

# Bigdata Analysis for Management

## Ch02. Hello Python!

---

맹 윤 호

Yunho Maeng

Latest update : 2018-09-11

[yunho0130@gmail.com](mailto:yunho0130@gmail.com)

# Ch02. Hello Python!

---

# Anaconda Download & Install

<https://www.continuum.io/downloads>

- ✓ 데이터 분석에 필요한 많은 Package들을 기본으로 포함하고 있어서 의존성 관리에 용이함

The screenshot shows the official Anaconda download page at <https://www.continuum.io/downloads>. The page features a large green 'DOWNLOAD ANACONDA NOW' button on the left. On the right, there's a decorative graphic of green and blue geometric shapes. Below the graphic, three download links are shown: 'Download for Windows', 'Download for OSX' (which is currently selected), and 'Download for Linux'. The 'Download for Windows' section is highlighted with a red border. It contains a heading 'Anaconda 4.2.0 For Windows', a brief description about the BSD license, a 'Changelog' link, and a list of installation steps. It also includes two download buttons: a large green '64-BIT INSTALLER (391M)' and a smaller light green '32-BIT INSTALLER (333M)'. Further down, there's a note about older versions and long-term support. The overall background has a subtle geometric pattern.

Log In ↗ Support Contact

ANACONDA Powered by Continuum Analytics COMMUNITY CONSULTING TRAINING ABOUT RESOURCES

DOWNLOAD ANACONDA NOW

Discover What #OpenDataScienceMeans At Anaconda.com

GET SUPERPOWERS WITH ANACONDA

Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular Python, R and Scala packages for data science.

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](#) ↗

For older versions of Anaconda installers, see the [Anaconda installer archive](#) ↗  
For long-term support of the packages found in the Anaconda archives, please [contact us](#).

Python 3.5 version

64-BIT INSTALLER (391M)

32-BIT INSTALLER (333M)

Python 2.7 version

64-BIT INSTALLER (381M)

32-BIT INSTALLER (324M)

# Anaconda Download & Install

<https://www.continuum.io/downloads>

## ✓ 리눅스 용 설치방법

The screenshot shows the Anaconda download page for Linux. At the top, there are three download links: "Download for Windows", "Download for OSX", and "Download for Linux". The "Download for Linux" link is highlighted with a blue border. Below this, the heading "Anaconda 4.2.0" is displayed in green, followed by "For Linux". A note states: "Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution." Below this, there is a "Changelog" link. The main content area is divided into two sections based on Python version: "Python 3.5 version" and "Python 2.7 version". Each section contains two download links: "64-BIT INSTALLER (455M)" and "32-BIT INSTALLER (373M)". In the "Python 3.5 version" section, the "64-BIT INSTALLER" link is highlighted with a green background. In the "Python 2.7 version" section, the "64-BIT INSTALLER" link is highlighted with a blue background. The "Python 3.5 version" section also contains a red box around the "Python 3.5 version" heading and the "bash Anaconda3-4.2.0-Linux-x86\_64.sh" command. The "Python 2.7 version" section contains the "bash Anaconda2-4.2.0-Linux-x86\_64.sh" command. A note at the bottom says: "NOTE: Include the "bash" command even if you are not using the bash shell."

Download for Windows    Download for OSX    Download for Linux

**Anaconda 4.2.0**

**For Linux**

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. In your terminal window type one of the below and follow the instructions.

**Python 3.5 version**

`bash Anaconda3-4.2.0-Linux-x86_64.sh`

**Python 2.7 version**

`bash Anaconda2-4.2.0-Linux-x86_64.sh`

NOTE: Include the "bash" command even if you are not using the bash shell.

**Python 3.5 version**

**64-BIT INSTALLER (455M)**

**32-BIT INSTALLER (373M)**

**Python 2.7 version**

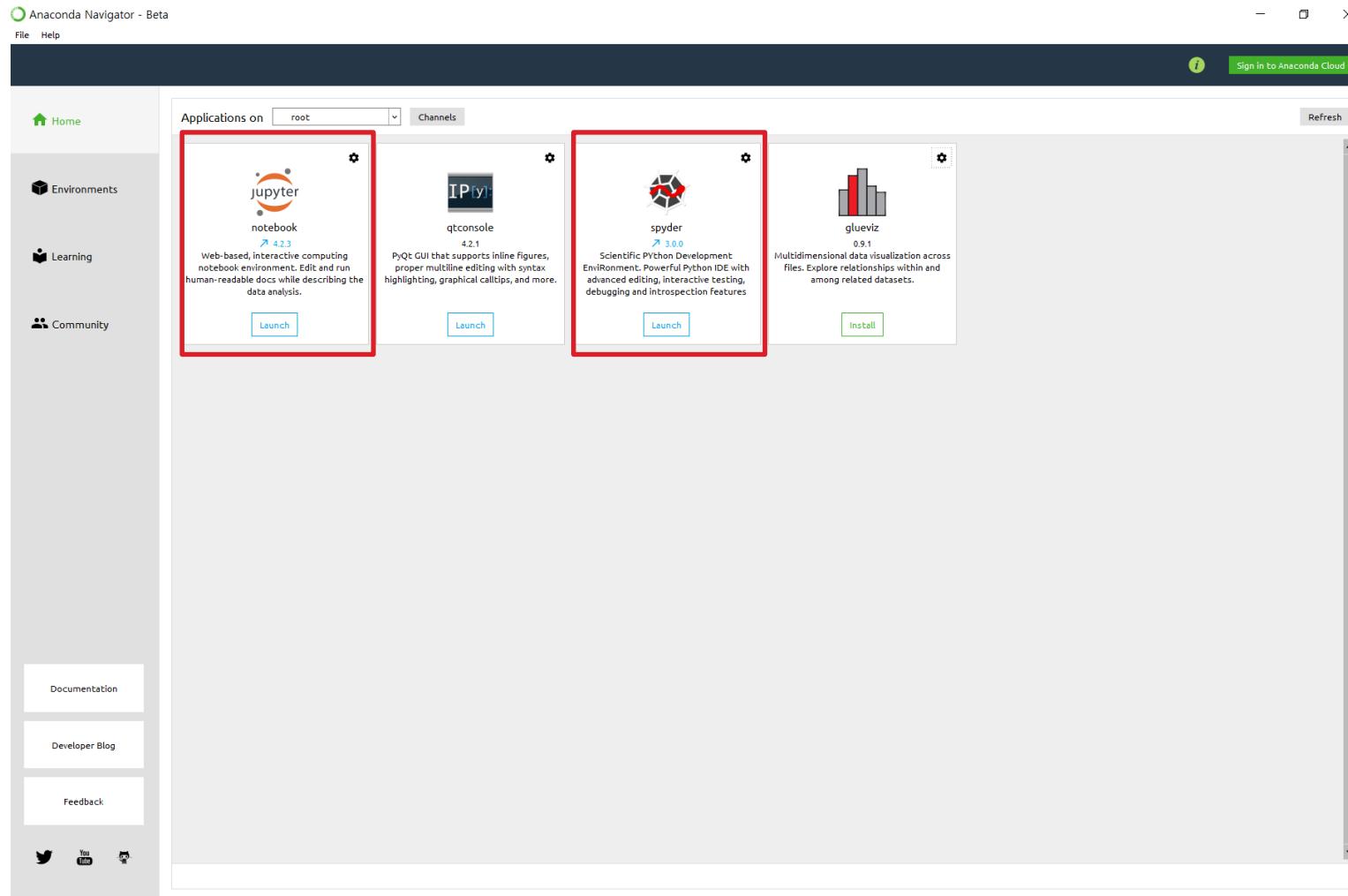
**64-BIT INSTALLER (446M)**

**32-BIT INSTALLER (365M)**

# Anaconda Download & Install

<https://www.continuum.io/downloads>

- ✓ 설치시, 환경변수를 자동으로 잡아주는 등 편리한 부분이 많음.



