

# 파이선 개요

강병준

# Python역사

1. 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 대화식 인터프리터 언어
  2. 귀도가 좋아하는 코미디 쇼인 '몬티 파이썬의 날아다니는 서커스'에서 따 왔다고 함
  3. 파이썬의 사전적 의미는 고대 신화에 나오는 파르나소스산의 동굴에 살 던 큰 뱀을 의미함
  4. 플랫폼에 독립적, 간단하고 쉬운 문법,
  5. 메모리 관리를 자동으로 수행, 많은 라이브러리 제공
- 구글에서 만든 소프트웨어의 50%이상이 파이썬으로 작성되었다는 이야기를 들었다. 이외에도 많이 알려진 예를 몇 가지 들자면 온라인 사진 공유 서비스 인스타그램(Instagram), 파일 동기화 서비스 드롭박스(Dropbox)등이 있다.
  - 또한 파이썬 프로그램은 공동 작업과 유지 보수가 매우 쉽고 편하다. 그 때문에 이미 다른 언어로 작성된 많은 프로그램과 모듈이 파이썬으로 재구성되고 있다. 국내에서도 그 가치를 인정받아 사용자 층이 더욱 넓어지고 있고, 파이썬을 사용해 프로그램을 개발하는 업체들 또한 늘어 가고 있는 추세이다.

# 파이썬 특징

## (1) 쉽고 간결

C 언어에서는 "Hello world"를 출력하기 위해 6~8줄이나 써야 하는 반면, 파이썬에서는 그저 `print("Hello world")` 한 줄만 적으면 됩니다. 프로그래밍 언어가 복잡하면 문제 해결능력을 기르는 것보다, 어려운 프로그래밍 문법 자체를 익히는데 초점이 맞춰지게 됩니다. 하지만 파이썬은 간결한 언어로써, '컴퓨터 과학'의 기초와 문제 해결능력에 집중할 수 있게 해줍니다.

## (2) 활용성이 높음

단순한 프로그래밍 이외에 데이터 분석에도 파이썬을 활용할 수 있습니다. [BeautifulSoup](#), [NumPy](#), [Pandas](#) 등 데이터 수집 및 분석을 위한 훌륭한 라이브러리들이 있습니다.

## (3) 많이 사용

미국 대학 중 상위 39개 컴퓨터 학과에서 첫 프로그래밍 언어로 파이썬을 선택하는 비율이 약 30%로 1위입니다. 구글의 경우 유튜브를 포함한 대부분의 서비스를 파이썬을 활용하고 있으며, 드롭박스 또한 파이썬으로 제작되었습니다. 우리나라에서 널리 쓰인지는 얼마되지 않았지만, 실리콘밸리에서 파이썬은 최근 가장 인기있는 언어 중 하나입니다.

## (4) 개발 속도가 빠름

파이썬의 쉽고 간결한 특징 때문에, 다른 언어에 비해 개발 속도가 매우 빠릅니다. 파이썬의 개발 속도가 빨라서, "Life is too short, You need Python".이라는 말이 유행처럼 생겨날 정도였습니다.

# Python

## 파이썬의 구현

1. C파이썬: C로 작성된 인터프리터로 일반적인 파이썬
2. 스택리스 파이썬 : C 스택을 사용하지 않는 인터프리터.
3. 자이썬 : 자바 가상 머신 용 인터프리터.  
과거에는 제이파이썬(JPython)이라고 불렸다.
4. IronPython : .NET 플랫폼 용 인터프리터.
5. PyPy : 파이썬으로 작성된 파이썬 인터프리터

## 사용 가능 플랫폼

1. 마이크로소프트 윈도우(9x/NT 계열은 최신판, 3.1 및 MS-DOS는 옛 버전만)
2. 매킨토시(맥 OS 9 이전, 맥 OS X 이후 포함)
3. 각종 유닉스
4. 리눅스
5. 팜 OS
6. 노키아 시리즈 60

# 파이썬의 활용 분야 및 버전

## 파이썬 활용분야

1. GUI Programming
2. Web Programming
3. Database Programming
4. Text 처리
5. 수치 연산 Programming: Numeric Python 모듈 이용
6. 병렬 연산 Programming
7. 사물인터넷
8. C,C++ 과 결합한 프로그래밍

## 파이썬의 버전

1. 파이썬은 2.x 버전과 3.x 버전이 있는데 파이썬은 2.x버전 과 3.x버전 간의 호환성을 유지하고 있지 않습니다.
2. 변화
  - ◆print가 함수로 변경
  - ◆long 형이 없어지고 정수는 int로 통일
  - ◆int / int 의 결과는 float
  - ◆문자열이 string과 unicode로 구분되었는데 string 과 bytes로 구분

# 파이썬으로 할 수 있는 일

## 시스템 유틸리티 제작

- 파이썬은 운영체제(윈도우, 리눅스 등)의 시스템 명령어를 사용할 수 있는 각종 도구를 갖추고 있기 때문에 이를 바탕으로 갖가지 시스템 유틸리티를 만드는 데 유리하다. 실제로 여러분은 시스템에서 사용 중인 서로 다른 유틸리티성 프로그램을 하나로 뭉쳐서 큰 힘을 발휘하게 하는 프로그램들을 무수히 만들어낼 수 있다.
- ※ 유틸리티란 컴퓨터 사용에 도움을 주는 여러 소프트웨어를 말한다.

## GUI 프로그래밍

- GUI(Graphic User Interface) 프로그래밍이란 쉽게 말해 화면에 또 다른 윈도우 창을 만들고 그 창에 프로그램을 동작시킬 수 있는 메뉴나 버튼, 그림 등을 추가하는 것이다. 파이썬은 GUI 프로그래밍을 위한 도구들이 잘 갖추어져 있어 GUI 프로그램을 만들기 쉽다. 대표적인 예로 파이썬 프로그램과 함께 설치되는 Tkinter(티케이인터)가 있다. Tkinter를 사용하면 단 5줄의 소스 코드만으로 윈도우 창을 띄울 수 있다.

## C/C++와의 결합

- 파이썬은 접착(glue) 언어라고도 부르는데, 그 이유는 다른 언어와 잘 어울려 결합해서 사용할 수 있기 때문이다. C나 C++로 만든 프로그램을 파이썬에서 사용할 수 있으며, 파이썬으로 만든 프로그램 역시 C나 C++에서 사용할 수 있다.

# 파이썬으로 할 수 있는 일

## 웹 프로그래밍

- 일반적으로 익스플로러나 크롬, 파이어폭스 같은 브라우저로 인터넷을 사용하는데, 누구나 한 번쯤 웹 서핑을 하면서 게시판이나 방명록에 글을 남겨 본 적이 있을 것이다. 그러한 게시판이나 방명록을 바로 웹 프로그램이라고 한다. 파이썬은 웹 프로그램을 만들기에 매우 적합한 도구이며, 실제로 파이썬으로 제작한 웹 사이트는 셀 수 없을 정도로 많다.

## 수치 연산 프로그래밍

- 사실 파이썬은 수치 연산 프로그래밍에 적합한 언어는 아니다. 수치가 복잡하고 연산이 많다면 C 같은 언어로 하는 것이 더 빠르기 때문이다. 하지만 파이썬은 NumPy라는 수치 연산 모듈을 제공한다. 이 모듈은 C로 작성했기 때문에 파이썬에서도 수치 연산을 빠르게 할 수 있다.

## 데이터베이스 프로그래밍

- 파이썬은 사이베이스(Sybase), 인포믹스(Infomix), 오라클(Oracle), 마이에스큐엘(MySQL), 포스트그레스큐엘(PostgreSQL) 등의 데이터베이스에 접근하기 위한 도구를 제공한다. 또한 이런 굴직한 데이터베이스를 직접 사용하는 것 외에도 파이썬에는 재미있는 도구가 하나 더 있다. 바로 피클(pickle)이라는 모듈이다. 피클은 파이썬에서 사용하는 자료를 변형 없이 그대로 파일에 저장하고 불러오는 일을 맡아 한다. 이 책에서는 외장 함수에서 피클을 어떻게 사용하고 활용하는지에 대해서 알아본다..



# 파이썬으로 할 수 있는 일

## 데이터 분석, 사물 인터넷

- 파이썬으로 만든 판다스(Pandas) 모듈을 사용하면 데이터 분석을 더 쉽고 효과적으로 할 수 있다. 데이터 분석을 할 때 아직까지는 데이터 분석에 특화된 'R'이라는 언어를 많이 사용하고 있지만, 판다스가 등장한 이후로 파이썬을 사용하는 경우가 점점 증가하고 있다.
- 사물 인터넷 분야에서도 파이썬은 활용도가 높다. 한 예로 라즈베리파이(Raspberry Pi)는 리눅스 기반의 아주 작은 컴퓨터이다. 라즈베리파이를 사용하면 홈시어터나 아주 작은 게임기 등 여러 가지 재미있는 것들을 만들 수 있는데, 파이썬은 이 라즈베리파이를 제어하는 도구로 사용된다. 예를 들어 라즈베리파이에 연결된 모터를 작동시키거나 LED에 불이 들어오게 하는 일을 파이썬으로 할 수 있다.

## 파이썬으로 할 수 없는 일

### 시스템과 밀접한 프로그래밍 영역

- 파이썬으로 리눅스 같은 운영체제, 엄청난 횟수의 반복과 연산이 필요한 프로그램 또는 데이터 압축 알고리즘 개발 프로그램 등을 만드는 것은 어렵다. 즉 대단히 빠른 속도를 요구하거나 하드웨어를 직접 건드릴어야 하는 프로그램에는 어울리지 않는다.

### 모바일 프로그래밍

- 파이썬은 구글이 가장 많이 애용하는 언어이지만 파이썬으로 안드로이드 앱(App)을 개발하는 것은 아직 어렵다. 안드로이드에서 파이썬으로 만든 프로그램이 실행되도록 지원하긴 하지만 이것만으로 앱을 만들기에는 아직 역부족이다. 아이폰 앱을 개발하는 것 역시 파이썬으로 할 수 없다



# 방법1 : 파이썬 다운로드 및 설치

파이썬을 설치한다는 것은 파이썬 언어를 해석할 수 있는 interpreter라는 프로그램을 설치한다는 뜻이다. 인터프리터란 소스 코드(source code)를 읽고 해석하여 수행하는 프로그램이다. 최초의 파이썬 언어 인터프리터는 C 언어로 만들어졌다. C로 만든 파이썬은 CPython(씨 파이썬이라고 읽는다)이라고 한다. CPython은 C 혹은 C++ 언어로 된 바이너리 라이브러리를 직접 호출할 수 있기 때문에 C/C++ 라이브러리를 파이썬에서 쉽게 사용할 수 있는 장점이 있다. 수치 계산이나 데이터 분석용으로 파이썬을 사용할 때는 항상 CPython을 사용한다.

그 외에 자바(Java)로 구현된 자이썬(Jython), 닷넷(.NET)으로 구현된 IronPython, 자바스크립트로 구현된 스컬프트(Skulpt) 등의 파이썬도 있다. 파이썬은 자바 라이브러리나 닷넷 컴포넌트를 파이썬에서 직접 호출하여 사용할 때나 웹 브라우저에서 직접 파이썬을 실행할 때 쓰인다. 파이파이(PyPy)는 JIT(Just-In-Time) 컴파일러를 내장하여 반복되는 코드를 빠르게 실행할 수 있도록 한 파이썬이다.

