



# Data Analytics Based Python

## SECT2. 파이썬 설치 및 개발 환경

Innovation Growth Intensive Training  
Kim Jin Soo



- ◆ 파이썬 버전 선택하기
- ◆ 파이썬 설치하기
- ◆ 기본 실행 환경, 파이썬 커맨드라인, Command Line
- ◆ 파이썬 기본 통합 개발 환경, IDLE

cf. IDLE : Integrated DeveLopment Environment



## ❖ 파이썬2와 파이썬3이 공존

- 혼란스럽게도, 두 버전이 상호 연동이 안됨
- 주요 핵심 라이브러리들은 대부분 파이썬3으로 우선 변경
- 파이썬3은 파이썬2의 단점을 많이 제거
- 언젠가는 파이썬2의 업그레이드는 중단

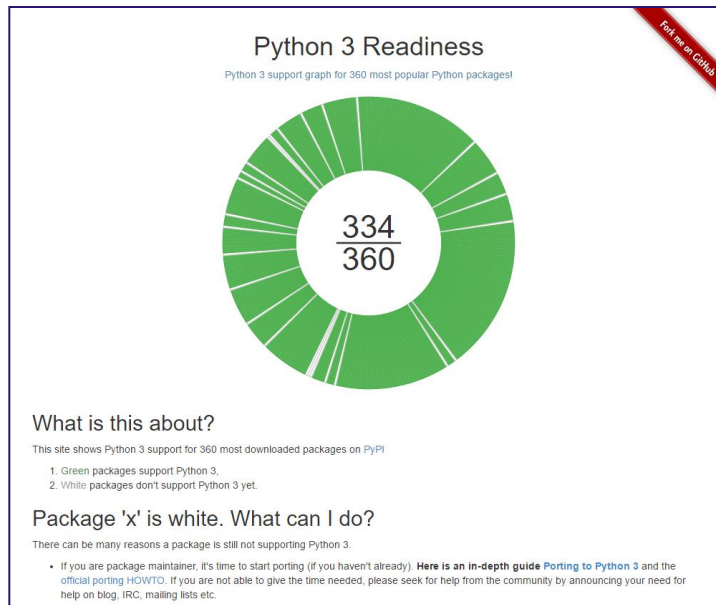


# 파이썬3 지원 유무 확인



## ❖ 파이썬 라이브러리 중 파이썬3 지원 유무 확인

- 참조 : <http://py3readiness.org/>
- 만약 파이썬3 버전을 사용하고 싶는데 라이브러리 의존성이 문제가 될 듯 하다면 일단 조사를 먼저 하는 것이 좋다.
- 모든 라이브러리의 업데이트 상황이 실시간으로 업데이트 되지 않음.
- 대개 파이썬 2.6이상, 파이썬 3.3 이상을 지원한다고 본다.



How do you identify Python 3 support?

This site utilizes a little tool, `canusepython3` created by Brett Cannon. Throw your requirements.txt file at it and it will tell you which packages support Python 3, and list out which don't.

Top 120	Top 120 - 240	Top 240 - 360
simplejson ✓	python-keystoneclient ✓	breadsticks ✓
setuputils ✓	traceback2 ✓	xmllint ✓
requests ✓	linecache2 ✓	reportlab ✓
six ✓	pandas ✓	hacking ✓
pip ✓	pycurl ✓	kazoo ✓
virtualenv ✓	python-memcached ✓	trollius ✓
boto ✓	isodate ✓	GitPython ✓
python-dateutil ✓	websocket-client ✓	hgtools ✓
pbr ✓	djangorestframework ✓	pluggy ✓
distribute ✓	python-daemon ✓	feedparser ✓
certifi ✓	suds ✓	python-magic ✓
pyasn1 ✓	python-novacient ✓	python-ldap X
pytz ✓	python-mimeparse ✓	oslo.log ✓
PyYAML ✓	netifaces ✓	Flask-SQLAlchemy ✓
wincertstore ✓	oslo.utils ✓	qds_sdk ✓
docutils ✓	oslo.i18n ✓	xmwt ✓
nose ✓	pyzmq ✓	unicodectsv ✓
Jinja2 ✓	ply ✓	termcolor ✓
lxml ✓	scikit-learn ✓	nosexcover ✓
MarkupSafe ✓	oslo.serialization ✓	iptools ✓

