
```

Ln: 5 Col: 4

그림 1-5 파이썬 실행 화면



여기서 잠깐

컴파일러
언어와
스크립트 언어

컴파일러(Compiler) 언어는 소스 코드를 실행 가능한 기계어로 일괄 번역한 후 번역이 완료된 파일(*.exe, *.class 등의 파일)을 실행하는 언어를 의미한다. 소스 코드를 기계어로 번역하는 과정을 컴파일(Compile)이라고 하며, 이 작업을 하는 프로그램을 컴파일러라고 한다. 대표적인 컴파일러 언어로는 C/C++, 자바 등이 있다. 이와 달리 스크립트 언어(또는 인터프리터 언어)는 소스 코드를 한 줄씩 읽어 실행되어 별도의 실행 파일이 생성되지 않는데, 이때 한 줄씩 처리하는 프로그램을 인터프리터(Interpreter)라고 한다. 대표적인 스크립트 언어로는 파이썬, 자바스크립트(JavaScript), 펄(Perl) 등이 있다.

일반적으로 컴파일러 언어는 소스 코드를 일괄적으로 기계어로 번역해 놓기 때문에 실행 속도가 스크립트 언어보다 빠르다. 하지만 컴파일러 언어는 배우는 데 시간이 오래 걸리는 반면, 스크립트 언어는 대부분 빠른 시간에 배울 수 있다는 장점이 있다.

여기서 잠깐



Hello World 프로그램

다른 프로그래밍 언어를 공부한 적이 있다면, 화면에 Hello, world!를 출력하는 Hello World 프로그램에 익숙할 것이다. 이 프로그램은 대부분의 프로그래밍 언어 책에서 처음으로 만드는 기본 예제이다. [그림 1-5]도 이런 맥락에서 작성한 것이다. 다음은 C나 자바로 작성된 Hello World 프로그램이다.

C 프로그램	자바 프로그램
<pre>#include <stdio.h> int main() { printf("Hello, world!\n"); return 0; }</pre>	<pre>public class HelloWorldApp { public static void main(String[] args) { System.out.println("Hello, world!"); } }</pre>

Hello World 프로그램은 1978년에 출판된 브라이언 커니핸과 데니스 리치가 쓴 『The C Programming Language』에서 처음 사용된 것으로 알려져 있다.

Section 03 파이썬 설치와 실행

■ 파이썬 다운로드 전 준비 과정(윈도의 확장명 표시)

- 윈도우 10은 파일 탐색기 실행→[보기] 메뉴 선택 → '파일 확장명' 체크