## 파이썬 다운로드 및 설치

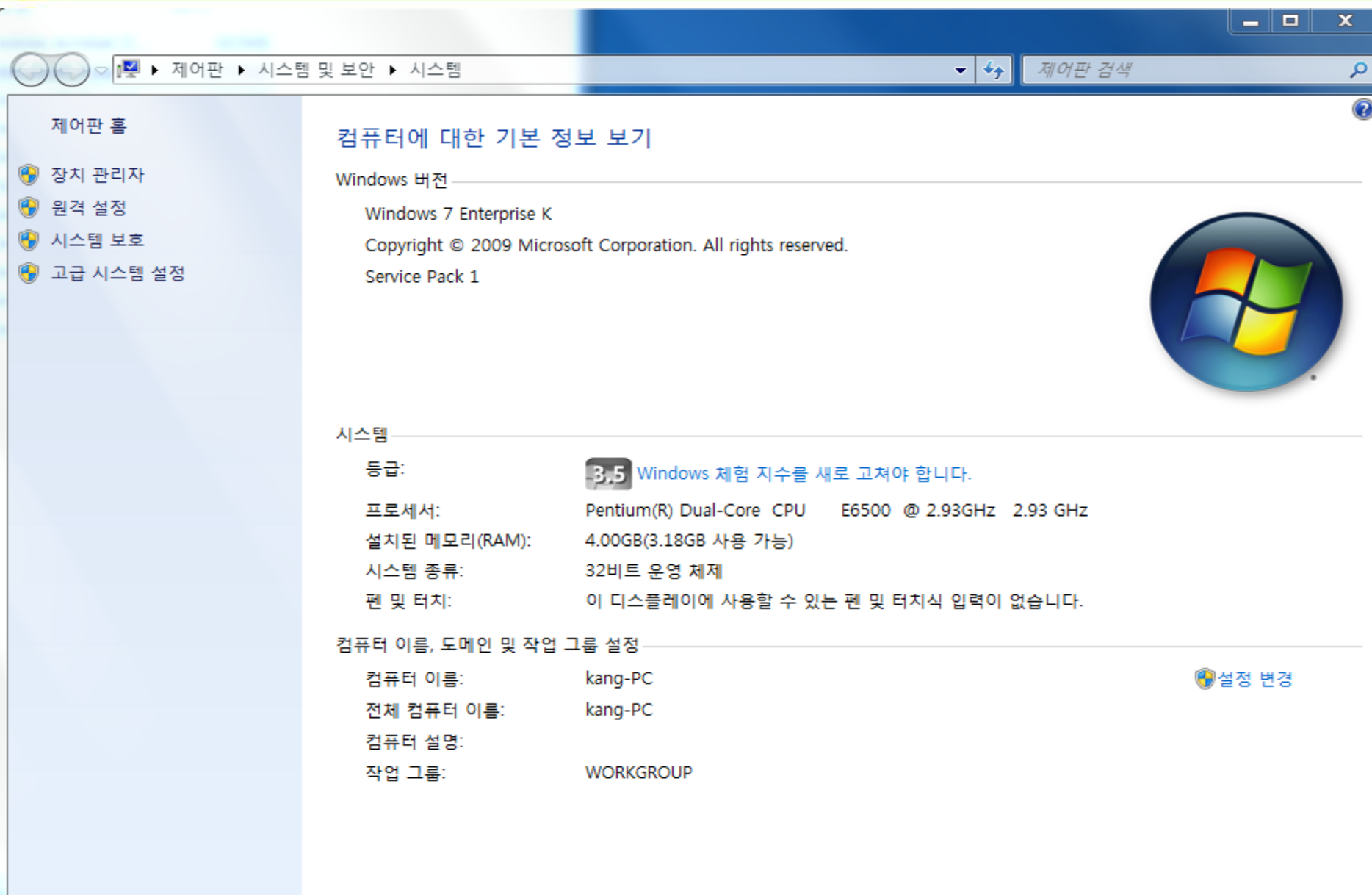
<https://www.python.org/downloads/>

윈도우 용을 설치할 때 파이썬 명령어가 있는 디렉토리를 path에 추가하는 옵션을 체크해야 이클립스에서 파이썬 작업이 가능합니다.

## 윈도우용 설치 파일은 32비트/64비트용이 따로 배포됨

어떤 버전을 사용할지 판단하기 위해서는 PC에 탑재되어 있는 윈도우의 아키텍처를 파악하는 것이 필요

# 윈도 버전 보기



The screenshot shows the Windows 7 'System' window in Korean. The title bar includes navigation buttons and a search box labeled '제어판 검색'. The left sidebar lists '제어판 홈' and various control panel links like '장치 관리자', '원격 설정', '시스템 보호', and '고급 시스템 설정'. The main content area is titled '컴퓨터에 대한 기본 정보 보기' and is divided into three sections: 'Windows 버전', '시스템', and '컴퓨터 이름, 도메인 및 작업 그룹 설정'. The 'Windows 버전' section shows 'Windows 7 Enterprise K' and copyright information. The '시스템' section displays hardware details like CPU (Pentium(R) Dual-Core CPU E6500 @ 2.93GHz), RAM (4.00GB), and system type (32-bit operating system). A yellow warning box indicates a Windows Experience Index of 3.5. The '컴퓨터 이름, 도메인 및 작업 그룹 설정' section shows the computer name as 'kang-PC' and the workgroup as 'WORKGROUP'. A '설정 변경' link is visible in the bottom right of this section.


제어판 홈

- 장치 관리자
- 원격 설정
- 시스템 보호
- 고급 시스템 설정

## 컴퓨터에 대한 기본 정보 보기

### Windows 버전

Windows 7 Enterprise K  
Copyright © 2009 Microsoft Corporation. All rights reserved.  
Service Pack 1



### 시스템

등급: **3.5** Windows 체험 지수를 새로 고쳐야 합니다.

프로세서: Pentium(R) Dual-Core CPU E6500 @ 2.93GHz 2.93 GHz

설치된 메모리(RAM): 4.00GB(3.18GB 사용 가능)

시스템 종류: 32비트 운영 체제

펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 및 터치식 입력이 없습니다.


### 컴퓨터 이름, 도메인 및 작업 그룹 설정

컴퓨터 이름: kang-PC

전체 컴퓨터 이름: kang-PC

컴퓨터 설명:

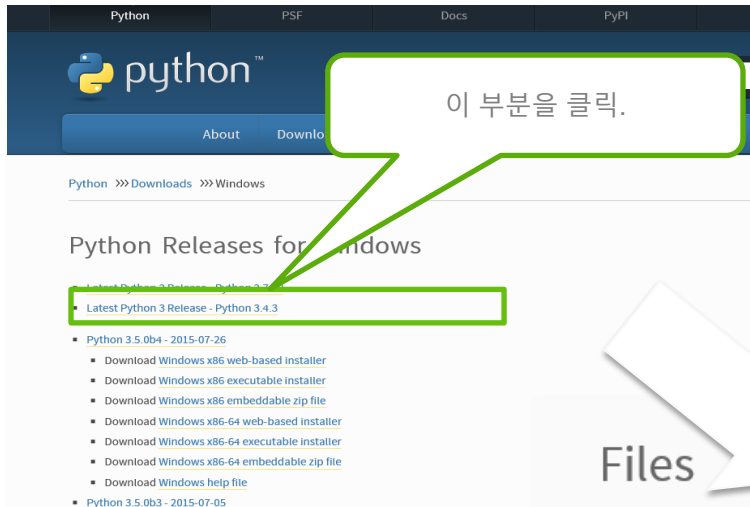
작업 그룹: WORKGROUP

 [설정 변경](#)

# 파이썬 설치하기

## ❖ 파이썬 다운로드 링크

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



## Files

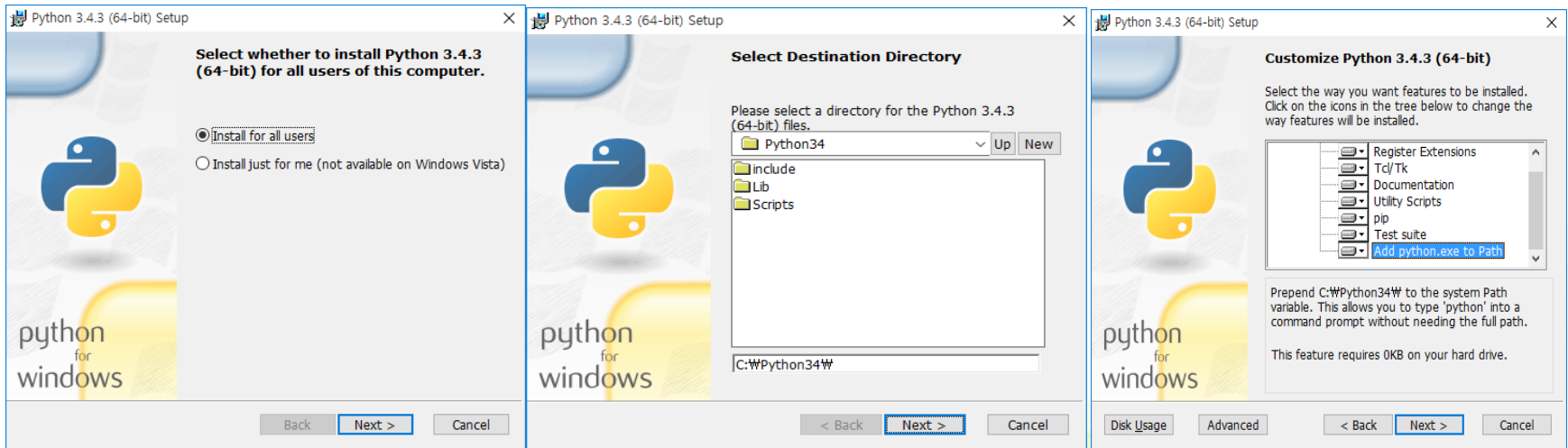
Version	Operating System	Description
<a href="#">Gzipped source tarball</a>	Source release	
<a href="#">XZ compressed source tarball</a>	Source release	
<a href="#">Mac OS X 32-bit i386/PPC installer</a>	Mac OS X	for Mac OS X 10.5 and later
<a href="#">Mac OS X 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later
<a href="#">Windows debug information files</a>	Windows	
<a href="#">Windows debug information files for 64-bit binaries</a>	Windows	
<a href="#">Windows help file</a>		
<a href="#">Windows x86-64 MSI installer</a>	Windows	for AMD64/EM64T/x64, not Itanium processors
<a href="#">Windows x86 MSI installer</a>		

64비트 윈도우용 설치 파일

32비트 윈도우용 설치 파일

# 파이썬 설치하기

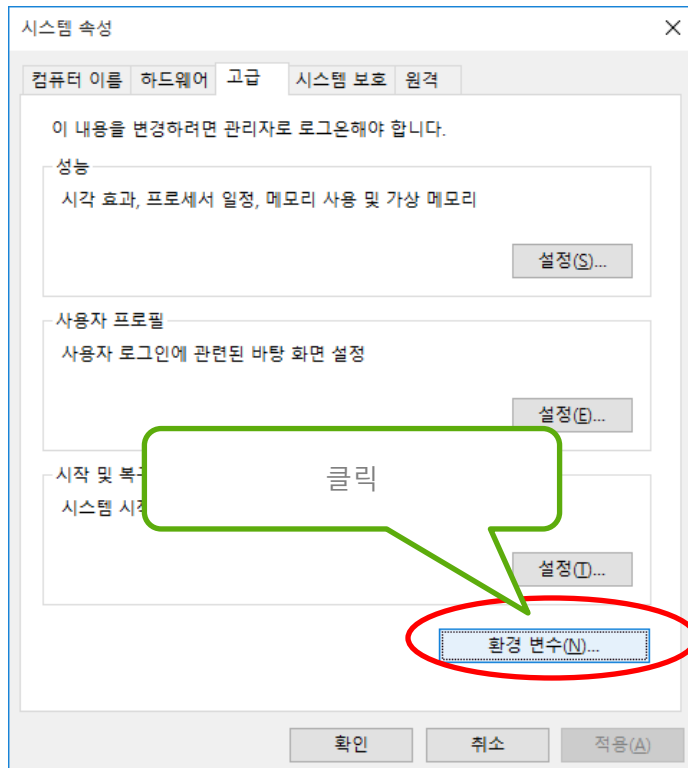
❖ 설치 파일을 클릭하여 설치 시작



# 파이썬 설치하기

## ❖ 환경 변수 설정 - 1

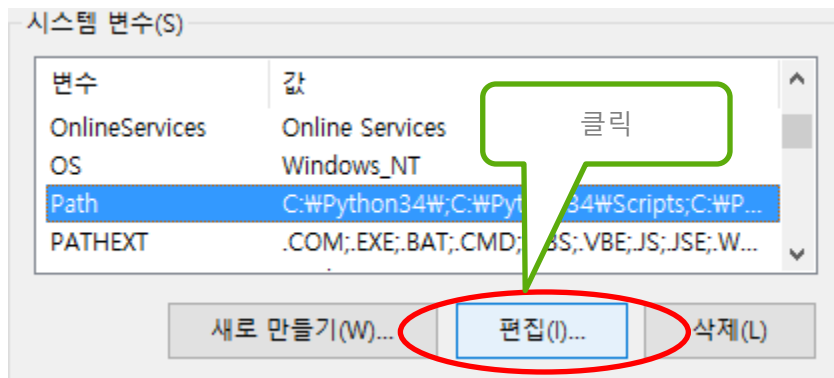
- 설치 마법사가 설치를 완료하면 python.exe와 부속 실행 파일(예:pip.exe)들을 어느 경로에 서나 실행할 수 있도록 Path 시스템 변수를 편집
- <시스템 속성> 창을 띄우기 위해 [윈도우]+[R] 키를 눌러 <실행> 창을 표시. <실행> 창이 나타나면 [열기] 입력 상자에 “sysdm.cpl”를 입력하고 [확인] 버튼을 클릭.



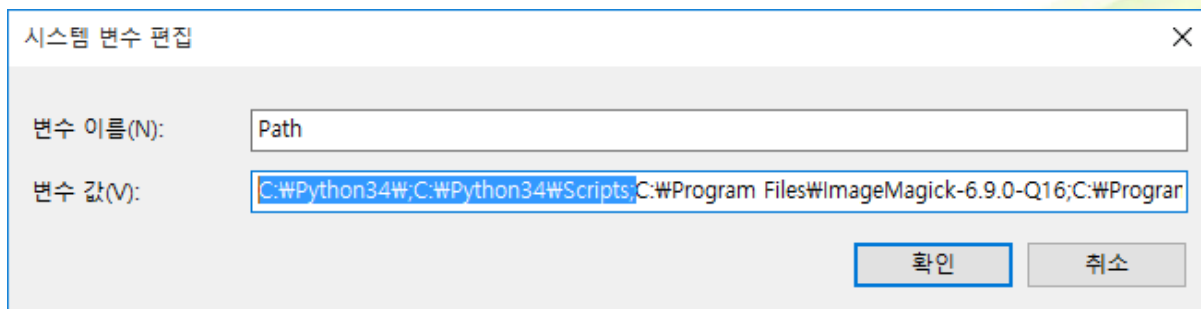
# 파이썬 설치하기

## ❖ 환경 변수 설정 - 2

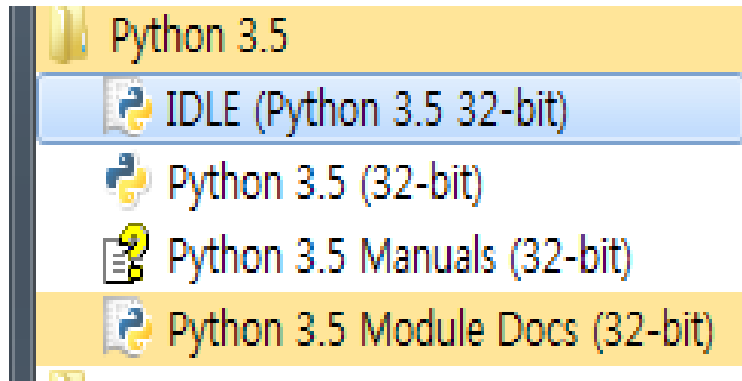
- <환경 변수> 대화상자가 나타나면 [시스템 변수] 그룹에서 “Path” 항목을 찾아 선택하고 [편집]을 선택.



