

Chapter 01. 파이썬 시작하기

01. 소개

02. 설치 및 개발 환경

03. 파이썬 프로그램 실행하기

1. 소개

1.1 파이썬이란?

- 1991년 귀도 반 로섬(Guido van Rossum) 이 발표한 인터프리터 언어
- 귀도가 구글에 입사하고 구글의 3대 개발언어 중 하나로 채택되면서 유명세를 탔다.
- 많은 상용 응용 프로그램에서 스크립트 언어로 채용
- Pyrex, Psycho, NumPy 등을 이용하면 수치를 빠르게 연산할 수 있기 때문에 과학 기술 컴퓨팅, 공학 분야에서도 많이 이용되고 있다.

1.2 특징

- 대화 기능의 인터프리터 언어
- 동적타이핑(동적인 데이터 타입 결정 지원)
- 플랫폼 독립적 언어(Unix, Linux, Window9x/NT 계열, DOS, OS/2, Mac OS, Nokia 등)
- 간단하고 쉬운 문법
- 가독성(코드블러의 들여쓰기 구분)
- 비교적 짧은 개발 시간
- 고수준의 내장 객체 자료형 제공(List, Dictionary, String, Turtle 등의 자료구조)
- 메모리 자동 관리
- 무료(비영리의 파이썬 소프트웨어 재단이 관리하는 개방형, 공동체 기반 개발 모델)
- 많은 수의 풍부한 라이브러리 제공
- 높은 확장성(접착성, Glue Language)
- 유니코드



1. 소개

1.3 파이썬의 종류

Cpython

C로 작성된 파이썬 인터프리터

Jython

Java로 작성된 파이썬 인터프리터

IronPython

.Net 플랫폼 용 파이썬 인터프리터, C#으로 구현

PyPy

파이썬으로 작성된 파이썬 인터프리터

1.4 활용 분야

GUI

Tcl/tk를 이용한 UI, wxPython(Window 인터페이스)

웹 프로그래밍

Django, Flask

인터넷 프로그래밍

socket 응용 프로그램, SOAP RPC 지원, 인터넷 프로토콜(FTP, SMTP, HTTP)