# Anaconda Download & Install

<https://www.continuum.io/downloads>

- ✓ Jupyter라는 Web기반 Notebook Interface를 지닌 커널 변경이 가능한 개발 환경을 함께 제공.(2,3 버전 함께 사용 가능)
- ✓ 문서에 코드를 바로 넣어 실행해 볼 수 있어서 많은 세미나에서 활용.
- ✓ 리눅스 기반 서버에 설치하여 다른 컴퓨터에서도 사용할 수도 있음 (Jupyter Hub)



```
In [ ]: # coding=utf-8
def ppap(item1='pen',item2='apple',item3='pineapple'):

    def song(item1):
        print 'I have a {}'.format(item1)
    def ah(item1, item2):
        print 'Ah~ {} {}'.format(item2, item1)
    def combine(item1, item2, item3):
        print '{}~ {}~ {}'.format(item2, item1, item3)

    song(item1) # I have a pen
    song(item2) # I have an apple
    ah(item1, item2) # Ah~ apple pen
    song(item1) # I have a pen
    song(item3) # I have a pineapple
    ah(item1, item3) # Ah~ pineapple pen
    combine(item1, item2, item3) # apple pen~ pineapple pen~ pen pineapple apple pen

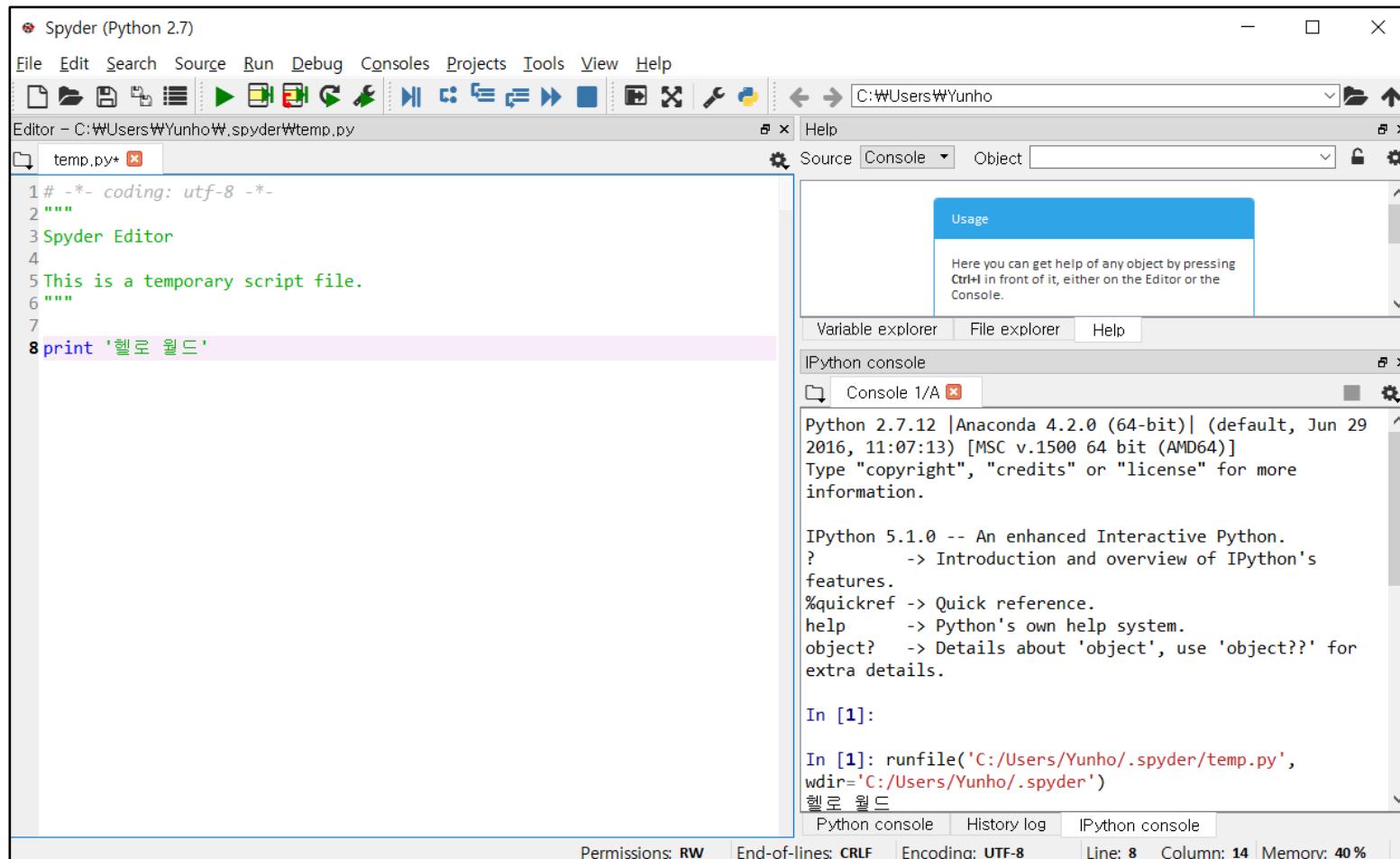
input_val_1 = raw_input("원하는 과일을 입력해 주세요: ")
input_val_2 = raw_input("원하는 두 번째 과일을 입력해주세요: ")

ppap(item2=input_val_1, item3=input_val_2)
```

# Anaconda Download & Install

<https://www.continuum.io/downloads>

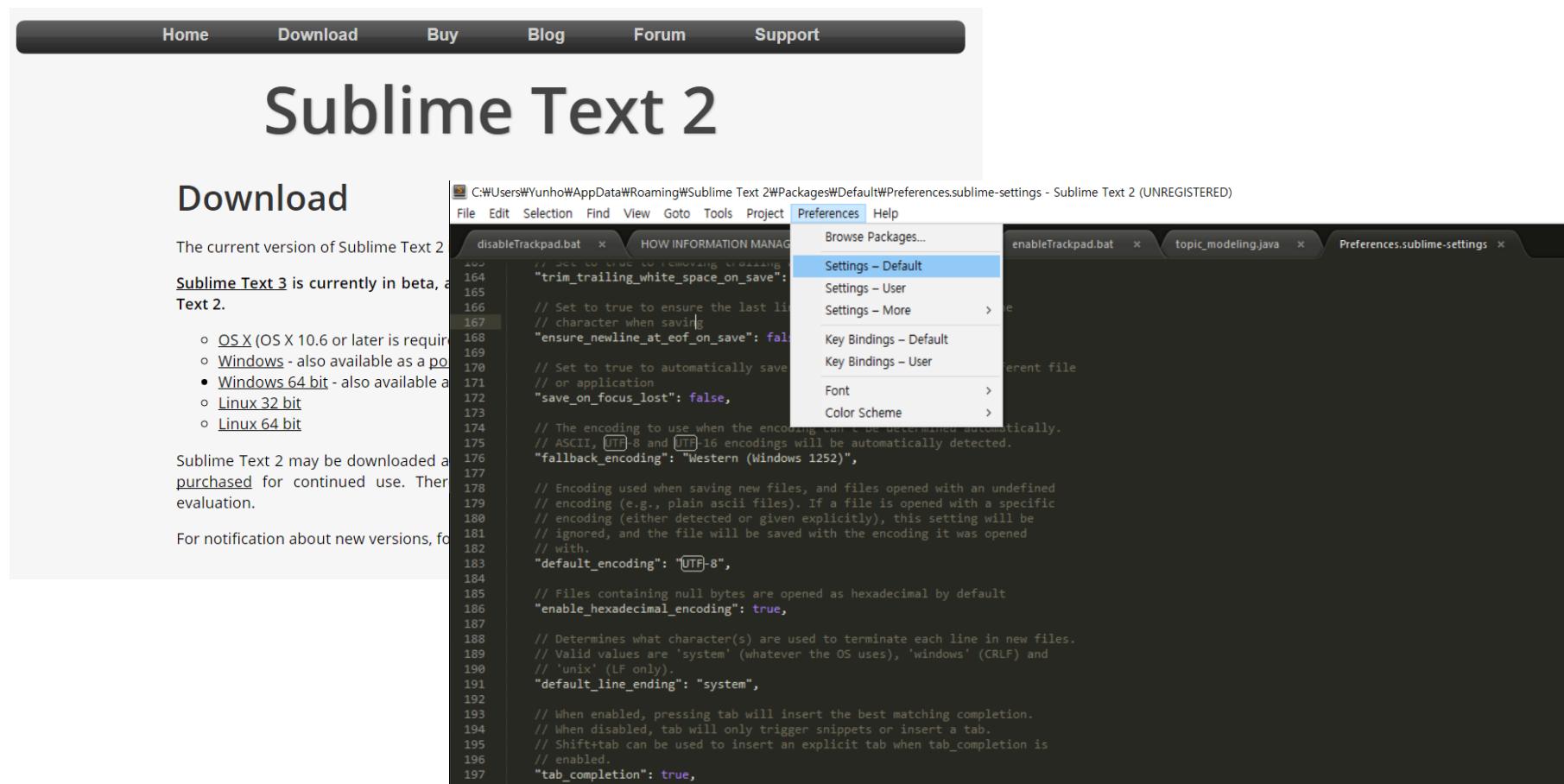
- ✓ Spyder라는 통합 개발 환경(IDE)을 지원



# SublimeText 2

<https://sublimetext.com/2>

- ✓ 대부분의 프로그래밍 언어의 문법 강조 기능을 지원하기 때문에 가벼운 코드 Editor를 급하게 사용할 때 편리함. (IDE vs Code Editor)
- ✓ 설치 이후, 한글 지원을 위해 기본 설정 변경 필요. [Setting]-[default\_encoding: UTF-8]