- 윈도우 7은 [구성]-[폴더 및 검색 옵션] 메뉴 선택 → [폴더 옵션] 대화상자의 [보기] 탭 클릭 → '알려진 파일 형식의 파일 확장명 숨기기'의 체크 표시 해제 → [확인] 버튼

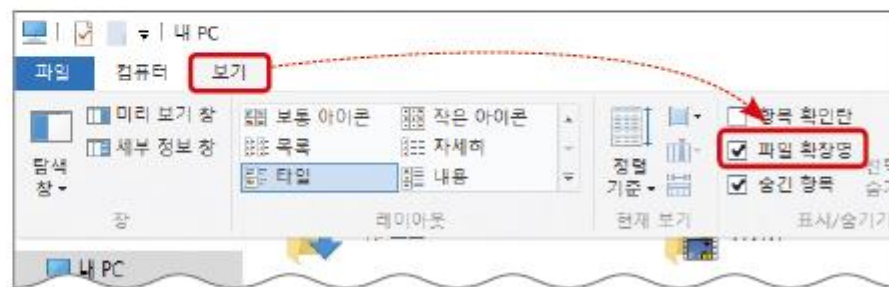
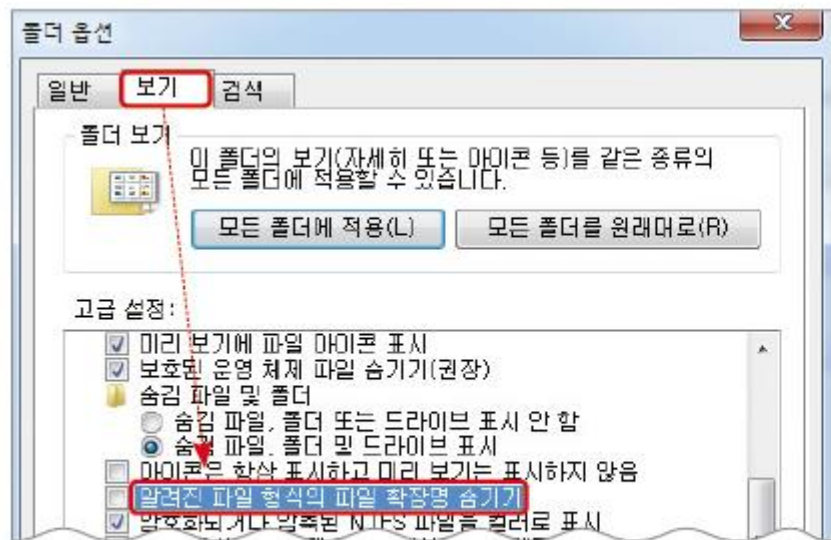


그림 1-6 윈도 7 및 윈도 10에서 확장명 표시

Section 03 파이썬 설치와 실행

■ 파이썬 다운로드

- <http://www.python.org/>에 접속 → [Downloads]-[Download Python 3.x.x] 클릭
→ 설치 파일인 python-3.x.x.exe를 원하는 위치에 저장

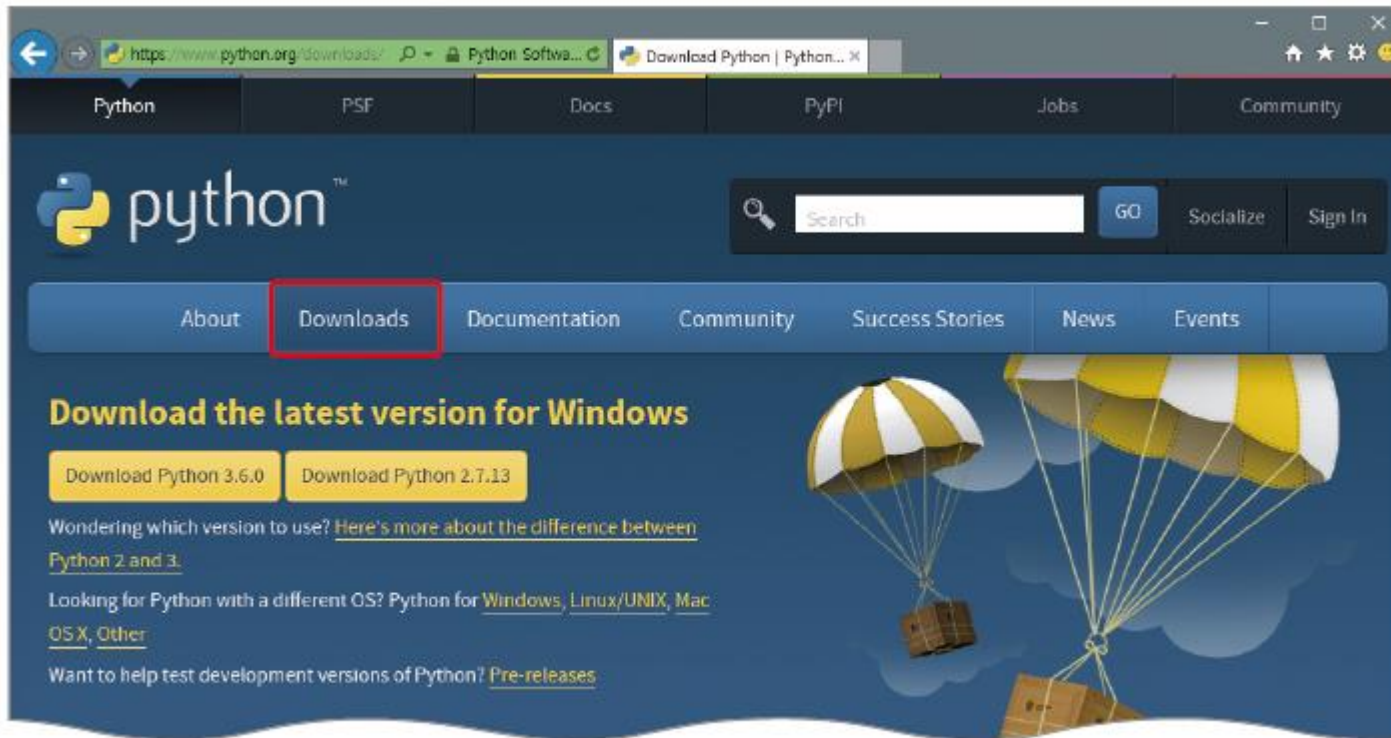


그림 1-7 파이썬 다운로드 화면

여기서 잠깐



파이썬 2.x와 3.x

파이썬은 크게 2.x 버전과 3.x 버전으로 나누는데, 이 책을 집필하는 시점에는 각각 2.7.13 버전과 3.6.0 버전까지 발표되었다. 이렇게 나누는 가장 큰 이유는 두 버전이 호환되지 않아 2.x로 작성된 코드를 3.x에서 사용하려면 코드를 일부 수정해야 하기 때문이다.

기존에 작성된 파이썬 코드들은 2.x로 된 것이 많다. 이런 코드들을 일괄적으로 3.x로 수정하기도 어렵고, 잘 작동하는 코드를 굳이 변경할 필요도 느끼지 못하기 때문이다. 하지만 파이썬 2.x가 2.7 버전을 마지막으로 더 이상 업데이트되지 않아 향후에는 3.x 버전만 업데이트될 것이다. 2.x에서 작성한 코드를 사용할 경우가 아니라면 가능한 3.x 버전을 사용하는 것이 최신의 파이썬 기능을 100% 활용할 수 있는 방법이다.

이 책에서도 집필 시점을 기준으로 최신인 3.6.0을 사용할 것이다. 이후에 더 버전을 업데이트해도 별 문제가 없을 것이라 생각되지만, 책과 완전히 동일한 버전을 사용하고 싶다면 이 책의 자료실 (<http://www.hanbit.co.kr/src/4359>)에서 python-3.6.0.exe 파일을 다운로드해서 사용한다.

Section 03 파이썬 설치와 실행

■ 파이썬 설치

- python-3.x.x.exe를 더블클릭 실행 → Add Python 3.6 to PATH에 체크
→ <Install Now> 버튼 클릭(<Customize Installation> 버튼으로 설치 폴더 변경 가능)
→ 설치 진행 → 설치를 마치면 <Close> 버튼 클릭