DB 프로그래밍

SQLite 내장, Oracle, DB2, Sybase, MySQL 등 각종 유명 DB 인터페이스 제공

1. 소개

1.4 활용 분야

텍스트 처리

문자열 처리가 뛰어남, 정규식, XML 처리

데이터 분석

Numpy, pandas 라이브러리를 활용한 데이터 분석

Matplotlib 라이브러리를 활용한 그래프 또는 2차원 데이터 시각화

SciPy를 활용한 과학/공학 계산

병렬연산

IPython을 이용한 병렬 연산

1.5 유명 프로젝트

1) BitTorrent, Trac, Yum

2) Flask, CherryPy, Django

3) GIMP, Maya, Paint Shop Pro

4) youtube.com, google groups, google maps, gmail

Chapter 01. 파이썬 시작하기

01. 소개

02. 설치 및 개발 환경

03. 파이썬 프로그램 실행하기

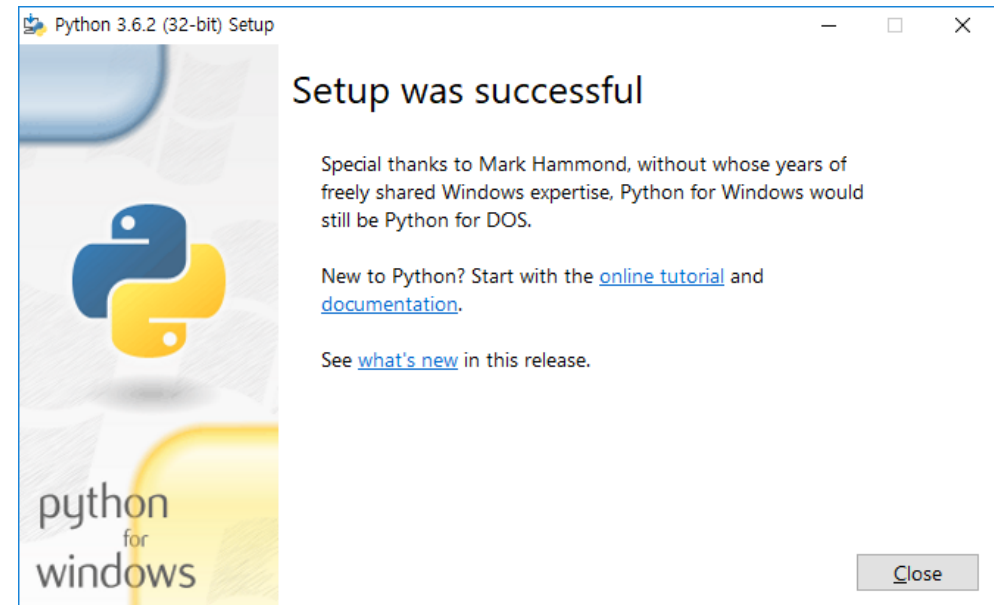
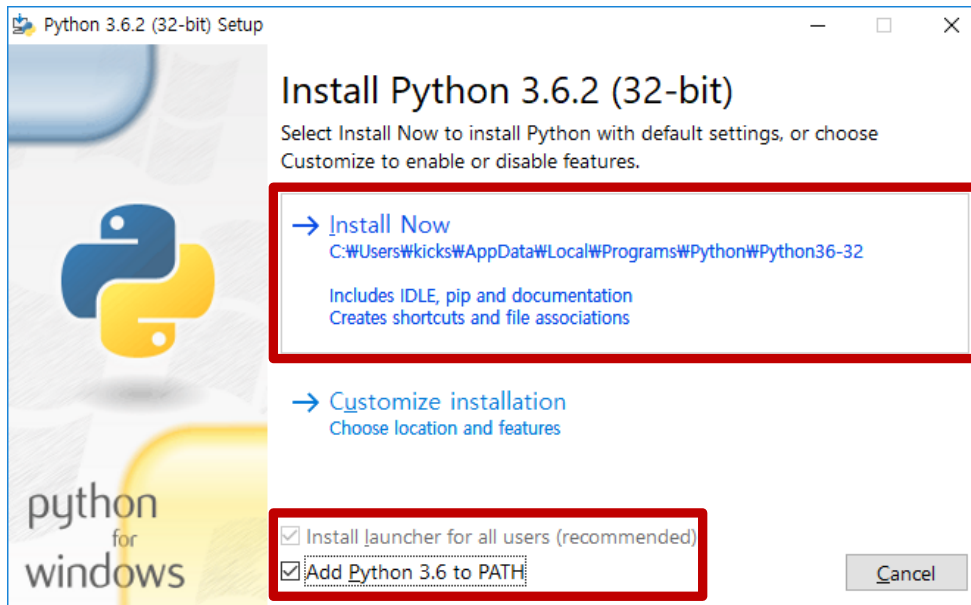
2. 설치 및 개발환경

2.1 다운로드

<https://www.python.org/downloads>

컴퓨터 환경에 맞는 최신 버전 다운로드 (2017년 9월 현재 3.6.x)

2.2 설치



2. 설치 및 개발환경

2.3 파이썬 설치 요소

Python(Command Line)

커맨드 라인에서 수행되는 대화식 인터프리터이다.

IDLE(Python GUI)

파이썬 표준 GUI 대화식 인터프리터

Module Docs

원하는 모듈을 쉽게 찾을 수 있는 도구

원하는 검색 단어를 입력하면 관련 모듈을 찾아준다.

Python Manuals

HTML 파일 형식의 파이썬 온라인 문서 모음.

주로 Library Reference 를 주로 참조한다.

2. 설치 및 개발환경

2.3 파이썬 설치 요소

Python(command line)

1. 도스창(cmd.exe) 를 실행 시키고 python를 실행한다.

