

## ○ IPython이란?

파이썬 공부 또는 코딩을 할 때 간단한 코드 테스트를 하기 위해서 Python Shell을 자주 사용하게 됩니다. Python Shell은 파이썬 인터프리터를 인터랙티브 모드로 사용할 수 있도록 해주는 어플리케이션입니다. 윈도우(Windows)의 커맨드(CMD, 명령프롬프트)나 맥(Mac OS), 리눅스(Rinux)의 터미널(Terminal)을 사용하는 것과 비슷한 개념입니다. 자바스크립트(JavaScript)로 개발을 해 본 분들이라면 자바스크립트의 콘솔(Console)과 같다고 보시면 됩니다. IPython은 기존의 Python Shell이 가진 편리하면서도 아주 강력한 기능들이 추가된 새로운 Shell이며, 또한 파이썬 유저들이 많이 사용하고 있는 쥬피터 노트북(Jupyter Notebook)의 커널로도 사용되고 있습니다.

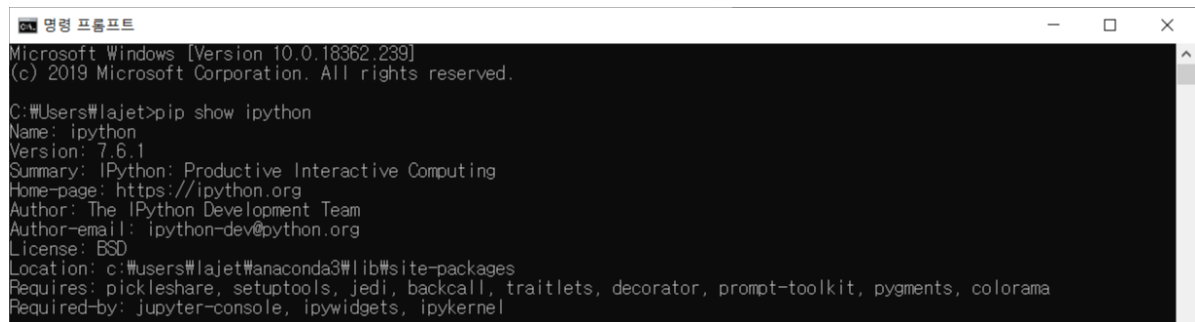
현재 아나콘다 파이썬을 설치하고 사용중이라면 파이썬 환경에 IPython 패키지가 이미 설치되어 있을 것입니다. 만약, 아나콘다 파이썬이 아닌 일반 파이썬을 설치했을 경우에는 PIP Package Manager를 통해서 간단히 설치를 할 수 있습니다.

## ○ IPython 설치 여부 확인 및 실행 방법

먼저 명령프롬프트를 실행하도록 합니다. IPython의 설치여부를 확인하려면 다음과 같은 명령어를 입력합니다.

```
pip show ipython
```

명령어를 입력한 후 엔터 버튼을 누르고, 현재 자신의 컴퓨터 또는 노트북에 IPython이 설치되어 있다면 아래와 같이 설치된 버전을 볼 수 있습니다.



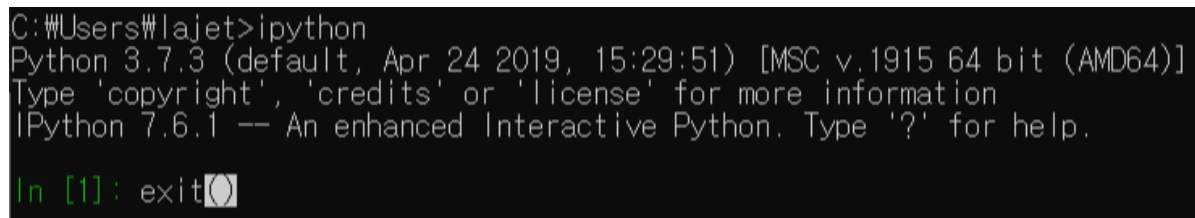
```
명령 프롬프트
Microsoft Windows [Version 10.0.18362.239]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lajet>pip show ipython
Name: ipython
Version: 7.6.1
Summary: IPython: Productive Interactive Computing
Home-page: https://ipython.org
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD
Location: c:\Users\lajet\anaconda3\lib\site-packages
Requires: pickleshare, setuptools, jedi, backcall, traitlets, decorator, prompt-toolkit, pygments, colorama
Required-by: jupyter-console, ipywidgets, ipykernel
```