# 파이썬 설치하기



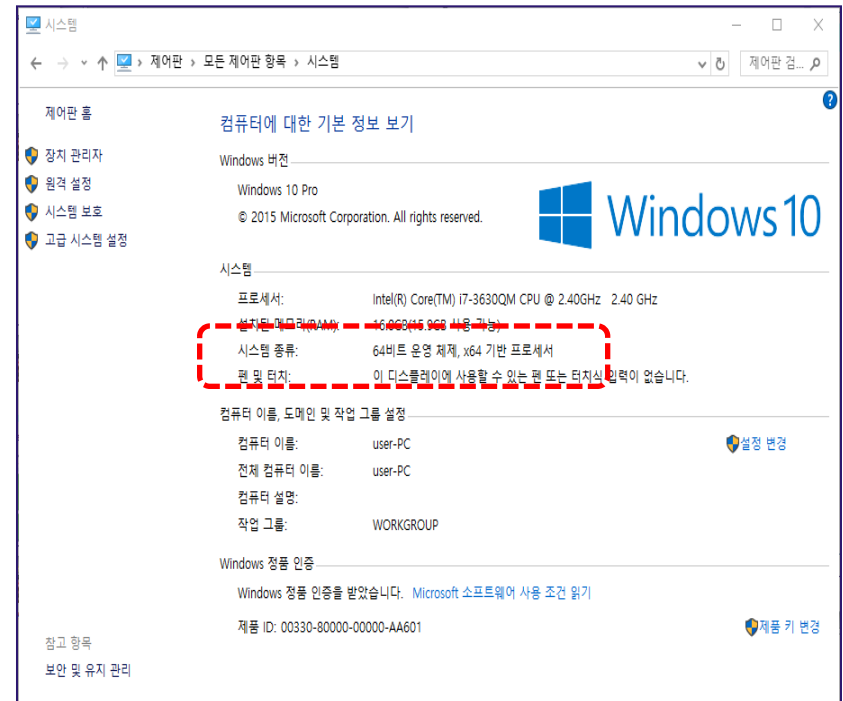
❖ 윈도우, 맥OS, 리눅스 운영체제에서 설치방법 조금씩 다르다.

## ❖ 윈도우 환경에서 설치

- 설치하기 전 시스템 종류 확인
- 32비트 운영체제/64비트 운영체제

## ❖ 32비트/64비트 운영체제와 프로그래밍의 관련성

- 전달할 수 있는 데이터의 양이 다름
  - 32비트 =  $2^{32}$  = 4,294,967,296 bit
  - 64비트 =  $2^{64}$  = 18,446,744,073,709,551,616 bit

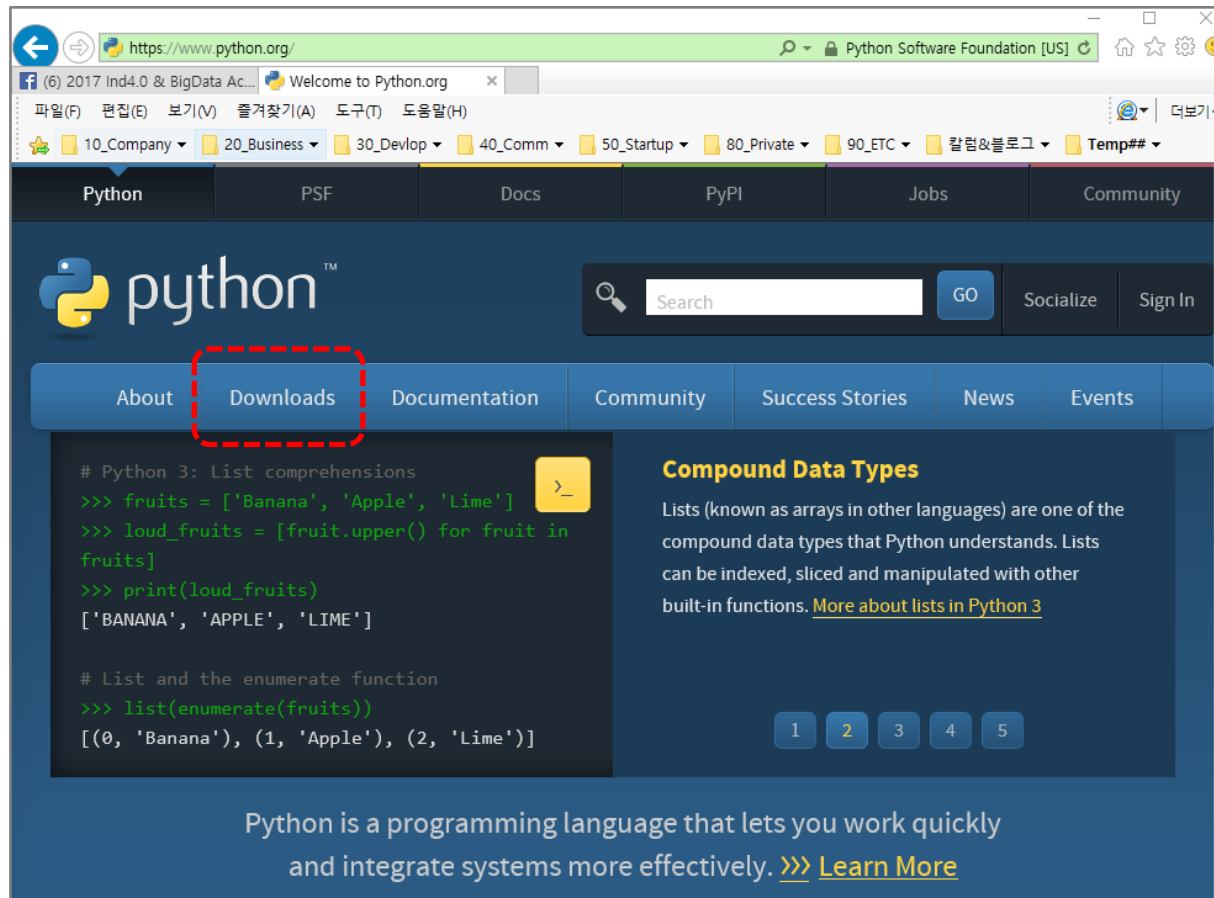


# 파이썬 설치파일 다운



## ❖ 다운로드 사이트

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





# 파이썬 설치파일 다운



## ❖ 최신의 안정적인 버전으로 릴리즈 된 파일 다운

- 릴리즈 버전의 a는 알파버전, b는 베타버전을 의미

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.6.1	2017-03-21	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.4.6	2017-01-17	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.5.3	2017-01-17	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.6.0	2016-12-23	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 2.7.13	2016-12-17	<a href="#">Download</a> <a href="#">Release Notes</a>