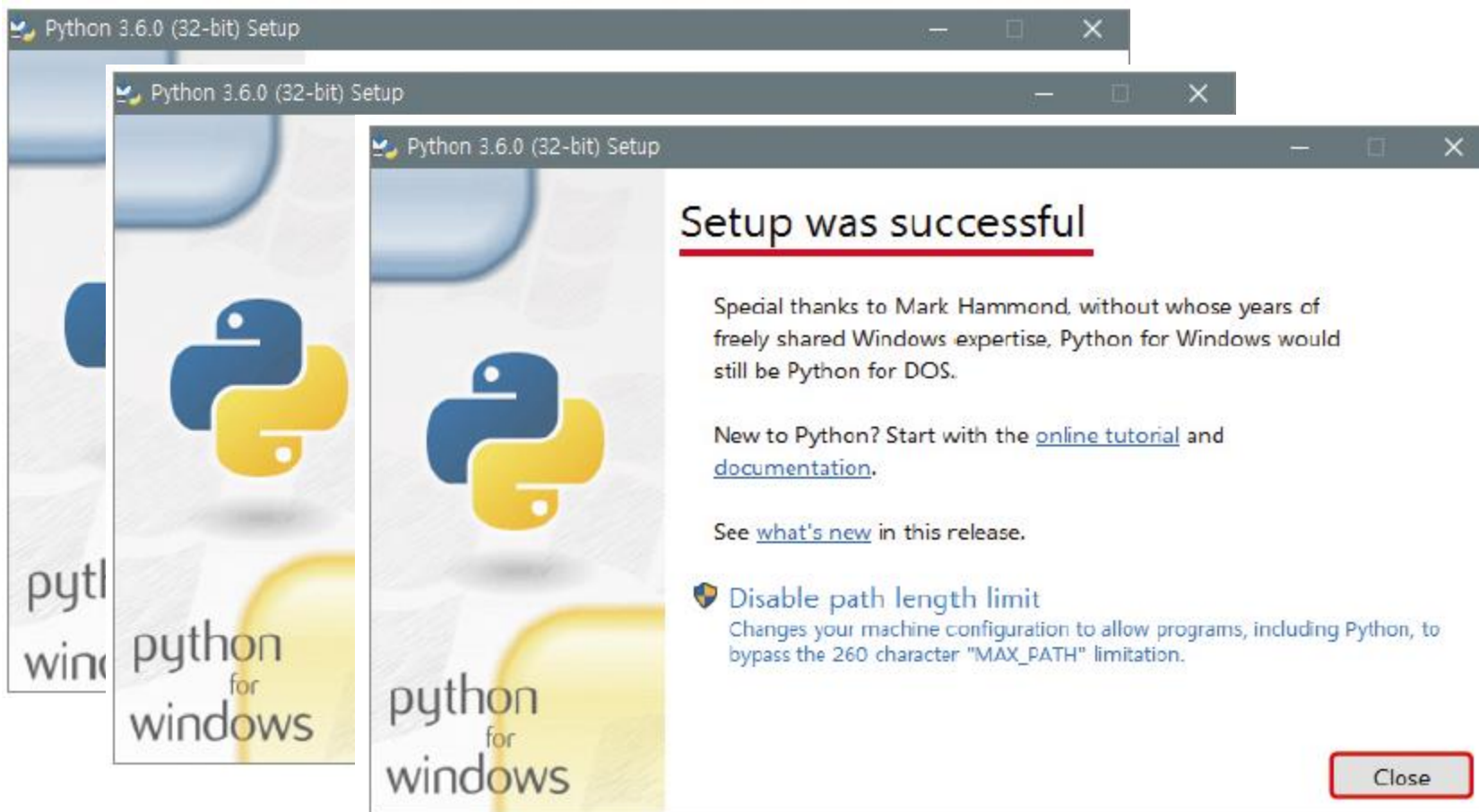


그림 1-8 파이썬 설치 화면

Section 03 파이썬 설치와 실행

■ 파이썬 실행

- 윈도우의 <시작> 버튼

→ [모든 프로그램]-[Python 3.6]-[IDLE (Python 3.6 32-bit)] 메뉴 선택

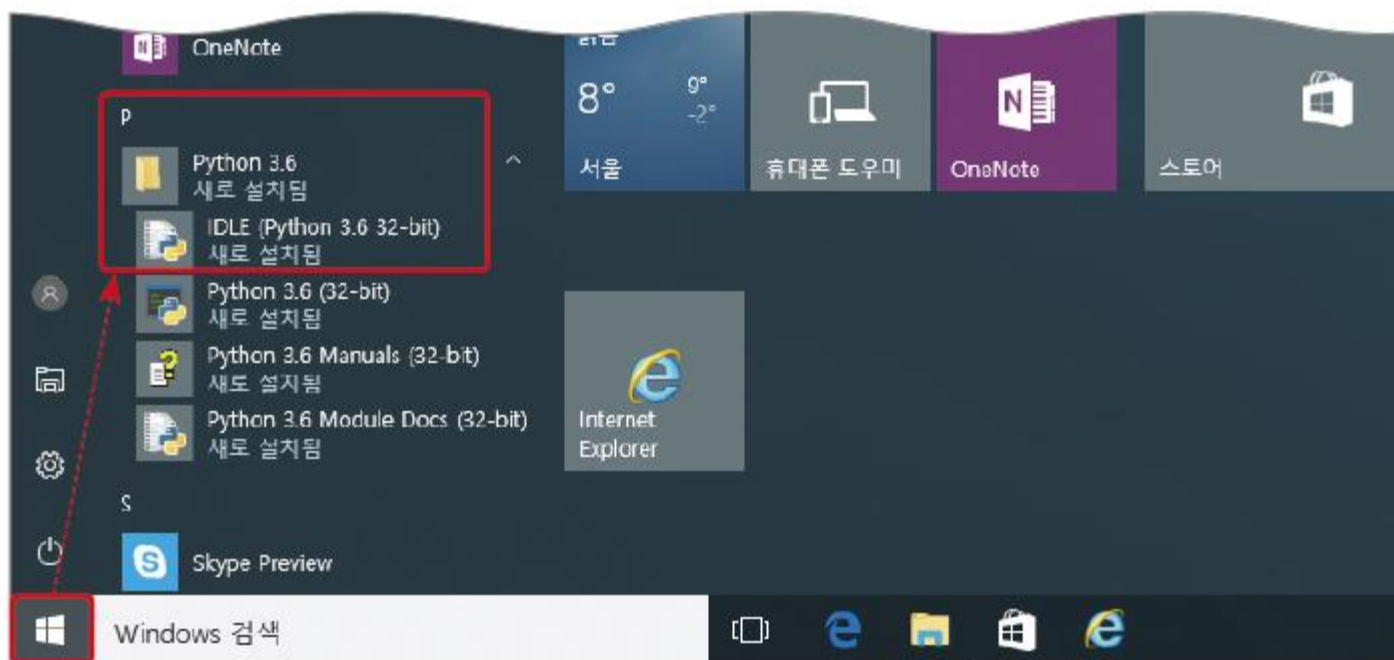


그림 1-9 IDLE 실행

Section 03 파이썬 설치와 실행

■ 파이썬 실행

- 윈도우의 <시작> 버튼

- [모든 프로그램]-[Python 3.6]-[IDLE (Python 3.6 32-bit)] 메뉴 선택

- IDLE이 시작되며 파이썬 셸(Python Shell)이 대화형 모드로 나타남

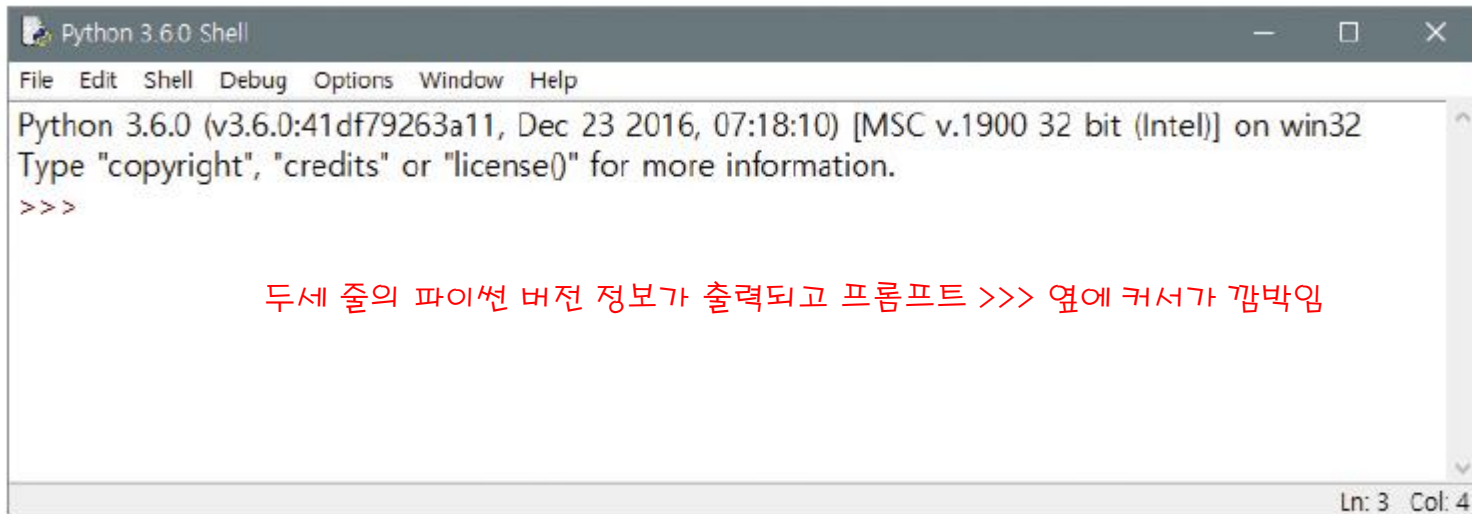


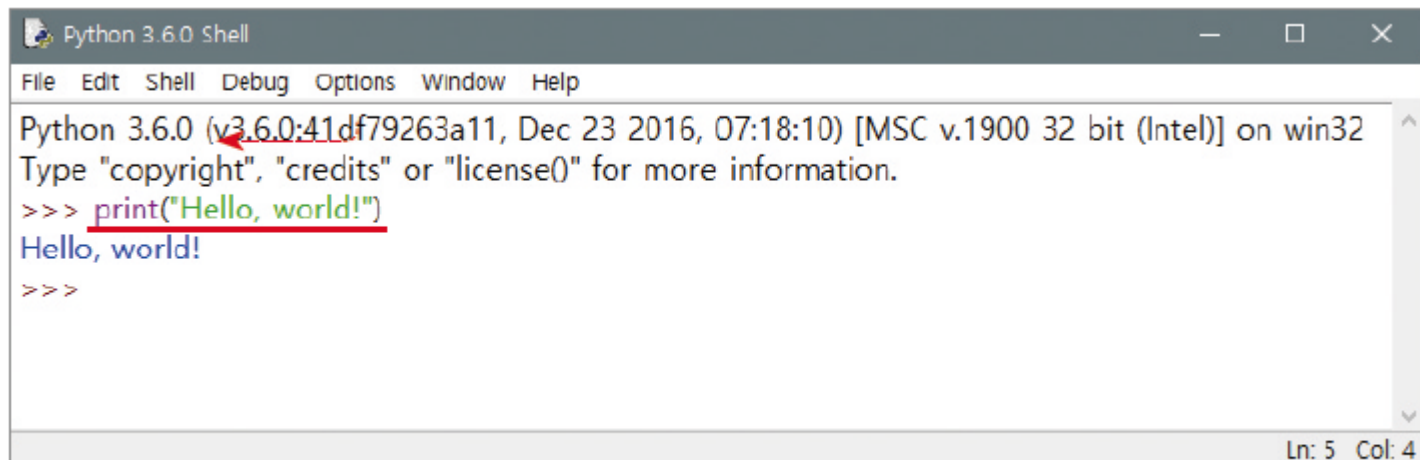
그림 1-10 IDLE 실행 화면

Section 03 파이썬 설치와 실행

■ 파이썬 코드 입력과 실행 예

■ 예1

- >>> 다음에 `print("Hello, world!")`를 입력하고 [Enter]



The screenshot shows a 'Python 3.6.0 Shell' window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The main text area displays the following content: 'Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32', 'Type "copyright", "credits" or "license()" for more information.', the prompt '>>>' followed by the command 'print("Hello, world!")', the output 'Hello, world!', and another prompt '>>>'. A red arrow points to the version number 'v3.6.0' in the first line. The status bar at the bottom right indicates 'Ln: 5 Col: 4'.

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello, world!")
Hello, world!
>>>
```