IPython 버전 확인

다음과 같은 명령어를 입력하여 직접 IPython을 실행하고, IPython shell이 실행되는지 확인해도 됩니다.

```
ipython
```



```
C:\Users\lajet>ipython
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.6.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: exit()
```

IPython 실행 및 종료

IPython을 종료하고자 할 때에는 위와 같이 `exit()`를 입력합니다. 현재 저의 컴퓨터에는 IPython이 설치되어 있으므로, 설치 방법에 대한 자료를 위해 가상 환경을 만들도록 하겠습니다. 가상 환경을 만드는 방법을 알고 싶으신 분은 아래의 글을 펼쳐주세요.

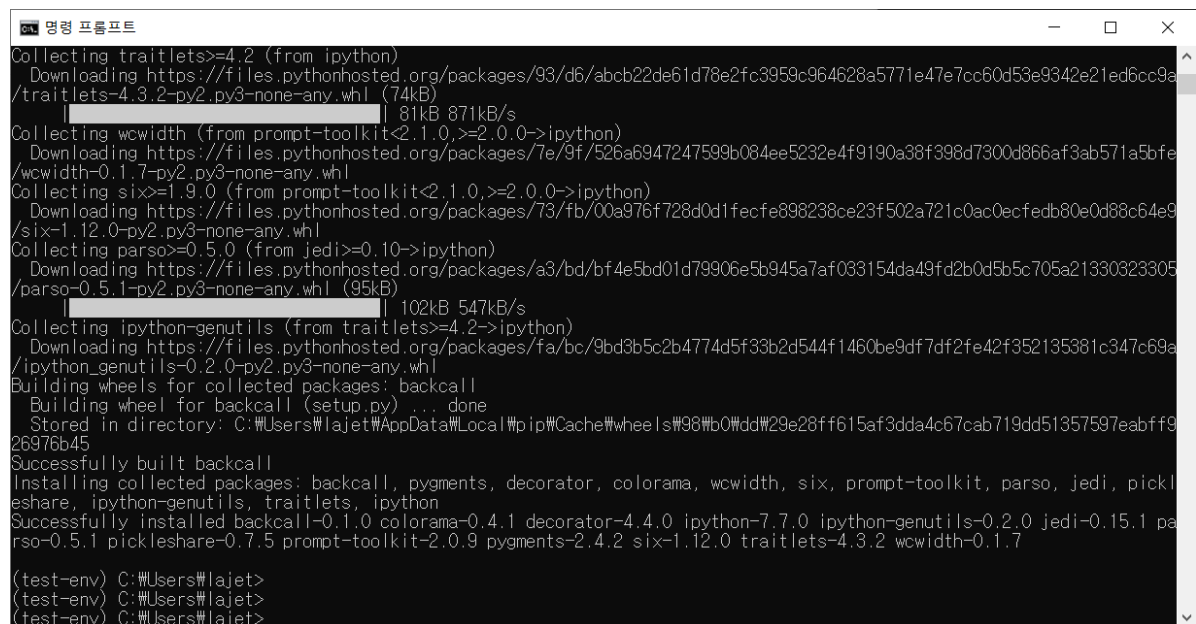
...더보기

## ○ IPython 설치 방법

다음과 같은 명령어를 입력하여 IPython 설치를 실행해주도록 합니다.

```
pip install ipython
```

아래와 같이 IPython이 설치된 것을 볼 수 있습니다.



```
명령 프롬프트
Collecting traitlets>=4.2 (from ipython)
  Downloading https://files.pythonhosted.org/packages/93/d6/abcb22de61d78e2fc3959c964628a5771e47e7cc60d53e9342e21ed6cc9a
/traits-4.3.2-py2.py3-none-any.whl (74kB)
    | 81kB 871kB/s
Collecting wcwidth (from prompt-toolkit<2.1.0,>=2.0.0->ipython)
  Downloading https://files.pythonhosted.org/packages/7e/9f/526a6947247599b084ee5232e4f9190a38f398d7300d866af3ab571a5bfe
/wcwidth-0.1.7-py2.py3-none-any.whl
Collecting six>=1.9.0 (from prompt-toolkit<2.1.0,>=2.0.0->ipython)
  Downloading https://files.pythonhosted.org/packages/73/fb/00a976f728d0d1fecfe898238ce23f502a721c0ac0ecfedb80e0d88c64e9
/six-1.12.0-py2.py3-none-any.whl
Collecting parso>=0.5.0 (from jedi>=0.10->ipython)
  Downloading https://files.pythonhosted.org/packages/a3/bd/bf4e5bd01d79906e5b945a7af033154da49fd2b0d5b5c705a21330323305
/parso-0.5.1-py2.py3-none-any.whl (95kB)
    | 102kB 547kB/s
Collecting ipython-genutils (from traitlets>=4.2->ipython)
  Downloading https://files.pythonhosted.org/packages/fa/bc/9bd3b5c2b4774d5f33b2d544f1460be9df7df2fe42f352135381c347c69a
/ipython_genutils-0.2.0-py2.py3-none-any.whl
Building wheels for collected packages: backcall
  Building wheel for backcall (setup.py) ... done
  Stored in directory: C:\Users\lajet\AppData\Local\Pip\Cache\wheels\98\bb\0\dd\29e28ff615af3dda4c67cab719dd51357597eabff9
26976b45
Successfully built backcall
Installing collected packages: backcall, pygments, decorator, colorama, wcwidth, six, prompt-toolkit, parso, jedi, pickl