The screenshot shows the Sublime Text 2 interface. At the top, there is a navigation bar with links: Home, Download, Buy, Blog, Forum, and Support. Below the navigation bar is the main title "Sublime Text 2". On the left, there is a "Download" section with information about the current version (Sublime Text 2) and download links for various operating systems. The right side of the screen shows a code editor window displaying a file named "Preferences.sublime-settings". A context menu is open over this file, with the "Settings - Default" option highlighted. The menu also includes options for "Browse Packages...", "Settings - User", "Settings - More", "Key Bindings - Default", "Key Bindings - User", "Font", and "Color Scheme". The code in the editor window is as follows:

```
193 // Set to true to ensure the last line character when saving.
194 "trim_trailing_white_space_on_save": false
195
196 // Set to true to automatically save
197 // or application
198 "ensure_newline_at_eof_on_save": false
199
200 // The encoding to use when the encoding can't be determined automatically.
201 // ASCII, UTF-8 and UTF-16 encodings will be automatically detected.
202 "fallback_encoding": "Western (Windows 1252)",
203
204 // Encoding used when saving new files, and files opened with an undefined
205 // encoding (e.g., plain ascii files). If a file is opened with a specific
206 // encoding (either detected or given explicitly), this setting will be
207 // ignored, and the file will be saved with the encoding it was opened
208 // with.
209 "default_encoding": "UTF-8",
210
211 // Files containing null bytes are opened as hexadecimal by default
212 "enable_hexadecimal_encoding": true,
213
214 // Determines what character(s) are used to terminate each line in new files.
215 // Valid values are 'system' (whatever the OS uses), 'windows' (CRLF) and
216 // 'unix' (LF only).
217 "default_line-ending": "system",
218
219 // When enabled, pressing tab will insert the best matching completion.
220 // When disabled, tab will only trigger snippets or insert a tab.
221 "tab_completion": true,
```

# Python 이란?

## 공식 소개글

파이썬은 배우기 쉽고, 강력한 프로그래밍 언어입니다. 파이썬은 효율적인 고수준 데이터 구조를 갖추고 있으며, 간단하지만 효과적인 객체 지향 프로그래밍 접근법 또한 갖추고 있습니다. 우아한 문법과 동적 타이핑, 그리고 인터프리팅 환경을 갖춘 파이썬은 다양한 분야, 다양한 플랫폼에서 사용될 수 있는 최적의 스크립팅, RAD(rapid application development - 빠른 프로그램 개발) 언어입니다

- ✓ 다른 개발 언어에 비해 배우기 쉽도록 설계되었음.  
(Learning Curve)



### File I/O in Java:

```
// get current directory  
File dir = new File(".");  
File fin=new File(dir.getCanonicalPath()  
+ File.separator + "Code.txt");  
  
FileInputStream fis =  
    new FileInputStream(fin);  
  
//Construct the BufferedReader object  
BufferedReader in = new BufferedReader  
(new InputStreamReader(fis));  
  
String aLine = null;  
while ((aLine = in.readLine()) != null)  
{//Process each line, here we count  
    empty lines  
    if (aLine.trim().length() == 0) {  
    }  
}  
// do not forget to close the buffer  
reader  
in.close();
```



### File I/O in Python:

```
myFile = open("/home/xiaoran/Desktop/  
test.txt")  
  
print myFile.read();
```

### 그림출처

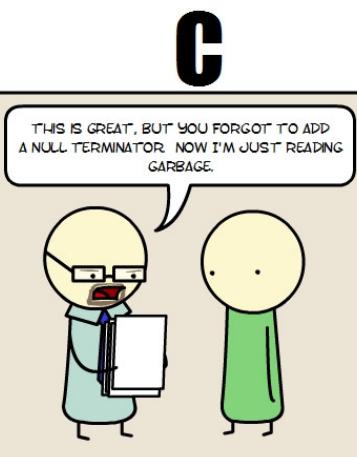
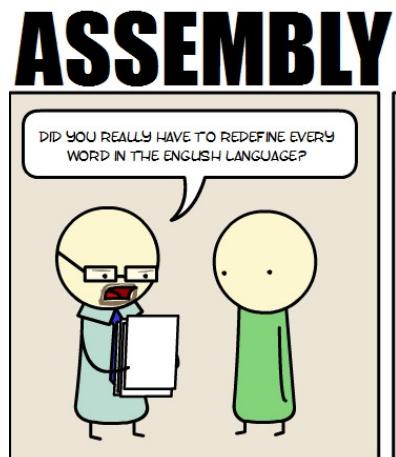
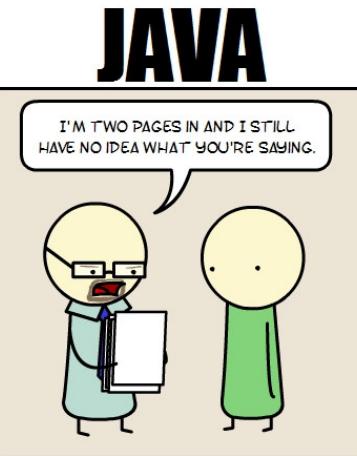
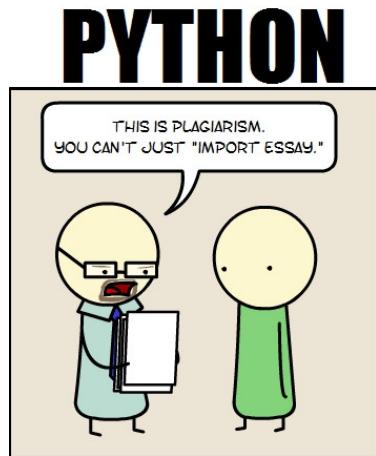
<http://blogs.perceptionsystem.com/infographic/java-vs-python-programming-language-productive/>

Life is too short, You need Python. 인생은 짧다, 그래서 파이썬이 필요하다.

# Python 이란?

## Awesome Python

파이썬에는 다양한 라이브러리, 프레임워크들이 존재하여 사용하기 간편함



`README.md`

### Awesome Python

A curated list of awesome Python frameworks, libraries, software and resources.

Inspired by [awesome-php](#).

- Awesome Python
  - Admin Panels
  - Algorithms and Design Patterns
  - Anti-spam
  - Asset Management
  - Audio
  - Authentication
  - Build Tools
  - Caching
  - ChatOps Tools
  - CMS 
  - Code Analyzers and Linter
  - Command-line Tools
  - Compatibility
  - Computer Vision
  - Concurrency and Parallelism
  - Configuration

**CMS** 

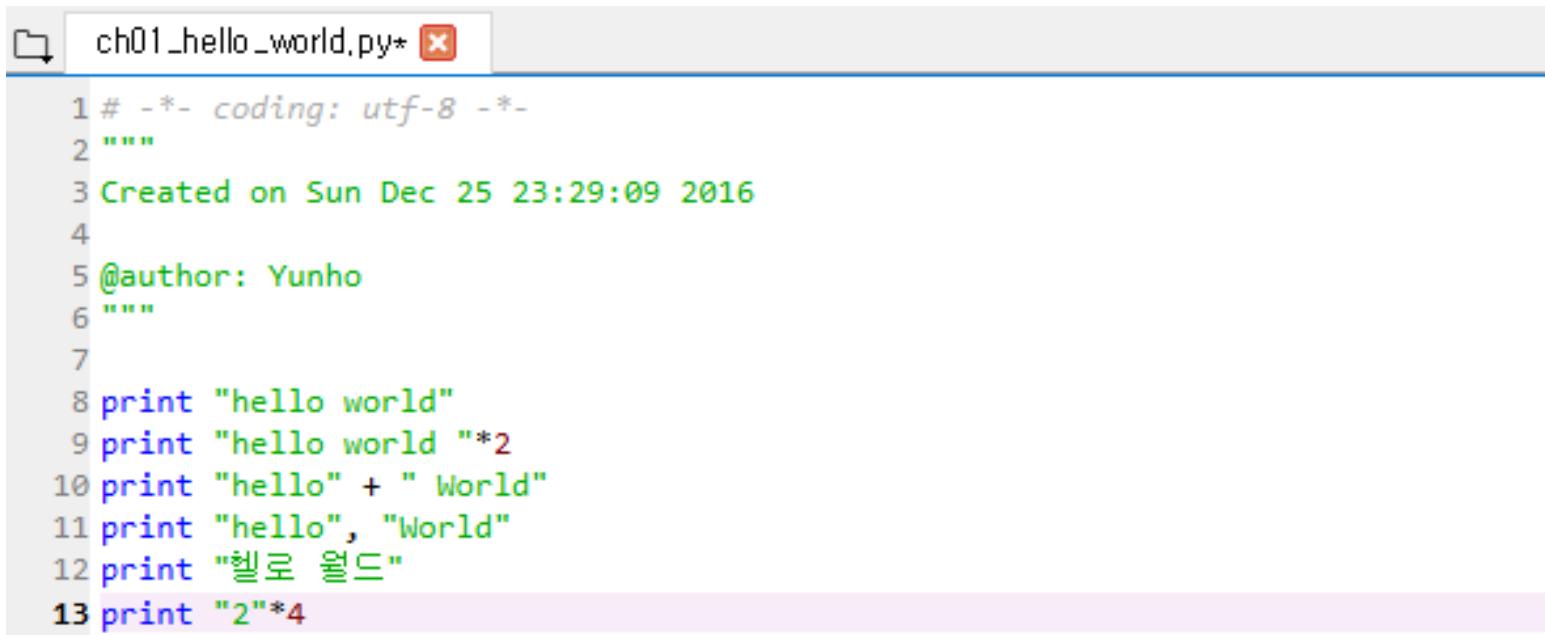
*Content Management Systems.*

- django-cms - An Open source enterprise CMS based on the Django.
- djedi-cms - A lightweight but yet powerful Django CMS with plugins, inline editing and performance in mind.
- FeinCMS - One of the most advanced Content Management Systems built on Django.
- Kotti - A high-level, Pythonic web application framework built on Pyramid.
- Mezzanine - A powerful, consistent, and flexible content management platform.
- Opps - A Django-based CMS for magazines, newspapers websites and portals with high-traffic.