Python Releases for Windows

- Latest Python 2 Release - Python 2.7.13
- Latest Python 3 Release - Python 3.6.1
- Python 3.6.2rc1 - 2017-06-17
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 embeddable zip file
  - Download Windows help file
- Python 3.6.1 - 2017-03-21
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 embeddable zip file
  - Download Windows help file
- Python 3.6.1rc1 - 2017-03-05
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 embeddable zip file



## ❖ 설치파일 (python-3.6.1.amd64.exe) 실행

- Install for all users 선택 후 [Next] 클릭
- 드라이브명 변경해서 설치 : C:\Python\Python36 위치 선정 후 [Next] 클릭
- 파이썬 설치 옵션을 설정
  - 마지막 리스트에 있는 'Add python.exe to Path' 부분의 'X' 박스 클릭
  - 파이썬 실행파일을 운영체제 환경변수에 추가
  - 수동으로 잡아 줄 수 있지만 실수할 수도 있으니 이 옵션을 선택
  - 'Will be installed on local hard drive'를 선택 후 [Next] 클릭
- 설치가 잘 되고 있는지 프로그래스바를 통해 확인
- 중간에 명령어 프롬프트가 잠시 떴다가 사라진다.  
설치 과정의 일부이니 신경 쓰지 않아도 된다.
- 정상적으로 마무리 되면, [Finish] 버튼을 클릭
- 설치 과정 종료~^^

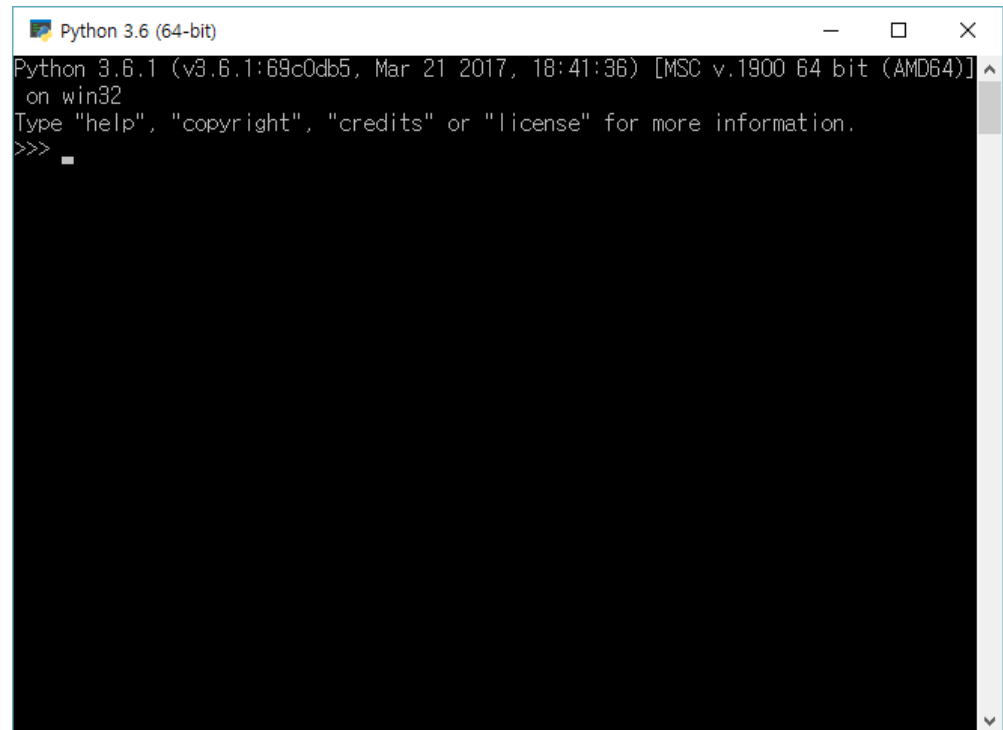
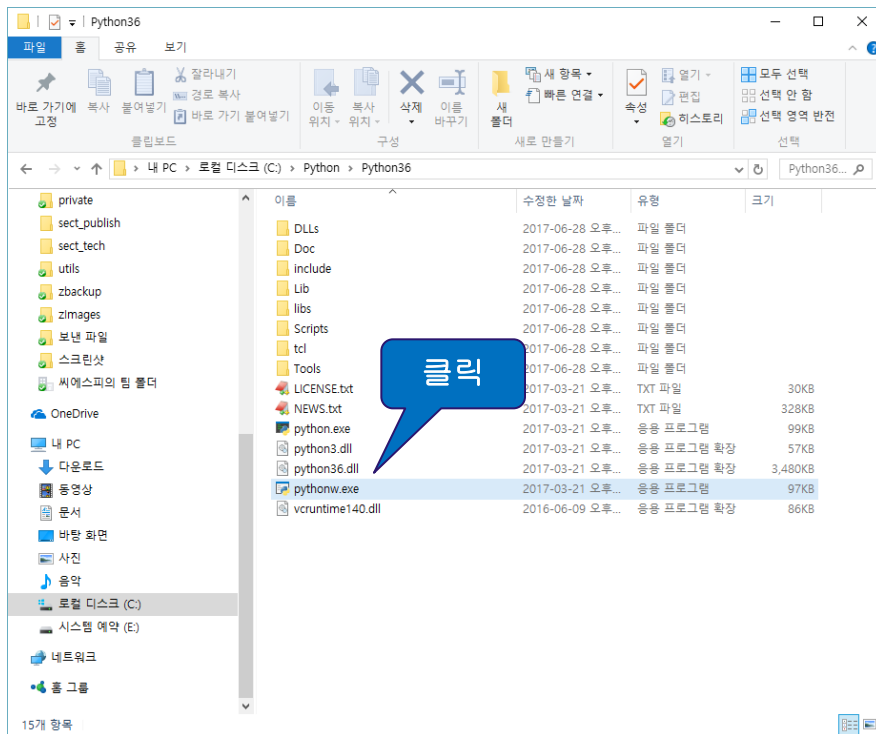