eshare, ipython-genutils, traitlets, ipython
Successfully installed backcall-0.1.0 colorama-0.4.1 decorator-4.4.0 ipython-7.7.0 ipython-genutils-0.2.0 jedi-0.15.1 pa
rso-0.5.1 pickleshare-0.7.5 prompt-toolkit-2.0.9 pygments-2.4.2 six-1.12.0 traitlets-4.3.2 wcwidth-0.1.7

(test-env) C:\Users\lajet>
(test-env) C:\Users\lajet>
(test-env) C:\Users\lajet>
```

IPython 설치 - 1

이전에 했던 과정과 동일하게 IPython의 설치 여부와 설치 버전을 확인해보니 정상적으로 설치된 것을 알 수 있습니다.

```
명령 프롬프트
/parso-0.5.1-py2.py3-none-any.whl (95kB)
| 102kB 547kB/s
Collecting ipython-genutils (from traitlets>=4.2->ipython)
  Downloading https://files.pythonhosted.org/packages/fa/bc/9bd3b5c2b4774d5f33b2d544f1460be9df7df2fe42f352135381c347c69a
/ipython_genutils-0.2.0-py2.py3-none-any.whl
Building wheels for collected packages: backcall
  Building wheel for backcall (setup.py) ... done
  Stored in directory: C:\Users\lajet\AppData\Local\Pip\Cache\wheels\98\bb\0\dd\29e28ff615af3dda4c67cab719dd51357597eabff9
26976b45
Successfully built backcall
Installing collected packages: backcall, pygments, decorator, colorama, wcwidth, six, prompt-toolkit, parso, jedi, pickl
eshare, ipython-genutils, traitlets, ipython
Successfully installed backcall-0.1.0 colorama-0.4.1 decorator-4.4.0 ipython-7.7.0 ipython-genutils-0.2.0 jedi-0.15.1 pa
rso-0.5.1 pickleshare-0.7.5 prompt-toolkit-2.0.9 pygments-2.4.2 six-1.12.0 traitlets-4.3.2 wcwidth-0.1.7

(test-env) C:\Users\lajet>
(test-env) C:\Users\lajet>
(test-env) C:\Users\lajet>pip show ipython
Name: ipython
Version: 7.7.0
Summary: IPython: Productive Interactive Computing
Home-page: https://ipython.org
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD
Location: c:\Users\lajet\anaconda3\envs\test-env\lib\site-packages
Requires: setuptools, pygments, jedi, backcall, decorator, pickleshare, traitlets, colorama, prompt-toolkit
Required-by:

(test-env) C:\Users\lajet>
```

## IPython 설치 - 2

다음과 같은 명령어를 입력하여 IPython과 Ipython 패키지를 사용하기 위해서 필요한 패키지의 설치 여  
부를 확인해보록 하겠습니다.

```
pip freeze
```