## 코드 (ch02\_hello\_python.py)

매우 간단하게 문자열을 출력할 수 있는 것을 확인할 수 있음

- ✓ 기본적으로 문자열 곱셈과 덧셈이 가능
- ✓ 한글 출력을 위한 주석 (utf-8 vs cp949), 주석은 #을 앞에 붙이면 되고, Spyder에서는 Ctrl+1을 단축키로 사용
- ✓ 숫자의 문자 형태에 대한 이해 필요



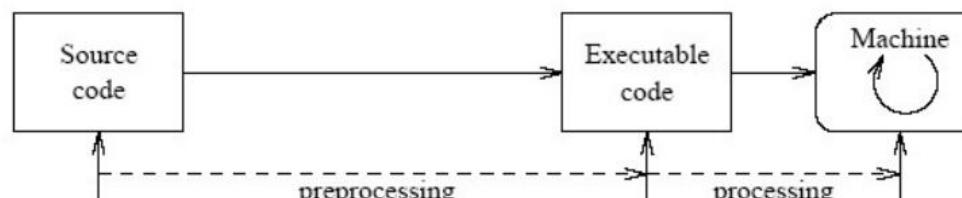
```
ch01_hello_world.py* [x]
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Dec 25 23:29:09 2016
4
5 @author: Yunho
6 """
7
8 print "hello world"
9 print "hello world "*2
10 print "hello" + " World"
11 print "hello", "World"
12 print "헬로 월드"
13 print "2"*4
```

# Python 이란?

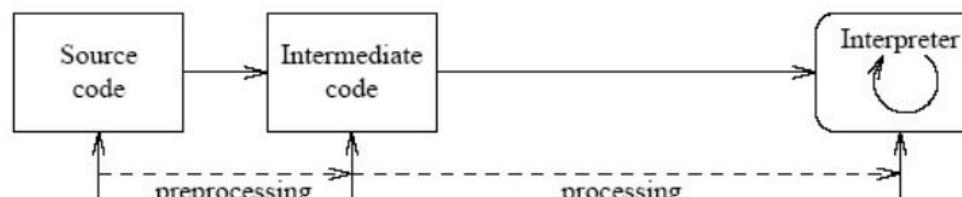
## 인터프리터 언어

컴파일 언어와는 다르게, 인터프리터가 **Line by line**으로 실행되는 타이밍에 코드를 실행하는 방식

- ✓ 파이썬에서는 바이트코드라는 중간 형태로 변환 뒤 인터프리터에서 실행
- ✓ 인터프리터 방식으로 인해 컴파일 언어에 비해 속도가 느린편. 많은 인터프리터 언어가 이를 극복하기 위해 JIT 컴파일러를 도입하고 있으나 Cpython의 경우 아직 도입되지 않은상황
- ✓ 고수준 언어 vs 저수준 언어



Compilation



Interpretation

그림 <http://www.circuitstoday.com/compilers-vs-interpreters-an-overview-of-the-differences>

# Python 이란?

## 동적타이핑(Dynamic Typing)

Python은 실행시간에 자료형을 검사하며, 동적으로 타입이 지정된다.

- ✓ Java나 C의 경우 변수선언 시점에 해당 자료형을 명시해주어야 하지만, Python은 그렇지 않음. 해당 변수에 어떤 값이 할당되느냐에 따라 동적으로 변할 수 있음
- ✓ Java에는 Generic이라는 기능이 이를 지원함 (단, 한 번 지정되면 Typecasting을 하지 않는 한 변하지 않음)

The image shows two side-by-side code snippets for file input/output. On the left, under 'File I/O in Java:', is Java code using BufferedReader and FileInputStream. On the right, under 'File I/O in Python:', is Python code using open(). Both snippets read from a file named 'test.txt' located on the desktop.

File I/O in Java:

```
// get current directory
File dir = new File(".");
File fin=new File(dir.getCanonicalPath()
    + File.separator + "Code.txt");

FileInputStream fis =
    new FileInputStream(fin);

//Construct the BufferedReader object
BufferedReader in = new BufferedReader
    (new InputStreamReader(fis));

String aLine = null;
while ((aLine = in.readLine()) != null)
{//Process each line, here we count
empty lines
    if (aLine.trim().length() == 0) {
    }
}
// do not forget to close the buffer
reader
in.close();
```

File I/O in Python:

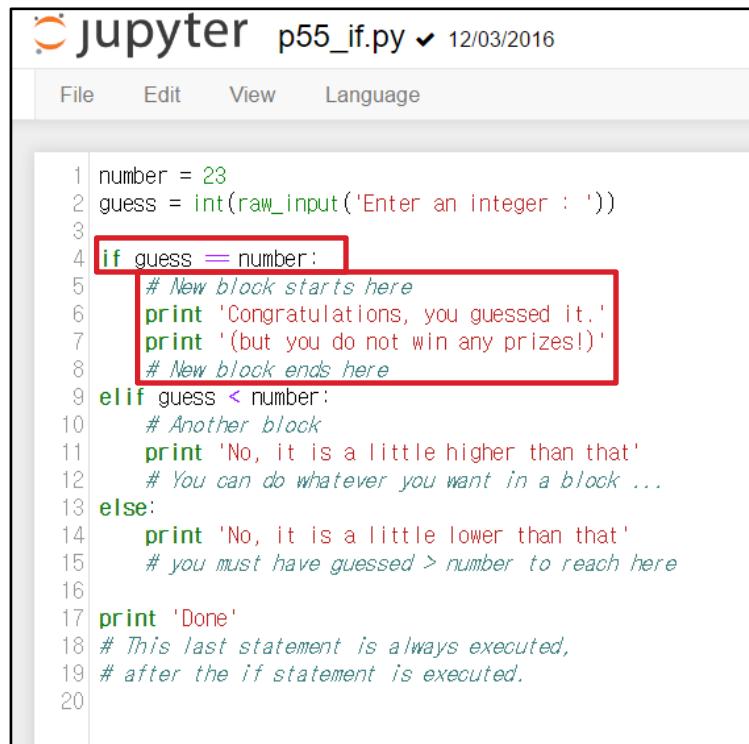
```
myFile = open("/home/xiaoran/Desktop/
test.txt")

print myFile.read();
```

# Python 이란?

## 우아한 문법을 위한 Coding Convention

- ✓ Coding Convention 이란, 프로그램을 작성할 때의 특정 스타일을 지칭하며, 협업과 가독성을 위해 하나의 일관성 있는 스타일을 사용하는 것을 지칭한다.
- ✓ Python은 Coding Block 구분을 들여쓰기(Indent)로 하며, 이는 선택사항이 아닌 필수임
- ✓ 들여쓰기는 Space 4개로 하는 것을 권장. IDE에서 Tab을 자동으로 바꿔주는 경우 있음



```

1 number = 23
2 guess = int(raw_input('Enter an integer : '))
3
4 if guess == number:
5     # New block starts here
6     print 'Congratulations, you guessed it.'
7     print '(but you do not win any prizes!)'
8     # New block ends here
9 elif guess < number:
10    # Another block
11    print 'No, it is a little higher than that'
12    # You can do whatever you want in a block ...
13 else:
14    print 'No, it is a little lower than that'
15    # you must have guessed > number to reach here
16
17 print 'Done'
18 # This last statement is always executed,
19 # after the if statement is executed.
20

```

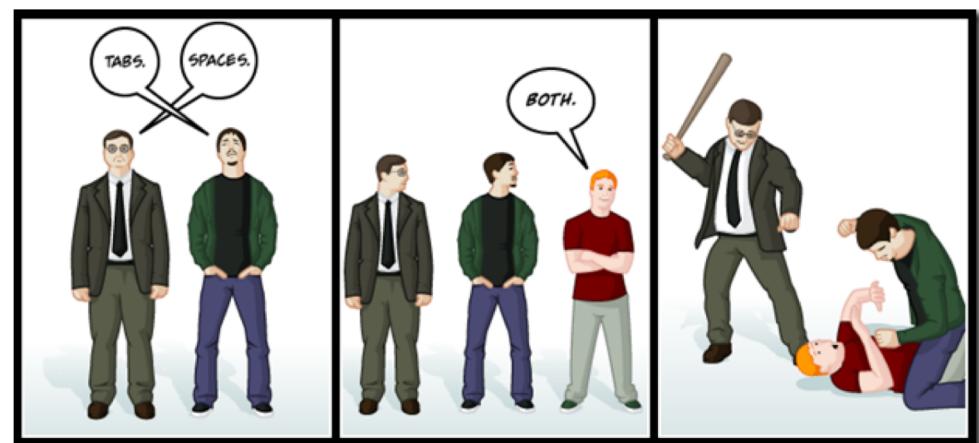


그림 참조: <https://til.chann.kr/python/coding-convention>

There are two types of people.