- 시스템 변수 편집 창에서 [변수 값] 항목에 파이썬 설치 디렉토리와 파이썬 스크립트 디렉토리를 추가. 예를 들어 파이썬 설치 디렉토리가 C:\Python34라면 파이썬 스크립트 디렉토리는 C:\Python34\Scripts



# Python



1. IDLE: 표준 GUI 대화식 인터프리터
2. Python: 콘솔 대화식 인터프리터
3. Manuals: HTML 형식의 파이썬 문서
4. Module Docs: 모듈을 검색하기 위한 도구



# 방법2. 아나콘다로 설치하기

현재 가장 널리 사용되는 파이썬 배포판은 '아나콘다(Anaconda)'다. 아나콘다는 여러 파이썬 배포판 중 가장 늦게 발표되었지만 무료이고 완성도가 뛰어나서 사실상의 표준(de facto standard) 파이썬 배포판이 되었다. 아나콘다 배포판은 모든 패키지가 컴파일이 필요 없는 바이너리(binary) 형태로 제공된다. 따라서 설치 속도가 빠르고 패키지 의존성을 관리해주므로 관리가 편리하다.

## 1. Anaconda 를 다운로드

<https://repo.anaconda.com/archive/>

<https://www.anaconda.com/distribution/>

<https://repo.continuum.io/archive/.winzip>

- 파이썬 배포판: Ipython, Jupyter notebook, Qt Console, Python 포함
- 데이터 분석, 수학, 과학 관련 다양한 라이브러리 포함

크롬에서 주소 사용

Anaconda 4.2.0 버전을 다운로드 받아서 설치한다.

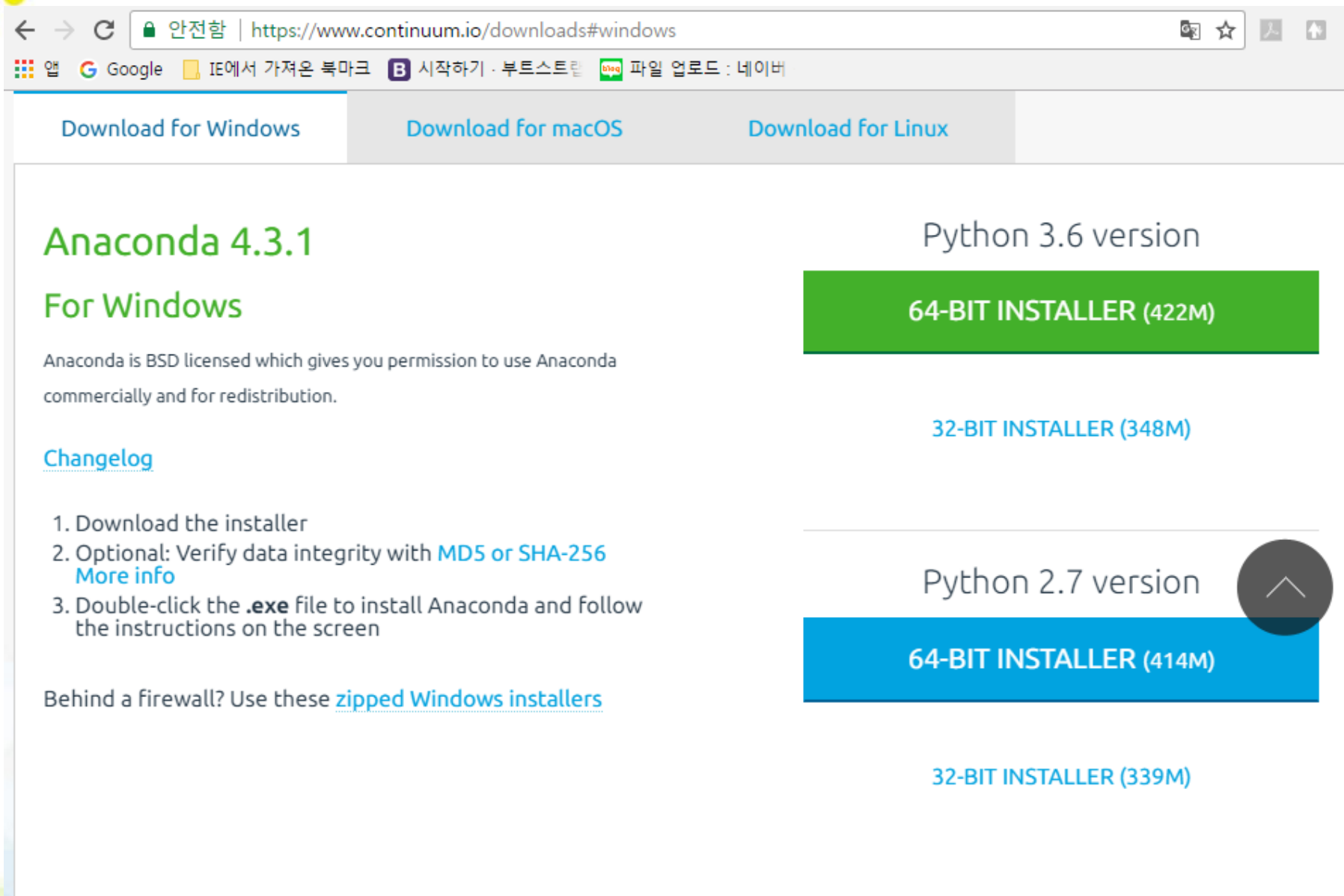
( Python 3.5 내장되어 있음)

C:\Program Files\Anaconda3 폴더에 설치됨

## 2. Anaconda 설치후

Anaconda Prompt 를 관리자 권한으로 실행한다.

# 방법2. 아나콘다로 설치하기



← → ↻ | 🔒 안전함 | <https://www.continuum.io/downloads#windows> | 📄 ☆ | 📄 📄 📄

📄 앱 | 📄 Google | 📄 IE에서 가져온 북마크 | 📄 시작하기 · 부트스트랩 | 📄 파일 업로드 : 네이버

Download for Windows | Download for macOS | Download for Linux

## Anaconda 4.3.1

### For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

[Changelog](#)

1. Download the installer
2. Optional: Verify data integrity with [MD5 or SHA-256](#)  
[More info](#)
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these [zipped Windows installers](#)

Python 3.6 version

**64-BIT INSTALLER (422M)**









32-BIT INSTALLER (348M)

Python 2.7 version

**64-BIT INSTALLER (414M)**

32-BIT INSTALLER (339M)

# 방법2. 아나콘다로 설치하기

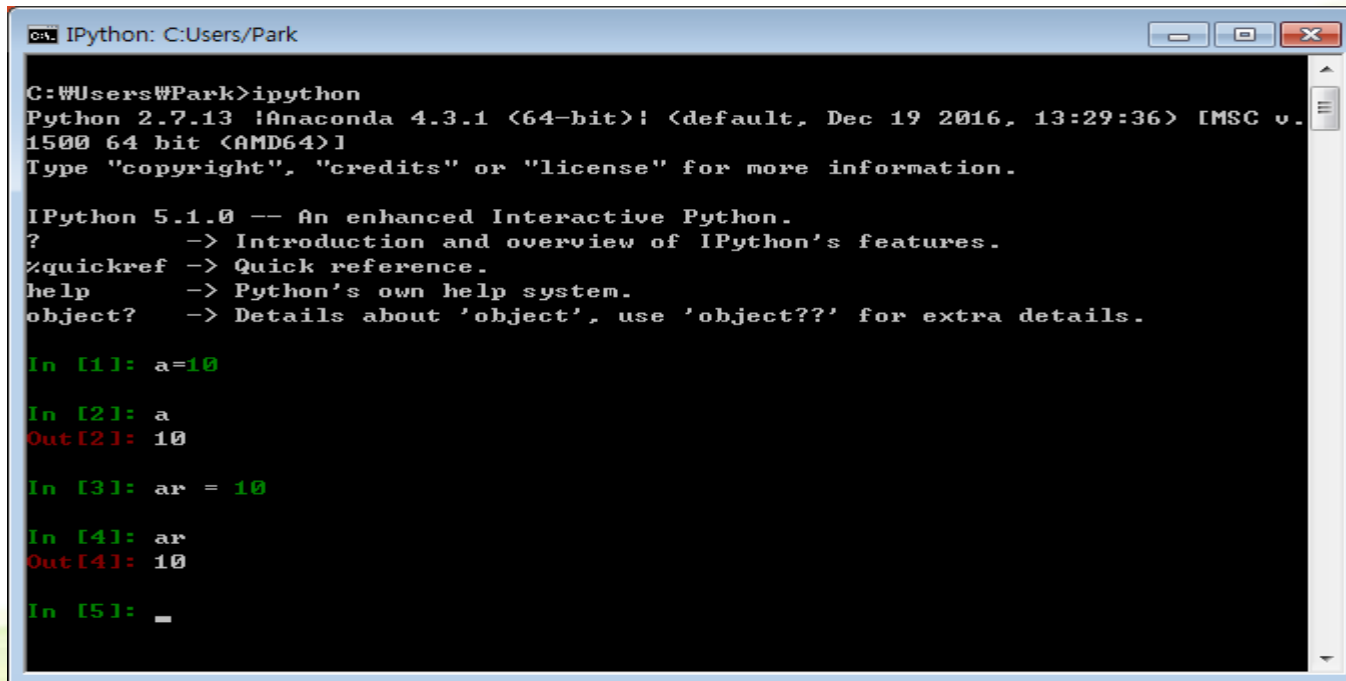
 <a href="https://repo.continuum.io/archive/winzip/">안전함   https://repo.continuum.io/archive/winzip/</a>   			
  IE에서 가져온 북마크  시작하기 · 부트스트랩  파일 업로드 : 네이버			
<h2>Index of /archive/winzip</h2>			
Filename	Size	Last Modified	MD5
../	-		<directory>
<a href="#">Anaconda2-4.3.1-Windows-x86.zip</a>	336.9M	2017-03-10 11:44:41	bb44d53bd11f6a42bf8bbc30060cece5
<a href="#">Anaconda2-4.3.1-Windows-x86_64.zip</a>	411.3M	2017-03-10 11:44:13	152aa56d31dd256f7ce9f9fe27246bc5
<a href="#">Anaconda3-4.3.1-Windows-x86.zip</a>	345.9M	2017-03-10 11:45:41	d729bfc5aa80f7a751016ecc5913f86a
<a href="#">Anaconda3-4.3.1-Windows-x86_64.zip</a>	419.7M	2017-03-10 11:45:12	70f2d4456c5d987cf01966912af5e9bf
<a href="#">Anaconda2-4.3.0.1-Windows-x86.zip</a>	336.0M	2017-02-03 10:26:52	6bb263ebb0f903d9177ed52cf1c7e761
<a href="#">Anaconda2-4.3.0.1-Windows-x86_64.zip</a>	410.4M	2017-02-03 10:26:35	358aef5a62d0d28593fe2a1a805f4a46
<a href="#">Anaconda3-4.3.0.1-Windows-x86.zip</a>	345.0M	2017-02-03 10:27:32	f237cc0a30b00415594317f88bd5efce
<a href="#">Anaconda3-4.3.0.1-Windows-x86_64.zip</a>	418.8M	2017-02-03 10:27:14	4402139144aca664dbbdf764e64d693c
<a href="#">Anaconda2-4.3.0-Windows-x86.zip</a>	336.0M	2017-01-31 10:49:57	e07306b8d09bf9cb0b732827acf62834
<a href="#">Anaconda2-4.3.0-Windows-x86_64.zip</a>	410.4M	2017-01-31 10:49:31	36763eb659bf32cefbl e0c1def4ddc8b
<a href="#">Anaconda3-4.3.0-Windows-x86.zip</a>	345.0M	2017-01-31 10:50:52	ff54f5c185742f6898d005b6f6ebb07b
<a href="#">Anaconda3-4.3.0-Windows-x86_64.zip</a>	418.8M	2017-01-31 10:50:25	e703a81b8d206ebbc9a48c6261188ac1
<a href="#">Anaconda2-4.2.0-Windows-x86.zip</a>	322.0M	2016-09-28 12:01:22	b2bd3bbdef28af3fbf1a3c5916caa88a
<a href="#">Anaconda2-4.2.0-Windows-x86_64.zip</a>	378.8M	2016-09-28 12:01:06	987a30a7c2d86bf2ac17c0e54e62cdc8
<a href="#">Anaconda3-4.2.0-Windows-x86.zip</a>	331.2M	2016-09-28 12:02:00	d6e415e79fa60d06b1532b3ac16eea67
<a href="#">Anaconda3-4.2.0-Windows-x86_64.zip</a>	389.1M	2016-09-28 12:01:43	582e9f9d2cee48666959ec5d82bd06af
<a href="#">Anaconda2-4.1.1-Windows-x86.zip</a>	284.1M	2016-07-08 11:32:17	bb6273f9c66b2cef9b28d9b79bf9d5d35c
<a href="#">Anaconda2-4.1.1-Windows-x86_64.zip</a>	339.2M	2016-07-08 11:32:02	6fff2bc4d0a2a124a3ed22a4eec3ac01
<a href="#">Anaconda3-4.1.1-Windows-x86.zip</a>	292.0M	2016-07-08 11:32:51	81801eff469ca3fe70f7f82c22700f2d
<a href="#">Anaconda3-4.1.1-Windows-x86_64.zip</a>	350.9M	2016-07-08 11:32:36	8daa929263a0694903c078f6b2871e0e
<a href="#">Anaconda2-4.1.0-Windows-x86.zip</a>	283.3M	2016-06-28 12:20:11	6edf05a36a1ab8d925dbf1b886e77aa4
<a href="#">Anaconda2-4.1.0-Windows-x86_64.zip</a>	338.1M	2016-06-28 12:19:56	000afe058833f27886829543b7076acf
<a href="#">Anaconda3-4.1.0-Windows-x86.zip</a>	290.7M	2016-06-28 12:20:44	e16202e16d1bfbe33c60a1f04fa3643
<a href="#">Anaconda3-4.1.0-Windows-x86_64.zip</a>	349.4M	2016-06-28 12:20:29	f064e26164627181b2704e2e7dfa0f8
<a href="#">Anaconda2-4.0.0-Windows-x86.zip</a>	279.2M	2016-03-29 11:33:14	3d5c75c977b0f6bc2220e4eah34109eed