아래와 같이 정상적으로 설치된 것을 볼 수 있습니다.

```
명령 프롬프트
(test-env) C:\Users\lajet>pip show ipython
Name: ipython
Version: 7.7.0
Summary: IPython: Productive Interactive Computing
Home-page: https://ipython.org
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD
Location: c:\Users\lajet\anaconda3\envs\test-env\lib\site-packages
Requires: setuptools, pygments, jedi, backcall, decorator, pickleshare, traitlets, colorama, prompt-toolkit
Required-by:

(test-env) C:\Users\lajet>pip freeze
backcall==0.1.0
certifi==2019.6.16
colorama==0.4.1
decorator==4.4.0
ipython==7.7.0
ipython-genutils==0.2.0
jedi==0.15.1
parso==0.5.1
pickleshare==0.7.5
prompt-toolkit==2.0.9
Pygments==2.4.2
six==1.12.0
traitlets==4.3.2
wcwidth==0.1.7
wincertstore==0.2

(test-env) C:\Users\lajet>
```

## IPython 설치 - 3

개인적으로 패키지 관리가 편리하고 손쉽게 가상환경 관리를 할 수 있는 아나콘다(Anaconda) 파이썬을  
설치하는 것을 권장합니다. 아나콘다 파이썬을 설치하는 것을 원하신다면 아래의 링크를 통해 방법을 확  
인하실 수 있습니다.

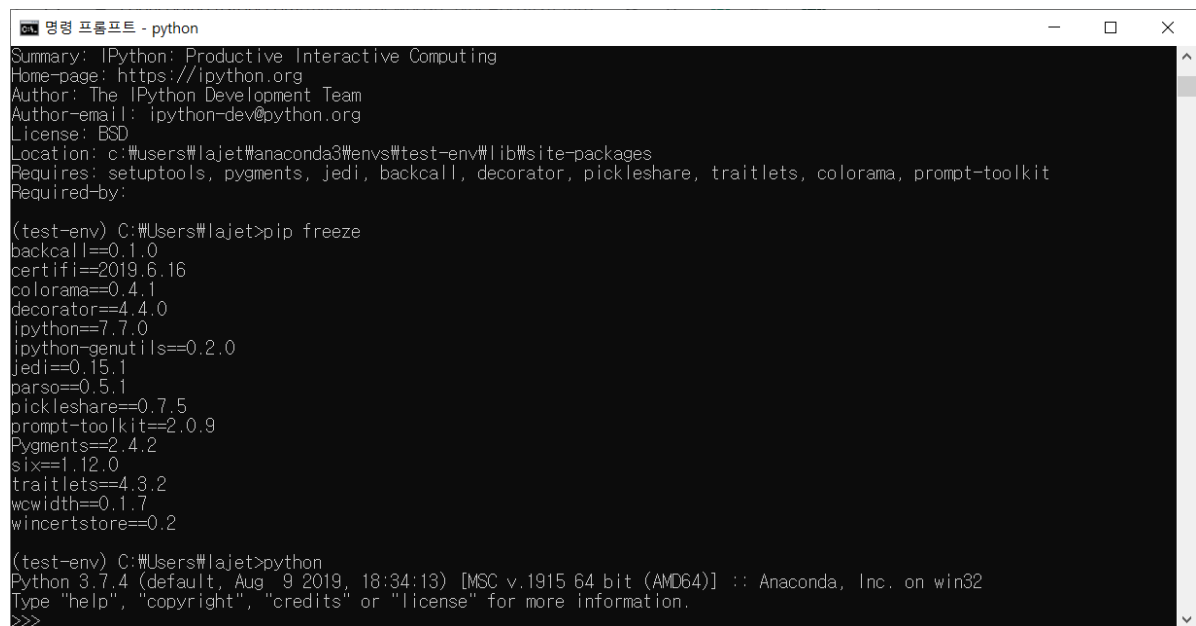
[2019/08/11 - Windows10 운영체제에서 파이썬\(Python\) 설치하기](#)

IPython 설치가 끝났으니, 이번에는 IPython 사용 방법에 대해서 알아보겠습니다. 먼저, 파이썬의 기본으로 설치된 Python Shell을 실행한 후에 IPython Shell을 실행하여 두 가지 Shell의 차이점을 보도록 하겠습니다.