# 탐색기에서 파이썬 실행



## ❖ 탐색기를 통해 설치된 경로로 이동

- 설치폴더인 C:\Python\Python36 디렉토리에서 'python.exe' 를 클릭



# 커맨드창에서 파이썬 실행



## ❖ 명령어 프롬프트를 열어서 확인

- [윈도우]-[실행] 창을 띄워서 'cmd' 입력 후 <Enter>키를 누른다.
- [실행] 창은 <윈도우>+<R>키를 눌러 열 수도 있다.
- 설치 과정 중에 설치 파일의 위치를 환경변수에 넣었으므로 바로 실행

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\User>echo %path%
C:\Java\jdk1.8.0_111\bin;C:\Program Files (x86)\Wizvera\Delfino;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\nodejs\;C:\Program Files (x86)\IDM Computer Solutions\UltraCompare\;C:\Git\cmd;C:\Git\mingw64\bin;C:\Git\usr\bin;C:\Program Files (x86)\Windows Kits\8.1\Windows Performance Toolkit\;C:\Python\Python36;C:\Python\Python36\Scripts;C:\Python\Anaconda3-4x;C:\Python\Anaconda3-4x\Library\mingw-w64\bin;C:\Python\Anaconda3-4x\Library\usr\bin;C:\Python\Anaconda3-4x\Library\bin;C:\Python\Anaconda3-4x\Scripts
```



## ❖ 명령어 프롬프트를 열어서 확인

- [윈도우]-[실행] 창을 띄워서 'cmd' 입력 후 <Enter>키를 누른다.
- [실행] 창은 <윈도우>+<R>키를 눌러 열 수도 있다.
- 설치 과정 중에 설치 파일의 위치를 환경변수에 넣었으므로 바로 실행

# 커맨드창에서 파이썬 실행



## ❖ 명령어 프롬프트를 열어서 확인

- [윈도우]-[실행] 창을 띄워서 'cmd' 입력 후 <Enter>키를 누른다.
- [실행] 창은 <윈도우>+<R>키를 눌러 열 수도 있다.
- 설치 과정 중에 설치 파일의 위치를 환경변수에 넣었으므로 바로 실행

```
C:\WINDOWS\system32\cmd.exe - python

C:\Users\User>python
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 20:20:57) [MSC v.1600 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> print("Hello, Python World!")
Hello, Python World!
>>>
```

# 커맨드창에서 파이썬 실행



## ❖ 명령어 프롬프트를 열어서 확인

- [윈도우]-[실행] 창을 띄워서 'cmd' 입력 후 <Enter>키를 누른다.
- [실행] 창은 <윈도우>+<R>키를 눌러 열 수도 있다.
- 설치 과정 중에 설치 파일의 위치를 환경변수에 넣었으므로 바로 실행

```
C:\WINDOWS\system32\cmd.exe

C:\Python\Python36>python
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> print('Hello, Python World!!')
Hello, Python World!!
>>>
>>> exit()