# IDE

1. ipython: 아나콘다를 설치하면 같이 설치
2. PyDev 플러그 인을 이용한 이클립스
3. Visual Studio의 파이썬 도구
4. PyCharm
5. Spider

## ipython

1. ipython을 이용한 코딩 및 실행
2. 객체의 이름만 입력하면 바로 출력



```
C:\Users\Park>ipython
Python 2.7.13 |Anaconda 4.3.1 (64-bit)| (default, Dec 19 2016, 13:29:36) [MSC v.
1500 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: a=10

In [2]: a
Out[2]: 10

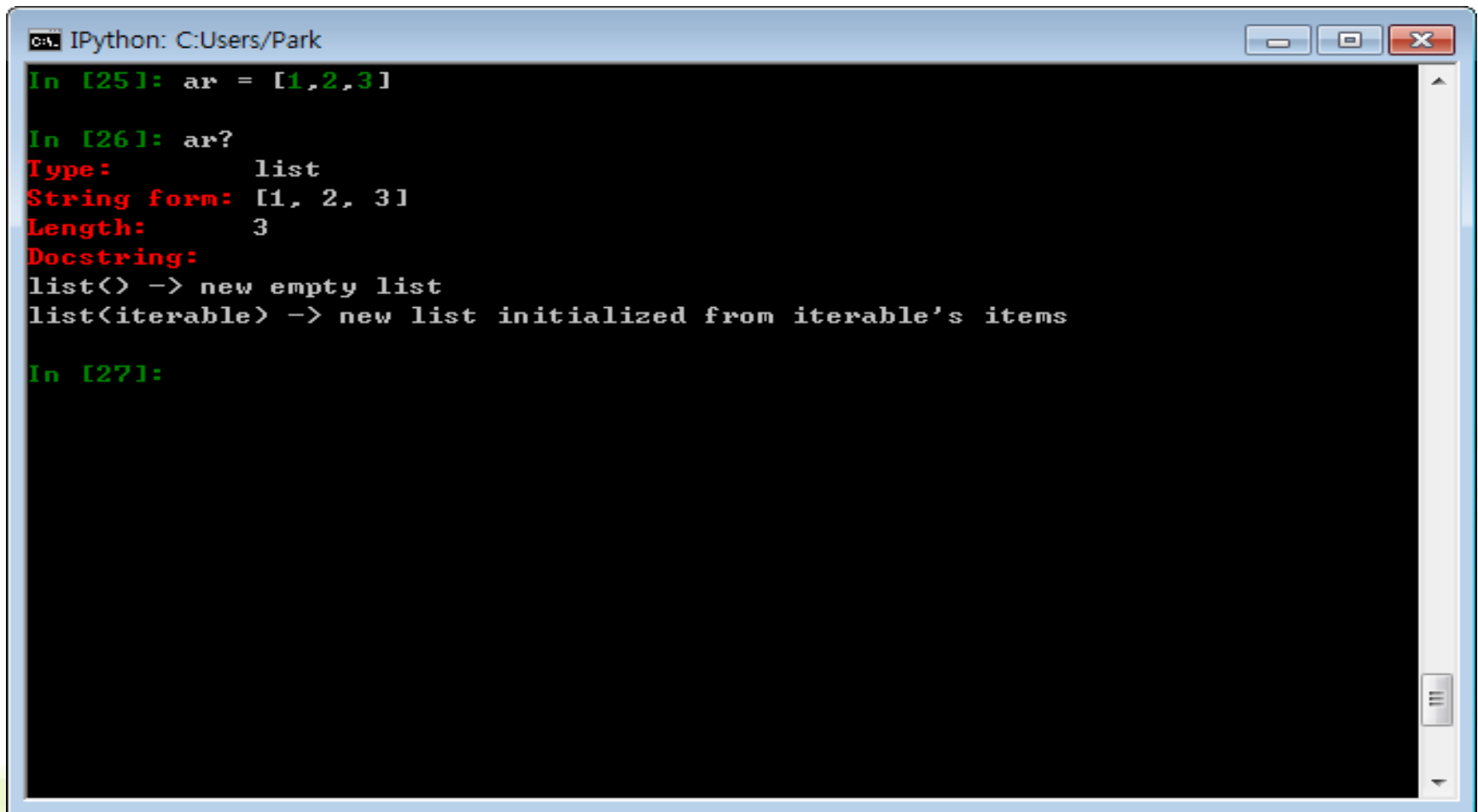
In [3]: ar = 10

In [4]: ar
Out[4]: 10

In [5]: _
```

# ipython

- ❖ 탭을 이용한 자동완성 기능 제공: 명령을 입력하는 동안 Tab을 누르면 현재 위치에서 사용할 수 있는 변수를 화면에 출력
- ❖ 변수 이름 앞이나 뒤에 ?를 입력하면 변수에 대한 정보를 출력
- ❖ 함수 이름 뒤에 ??를 입력하면 함수의 소스 코드 출력

A screenshot of an IPython terminal window titled "C:\ IPython: C:\Users\Park". The window has a black background with green and red text. It shows the execution of three commands: 1. "In [25]: ar = [1,2,3]" which assigns a list to the variable 'ar'. 2. "In [26]: ar?" which triggers IPython's help system, displaying the type ('list'), string form ('[1, 2, 3]'), length ('3'), and docstring for the 'list' object. 3. "In [27]:" which is an empty prompt line. The window includes standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
C:\ IPython: C:\Users\Park

In [25]: ar = [1,2,3]

In [26]: ar?
Type:      list
String form: [1, 2, 3]
Length:    3
Docstring:
list() -> new empty list
list(iterable) -> new list initialized from iterable's items

In [27]:
```

# ipython

## ❖ 작업 디렉토리 확인

In [1]: `import os`

In [2]: `os.getcwd()`

Out[2]: 'C:\Users\WWPark'

## ❖ 작업 디렉토리 변경

In [4]: `os.chdir('c:\python')`

## ❖ 단축키

Ctrl+P 또는 위 화살표 키: 명령어 히스토리를 역순으로 검색하기

Ctrl+N 또는 아래 화살표 키: 명령어 히스토리에서 최근 순으로 검색하기

Ctrl+R: readline 명령어 형식의 히스토리 검색(부분 매칭)하기

Ctrl+Shift+V: 클립보드에서 텍스트 붙여넣기

Ctrl+C: 현재 실행 중인 코드 중단하기

Ctrl+A: 커서의 줄의 처음으로 이동하기

Ctrl+E: 커서의 줄의 끝으로 이동하기

Ctrl+K: 커서가 놓은 곳부터 줄의 끝까지 텍스트 삭제하기

Ctrl+U: 현재 입력된 모든 텍스트 지우기

Ctrl+F: 커서를 앞으로 한 글자씩 이동하기

Ctrl+B: 커서를 뒤로 한 글자씩 이동하기

Ctrl+L: 화면 지우기

# Python

## 대화식으로 파이썬 실행

1. IDLE를 실행
  2. 아래 코드를 입력하고 Enter
- ```
>>> print("Hello Python")
```

파이썬 코드 입력 시 주의할 점은 반드시 첫 번째 열부터 입력해야 합니다.

들여쓰기를 하면 에러가 발생합니다.

IDLE을 종료할 때는 Ctrl + D



# 파이썬 쉘로 코딩하기

- . 직접 입력하고 출력해도 된다
- . 이것은 변수안에 어떤 데이터가 있는지 확인할 때 많이 사용
- . 숫자는 그냥, 문자는 "인용부호 사용