## ○ Python Shell 실행 방법

명령프롬프트 창에 다음과 같은 명령어를 입력하면 아래와 같이 본인의 운영체제에 설치된 파이썬의 버전이 출력되며, 동시에 Python Shell이 실행됩니다.

```
python
```



```
명령 프롬프트 - python
Summary: IPython: Productive Interactive Computing
Home-page: https://ipython.org
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD
Location: c:\Users\lajet\anaconda3\envs\test-env\lib\site-packages
Requires: setuptools, pygments, jedi, backcall, decorator, pickleshare, traitlets, colorama, prompt-toolkit
Required-by:

(test-env) C:\Users\lajet>pip freeze
backcall==0.1.0
certifi==2019.6.16
colorama==0.4.1
decorator==4.4.0
ipython==7.7.0
ipython-genutils==0.2.0
jedi==0.15.1
parso==0.5.1
pickleshare==0.7.5
prompt-toolkit==2.0.9
Pygments==2.4.2
six==1.12.0
traitlets==4.3.2
wcwidth==0.1.7
wincertstore==0.2

(test-env) C:\Users\lajet>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Python Shell 실행 방법 - 1

다음과 같이 간단하게 Welcome CODA를 출력해보는 코드를 입력해보니, 정상적으로 출력된 것을 볼 수 있습니다.

```
print("Welcome CODA")
```

```
명령 프롬프트 - python
Author: The IPython Development Team
Author-email: ipython-dev@python.org
License: BSD
Location: c:\Users\lajet\anaconda3\envs\test-env\lib\site-packages
Requires: setuptools, pygments, jedi, backcall, decorator, pickleshare, traitlets, colorama, prompt-toolkit
Required-by:

(test-env) C:\Users\lajet>pip freeze
backcall==0.1.0
certifi==2019.6.16
colorama==0.4.1
decorator==4.4.0
ipython==7.7.0
ipython-genutils==0.2.0
jedi==0.15.1
parso==0.5.1
pickleshare==0.7.5
prompt-toolkit==2.0.9
Pygments==2.4.2
six==1.12.0
traitlets==4.3.2
wcwidth==0.1.7
wincertstore==0.2

(test-env) C:\Users\lajet>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Welcome CODA")
Welcome CODA
>>>
```

## Python Shell 실행 방법 - 2

이렇게 Python Shell을 사용하여 코드를 작성하면 간단한 코드를 소스파일에 저장하여 실행할 필요 없이 간단하게 바로 실행할 수 있습니다. exit()를 입력하면 파이썬 셸을 종료할 수 있습니다.

## ○ IPython Shell 실행 방법 및 Python Shell과의 차이점

이번에는 IPython Shell을 실행해보도록 하겠습니다. 명령 프롬프트 창에 다음과 같은 명령어를 입력하면 IPython Shell이 실행됩니다.

```
ipython
```

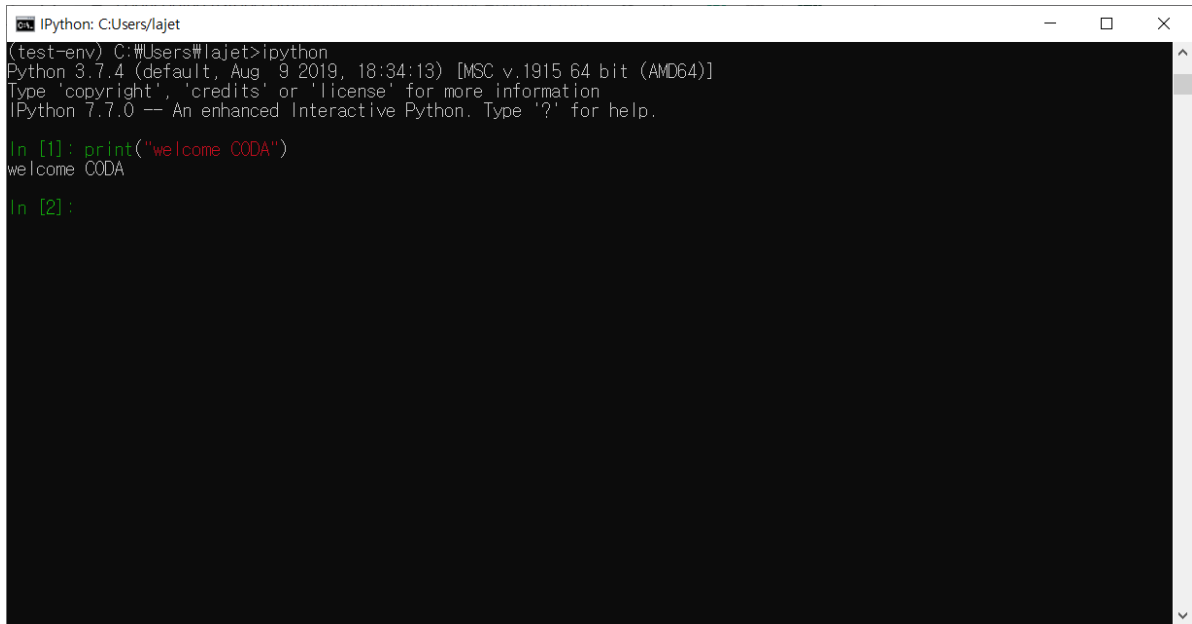
```
선택 IPython: C:\Users\lajet
(test-env) C:\Users\lajet>ipython
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information
IPython 7.7.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