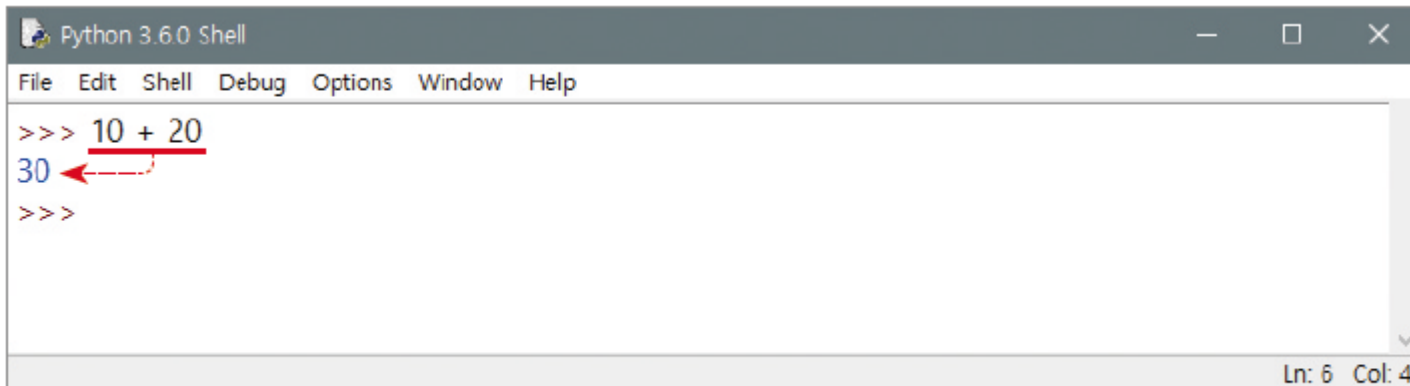
그림 1-11 코드 입력과 실행 예 1

Section 03 파이썬 설치와 실행

■ 파이썬 코드 입력과 실행 예

■ 예2

- >>> 다음에 다음 계산식을 입력하고 [Enter]



The screenshot shows a window titled "Python 3.6.0 Shell" with a menu bar containing "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area contains the following text:
 >>> 10 + 20
 30
 >>>
 A red underline is under "10 + 20", and a red arrow points from the underline to the number "30". The status bar at the bottom right shows "Ln: 6 Col: 4".