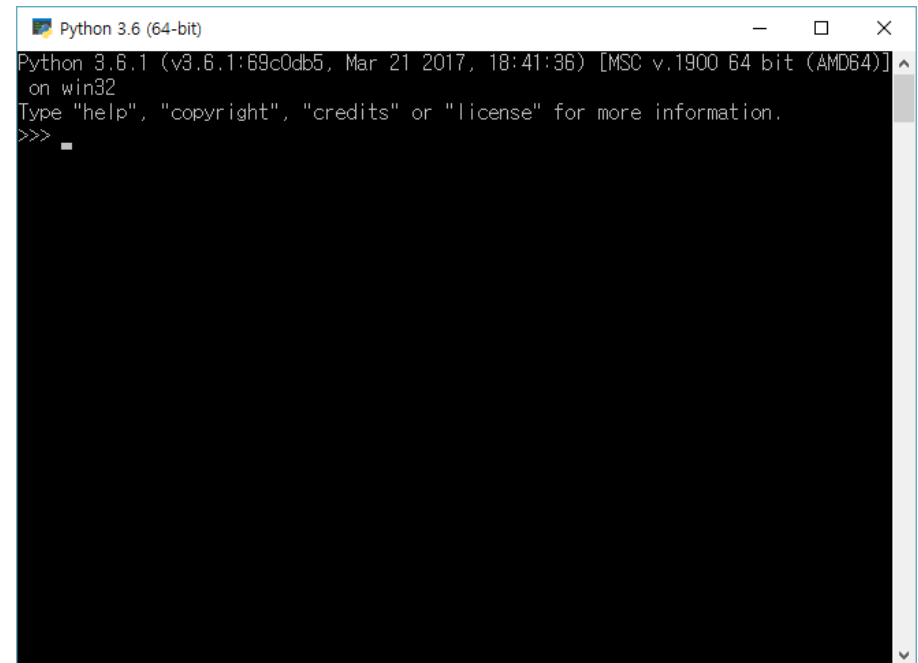
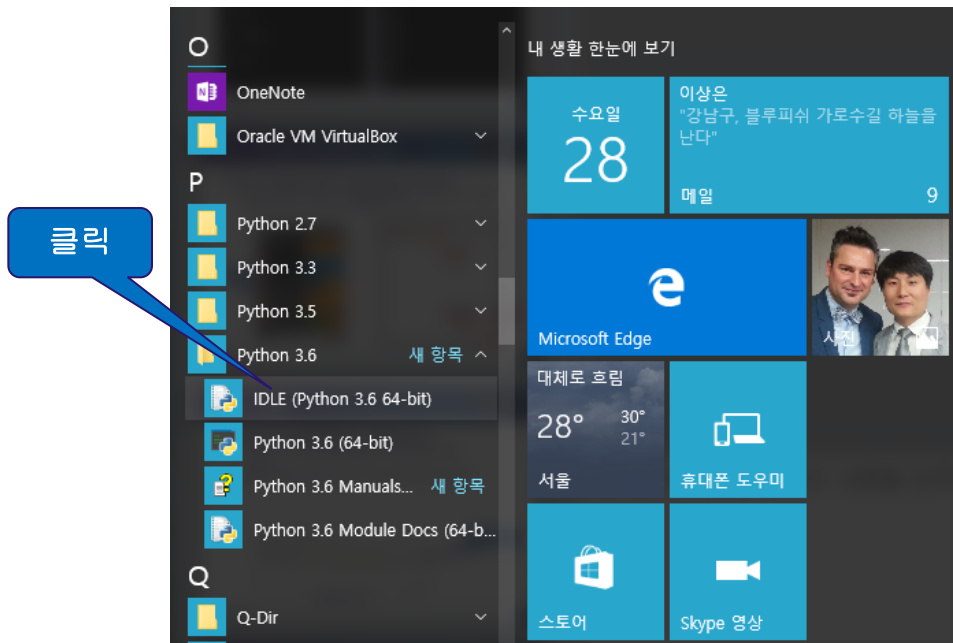
C:\Python\Python36>
```

# 기본 실행 환경, 파이썬 커맨드라인



## ❖ 파이썬 IDLE

- 커맨드라인을 기반으로 한 파이썬의 기본 실행 환경
- [윈도우] - [모든 프로그램] - [Python 3.6] - [Python 3.6 (64-bit)] 를 클릭해서 실행 할 수도 있다.





# 파이썬 커맨드라인에서 명령어 실행