if (Condition)  
{  
 Statements  
 /\*  
 \*...  
 \*/  
}

if (Condition) {  
 Statements  
 /\*  
 ...  
 \*/  
}

WA 9GAG.COM

# Python Style Guidelines



## Python의 철학과 Google Styles Guidelines을 바탕으로 Coding Convention에 근거하여 강의할 예정

- ✓ Perl의 '하나 이상의 해결법이 존재한다' 와는 정반대로 '가장 아름다운 하나의 답이 존재한다'라는 명제를 모토로 하고 있다. 이 모토 하에 다음과 같은 철학을 지니게 되었다.
- ✓ The Zen of Python[1]
  - Beautiful is better than ugly (아름다운 것이 추한 것보다 낫다)
  - Explicit is better than implicit (명시적인 것이 암시적인 것보다 낫다)
  - Simple is better than complex (단순한 것이 복잡한 것보다 낫다)
  - Complex is better than complicated (내부적으로 복잡한 것이 외부적으로 복잡한 것보다 낫다)

**Google Python Style Guide**

Revision 2.59

Amit Patel  
Antoine Picard  
Eugene Jhong  
Jeremy Hylton  
Matt Smart  
Mike Shields

Each style point has a summary for which additional information is available by toggling the accompanying arrow button that looks this way: . You may toggle all summaries with the big arrow button:

Toggle all summaries

**Table of Contents**

Python Language Rules	<a href="#">Lint</a> <a href="#">Imports</a> <a href="#">Packages</a> <a href="#">Exceptions</a> <a href="#">Global variables</a> <a href="#">Nested/Local/Inner Classes and Functions</a> <a href="#">List Comprehensions</a> <a href="#">Default Iterators and Operators</a> <a href="#">Generators</a> <a href="#">Lambda Functions</a> <a href="#">Conditional Expressions</a> <a href="#">Default Argument Values</a> <a href="#">Properties</a> <a href="#">True/False evaluations</a> <a href="#">Deprecated Language Features</a> <a href="#">Lexical Scoping</a> <a href="#">Function and Method Decorators</a> <a href="#">Threading</a> <a href="#">Power Features</a>
Python Style Rules	<a href="#">Semicolons</a> <a href="#">Line length</a> <a href="#">Parentheses</a> <a href="#">Indentation</a> <a href="#">Blank Lines</a> <a href="#">Whitespace</a> <a href="#">Shebang Line</a> <a href="#">Comments</a> <a href="#">Classes</a> <a href="#">Strings</a> <a href="#">Files and Sockets</a> <a href="#">TODO Comments</a> <a href="#">Imports formatting</a> <a href="#">Statements</a> <a href="#">Access Control</a> <a href="#">Naming</a> <a href="#">Main</a>

[1] <https://github.com/chohankyun/python-django-study/wiki/00.-python-%EC%9D%B4%EB%9E%80>

[2] Google Python Style Guide <https://google.github.io/styleguide/pyguide.html>

## 코드 (ch02\_coding\_convention.py)

---

```
8 # semicolons
9 # No 두 줄을 한 줄로 만드는 ; 사용하지 말 것
10 print "hello"; print "world"
11
12 # Statements - 1줄에 1개의 statement가 원칙
13
14 # No 끝나는 부분 수직을 맞출 필요 없음.
15 str1 = 'hello'
16 str24 = 'world world'
17
18 # Yes
19 str1='hello'
20 str24='world world'
21
22 # Line Length
23
24 # No
25 print "hellohellohellohellohellohellohellohellohellohel
```

## 코드 (ch02\_coding\_convention.py)

---

```
27 # Yes1
28 str = ('This will build a very long long '
29       'long long long long long long string')
30 print str
31
32 # Yes2
33 if (width == 0 and height == 0 and
34     color == 'red' and emphasis == 'strong'):
35
36 # Yes3 - Long URLs은 길게 써도 무관
37 # See details at
38 # https://www.example.com/us/developer/documentation/api/content
39
40 # Block and Inline Comments
41 # No - 코드 끝에 주석을 다는 경우는 자제하라. 위에 주석을 첨부하라.
42 print "hello world" # This will build a very Long Long comments
43 # Yes - 공백 2칸단위, 짧은 주석
44 print "hello world" # Short comments
```

# Python Style Guidelines



Python의 철학과 Google Styles Guidelines을 바탕으로 Coding Convention에 근거하여 강의할 예정

✓ Naming

- 스네이크

module\_name, package\_name, method\_name, function\_name, global\_var\_name, instance\_var\_name, function\_parameter\_name, local\_var\_name.

- 카멜

ClassName, ExceptionName,

- 그 외

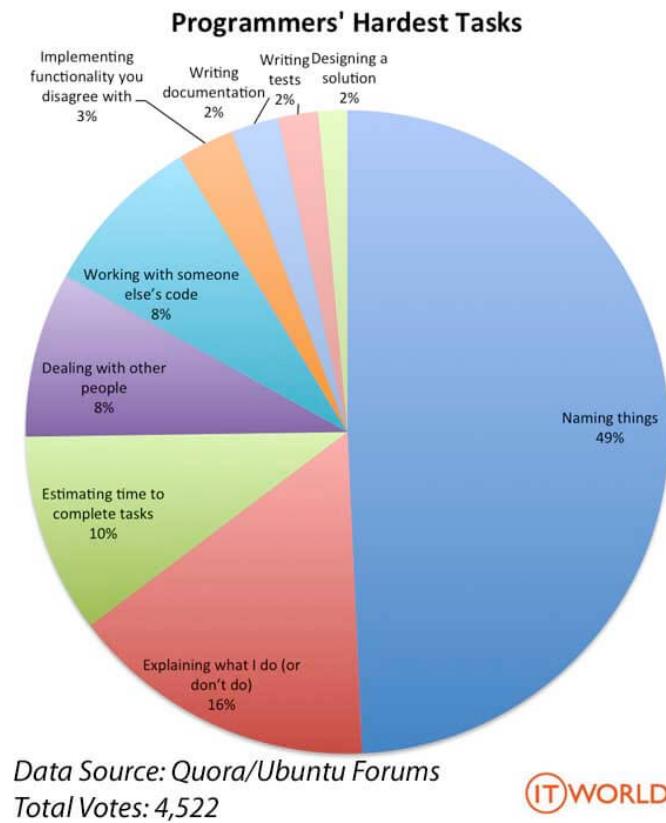
GLOBAL\_CONSTANT\_NAME,

✓ 피할 것

- 카운터나 iterator의 예외에 하나의 Character를 사용하는 것

- package나 module 명에 dash(-)를 쓰는 것

- 파이썬 예약어를 따로 선언하는 것. e.g. \_self\_



# Version 2 vs 3

**Python의 경우 2.7x 버전과 3x 버전간의 호환성 이슈가 있음.**

- ✓ Anaconda에 기본으로 설치되어 있는 Jupyter를 활용하면 2버전과 3버전 둘 다 활용가능

## 자동 형 변환

파이썬 3의 경우 숫자연산 시 자동으로 형 변환이 된다.

파이썬 3 버전의 예

```
>>> 3 / 4
0.75
```

파이썬 2.7 버전의 예

```
>>> 3 / 4
0
>>> 3 / 4.0
0.75
```

## input

파이썬 3 버전의 `input` 내장함수와 파이썬 2.7버전의 `raw_input` 내장함수는 동일하다. 기존 파이썬 2.7의 `input` 내장함수는 파이썬 3부터는 더이상 지원되지 않는다.

파이썬 3 버전의 예

```
>>> name = input("이름을 입력하세요:")
```

파이썬 2.7 버전의 예

```
>>> name = raw_input("이름을 입력하세요:")
```

## 에러처리

`try ... except...` 에러 처리 시 에러 변수명을 표기하는 방식이 파이썬 버전 3과 버전 2.7이 서로 다르다.

파이썬 3 버전의 예

```
try:
    4 / 0
except ZeroDivisionError as e:
    print(e)
```

파이썬 2.7 버전의 예

```
try:
    4 / 0
except ZeroDivisionError, e:
    print e
```

에러변수 설정 시 파이썬 3 버전은 `as` 를 2.7 버전은 콤마(`,`)를 사용한다.

## 기본 인코딩 방식 변경

```
# -*- coding: utf8 -*-
```

- ✓ 한글 입력시 위의 인코딩 표기를 해야 했으나, 3버전 부터 utf8로 고정

## Print

- ✓ 2.7 버전에서도 `print('hello python')` 작동 합니다.