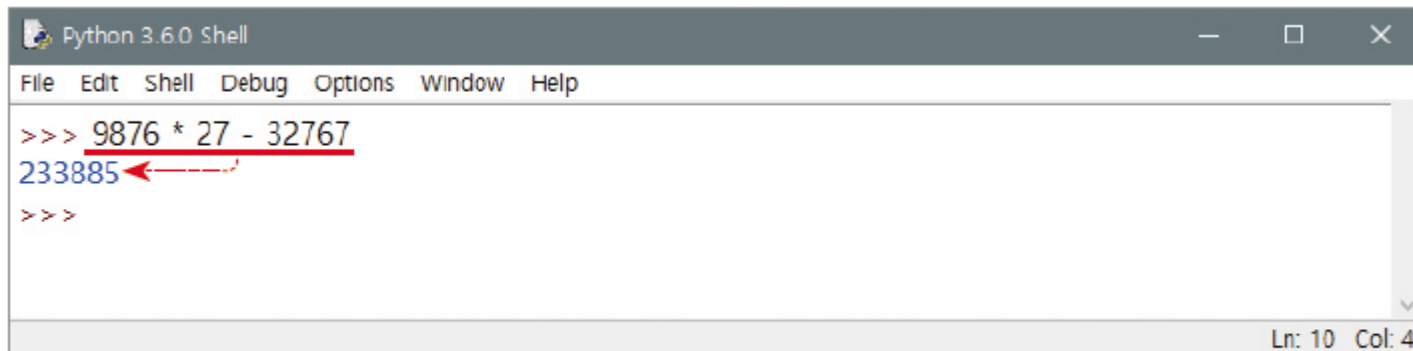
그림 1-12 코드 입력과 실행 예 2

Section 03 파이썬 설치와 실행

■ 파이썬 코드 입력과 실행 예

■ 예3

- >>> 다음에 다음 계산식을 입력하고 [Enter]



The screenshot shows a Python 3.6.0 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The command prompt shows the input `>>> 9876 * 27 - 32767` on a line that is underlined in red. The output `233885` is displayed on the next line, with a red arrow pointing from the end of the input line to the output. The prompt `>>>` is shown on the following line. The status bar at the bottom right indicates 'Ln: 10 Col: 4'.

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>> 9876 * 27 - 32767
233885
>>>
```

그림 1-13 코드 입력과 실행 예 3

Section 03 파이썬 설치와 실행

- 파이썬 IDLE 종료
 - [File]-[Exit] 메뉴 선택

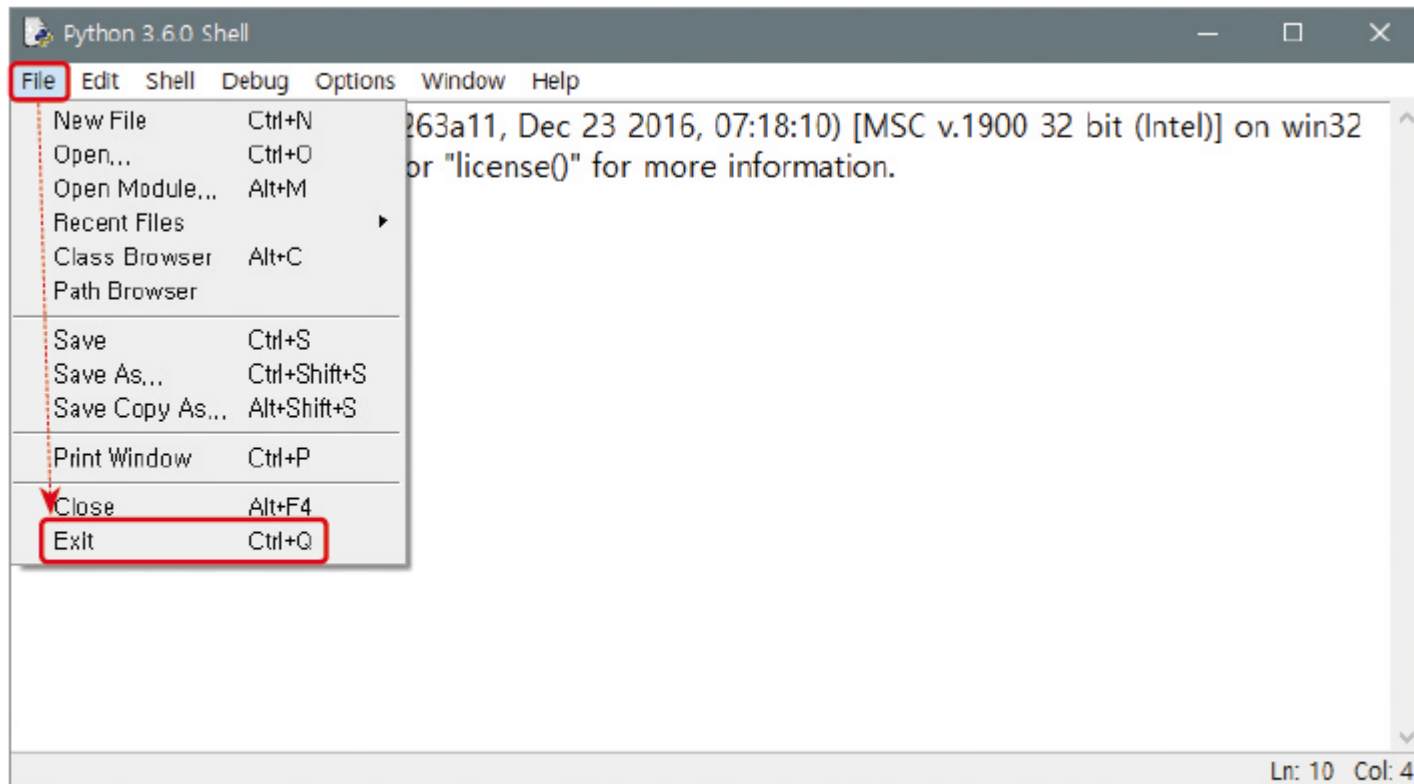


그림 1-14 IDLE 종료

스크립트 모드

- 코드가 복잡해지면 인터프리트 모드는 번거롭다.

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '%s [%s]' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s' % ast[1]  
        else:  
            print ''  
    else:  
        print '['  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print '%s -> {' % nodename,  
        for name in children:  
            print '%s' % name,
```