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\kicks>python
Python 3.6.2 (v3.6.2:5fd33b5, Jul  8 2017, 04:14:34) [MSC v.1900 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print( "Hello World" )
Hello World
>>> 10 + 20
30
>>> 2**10
1024
>>>
```

2. 간단한 파이썬 명령들을 수행해 보자.

3. python를 실행 할 때 파이썬 스크립트 파일을 지정하지 않으면 대화식 인터프리터가 실행된다.

2. 설치 및 개발환경

2.3 파이썬 설치 요소

Module Docs

1. Python Module Docs 를 시행시키면 웹서버가 실행 된다.
2. localhost 와 방금 실행된 웹 서버의 서비스 포트로 접근 하면 다음과 같은 화면을 볼 수 있다.

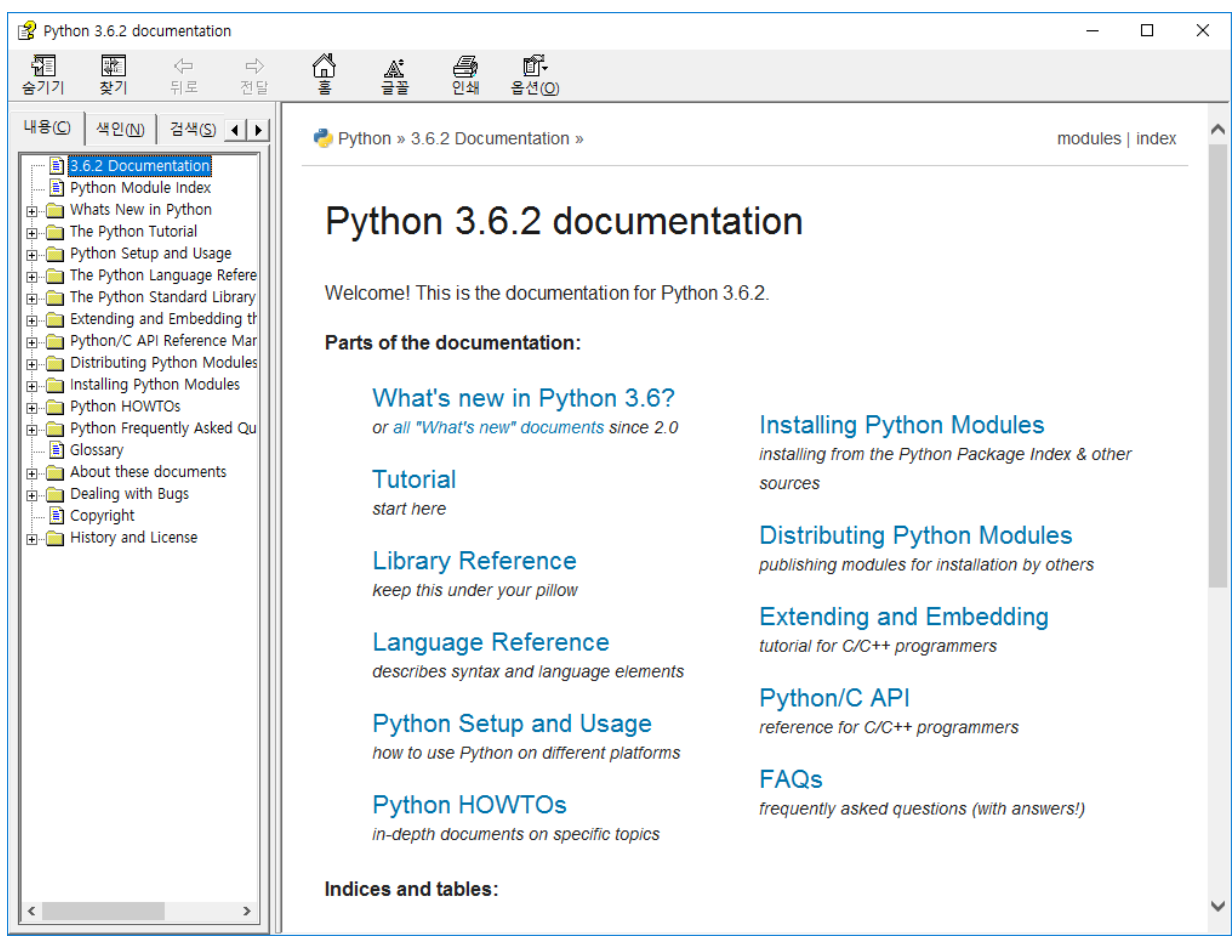


2. 설치 및 개발환경

2.3 파이썬 설치 요소

Python Manuals

1. 파이썬의 CHM 파일 형식의 도움말을 실행 시킨다.

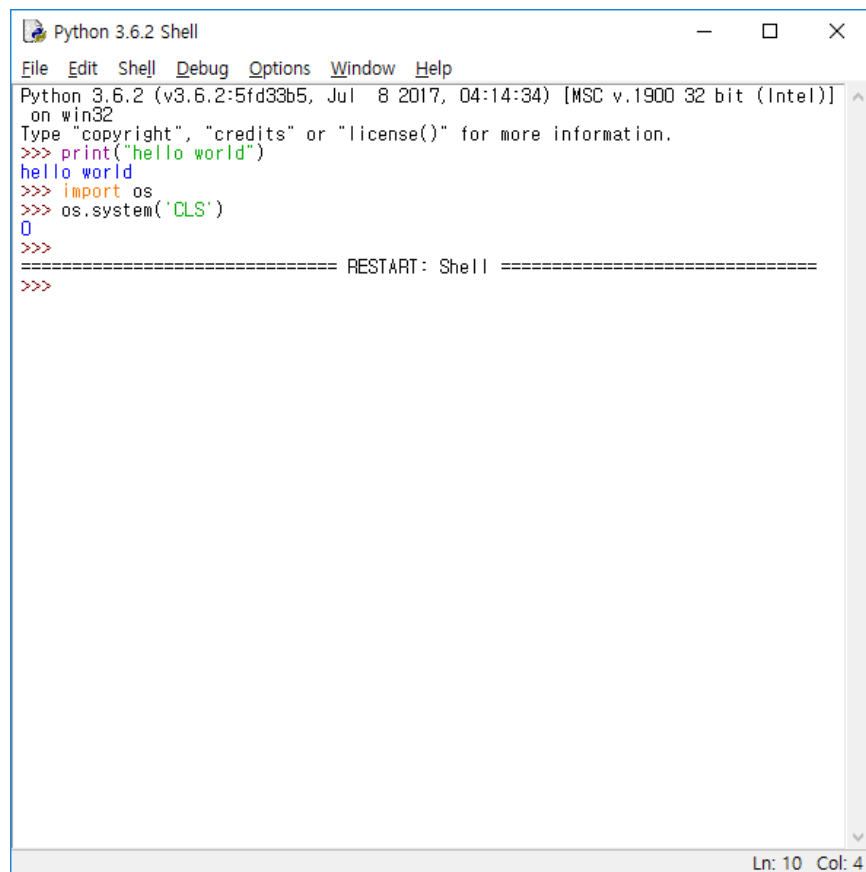


2. 설치 및 개발환경

2.3 파이썬 설치 요소

IDLE(Python GUI)

1. 도스창에서 실행시킨 커맨드라인 대화식 인터프리터와 비슷한 기능을 제공한다.
2. 추가적으로 자동완성 기능, 디버깅 기능과 파일 저장 등의 편리한 기능을 제공한다.
3. IDLE 사용을 권장