## 줄 바꿈 방지

- ✓ Google Style Guide에 의하면 쓰지 않는 것을 추천

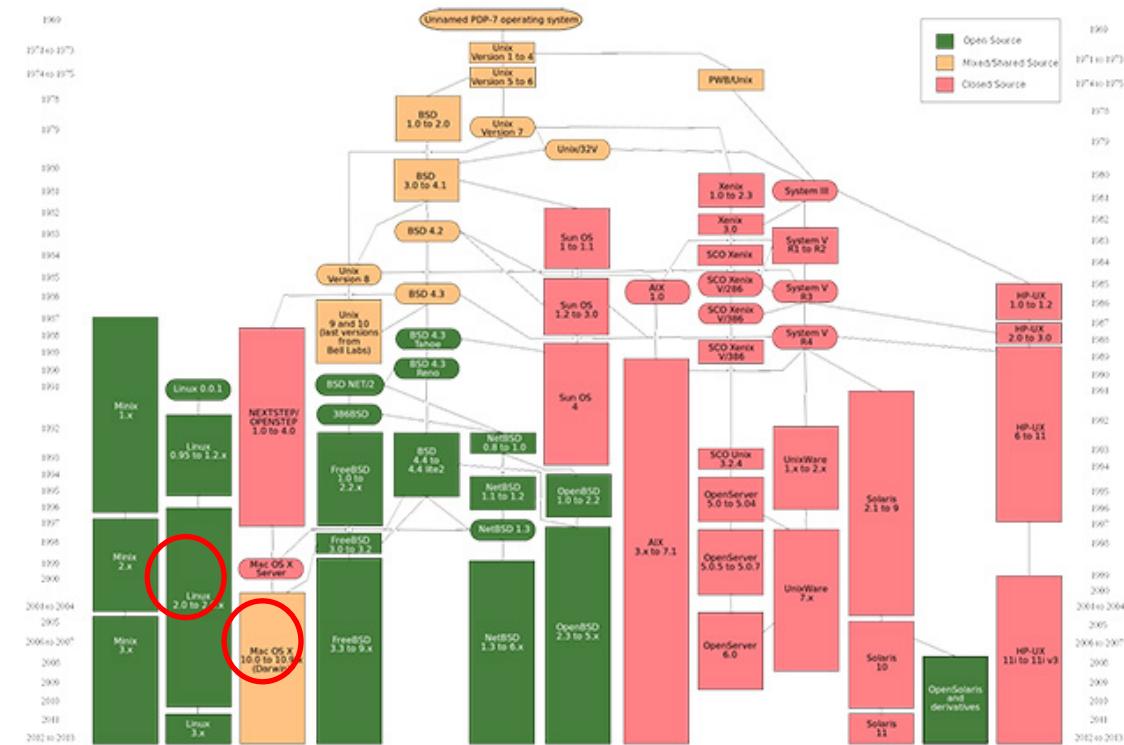
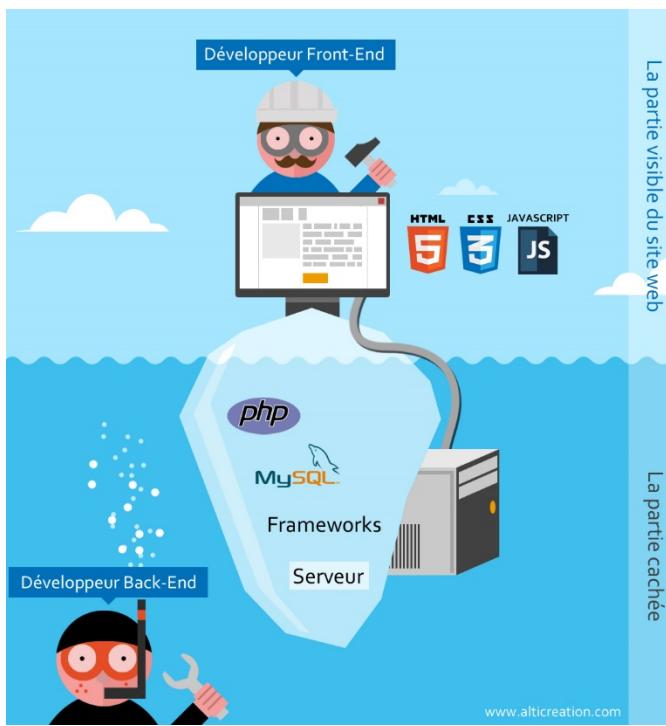
# Docker

---

# Docker

## 개발자들이 MAC을 쓰는 이유?

- ✓ MacOS와 Linux는 같은 BSD Unix 계열
- ✓ 웹은 프론트엔드와 백엔드로 나뉘어져 있는데, 백엔드를 편집하려면 서버사이드 개발 환경에서 작업을 해야함. 대부분의 경우가 Unix계열 환경임.

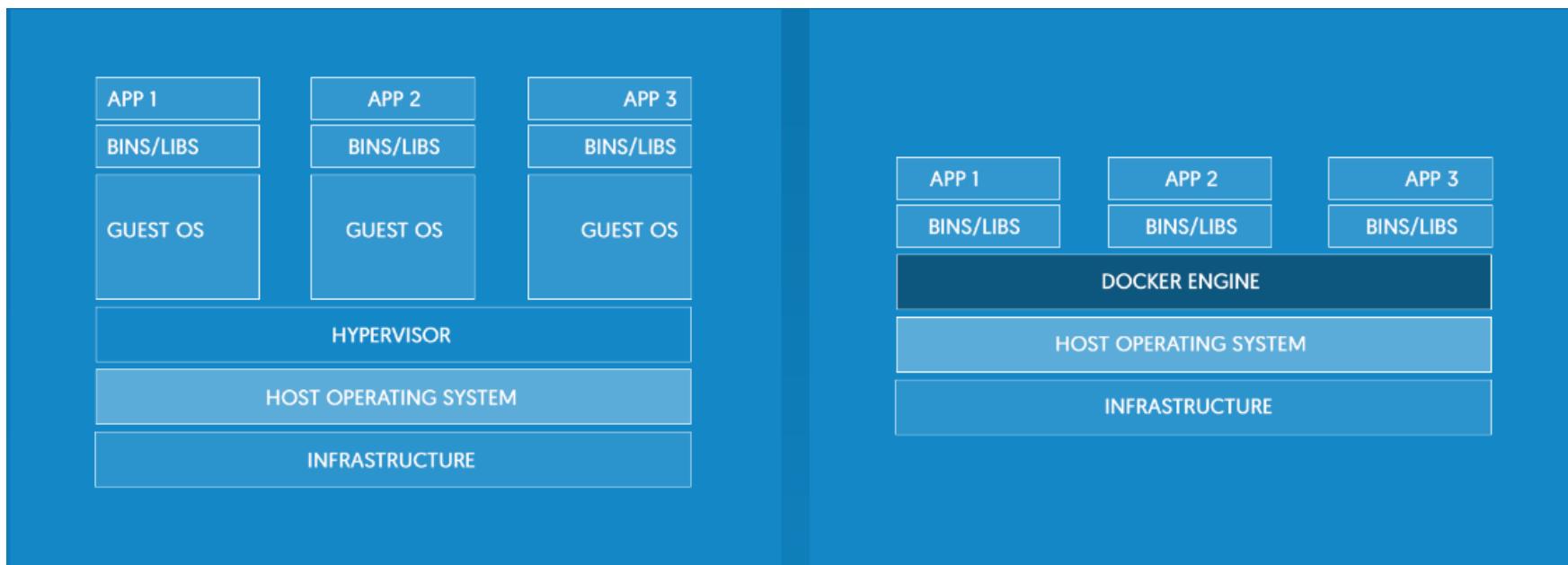


Unix Family Tree [http://keithcu.com/wordpress/?page\\_id=1165](http://keithcu.com/wordpress/?page_id=1165)

# Docker

## 항상 Guest OS 띄워야 하는 가상머신과는 달리 Docker Engine으로 통합하여 성능향상

- ✓ Docker는 호스트 OS에서 개발 환경 등을 통째로 변경하지 않고 컨테이너 Container화 시켜서 다른 컴퓨터로 배포할 수 있음.
- ✓ 도커를 활용하면, 기존의 개발환경에 영향을 주지 않으면서 개발 환경을 구축할 수 있다는 장점이 있음.
- ✓ Backend 환경(MacOS/Linux)에서만 가능했던 작업들이 상당수 Docker에 의해 작업이 가능함.



Docker Official homepage, <https://www.docker.com/what-docker>

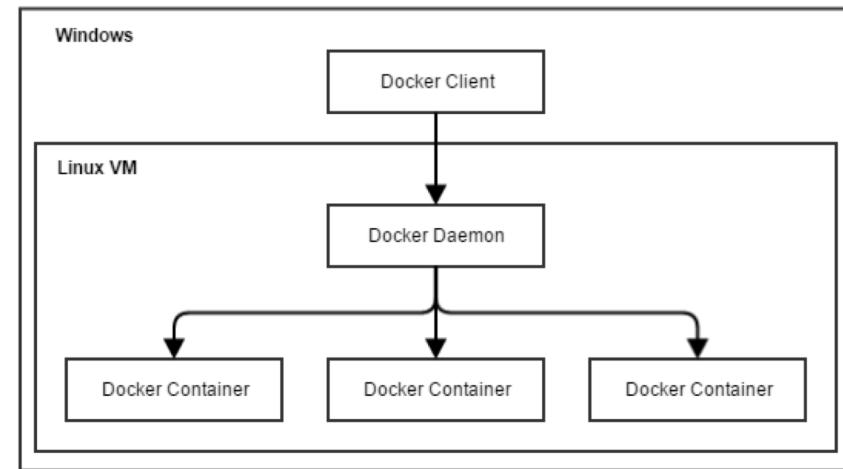
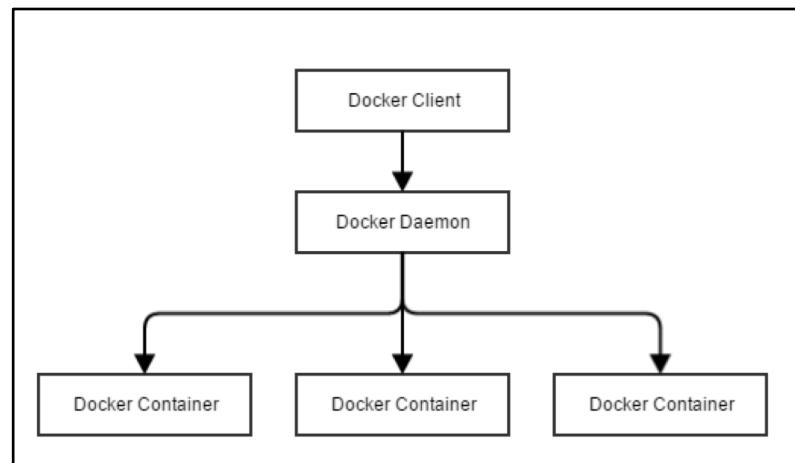
참고자료: 이재홍(2015) <http://www.slideshare.net/pyrasis/docker-fordummies-44424016>

Docker를 통해 격리된 개발 환경을 만들 수 있으며, 해당 개발 환경을 재사용 및 배포 할 수 있음.