```
>>> print("hello world");  
hello world
```

```
>>> 123  
123
```

```
>>> "안녕하세요"  
'안녕하세요'
```

```
>>> '안녕하세요'  
'안녕하세요'
```

```
>>>
```

# 1. Python

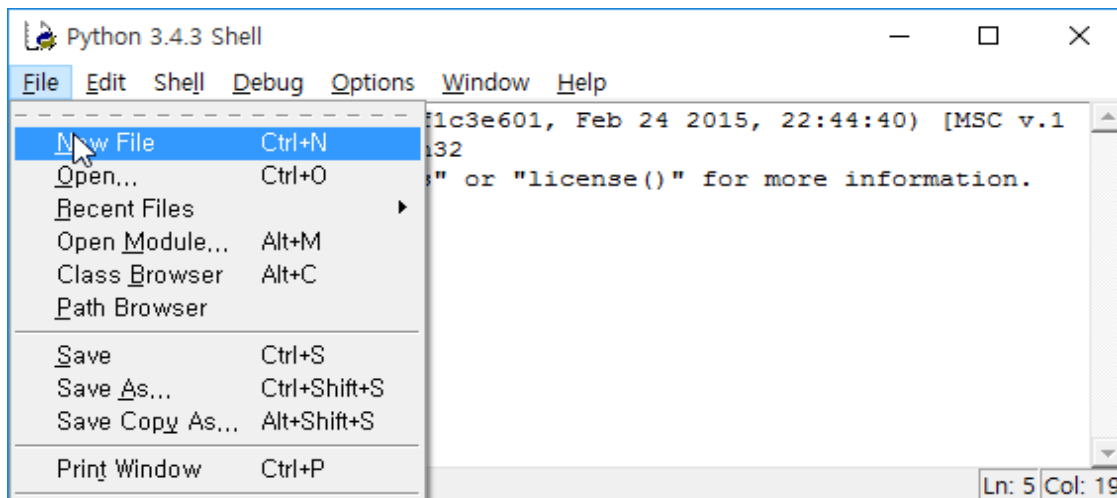
## 파일로 저장하고 파이썬 실행

1. IDLE를 실행
2. [File] – [New File]을 선택
3. 코드를 작성하고 [File] – [Save] : 파일명의 확장자는 py  
`print('File Save')`  
`print('Python is Fun')`
4. [Run] – [Run Module] 또는 콘솔 창에서 저장된 곳으로 이동한 후  
python 파일명

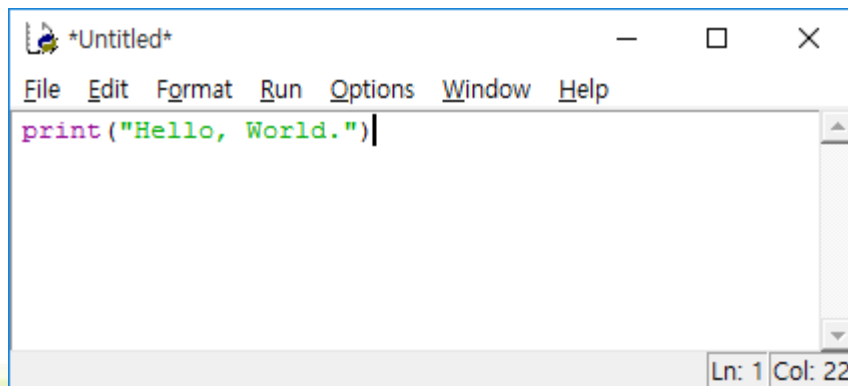
# 코드 편집기로 코딩하기

## ❖ 코드 편집기로 코딩하기 - 1

- IDLE 창에서 [File] → [New File] 메뉴 항목을 클릭하여 코드 편집기 실행



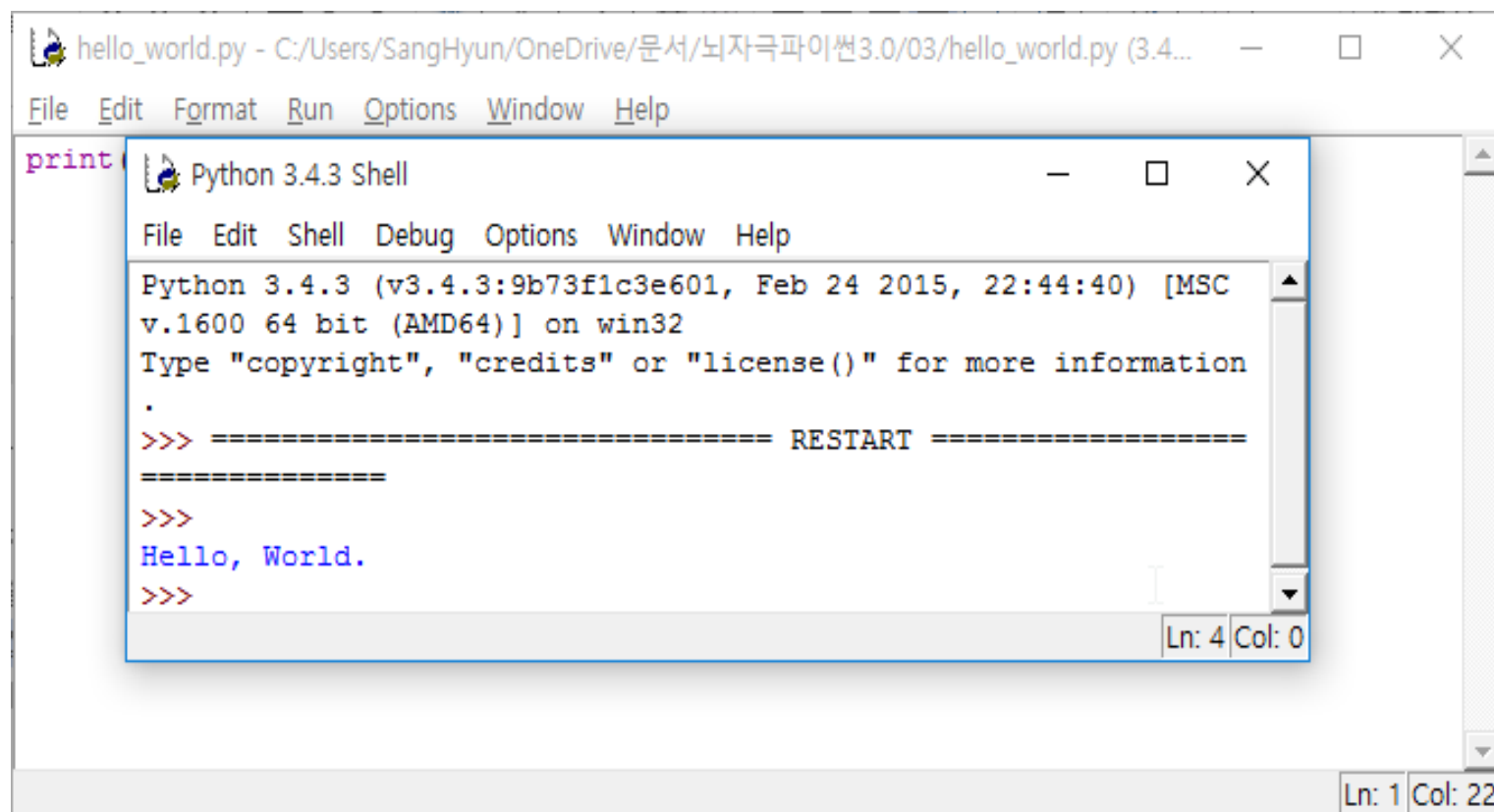
- 다음과 같이 코드를 입력하고 [Ctrl]+[S]키를 눌러서 파일 저장.
- 파일 이름은 hello\_world.py



# 코드 편집기로 코딩하기

## ❖ 코드 편집기로 코딩하기 - 2

- 코드 편집기가 활성화된 상태에서 [F5] 버튼을 클릭



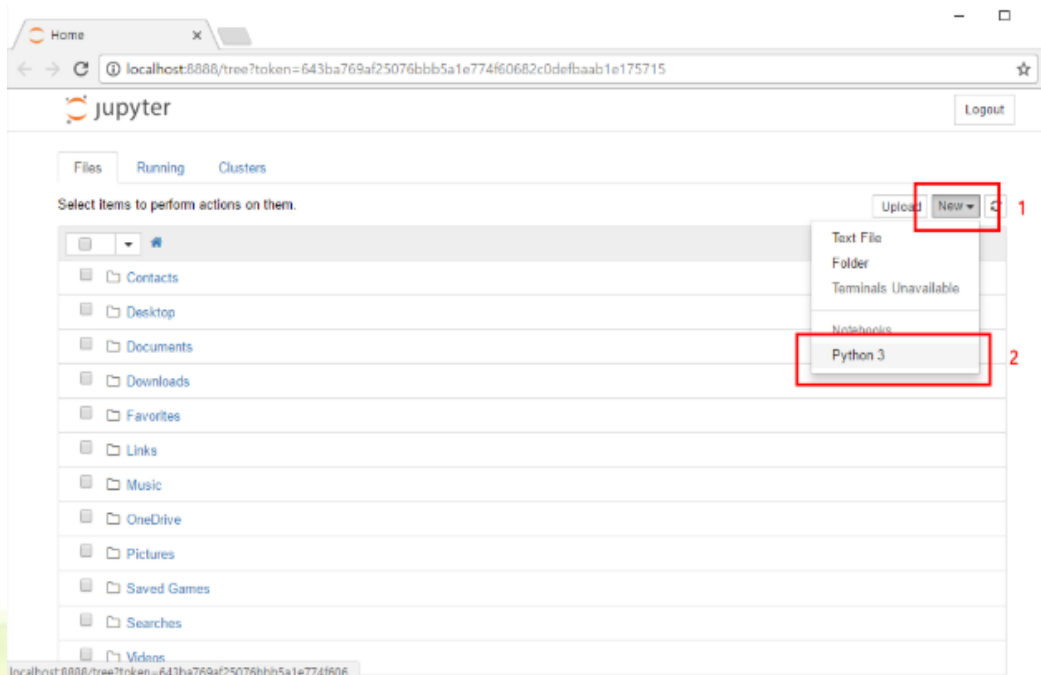
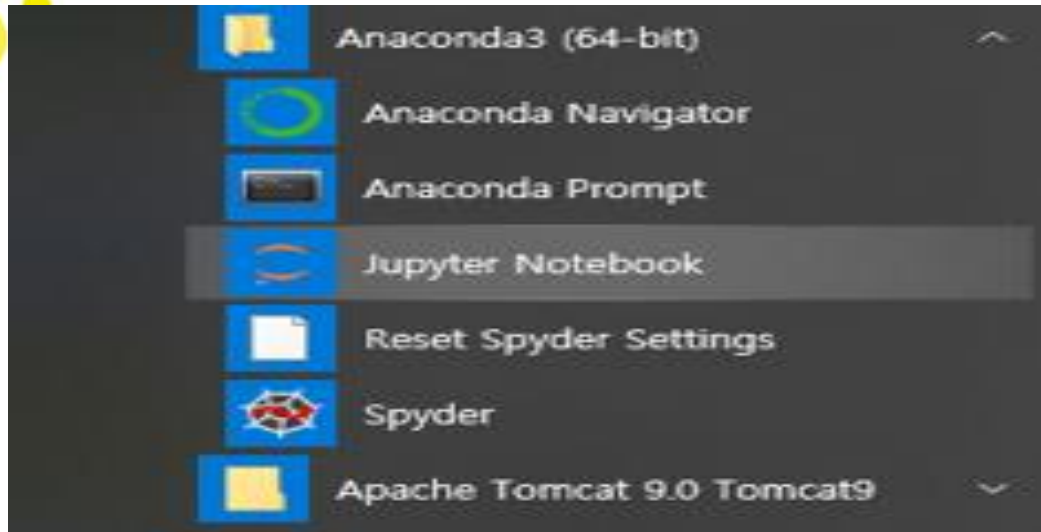
The screenshot shows a Python IDE window titled 'hello\_world.py - C:/Users/SangHyun/OneDrive/문서/뇌자극파이썬3.0/03/hello\_world.py (3.4...'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor displays the line `print`. Overlaid on the editor is a 'Python 3.4.3 Shell' window. The shell window's menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell output shows the Python version and architecture: 'Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32'. It prompts the user to type 'copyright', 'credits', or 'license()' for more information. A separator line with 'RESTART' in the center is shown. Below this, the prompt `>>>` is followed by the output 'Hello, World.' and another `>>>` prompt. The shell window's status bar at the bottom right shows 'Ln: 4 Col: 0'. The main IDE window's status bar at the bottom right shows 'Ln: 1 Col: 22'.

```
hello_world.py - C:/Users/SangHyun/OneDrive/문서/뇌자극파이썬3.0/03/hello_world.py (3.4...
File Edit Format Run Options Window Help
print

Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC
v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information
.
>>> ===== RESTART =====
>>>
Hello, World.
>>>
Ln: 4 Col: 0

Ln: 1 Col: 22
```

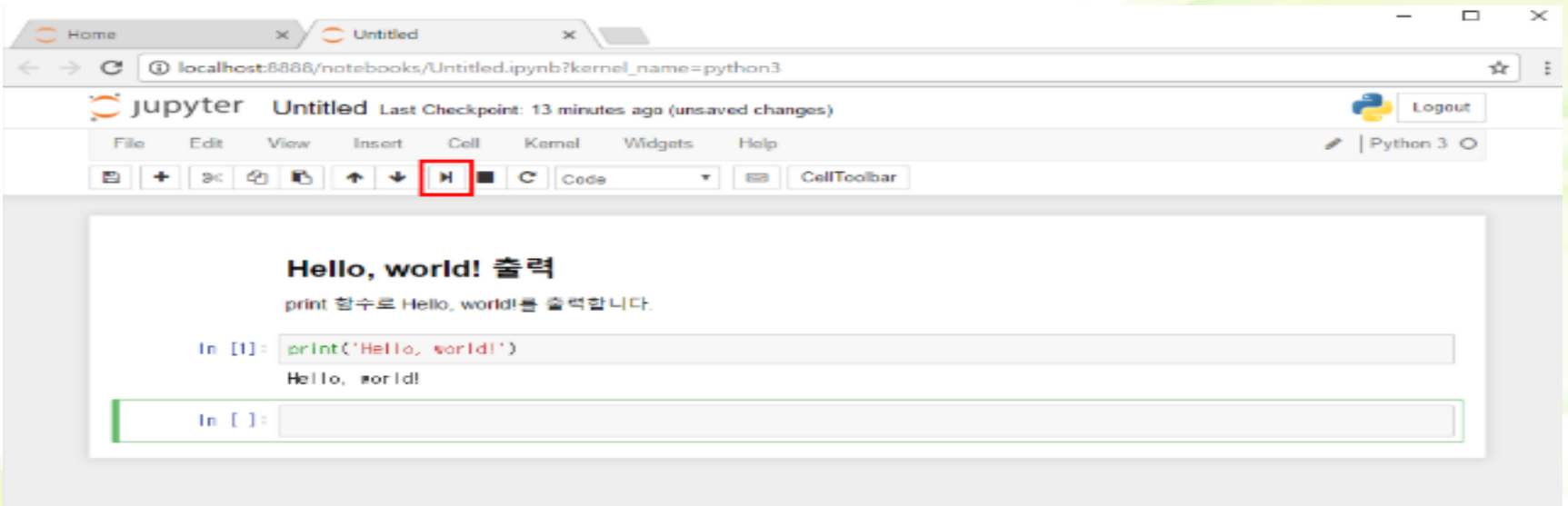
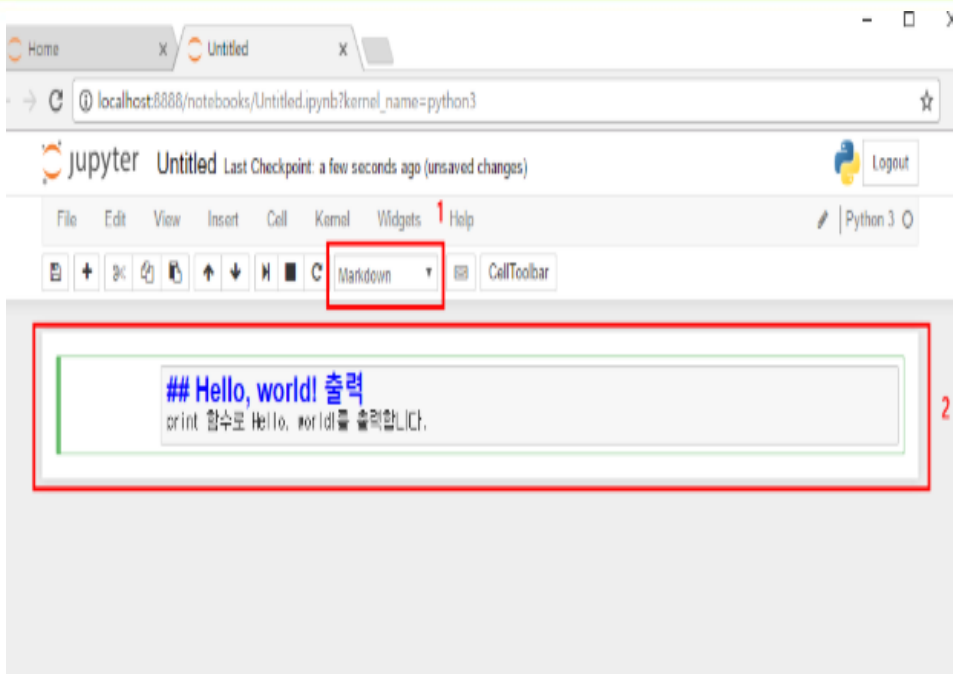
# Jupyter notebook



- 시작 > Anaconda3 (64-bit) > Jupyter Notebook을 클릭합니다. 또는, 명령 프롬프트를 실행(윈도우 키+R을 누른 뒤 cmd를 입력)한 뒤 다음 명령을 입력합니다.
- C:\Users\<사용자계정>\Anaconda3\python.exe -m notebook
- 명령을 실행하면 웹 브라우저에 주피터 노트북이 표시됩니다. 이 화면에서 파이썬 노트북을 만들어보겠습니다. 오른쪽 New 버튼을 클릭한 뒤 Python 3을 클릭합니다.

# Jupyter notebook

- 노트북의 제목을 바꾸려면 맨 위 Jupyter 로고 옆의 Untitled 를 클릭하면 됩니다. 그리고 메뉴의 File > Save and Checkpoint를 클릭하면 노트북이 파일로 저장됩니다. 이번 실습에서는 노트북의 제목을 바꾸지 않았으므로 C:\Users ₩<사용자계정> 폴더에 Untitled.ipynb 파일로 저장됩니다(노트북이름.ipynb 형식)



# 파이썬 기초 문법 따라 해보기

## 사칙연산

- 1 **더하기(+)** 2는 3이라는 값을 출력해 보자. 보통 계산기 사용하듯 더하기 기호만 넣어 주면 된다.
- `>>> 1 + 2` 3 **나눗셈(/)**과 **곱셈(\*)** 역시 예상한 대로 결과값을 보여준다.
- `>>> 3 / 2.4` 1.25 `>>> 3 * 9` 27 우리가 일반적으로 알고 있는 ÷ 기호나 × 기호가 아닌 것에 주의하자.

## 변수에 숫자 대입하고 계산하기

- `>>> a = 1` `>>> b = 2` `>>> a + b` 3 a에 1을, b에 2를 대입한 다음 a와 b를 더하면 3이라는 결과값을 보여 준다.

## 변수에 문자 대입하고 출력하기

- `>>> a = "Python"` `>>> print(a)` Python a 변수에 Python이라는 값을 대입한 다음 `print(a)`라고 작성하면 a 값을 출력한다.
- ※ 파이썬은 대소문자를 구별한다. `print`를 `PRINT`로 쓰면 정의되지 않았다는 오류 메시지가 나온다.
- 또는 다음과 같이 `print`문을 생략하고 변수 이름 a만 입력하여 a의 값을 출력할 수도 있다.
- `>>> a = "Python"`