## IPython Shell 실행 방법

이어서 이전에 출력했던 Welcome CODA를 출력해보도록 하겠습니다.

```
print("welcome CODA")
```



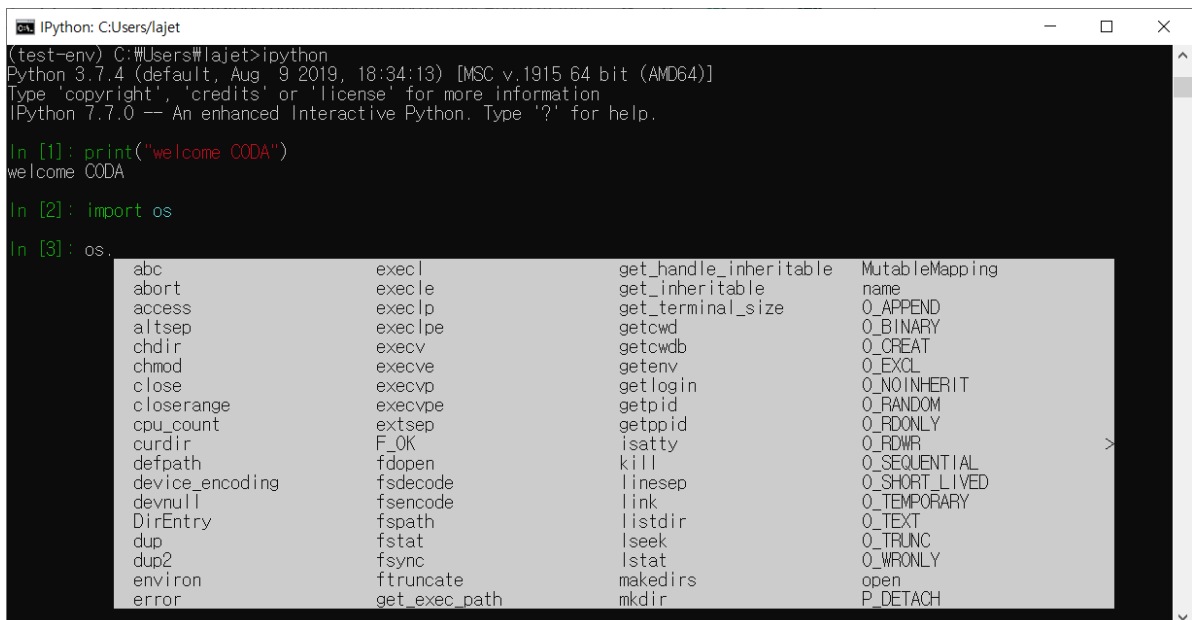
```
IPython: C:\Users\lajet
(test-env) C:\Users\lajet>ipython
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.7.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("welcome CODA")
welcome CODA

In [2]:
```