```
>>> print("Hello, Daniel !!")
```

```
>>> 4+5
```

```
>>> 5-8
```

```
>>> 8*9
```

```
>>> 100/20
```

```
>>> 5**2
```

```
>>> for x in range(10):
```

```
...     print(x, x**2)
```

```
...
```

```
0 0
```

```
1 1
```

```
2 4
```

```
3 9
```

```
4 16
```

```
5 25
```

```
6 36
```

```
7 49
```

```
8 64
```

```
9 81
```

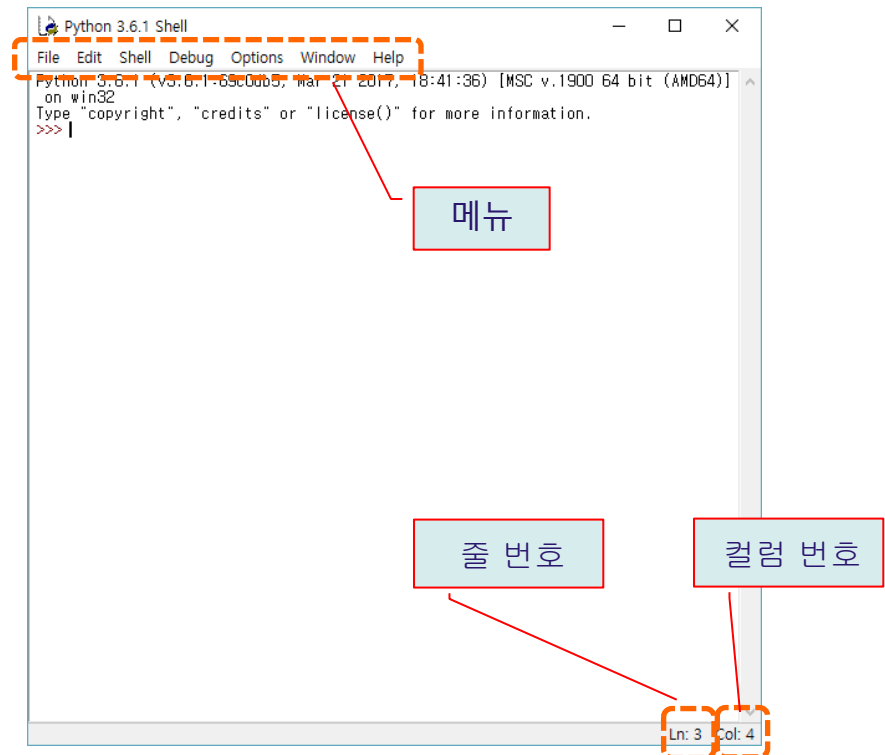
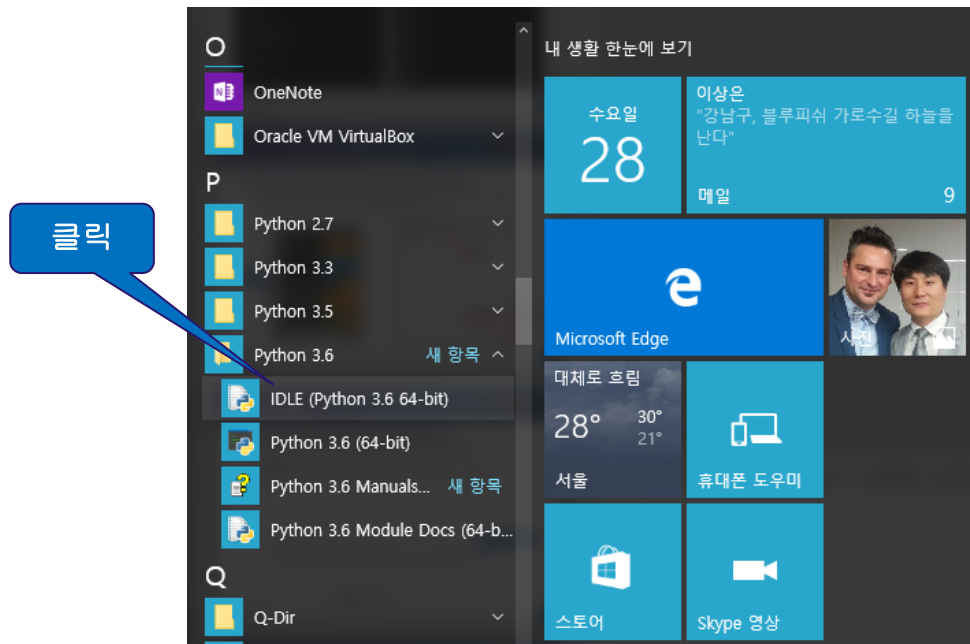
```
>>> _
```

# 파이썬 기본 통합 개발 환경, IDLE



## ❖ IDLE, Integrated DeveLopment Environment

- 파이썬 IDLE는 [윈도우] – [모든 프로그램] – [Python 3.6 – [IDLE (Python 3.6 64-bit)] 를 클릭해서 실행





## ❖ IDLE에서 표준 출력문을 하나 출력 해보자.

- 'print'까지 입력하고 좌측 종괄호 ( 까지만 입력해 보자.  
**print (**
- 출력문 도움말 팝업 텍스트가 에디터 창 위로 튀어 나온다.