```
>>> a
```

```
Python
```



# 파이썬 기초 문법 따라 해보기

```
>>> for a in [1, 2, 3]:
```

```
... print(a)
```

```
...
```

```
1 2 3
```

for문을 사용하면 실행해야 할 문장을 여러 번 반복해서 실행시킬 수 있다. 위 예는 대괄호([ ])사이에 있는 값들을 하나씩 출력한다. 위 코드의 의미는 "[1, 2, 3] 리스트의 앞에서부터 하나씩 꺼내어 a 변수에 대입한 후 print(a)를 수행하라"이다. 당연히 a에 차례로 1, 2, 3이라는 값이 대입되며 print(a)에 의해서 그 값이 차례대로 출력된다.

## 반복문 while

- 다음은 **while**을 사용하는 예이다.

```
>>> i = 0
```

```
>>> while i < 3:
```

```
... i=i+1
```

```
... print(i)
```

```
...
```

1 2 3 while이라는 영어 단어는 "~인 동안"이란 뜻이다. for문과 마찬가지로 반복해서 문장을 수행할 수 있도록 해준다. 위 예제는 i 값이 3보다 작은 동안 i=i+1과 print(i)를 수행하라는 말이다. i=i+1이라는 문장은 i의 값을 1씩 더하게 한다. i 값이 3보다 커지게 되면 while문을 빠져나가게 된다.

# 구글 코랩 사용법

구글 colaboratory 서비스, 줄여서 코랩(colab)이라고 부르는 웹 서비스는 주피터 노트북을 구글 서버에서 가동시키고 사용자가 마음대로 사용할 수 있도록 무료로 제공한다. 이 서비스를 이용하려면 구글 지메일(gmail) 계정을 가지고 있어야 한다. 지메일 계정이 있다면 다음과 같이 코랩을 사용할 수 있다.

코랩 노트북은 다음과 같은 장점이 있다.

별도의 파이썬 설치 없이 웹 브라우저 만을 이용해 주피터 노트북과 같은 작업을 할 수 있다.

다른 사용자와 공유가 쉬워 연구 및 교육용으로 많이 사용된다.

Tensorflow, keras, matplotlib, scikit-learn, pandas 등 데이터 분석에 많이 사용되는 패키지들이 미리 설치되어 있다.

무료로 GPU를 사용할 수 있다.

구글 독스나 구글 스프레드시트 등과 같은 식으로 공유와 편집이 가능하다. 만약 두 명 이상의 사람이 동시에 같은 파일을 수정 하더라도 변경사항이 모든 사람에게 즉시 표시된다.

## 구글 코랩 시작하기

구글 코랩을 사용하려면 지메일 계정이 필요하다. 지메일 계정이 있으면 로그인을 하고 지메일 계정이 없다면 다음 웹사이트에서 지메일을 만든 다음에 로그인한다.

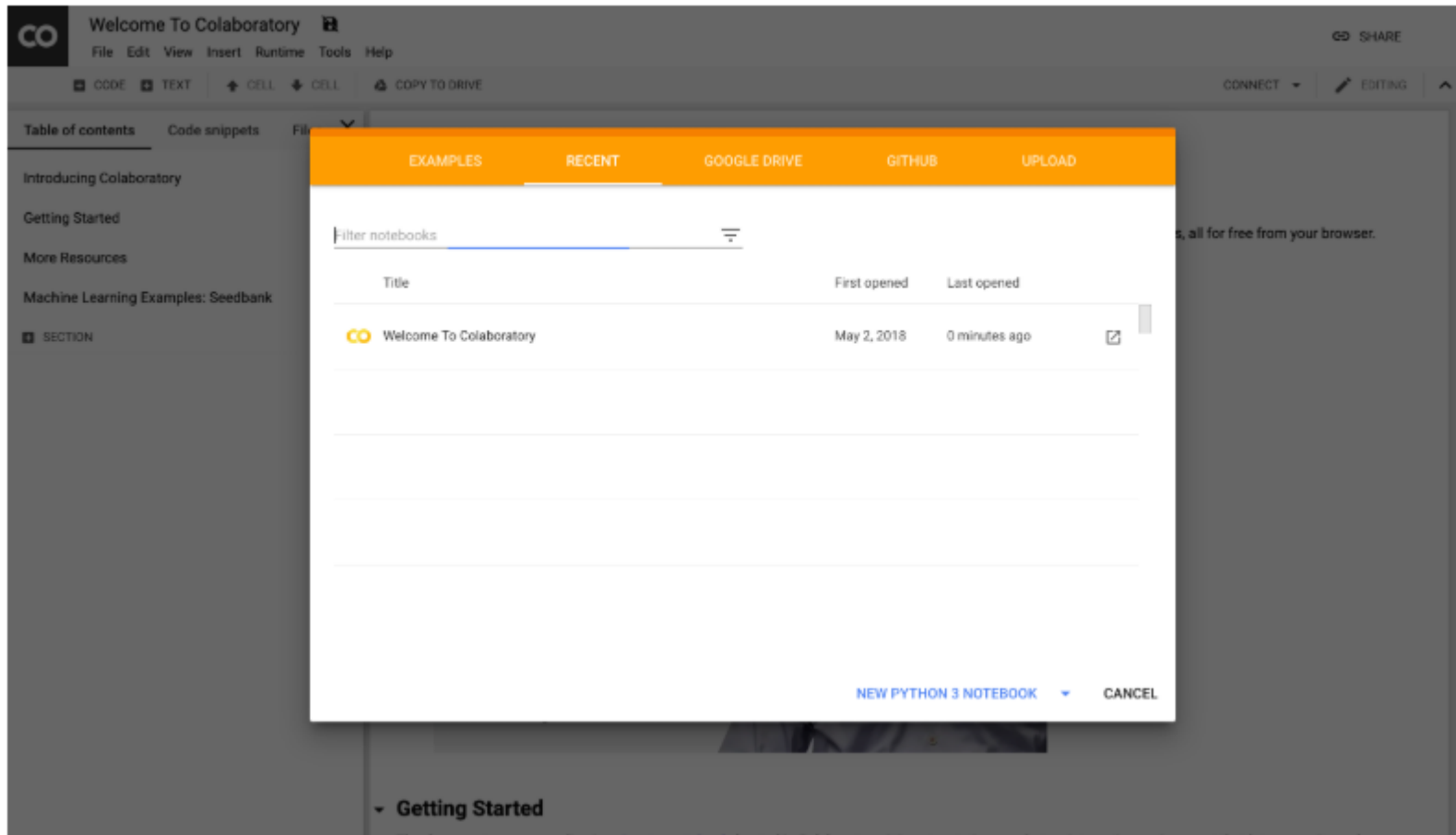
# 구글 코랩 사용법

<https://www.google.com/gmail/>

로그인을 한 후에는 웹 브라우저를 열고 다음 웹 페이지 주소로 접속한다.

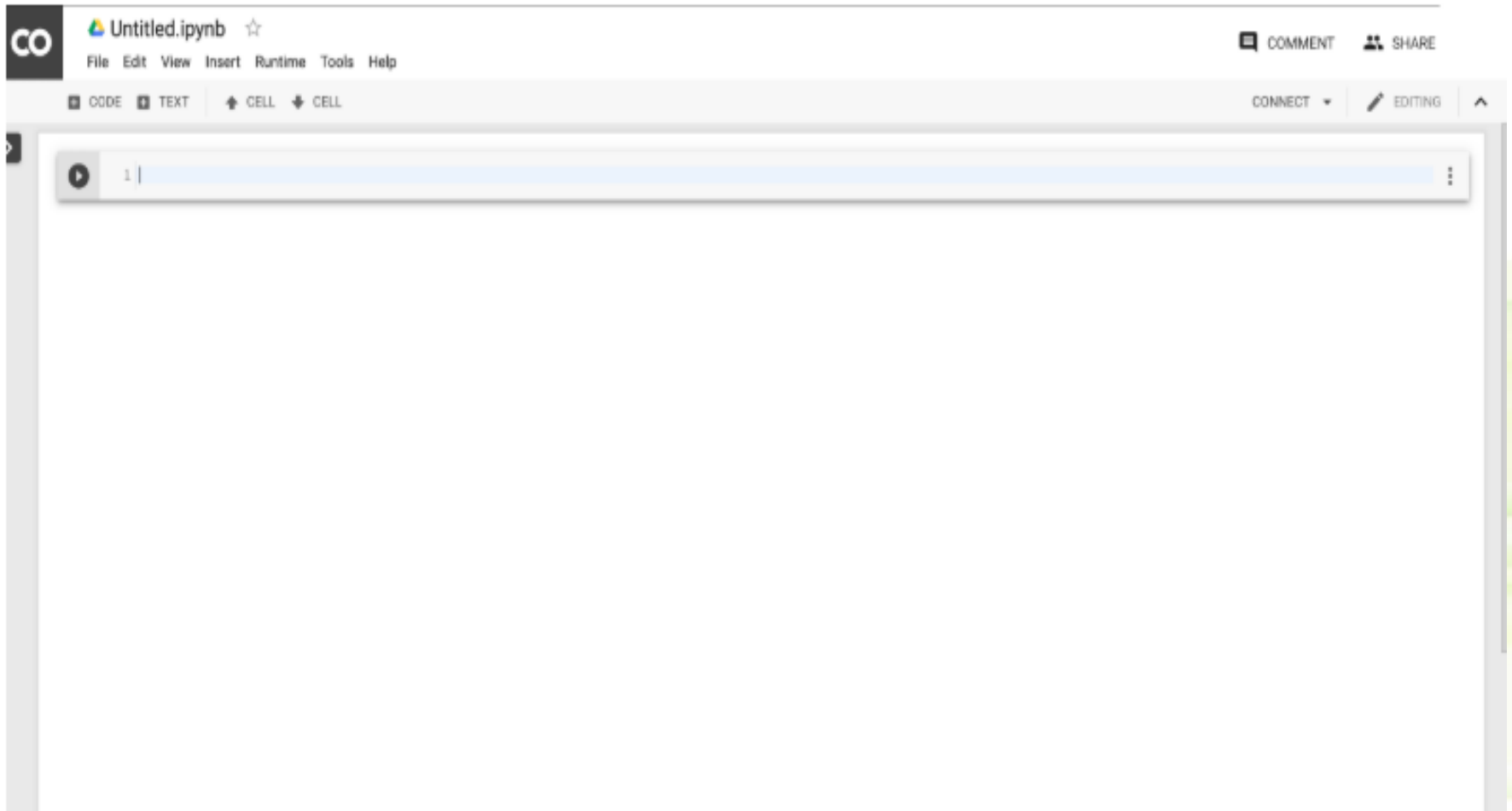
<http://colab.research.google.com>

그러면 다음 그림과 같은 시작 페이지가 나온다.



# 구글 코랩 사용법

메뉴에서 '파일' > '새 Python 3 노트' 명령을 선택하거나 다이얼로그 아래의 "새 PYTHON 3 노트" 버튼을 누르면 다음 그림처럼 새 노트북을 사용할 수 있다. 사용법은 주피터 노트북과 같다



# 파이썬 패키지 설치하기

## 라이브러리와 패키지

집을 지을 때, 시멘트나 철근이 필요하다고 해서 집을 짓는 사람이 시멘트 제작이나 철광석 제련부터 할 수는 없다. 부품이나 재료는 이미 만들어진 것을 사서 쓰는 것이 효율적이다. 프로그램도 마찬가지로 필요한 모든 부분을 한 사람이 만드는 경우는 거의 없으며 여러가지 프로그램에 공통적으로 사용되는 부분은 해당 분야의 전문가가 미리 만들어 놓은 프로그램을 가져다 사용하는 것이 현명한 선택이다. 이처럼 다른 프로그램 제작에 사용하기 위해 미리 만들어진 프로그램의 집합을 라이브러리(library)라고 한다. 파이썬에서는 라이브러리라는 용어 대신 패키지(package)라는 말을 주로 사용한다.

## 패키지 설치

일부 패키지는 파이썬을 설치할 때 함께 설치되는데 이러한 패키지를 표준 패키지(standard package)라고 한다. 그러나 대부분의 패키지는 파이썬을 설치한 후에 별도로 설치한다. 아나콘다 배포판을 설치하면 다양한 패키지들을 추가로 기본 설치해준다. 그러나 그 외에 추가로 패키지를 더 사용해야 할 때는 사용자가 수동으로 설치해야 한다.

C, C++, 자바 같은 언어는 새로운 라이브러리를 설치할 때 그 라이브러리를 제공하는 웹사이트를 일일이 방문하여 소스코드나 바이너리를 내려 받아야 한다. 반면 파이썬은 모든 패키지를 한 서버에 등록해서 간단한 명령어만으로 패키지를 쉽게 내려 받아 설치할 수 있다..

# 파이썬 패키지 설치하기

- 기본적으로 모든 파이썬 패키지는 PyPI 서버(<https://pypi.org/>) 라는 공식 패키지 서버에 등록된다. 하지만 아나콘다 배포판의 경우에는 PyPI가 아닌 독자적인 패키지 서버(<https://anaconda.org/>)를 운영하고 있으며 운영체제별로 미리 컴파일된 패키지를 제공하는 대신 PyPI보다는 제공하는 패키지의 수가 적다

## 패키지 관리자

패키지를 설치하기 위한 프로그램을 패키지 관리자라고 한다. 파이썬에서는 pip라고 하는 패키지 관리자를 기본으로 제공하지만, 아나콘다 배포판에서는 conda라고 하는 아나콘다 전용 패키지 관리자도 제공한다. conda는 아나콘다 배포판이 설치될 때 자동으로 함께 설치된다.

## pip를 사용하여 설치된 패키지를 확인하기

ip 패키지 관리자를 사용할 때는 pip list 명령으로 설치된 패키지 목록을 볼 수 있다.