IPython Shell 기능 - 1

위와 같이 일반 Python Shell과는 달리 코드의 가독성을 높여주는 코드 하이라이팅(Highlighting) 기능이 탑재된 것을 볼 수 있습니다. 또한 IPython은 일반 Python Shell에서 제공하지 않은 자동 완성(Tab Completion) 기능을 제공합니다. 예를 들어, 아래와 같이 모듈을 입력한 후 Tab 키를 누르면 모듈에 포함되어 있는 모든 함수를 보여줍니다.



```
IPython: C:\Users\lajet
(test-env) C:\Users\lajet>ipython
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.7.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("welcome CODA")
welcome CODA

In [2]: import os

In [3]: os.
abc          execl          get_handle_inheritable  MutableMapping
abort        execl         get_inheritable         name
access       execlp        get_terminal_size       O_APPEND
altsep       execlpe       getcwd                  O_BINARY
chdir        execlv        getcwddb                O_CREAT
chmod        execlv        getenv                  O_EXCL
close        execlvp       getlogin                O_NOINHERIT
closerange   execlvpe      getpid                 O_RANDOM
cpu_count    extsep        getppid                 O_RDONLY
curdir       F_OK          isatty                  O_RDWR
defpath      fdopen        kill                    O_SEQUENTIAL
device_encoding fsdecode      linesep                 O_SHORT_LIVED
devnull      fsencode      link                    O_TEMPORARY
DirEntry     fspath        listdir                 O_TEXT
dup          fstat         lseek                   O_TRUNC
dup2         fsync         lstat                   O_WRONLY
environ      ftruncate     makedirs                open
error        get_exec_path mkdir                  P_DETACH
```

IPython Shell 기능 - 2

또한 for문을 입력할 때 자동으로 코드 들여넣기를 해주는 기능도 있습니다.

```
IPython: C:\Users\lajet
(test-env) C:\Users\lajet>ipython
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.7.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("welcome CODA")
welcome CODA

In [2]: import os

In [3]: for i in range(10):
...:     print(i)
...:
0
1
2
3
4
5
6
7
8
9

In [4]:
```

### IPython Shell 기능 - 3

물음표를 사용하여 모듈, 함수, 메소드의 설명을 참조할 수 있습니다.

```
IPython: C:\Users\lajet
9

In [4]: os?
Type:      module
String form: <module 'os' from 'c:\Users\lajet\Anaconda3\envs\test-env\lib\os.py'>
File:      c:\Users\lajet\Anaconda3\envs\test-env\lib\os.py
Docstring:
OS routines for NT or Posix depending on what system we're on.

This exports:
- all functions from posix or nt, e.g. unlink, stat, etc.
- os.path is either posixpath or ntpath
- os.name is either 'posix' or 'nt'
- os.getcwd() is a string representing the current directory (always '.')
- os.pardir is a string representing the parent directory (always '..')
- os.sep is the (or a most common) pathname separator ('/' or '\\')
- os.extsep is the extension separator (always '.')
- os.altsep is the alternate pathname separator (None or '/')
- os.pathsep is the component separator used in $PATH etc
- os.linesep is the line separator in text files ('\r' or '\n' or '\r\n')
- os.defpath is the default search path for executables
- os.devnull is the file path of the null device ('/dev/null', etc.)

Programs that import and use 'os' stand a better chance of being
portable between different platforms. Of course, they must then
only use functions that are defined by all platforms (e.g., unlink
and opendir), and leave all pathname manipulation to os.path
(e.g., split and join).

In [5]:
```

### IPython Shell 기능 - 4

마지막으로 IPython이 제공하는 매직 함수 중에서 몇 가지를 살펴보도록 하겠습니다. 파이썬 모듈을 import 해야만 사용할 수 있는 것을 매직 함수를 사용하면 간단히 사용할 수 있도록 해줍니다. 먼저 아래와 같은 코드를 입력하여 현재 폴더의 파일의 목록을 불러오도록 하겠습니다.

```
%ls
```

```
선택 IPython: C:\Users\lajet

In [5]: %ls
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: A6B1-8ADD

C:\Users\lajet 디렉터리

2019-08-14 오후 07:57 <DIR> .
2019-08-14 오후 07:57 <DIR> ..
2019-08-14 오후 07:29 <DIR> .atom
2019-08-14 오후 07:57 <DIR> .conda
2019-08-10 오후 11:27 <DIR> .idlerc
2019-08-11 오후 10:05 <DIR> .ipynb_checkpoints
2019-08-11 오후 09:59 <DIR> .ipython
2019-08-11 오후 10:02 <DIR> .jupyter
2019-08-11 오후 09:58 7 .python_history
2019-08-06 오후 09:56 <DIR> 3D Objects
2019-08-11 오후 09:48 <DIR> Anaconda3
2019-08-06 오후 09:56 <DIR> Contacts
2019-08-14 오후 05:11 <DIR> Desktop
2019-08-14 오후 04:58 <DIR> Documents
2019-08-14 오후 04:57 <DIR> Downloads
2019-08-06 오후 10:01 <DIR> Favorites
2019-08-06 오후 09:56 <DIR> Links
2019-08-06 오후 09:56 <DIR> Music
2019-08-10 오후 04:23 <DIR> OneDrive
2019-08-06 오후 09:58 <DIR> Pictures
2019-08-06 오후 09:56 <DIR> Saved Games
2019-08-06 오후 09:57 <DIR> Searches
2019-08-11 오후 10:07 798 Untitled.ipynb
```