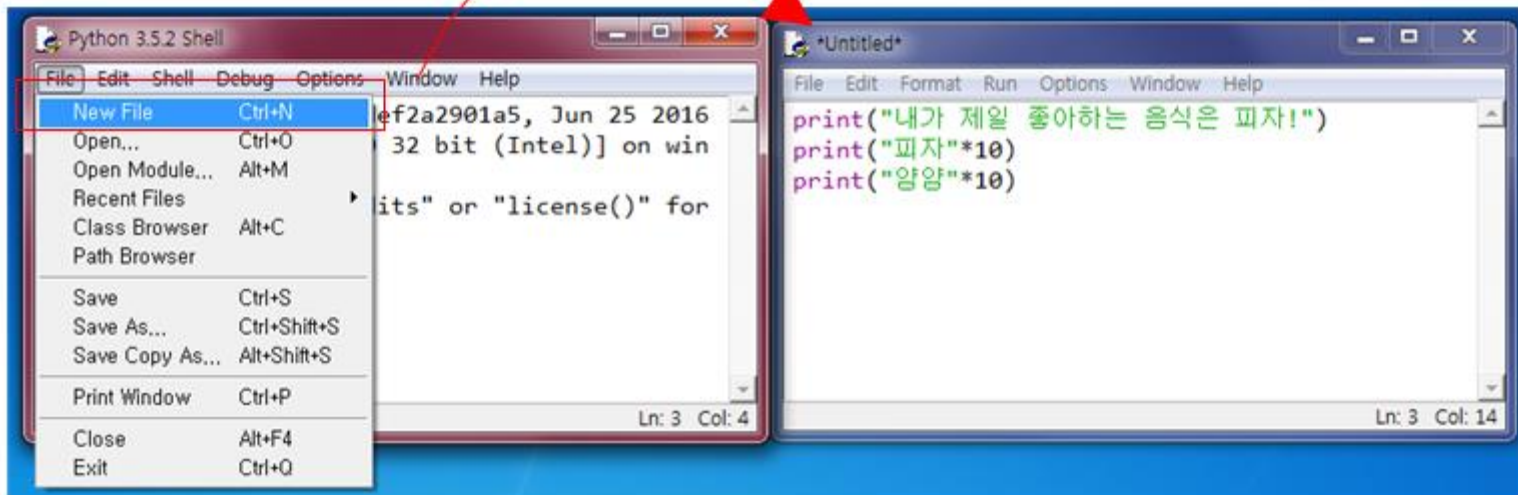
소스 파일 작성하기

- 텍스트 에디터를 이용하여 명령어들을 파일에 저장한 후에 파일을 읽어서 명령어들을 하나씩 실행하는 방법이 있다. 명령어들이 저장된 파일을 **소스 파일(source file)**이라고 한다.



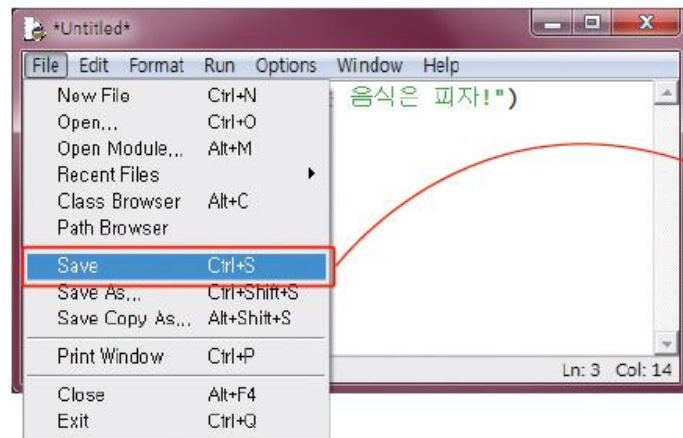
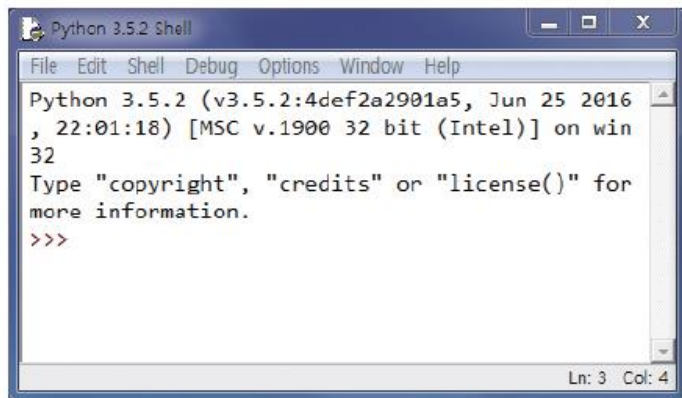
IDLE를 이용한 소스 파일 작성

- 파이썬 셸의 메뉴 중에서 [File] -> [New File]을 선택한다.