# Docker

## 원도우에서 구동되는 Docker는 VirtualBox 기반의 Linux VM에서 작동함

- ✓ MAC이나 Linux 기반의 PC에서 보다 성능이 저하될 수 있으나, 격리된 개발 환경을 재사용할 수 있다는 장점은 여전히 지니고 있음.



Tom (2016) <http://pseg.or.kr/pseg/infoinstall/6076>

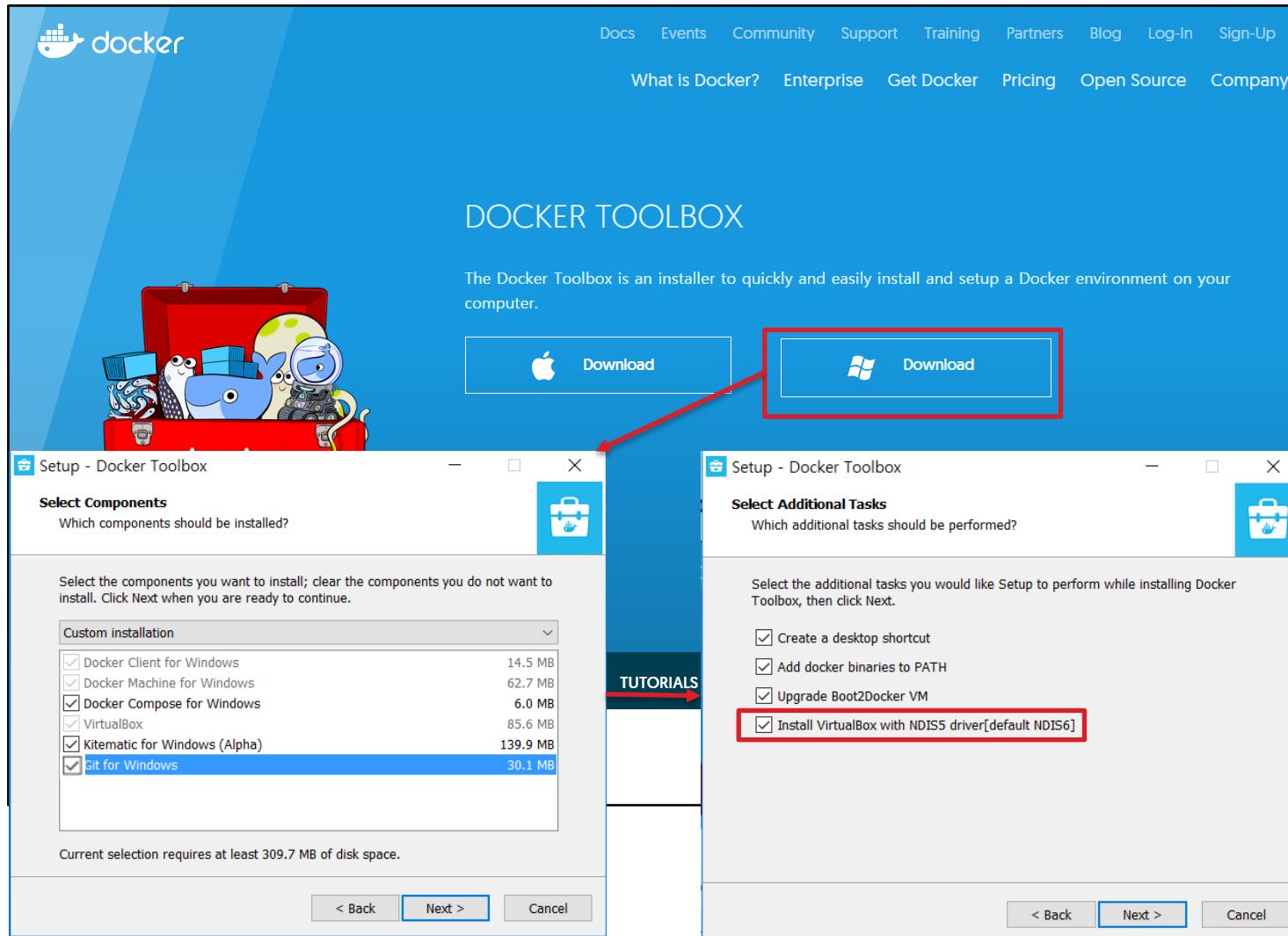
## Docker를 본 수업에서 사용하는 이유.

- ✓ 간편하게 Linux상의 개발 환경을 구현하기 위해
- ✓ 추후 Web 실습 수업과 특정 개발 환경하에서 개발을 진행할 시, 유용하게 활용할 수 있음.

# Docker Install

<https://www.docker.com/products/docker-toolbox>

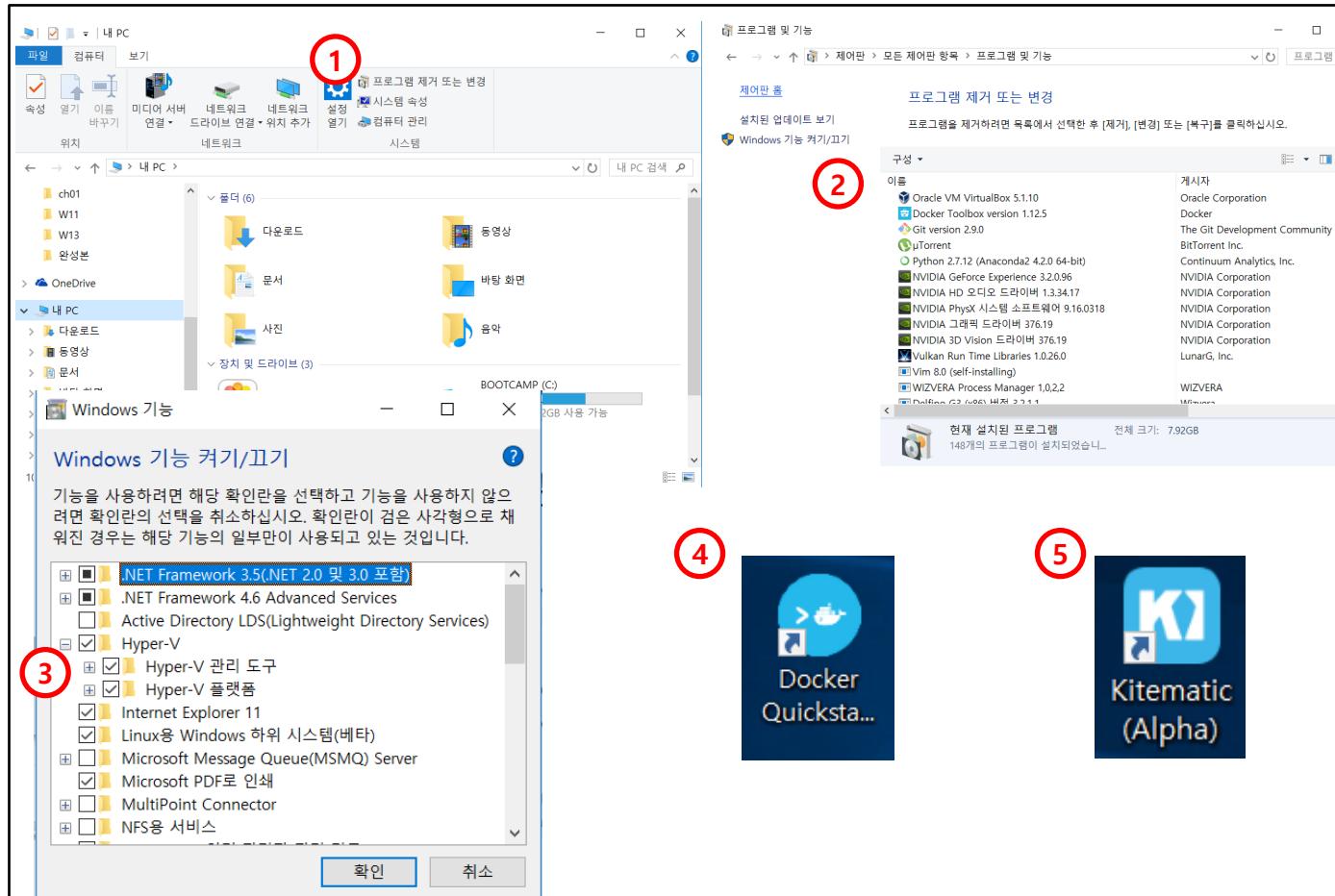
## ✓ Docker Toolbox Download



# Docker Install

<https://www.docker.com/products/docker-toolbox>

- ✓ 윈도우에서 지원하는 Hyper-V 기능 켜기
- ✓ 각종 금융사이트에서 설치되어 트레이 아이콘에서 실행되고 있는 보안 프로그램 실행 일시중지