## IPython Shell 기능 - 5

이번에는 다음과 같은 코드를 입력하여 test라는 이름의 폴더를 생성한 후, 생성한 폴더로 이동해보겠습니다.

```
%mkdir test
%cd test
```

```
IPython: C:\lajet\test

2019-08-11 오후 10:05 <DIR> .ipynb_checkpoints
2019-08-11 오후 09:59 <DIR> .ipython
2019-08-11 오후 10:02 <DIR> .jupyter
2019-08-11 오후 09:58 7 .python_history
2019-08-06 오후 09:56 <DIR> 3D Objects
2019-08-11 오후 09:48 <DIR> Anaconda3
2019-08-06 오후 09:56 <DIR> Contacts
2019-08-14 오후 05:11 <DIR> Desktop
2019-08-14 오후 04:58 <DIR> Documents
2019-08-14 오후 04:57 <DIR> Downloads
2019-08-06 오후 10:01 <DIR> Favorites
2019-08-06 오후 09:56 <DIR> Links
2019-08-06 오후 09:56 <DIR> Music
2019-08-10 오후 04:23 <DIR> OneDrive
2019-08-06 오후 09:58 <DIR> Pictures
2019-08-06 오후 09:56 <DIR> Saved Games
2019-08-06 오후 09:57 <DIR> Searches
2019-08-11 오후 10:07 798 Untitled.ipynb
2019-08-06 오후 10:32 <DIR> Videos
22개 파일 805 바이트
22개 디렉터리 94,375,604,224 바이트 남음

In [6]: mkdir test
In [7]: cd test
C:\Users\lajet\test

In [8]: pwd
Out[8]: 'C:\Users\lajet\test'
```

## IPython Shell 기능 - 6

이 외에도 편리한 매직 함수를 많이 제공하고 있는데, 다음과 같은 코드를 입력하여 매직 함수의 목록과 정보를 볼 수 있습니다.

```
%quickref
```



```
IPython: C:\ajet/test
In [10]: %quickref

IPython -- An enhanced Interactive Python - Quick Reference Card
=====

obj?, obj??      : Get help, or more help for object (also works as
                  ?obj, ??obj).
?foo.*abc*       : List names in 'foo' containing 'abc' in them.
%magic           : Information about IPython's 'magic' % functions.

Magic functions are prefixed by % or %, and typically take their arguments
without parentheses, quotes or even commas for convenience. Line magics take a
single % and cell magics are prefixed with two %.

Example magic function calls:

%alias d ls -F    : 'd' is now an alias for 'ls -F'
alias d ls -F    : Works if 'alias' not a python name
alist = %alias    : Get list of aliases to 'alist'
cd /usr/share    : Obvious. cd ~<tab> to choose from visited dirs.
%cd??           : See help AND source for magic %cd
%timeit x=10     : time the 'x=10' statement with high precision.
%%timeit x=2**100 : time 'x**100' with a setup of 'x=2**100'; setup code is not
x**100           : counted. This is an example of a cell magic.

System commands:

!cp a.txt b/     : System command escape, calls os.system()
```

## IPython Shell 기능 - 7

앞서 말씀드렸듯이 Python Shell은 개발을 할 때 코드 테스트 또는 모듈 테스트 등에 주로 사용되며, 공부를 하면서도 간단히 코드를 테스트해볼 수 있는 편리한 도구이므로 반드시 익숙해져야 할 과제 중 하나입니다.