```
*Python 3.6.1 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(
    print(value, ..., sep=' ', end='\\n', file=sys.stdout, flush=False)
```

출력문 도움말

<표준 출력문 입력하기>



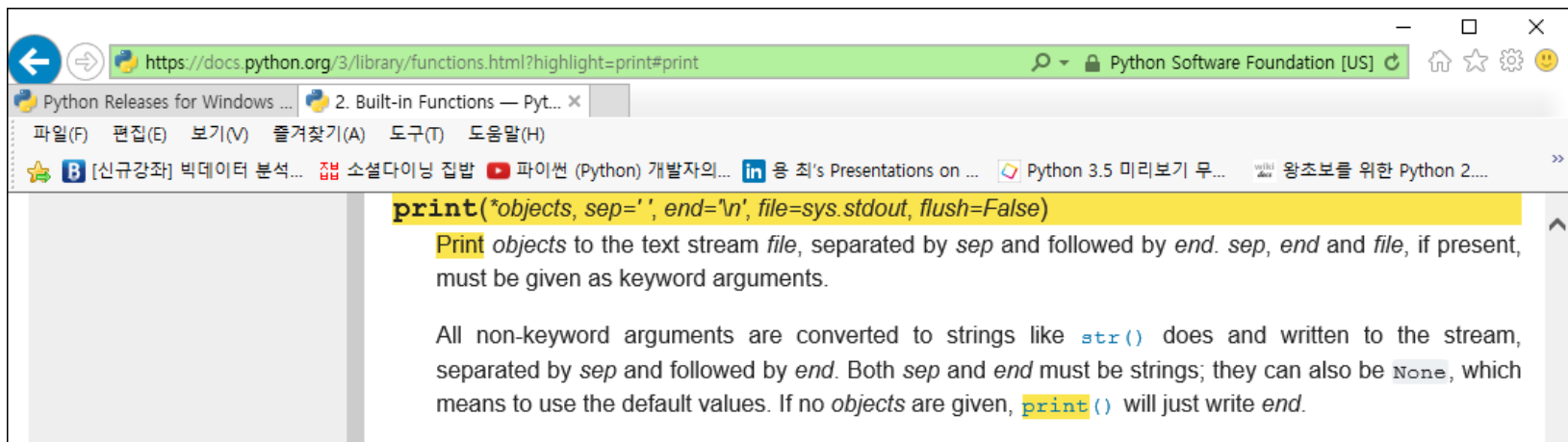
## ❖ IDLE에서 표준 출력문을 하나 출력 해보자.

- 'print'까지 입력하고 좌측 중괄호 ( 까지만 입력해 보자.

**print (**

- 출력문 도움말 팝업 텍스트가 에디터 창 위로 튀어 나온다.
- 이 문법은 파이썬 공식 문서에 기재되어 있는 문법을 호출하여 개발할 때 참고할 수 있도록 돕고 있다.

<https://docs.python.org/3/library/functions.html?highlight=print#print>

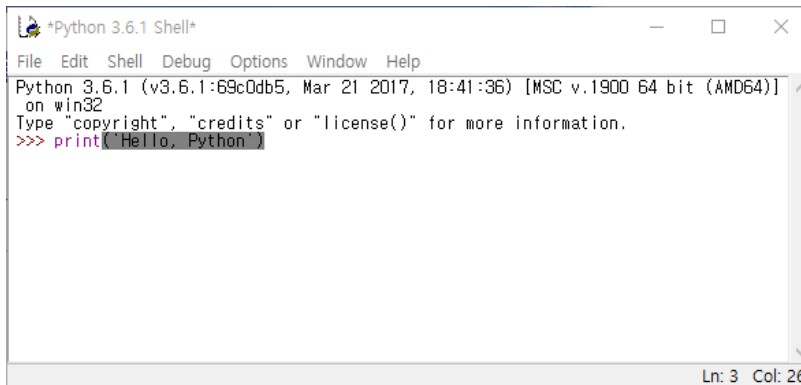


<'print' 함수 상세 설명 링크>



## ❖ IDLE에서 표준 출력문을 하나 출력 해보자.

- 원하는 출력문을 끝까지 입력하고 우측 종괄호까지 입력 후 <Enter>  
`print ('Hello, Python')`

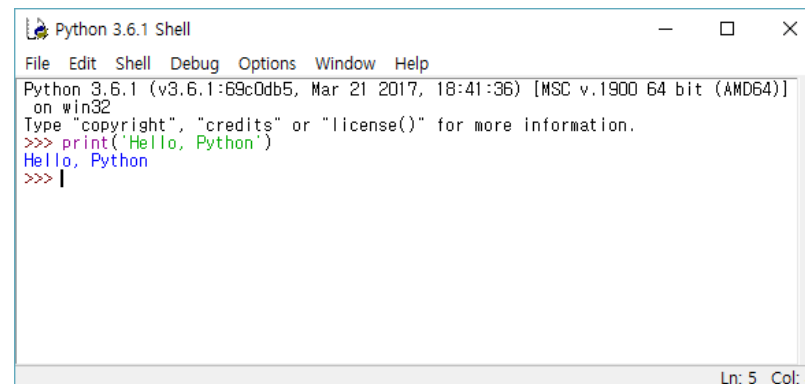


```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, Python')
```

회색음영이 표기되는 것을 확인할 수 있다.  
→ 이는 시작한 종괄호가 잘 닫혔다는 의미다.

### <출력문 입력 후 대기하기>

파이썬 커맨드라인 실행에 비해 달라진 것이 눈에 보인다.  
→ 바로 '글자 색'이다.  
이는 가독성 향상에 도움이 된다.



```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, Python')
Hello, Python
>>> |
```

### <출력문 실행하기>

# Undo 와 Redo 기능



## ❖ Undo : 수정한 명령어에 대한 실행 취소 기능

- <Ctrl>+<Z> 키

```
*Python 3.6.1 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, Python')
Hello, Python
>>> print('Hello, Python')
Ln: 6 Col: 26
```

출력문 변경 후 취소(Undo)하기

<Ctrl>+<Z> 키를 8번 눌러보면  
'Python'를 입력한 것을 실행취소가  
된다.

새로이 'Bigdata'를 입력해보자.

## ❖ Redo : 취소한 명령어를 다시 수행하는 기능

- <Ctrl>+<Shift>+<Z> 키