```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello world")
hello world
>>> import os
>>> os.system('CLS')
0
>>>
>>> ===== RESTART: Shell =====
>>>
```

Ln: 10 Col: 4

2. 설치 및 개발환경

2.3 파이썬 설치 요소

IDLE(Python GUI)

4. 간단한 명령은 IDLE을 이용하면 되지만, 보다 복잡한 코드는 파일로 저장해서 실행한다.

[실습 Hello World]

IDLE에서 다음코드를 작성하고 실행하고 에러가 없으면 hello.py 로 저장하고 Command Line에서 파일을 실행시켜 보자.

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

```
Microsoft Windows [Version 10.0.15063]  
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
D:\>python hello.py  
hello World
```

Chapter 01. 파이썬 시작하기

01. 소개

02. 설치 및 개발 환경

03. 프로그램 작성과 실행

3. 프로그램 작성과 실행

- 파이썬 프로그램의 작성 시, 가장 주의해야 할 것은 들여쓰기 이다 (파이썬 구문의 가장 큰 특징)
- 파이썬 프로그램은 대화식 모드와 스크립트 파일 실행 두가지 방식이 있다.

3.1 대화식 모드(Interactive Mode)에서 실행하기

[예제] 파이썬 버전 확인

```
>>> import sys
>>> sys.version
'3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]'
>>> sys.version_info
sys.version_info(major=3, minor=6, micro=2, releaselevel='final', serial=0)
>>>
```

3. 프로그램 작성과 실행

[예제] 파이썬 구문 작성 해보기