```
$ pip list
```

| Package        | Version |
|----------------|---------|
| -----          |         |
| absl-py        | 0.7.1   |
| alabaster      | 0.7.12  |
| anaconda-clean | 1.0     |
| ... (이하 생략)    |         |

# 파이썬 패키지 설치하기

pip install 패키지이름

예를 들어 quandl이라는 이름의 패키지를 설치하려면 pip install quandl

~\$ pip install quandl

Collecting quandl

Downloading https://

...(중략)...

Successfully uninstalled more-itertools-7.0.0

Successfully installed inflection-0.3.1 more-itertools-5.0.0 ndg-httpsclient-0.5.1  
quandl-3.4.8

dockeruser@7f62df393a47:~\$

대부분의 패키지는 내부적으로는 다른 패키지의 기능을 사용한다. 예를 들어 만약 패키지 A가 패키지 B의 기능을 사용하고 있다면 패키지 B가 설치되어 있을 때만 패키지 A를 사용할 수 있다. 이때 패키지 A가 패키지 B에 의존(depend)한다고 말한다. pip로 패키지를 설치하면 그 패키지가 의존하는 다른 패키지도 자동으로 설치한다. 위에서 quandl 패키지를 설치하였더니 more-itertools, ndg-httpsclient를 같이 설치한 것을 볼 수 있다.



# 파이썬 패키지 설치하기

## conda를 이용하여 패키지를 설치하기

아나콘다 배포판에서 제공하는 conda 패키지 관리자는 의존성 관리 성능이 pip보다 뛰어나다. 설치하거나 제거할 패키지간의 의존성을 파악하고 만약 설치하고자 하는 패키지가 의존하고 있는 다른 패키지가 설치되어 있지 않거나 버전이 맞지 않으면 필요한 패키지를 설치해 주는 것 이외에도 기존의 패키지를 자동으로 업그레이드하거나 다운그레이드해 준다.

conda install 패키지이름

예를 들어 flake8이라는 이름의 패키지를 설치하려면 다음과 같이 conda install flake8이라고 명령한다.

중간에 Proceed ([y]/n)?라고 진행여부를 물어보면 y 키를 누르면 된다.

```
$ conda install flake8
```

```
Collecting package metadata: done
```

```
environment location: /home/dockeruser/anaconda3
```

| package      | build  |        |
|--------------|--------|--------|
| conda-4.6.14 | py37_0 | 2.1 MB |
| flake8-3.7.7 | py37_0 | 131 KB |
| Total:       |        | 2.2 MB |

# 파이썬 패키지 설치하기

The following NEW packages will be INSTALLED:

flake8 pkgs/main/linux-64::flake8-3.7.7-py37\_0

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates conda-forge::ca-certificates-2019.3.9~ --> pkgs/main::ca-certificates-2019.1.23-0

conda conda-forge --> pkgs/main

openssl conda-forge::openssl-1.1.1b-h14c3975\_1 -->  
pkgs/main::openssl-1.1.1b-h7b6447c\_1

Proceed ([y]/n)?

Downloading and Extracting Packages

flake8-3.7.7 | 131 KB |  
##### | 100%

conda-4.6.14 | 2.1 MB |  
##### | 100%

Preparing transaction: done

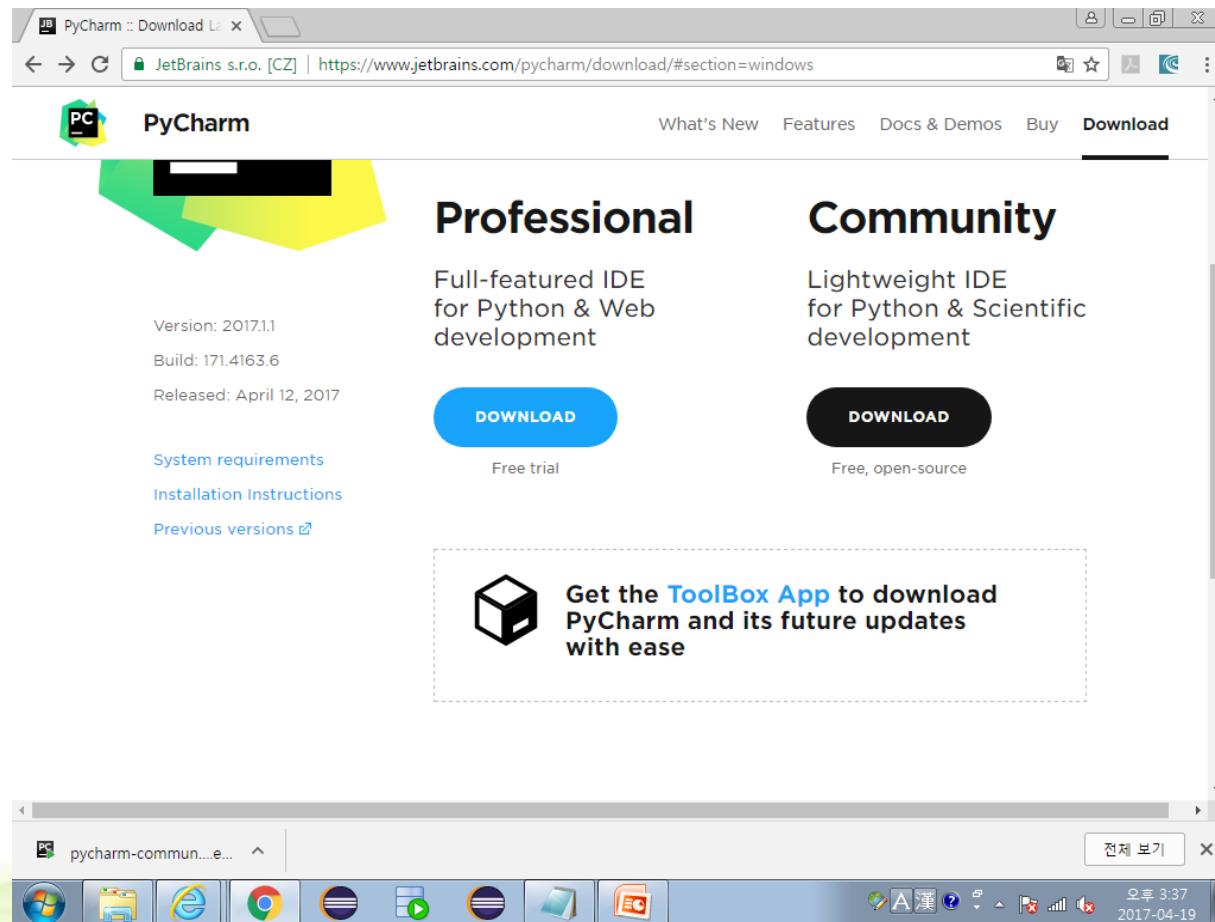
Verifying transaction: done

Executing transaction: done

conda 패키지 관리자가 flake8 설치에 필요한 다른 패키지까지도 자동으로 설치하거나 업그레이드 해 주는 것을 볼 수 있다.

# Pycharm설치

- ❖ Pycharm은 IntelliJ, CLion 등을 제공하는 JetBrains에서 배포하는 프로그램
- ❖ <http://www.jetbrains.com/pycharm/download/#section=windows> 에서 Community 버전을 선택해서 다운로드 후 설치



# PyCham 개발 환경 구축

## 1) 줄번호 표시

Configure - Settings - IDE Settings 에서

Editor - General - Appearance - Show line numbers에 체크

## 2) Fonts 설정

Configure - Settings - IDE Settings 에서

Editor - Color & Fonts - Font 선택

Save As 버튼 클릭해서 새로운 Schema name 입력(f)후 폰트 size설정

## 3) 단축키 변경 (eclipse 단축키로 변경)

Configure - Settings - IDE Settings 에서

Keymap - 우측화면에서 Keymas : eclipse 선택

## 4) Auto Import

Configure - Settings - IDE Settings 에서

Editor - General - Auto Import

Show Import popup

Optimize imports on the fly

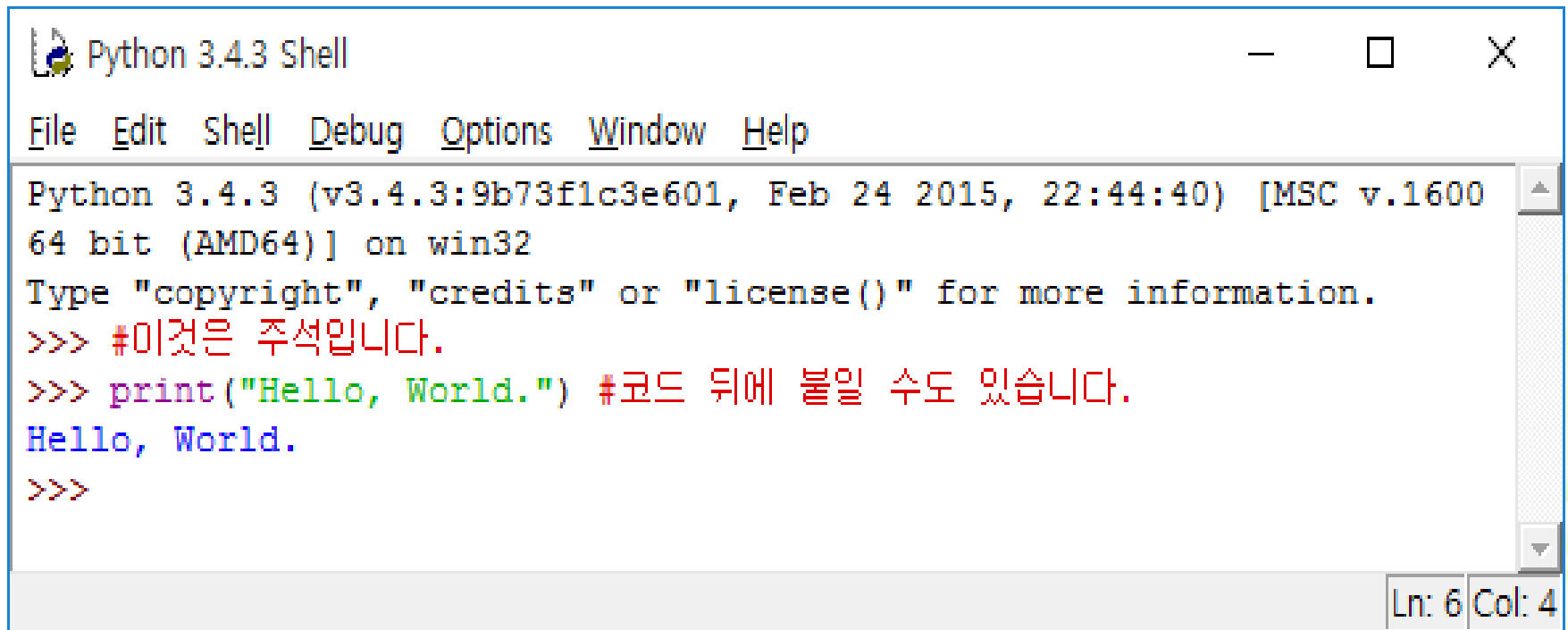
Add unambiguous imports on the fly

# 파이썬과 다른 언어의 다른 차이점

- ✓ 구조적 프로그래밍 언어의 대부분은 특정한 부분을 구분 짓기 위해서 특정한 기호를 사용하여 그 시작과 끝을 표시하는 엄격한 구조를 따라야만 했는데 파이썬은 이러한 명령어가 없습니다.
- ✓ 파이썬에서는 코드의 구조를 정의하기 위해 기호 대신에 들여쓰기를 사용
- ✓ Java 나 C와 같은 언어에서 코드 블록을 여닫기 위하여 사용하는 중괄호 대신 파이썬에서는 공백을 사용하며 이는 코드를 읽기 쉽게하고 코드 내에서 불필요한 기호의 사용을 줄여줍니다.
- ✓ 코드의 정렬과 구성이 엄격하게 제한되는데, 네 칸를 들여쓰는 것이 표준이기는 해도 프로그래머가 들여쓰기의 규칙을 정할 수 있습니다.
- ✓ 하나의 문장을 종료할 때 종료기호는 없음
- ✓ 하나의 줄에 2개 이상의 명령어를 사용할 때 명령어를 구분하기 위한 용도로 ;을 사용

# 소스 코드에 설명 달기 : #

- ❖ 파이썬에서 기호 #은 주석(註釋 : Comment)을 뜻함
- ❖ 주석 : 프로그래밍에서는 소스 코드를 쉽게 풀이한 글이라는 뜻



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> #이것은 주석입니다.
>>> print("Hello, World.") #코드 뒤에 붙일 수도 있습니다.
Hello, World.
>>>
```