```
*Python 3.6.1 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, Python')
Hello, Python
>>> print('Hello, BigData')
Ln: 6 Col: 27
```

출력문 다시 실행(Redo)하기

새로이 'Bigdata'를 입력한 것을  
실행취소를 한 후에,  
<Ctrl>+<Shift>+<Z> 키를 9번  
눌러보면 다시 'Bigdata'로 돌아간다.

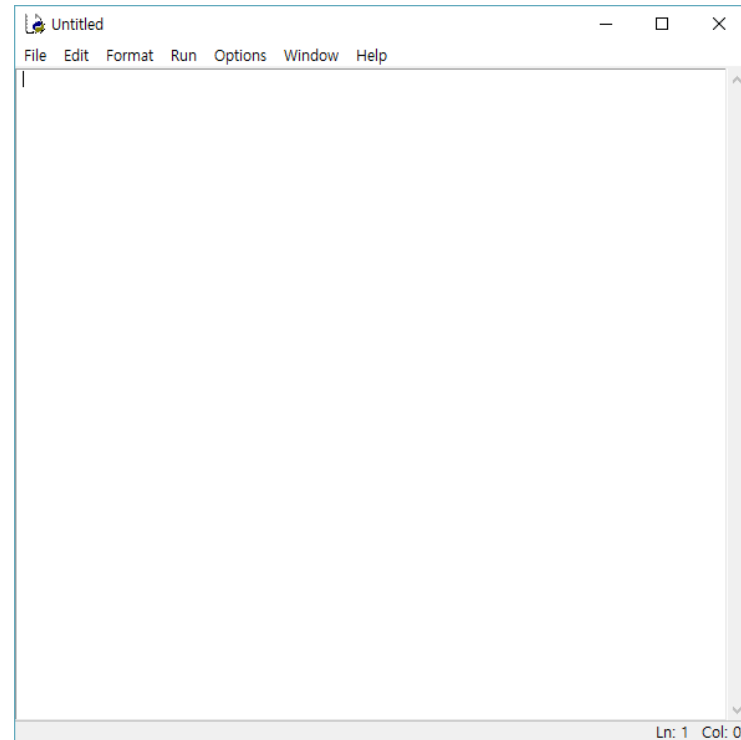
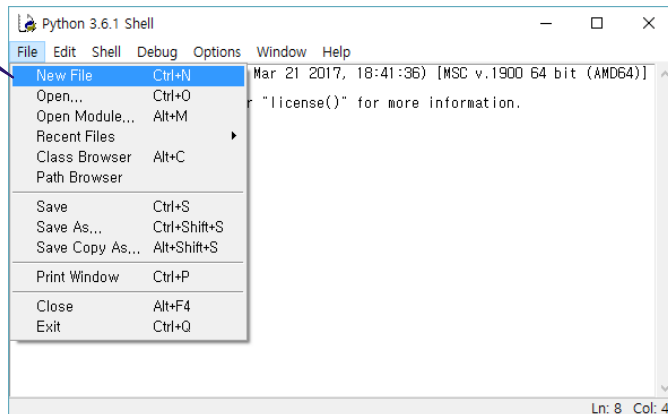




## ❖ 소스 코드를 수집하는 창과 실행하는 창을 분리

- IDLE 실행 화면에서 <Ctrl>+<N>키를 누르거나, 메뉴 중 [File] - [New File]를 클릭
- 메모장과 같은 깨끗한 화면의 창이 뜨는 것을 확인

클릭



<IDLE 새 창 초기 화면>



- ❖ IDLE 소스 수정 창에서 코드 입력 및 수정 작업
- ❖ 실행하려면 소스를 저장 후, 실행하기
  - 메뉴에서 [Run] – [Run Module]를 클릭하거나 <F5>키를 누른다.

```
*exam.py - C:/Python/Python36/exam.py (3.6.1)*
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

for num in range(10):
    print(num, num+2)
```

소스 수정 창

Ln: 6 Col: 0

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello, Python')
Hello, Python
>>>
>>> print('Hello, BigData')
Hello, BigData
>>>
===== RESTART: C:/Python/Python36/exam.py =====
0 0
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
>>>
```

소스 실행 결과 확인  
IDLE 쉘 창

Ln: 20 Col: 4

<IDLE 소스 수정 창(좌측)과 소스 실행 창(우측)을 함께 보기>



## ❖ 파이썬 설치 버전을 선택한 뒤 다양한 운영체제(윈도우, 맥OS, 리눅스) 에서 파이썬 설치

- 맥OS나 리눅스에는 파이썬2가 기본적으로 설치되어 있다.
- 특히, 리눅스에서는 파이썬3까지 설치되어 있다.

## ❖ 파이썬 개발환경

- 커맨드 창 : 명령어 프롬프트를 열어서 확인
- 파이썬 쉘 : 파이썬 커맨드라인기반 실행,
- 파이썬 IDLE : 파이썬 기본 통합 개발 환경

## ❖ 파이썬 버전 명명 규칙

- 현재 설치된 파이썬 버전은 3.6.1 버전이다. (A.B.C 형태)
  - A : Major 버전 번호, 중대한 변화가 생겼을 때에만 증가
  - B : Minor 버전 번호, 세상을 떠들썩하게 할 정도가 아닌 변화인 경우 증가
  - C : Micro 버전 번호, 버그가 수정되어 릴리즈 되는 경우에 해당
- 알파, 베타, 릴리즈 후보 버전을 표기하기 위한 접미사 (Suffix) 사용
  - 'aN'은 알파 버전, 'bN'은 베타 버전, 'rcN'은 릴리즈 후보인 경우