```
>>> print('Hello World')
```

```
Hello World
```

```
>>> sum = 10 + 20
```

```
>>> print( sum )
```

```
30
```

```
>>> 10 + 20
```

```
30
```

```
>>> “Hello World”
```

```
Hello World
```

3. 프로그램 작성과 실행

[예제] 산술 연산하기

```
>>> 4 + 5
9
>>> 12 - 32
-20
>>> 3/2
1.5
>>> 3//2
1
>>> 9%5
4
>>> -9%5
1
>>> divmod(-3, 2)
(-2, 1)
>>> 9.0 / 5.0
1.8
>>> 9 / 5.
1.8
>>> a = 2 + 3j
>>> b = 4 + 5j
>>> a + b
(6+8j)
>>> from cmath import *
>>> sqrt(-2)
1.4142135623730951j
```


3. 프로그램 작성과 실행

[예제] 주석 처리

```
>>> #주석은 #로 시작합니다.  
>>> #이 후의 문자는 해석되지 않습니다.  
>>>
```

[예제] 문자열 연산

```
>>> a = 'My Name is'  
>>> n = "Python"  
>>> a + n  
'My Name is Python'  
>>> a.upper()  
'MY NAME IS'  
>>> a.split()  
['My', 'Name', 'is']  
>>> ":".join( a.split() )  
'My:Name:is'  
>>>
```

마지막 결과 문자열을 다시 ['My', 'Name', 'is']로 쪼개어 보자.

3. 프로그램 작성과 실행

[예제] 파이썬 종료하기

```
>>> ^D
```

Ctrl + D 키를 눌러 종료 시킨다.

또는, 프로그램(스크립트)으로 종료 할 때는

```
import sys
sys.exit()
#또는 SystemExit 예외를 발생 시킨다.
raise SystemExit
```

3. 프로그램 작성과 실행

3.2 들여쓰기 규칙

- 들여쓰기(indentation)은 파이썬 구문의 가장 큰 특징이다.
- 가독성을 높인다.

1) 가장 바깥쪽에 있는 블록의 코드는 1열부터 시작해야 한다.

```
>>> a=1
>>> a=1
SyntaxError: unexpected indent
```

2) 내부 블록은 같은 거리만큼 들여 쓰기를 해야 한다.

```
>>> if( a > 1 ):
    print( "big" )
    print( "really?" )

SyntaxError: unexpected indent
>>>
```

3) 블록은 들여쓰기로 결정된다.

4) 탭과 공백은 함께 쓰는 것은 바람직 하지 못하다(비추천)

5) 들여쓰기 간격은 일정하기만 하면 된다(4 spaces 추천)

3. 프로그램 작성과 실행

[예제] 들어쓰기 연습

```
>>> def int2digit(n, base):  
    res = ""  
    while( n > 0 ):  
        n, r = divmod(n, base)  
        res = str(r) + res  
    return res  
  