Ln: 6 Col: 4

# Python Basic

## 파이썬과 다른 언어의 또 다른 차이점은 코딩 구조

1. 구조적 프로그래밍 언어의 대부분은 특정한 부분을 구분 짓기 위해서 특정한 기호를 사용하여 그 시작과 끝을 표시하는 엄격한 구조를 따라야만 했습니다.
2. 파이썬에서는 코드의 구조를 정의하기 위해 기호 대신에 들여쓰기를 사용
3. 자바 나 C와 같은 언어에서 코드 블록을 여닫기 위하여 사용하는 중괄호 대신 파이썬에서는 공백을 사용하며 이는 코드를 읽기 쉽게하고 코드 내에서 불필요한 기호의 사용을 줄여줍니다.
4. 그러한 규칙으로 인해 코드의 정렬과 구성이 엄격하게 제한되는데, 네 칸를 들여쓰는 것이 표준이기는 해도 프로그래머가 들여쓰기의 규칙을 정할 수 있습니다.

# Python Basic

## 자바코드

```
int x = 100;
if(x > 0){
    System.out.println("x가 0보다 큼니다.");
} else {
    System.out.println("x가 0보다 작습니다.");
}
```

## 파이썬 코드

```
x = 100
if x > 0:
    print('x가 0보다 큼니다.')
else:
    print('x가 0보다 크지 않습니다.')
```



# Python Basic

❖ 도움말을 실행하고자 하는 경우에는 [F1] 키를 눌러도 되고 IDLE 창에서 help(도움말을 얻고자 하는 명령)

❖ Python의 예약어

아래처럼 작성하면 예약어 확인 가능

```
>>> import keyword
```

```
>>> keyword.kwlist
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue',  
'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',  
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return',  
'try', 'while', 'with', 'yield']
```

# Python Basic

1. 자료형(Data Type): 데이터를 저장하는 방법
2. 파이썬의 자료형

## 저장방법에 따른 분류

- Direct 형: 직접 데이터를 표현하는 자료형
- Sequence 형: 다른 데이터를 포함하는 자료형으로 순서가 있는 자료형
- Mapping형: 다른 데이터를 포함하는 자료형으로 키와 값으로 구분해서 저장하는 자료형
- Set형: 순서가 없고 자료의 중복이 없는 자료형

## 변경 가능성에 따른 분류

- 변경 가능(Mutable): list, dict, set
- 변경 불가능형(Immutable): int, float, complex, str, tuple, frozenset

## 데이터 저장 개수에 따른 분류

- 하나의 객체만 저장: str, bytes, bytearray, int, float, complex
- 여러 개의 객체를 저장: list, tuple, dict, set, frozenset

# Python Basic

## 상수(Constant, Literal):값을 변경할 수 없는 데이터

### 1. Number

- 정수:10진수(107), 8진수(0o107), 16진수(0x107, 0X107), 2진수(0b101), long(107L)
- 실수:1.2, 0.12e1, 0.12E1 -> 주의:지수부는 정수일 수 있으나 실수 일수는 없음
- 복소수: 허수부 뒤에 j 나 J 사용 4+5j, 7-2j

### 2. String:' 문자열' 또는 "문자열" 의 형태로 기재하며 여러 줄의 문자열을 하나의 문자열로 만들 때는 ''' 문자열''' 또는 """"문자열""""

### 3. List:[데이터 나열]

### 4. Tuple:(데이터 나열)

### 5. Dictionary:{키:값...}

### 6. bool 상수:True(1), False(0)

### 7. 제어문자:\n(줄바꿈), \t(탭)

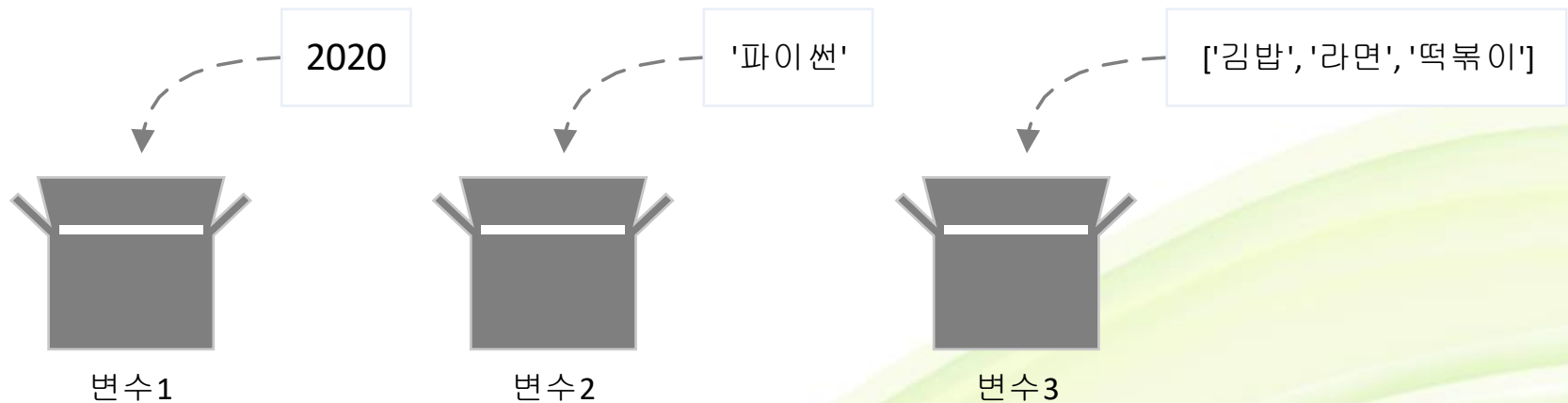
## 2. Python Basic

1. 변수(Variable): 데이터를 저장할 수 있는 공간에 붙인 이름
2. Python에서는 변수를 만들 때 자료형을 기재하지 않아도 됩니다.
3. 변수에 값을 할당 할 때 선언은 자동으로 이루어집니다.
4. 등호 (=)는 변수에 값을 할당하는 데 사용됩니다.
5. 변수 명명 규칙
  - 1) 유니코드 문자나 밑줄로 시작
  - 2) 이름에 공백이 없어야 함
  - 3) 아스키코드의 특수문자는 사용 못함
  - 4) 예약어는 변수 이름으로 사용 못함
  - 5) 한글은 가능하지만 잘 사용하지 않음
6. 주석은 # 뒤에 기재하면 됩니다.

# 변수

## ❖ 변수란?

- 변수는 데이터를 담는 메모리 공간
- 변수에는 수, 텍스트, 목록, 이미지 데이터 등을 담을 수 있음.

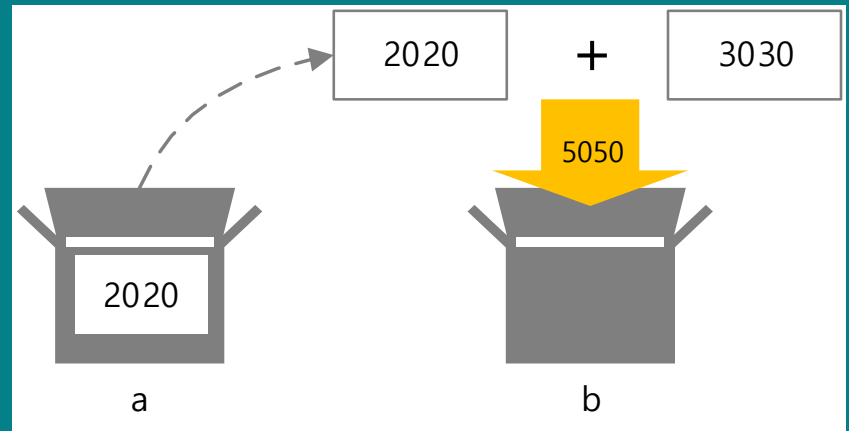


# 변수

## ❖ 실습 1

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:
40) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more in
formation.
>>> a = 2020
>>> a
2020
>>> b = a + 3030
>>> b
5050
>>> |
```

b = a + 3030 코드에서 파이썬은 a에 담겨있는 2020을 3030과 더한 후 변수 b에 저장

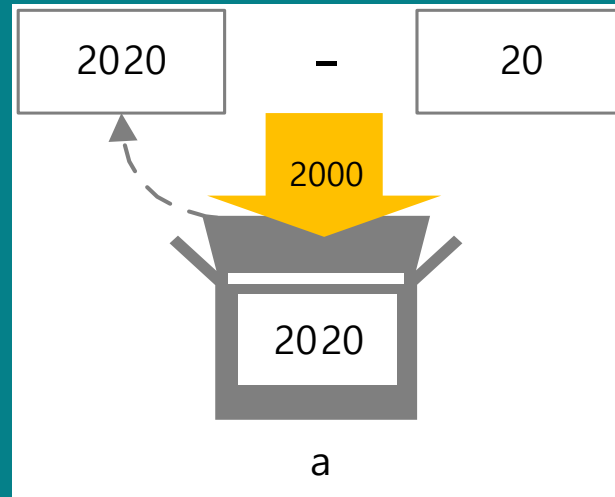


# 변수

## ❖ 실습 2

```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license()" for more in
formation.
>>> a = 2020
>>> a
2020
>>> b = a + 3030
>>> b
5050
>>> a = a - 20
>>> a
2000
>>> |
```

a에 담겨있는 2020에서 20을 뺀 값을 다시 a에 저장



# Python Basic

```
counter = 100      # 정수 할당
```

```
miles  = 1000.0    # 부동 소수점
```

```
name   = "홍길동"  # 문자열
```

```
print(counter)
```

```
print(miles)
```

```
print(name)
```



# Python Basic

1. 여러 줄을 하나의 문장으로 만들고자 할 때는 중간에 줄 연속 문자(w)를 추가해주면 됩니다.
  2. [], {}에 포함 된 문이나 () 괄호는 줄 연속 문자를 사용할 필요가 없습니다.
  3. 하나의 줄에서 문장을 구분하는 연산자는 ;
  4. +연산자: 문자열의 경우는 연결을 시키고 숫자의 경우는 덧셈
  5. \*연산자: 문자열의 경우는 반복을 나타내고 숫자의 경우는 곱셈
  6. \*\*연산자: 거듭제곱( $3^{**}3$  은 3의 3제곱)
  7. /연산자: 나눗셈을 한 결과
  8. //연산자: 몫을 정수로 구해줍니다.
- ❖ 여러 개를 하나의 값을 대입 가능  
 $a = b = 0$
  - ❖ 연속해서 여러 개의 변수에 값 할당 가능  
 $c, d = 3, 4$
  - ❖ 값 교환  
 $\text{변수1, 변수2} = \text{변수2, 변수1}$

# Python Basic

## ❖ 샘플 코드

```
>>> x=y=z=0
```

```
>>> x
```

```
0
```

```
>>> y
```

```
0
```

```
>>> z
```

```
0
```

```
>>> c,d=3,4
```

```
>>> c,d = d,c
```

```
>>> c
```

```
4
```

```
>>> d
```

```
3
```

```
>>>
```