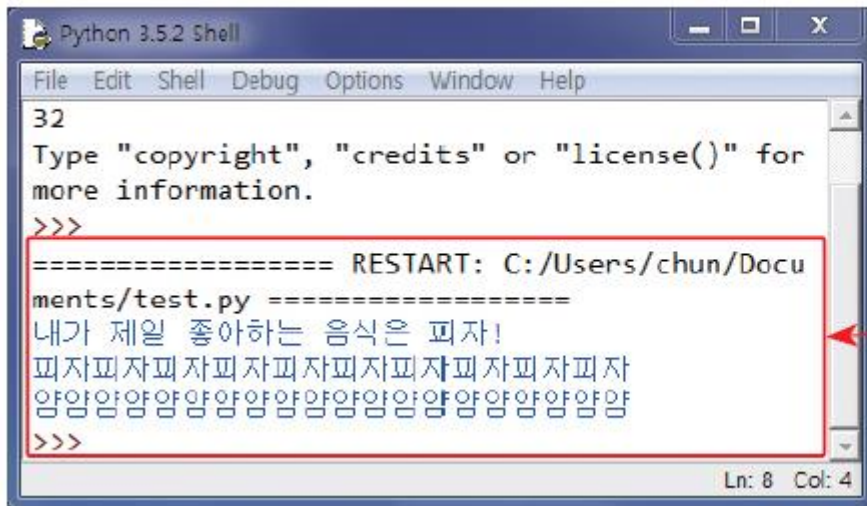
IDLE를 이용한 소스 파일 저장

- 텍스트 에디터의 [File]->[Save] 메뉴를 선택하여 코드를 파일로 저장한다.



IDLE를 이용한 소스 파일 실행

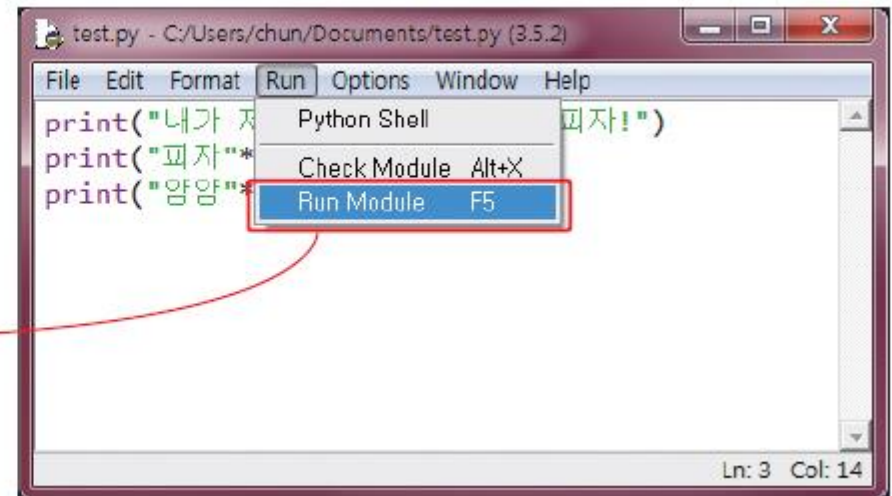
- 파이썬 셸의 메뉴 중에서 메뉴 [Run]->[Run Module]을 선택한다.



Python 3.5.2 Shell

```
File Edit Shell Debug Options Window Help
32
Type "copyright", "credits" or "license()" for
more information.
>>>
===== RESTART: C:/Users/chun/Docu
ments/test.py =====
내가 제일 좋아하는 음식은 피자!
피자피자피자피자피자피자피자피자피자
암암암암암암암암암암암암암암암암암
>>>
```

Ln: 8 Col: 4



test.py - C:/Users/chun/Documents/test.py (3.5.2)

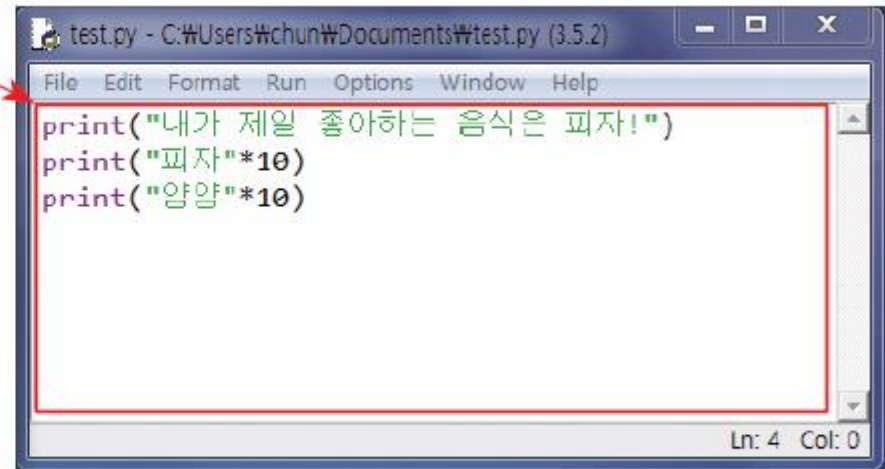
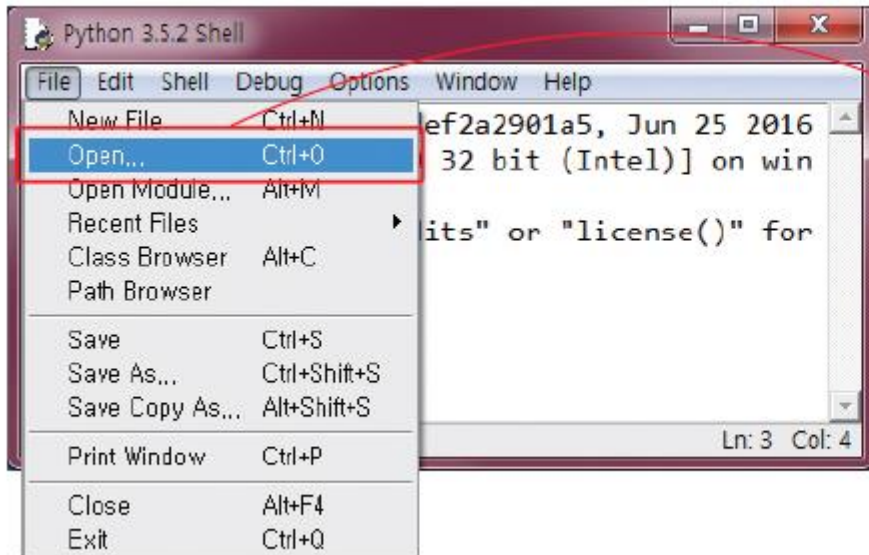
```
File Edit Format Run Options Window Help
print("내가 제일 좋아하는 음식은 피자!")
print("피자"*10)
print("암암"*10)
```

Python Shell
Check Module Alt+X
Run Module F5

Ln: 3 Col: 14

소스 파일 다시 열기

- IDLE의 [File]->[Open] 메뉴를 선택한다. 우리가 저장하였던 폴더로 가서 원하는 파일을 선택한다. 텍스트 에디터가 나오고 우리가 입력하였던 소스가 다시 보일 것이다.