>>> int2digit( 22, 2 )  
'10110'  
>>>
```

3. 프로그램 작성과 실행

3.3 프로그램(스크립트)로 실행하기

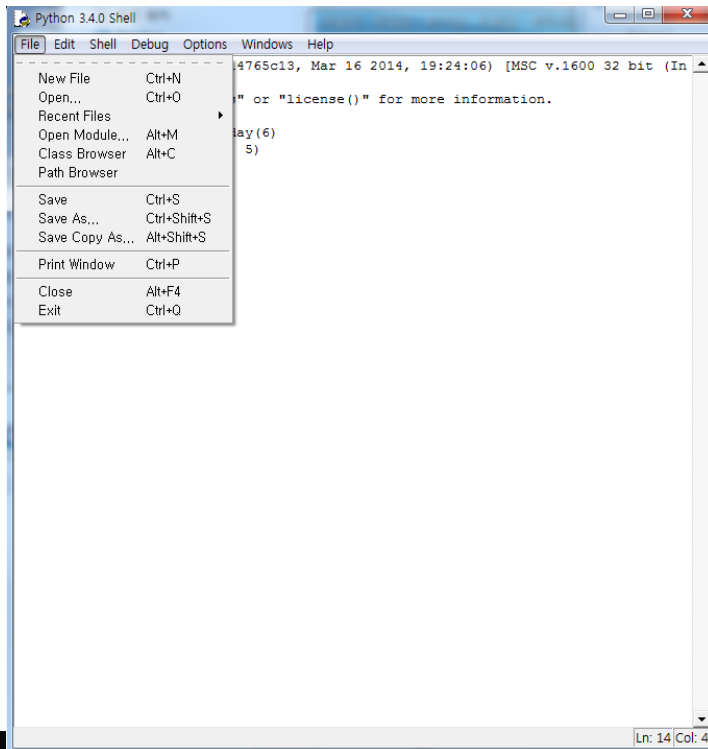
파일에 프로그램을 작성하고 한 번에 실행하는 방식으로 보통 파이썬으로 프로그램을 작성한 후 실행하는 방식이다.

IDLE에서 실행하기

[실습 cal.py]

이 번달 달력 화면을 표시하는 프로그램 작성하기

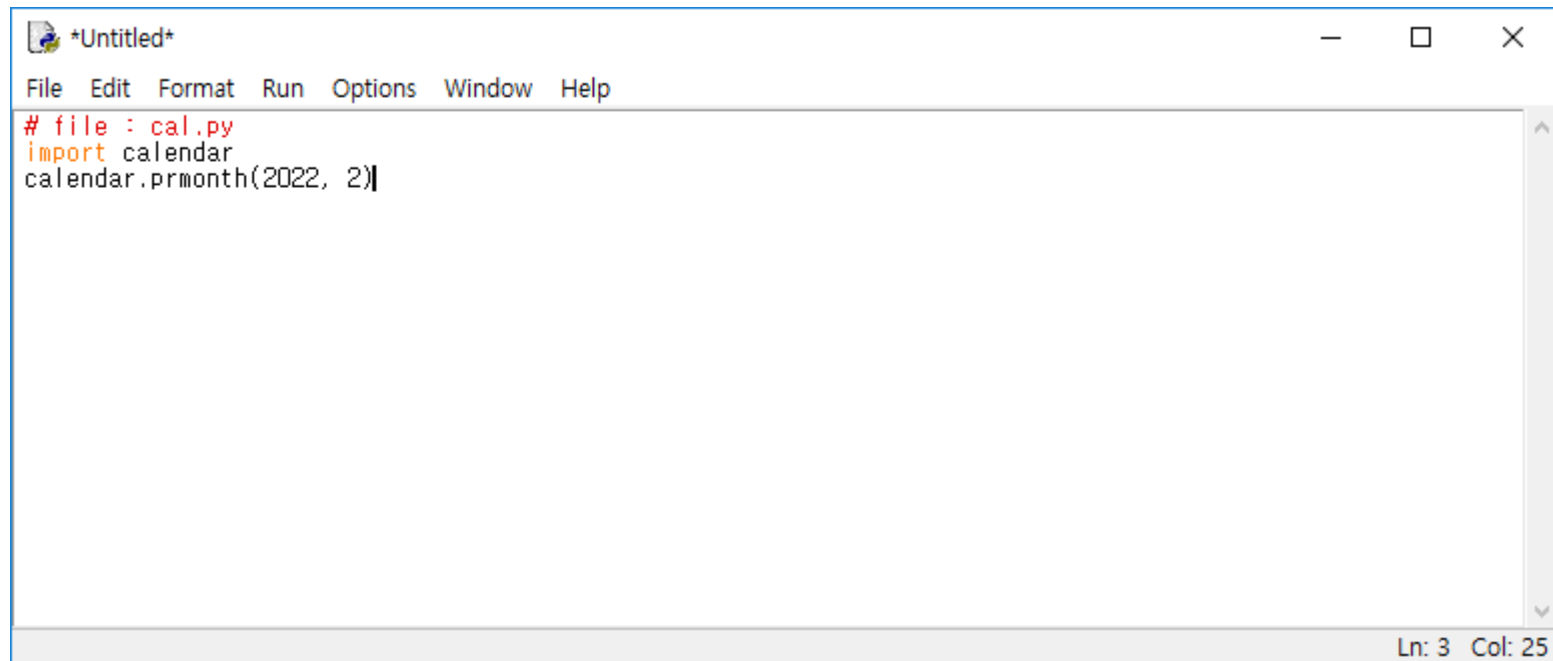
1. 에디터 실행하기 (File -> New File 선택, 또는 Ctrl + n)



3. 프로그램 작성과 실행

2. 코드 작성

3. cal.py 이름으로 저장 (File->Save, File->Save As, 또는 Ctrl + s)

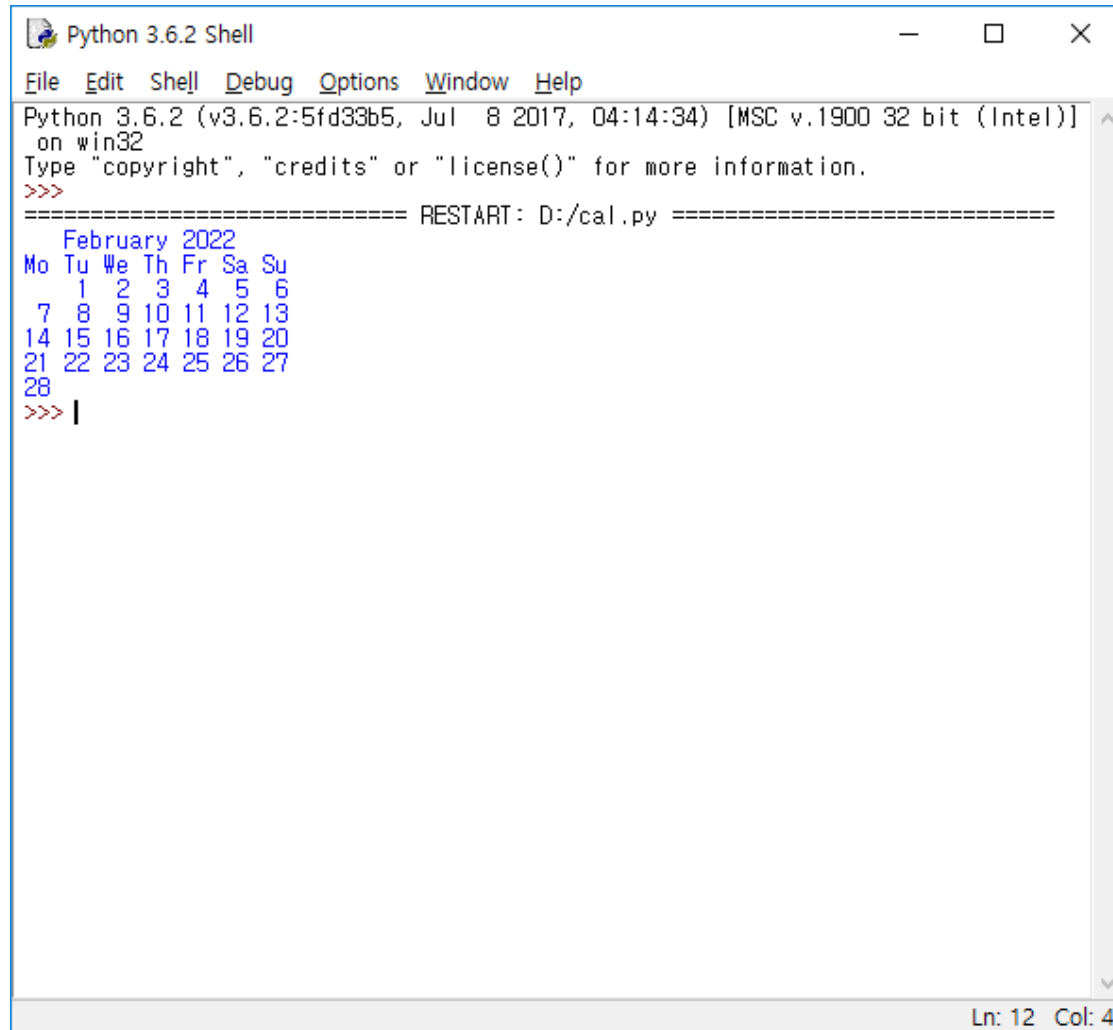


```
# file : cal.py
import calendar
calendar.prmonth(2022, 2)
```

Ln: 3 Col: 25

3. 프로그램 작성과 실행

4. 실행(Run -> Run Module, Ctrl + F5)



```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/cal.py =====
February 2022
Mo Tu We Th Fr Sa Su
  1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28
>>> |
```

Ln: 12 Col: 4