문법적인 오류

```
>>> pront("Hello World")
```

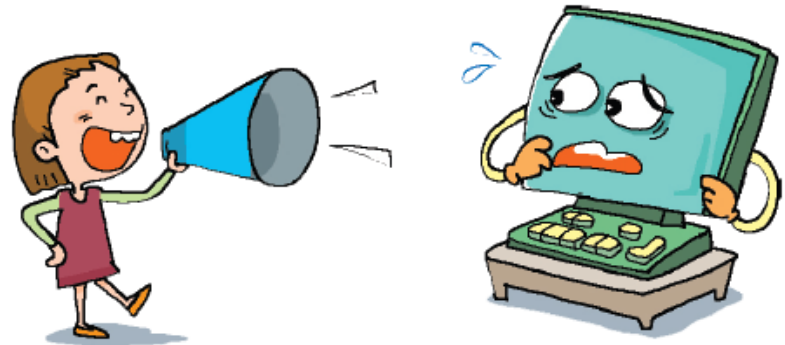
```
SyntaxError: invalid syntax
```

```
>>> 1 +
```

```
SyntaxError: invalid syntax
```

```
>>> 3 +* 2
```

```
SyntaxError: invalid syntax
```



실행시 오류

```
print("안녕하세요? 파이썬에 오신 것을 환영합니다!")  
print("프로그래밍 공부를 즐기셨으면 합니다.")  
print("안녕!" + 3)
```

```
===== RESTART: D:\Ws.py
```

```
=====
```

```
안녕하세요? 파이썬에 오신 것을 환영합니다!  
프로그래밍 공부를 즐기셨으면 합니다.
```

```
Traceback (most recent call last):
```

```
File "D:\Ws.py", line 3, in <module>
```

```
print("안녕!" + 3)
```

```
TypeError: Can't convert 'int' object to str implicitly
```

파이썬으로 무엇을 만들 수 있을까?

- 아래의 소스를 입력하고 실행해보자.

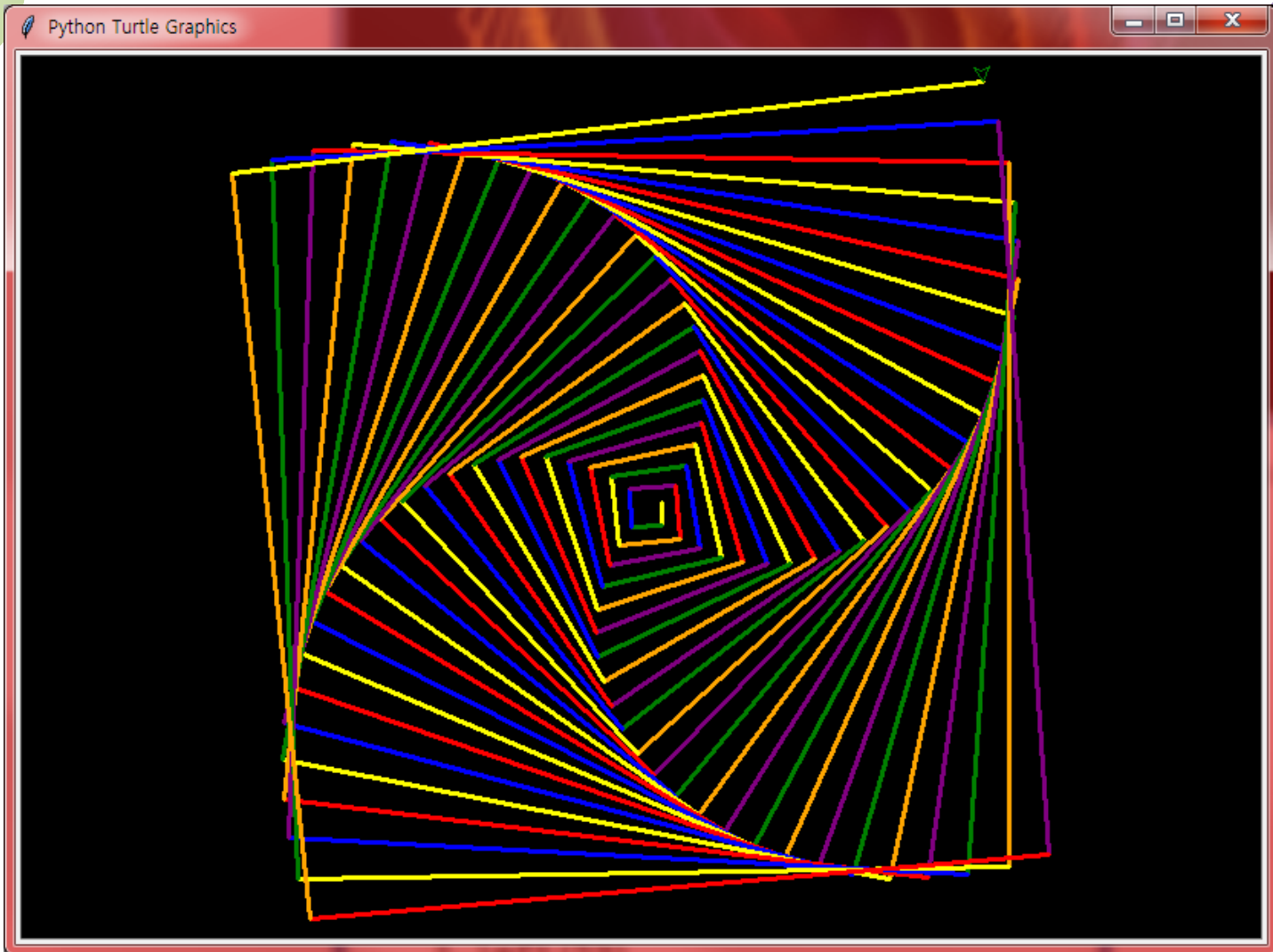
```
import turtle

colors = ["red", "purple", "blue", "green", "yellow", "orange" ]
t = turtle.Turtle()

turtle.bgcolor("black")
t.speed(0)
t.width(3)
length = 10

while length < 500:
    t.forward(length)
    t.pencolor(colors[length%6])
    t.right (89)
    length += 5
```

실행 결과는?





Thank You