Description	
① [내 컴퓨터] - [프로그램 제거 또는 변경]	
② [Windows 기능 켜기/끄기]	
③ Hyper-V 항목에 관리 도구, 플랫폼의 모든 하위 항목에 체크표시 확인. (a) 항목이 존재하지 않는 경우 Hyper-V를 지원하지 않는 컴퓨터 혹은 윈도우 버전임 (b) Hyper-V 하위 항목 중 일부만 선택 가능시, [일부항목체크]-[재부팅]-[남은항목체크]-[재부팅]하면 모든 항목에 체크 되어 있는 걸 확인할 수 있음.	
④ [Docker Quickstart Terminal] 실행 후, [Kitematic (Alpha)] 실행	

# Docker Install

<https://www.docker.com/products/docker-toolbox>

## ✓ Docker 설치 확인하기

The terminal window shows the following sequence:

- ① Docker logo icon in the top right corner.
- ② The command `$ docker run hello-world` is typed into the terminal.
- ③ The output of the command is displayed:

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker Hub account:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

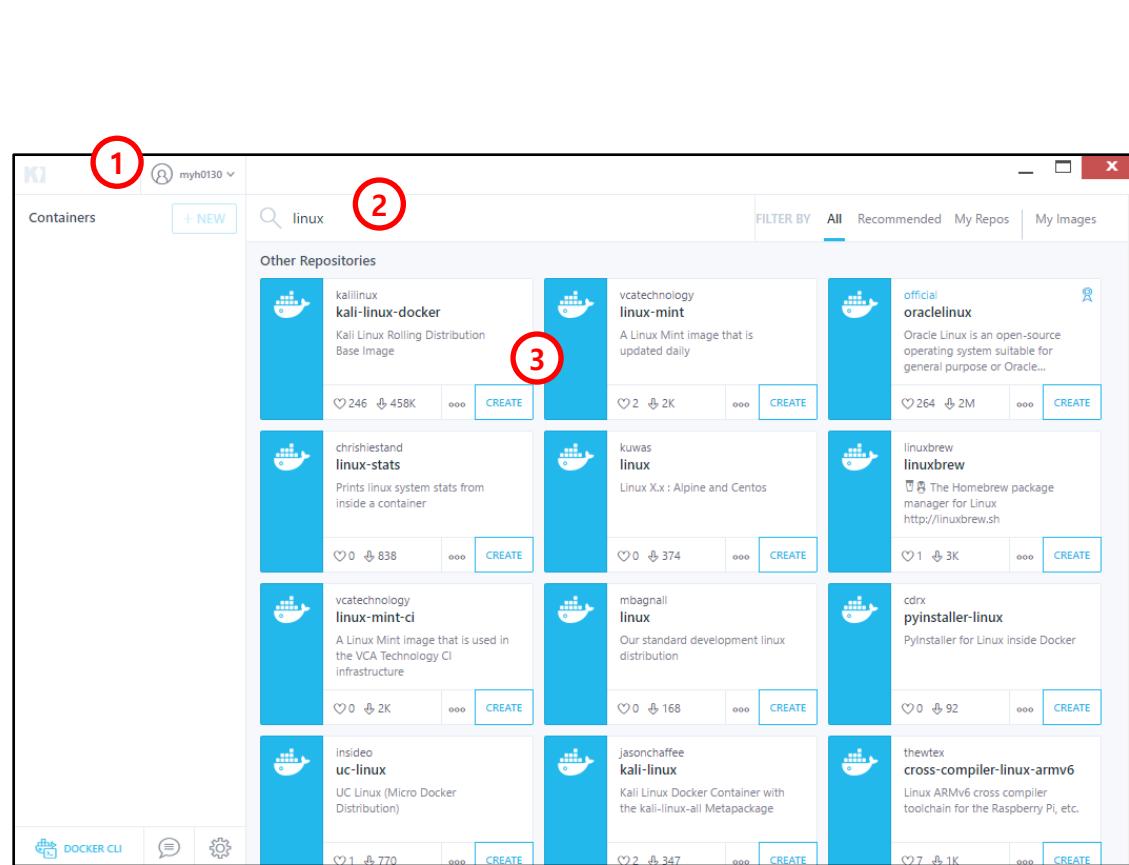
### Description

- ① 정상적으로 Docker가 설치되었을 시 화면
- ② \$ docker run hello-world 입력
- ③ 정상적으로 Docker가 작동한다면 [Hello from Docker!] 메시지를 확인할 수 있음

# Docker Install

<https://www.docker.com/products/docker-toolbox>

- ✓ Kitematic은 GUI 기반으로 Docker Image를 검색하고 Container를 간편하게 설치할 수 있는 Tool



## Description

- 회원가입을 진행하면, 자신이 만든 Docker Image를 Docker Hub에 배포할 수 있음.
- 원하는 Docker Image를 찾기 위해 검색. Official 마크가 붙은 것은 공식적으로 해당 기술 Stack의 개발사가 제공하는 Image
- [Create]를 누르면 Container가 생성된다.

# Docker Cheat Sheet

[https://www.docker.com/sites/default/files/Docker\\_CheatSheet\\_08.09.2016\\_0.pdf](https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf)



**Docker Cheat Sheet**

**ORCHESTRATE**

Initialize swarm mode and listen on a specific interface <code>docker swarm init --advertise-addr 10.1.0.2</code>	Create a service from an image exposed on a specific port and deploy 3 instances <code>docker service create --replicas 3 -p 80:80 --name web nginx</code>
Join an existing swarm as a manager node <code>docker swarm join --token &lt;manager-token&gt; 10.1.0.2:2377</code>	List the services running in a swarm <code>docker service ls</code>
Join an existing swarm as a worker node <code>docker swarm join --token &lt;worker-token&gt; 10.1.0.2:2377</code>	Scale a service <code>docker service scale web=5</code>
List the nodes participating in a swarm <code>docker node ls</code>	List the tasks of a service <code>docker service ps web</code>

**BUILD**

Build an image from the Dockerfile in the current directory and tag the image  
`docker build -t myapp:1.0 .`

List all images that are locally stored with the Docker engine  
`docker images`

Delete an image from the local image store  
`docker rmi alpine:3.4`

**SHIP**

Pull an image from a registry  
`docker pull alpine:3.4`

Retag a local image with a new image name and tag  
`docker tag alpine:3.4 myrepo/myalpine:3.4`

Log in to a registry (the Docker Hub by default)  
`docker login my.registry.com:8000`

Push an image to a registry  
`docker push myrepo/myalpine:3.4`

**RUN**

```
docker run
  --rm remove container automatically after it exits
  -it connect the container to terminal
  --name web name the container
  -p 5000:80 expose port 5000 externally and map to port 80
  -v ~/dev:/code create a host mapped volume inside the container
  alpine:3.4 the image from which the container is instantiated
  /bin/sh the command to run inside the container
```

Stop a running container through SIGTERM  
`docker stop web`

Stop a running container through SIGKILL  
`docker kill web`

Create an overlay network and specify a subnet  
`docker network create --subnet 10.1.0.0/24 --gateway 10.1.0.1 -d overlay mynet`

List the networks  
`docker network ls`

List the running containers  
`docker ps`

Delete all running and stopped containers  
`docker rm -f $(docker ps -aq)`

Create a new bash process inside the container and connect it to the terminal  
`docker exec -it web bash`

Print the last 100 lines of a container's logs  
`docker logs --tail 100 web`

# Running on Linux Server

---

# Running on Linux Server (실습)

K-big 실습 서버 사용 요청 [https://kbig.kr/?q=infra/edu\\_practice\\_apply](https://kbig.kr/?q=infra/edu_practice_apply)

The image shows two screenshots of the K-ICT Big Data Center website. The top screenshot is a navigation menu with a red box around the 'Education Practice' link under the 'Infra' category. The bottom screenshot is a detailed view of the 'Practice Application' page, also with a red box around the same link.

**K-ICT 빅데이터센터 BIG DATA CENTER**

검색어를 입력하세요.  검색

미래창조과학부 NIA 한국정보화진흥원

센터소개 인프라 데이터셋 클리어링 지식정보공유 센터이용지원 비식별조치 | 개방형 문제해결 플랫폼 소셜분석체험

**인프라 Infra**

소개

분석활용 인프라

인프라 예약  
인프라 관리

**교육실습 인프라**

교육실습 강사  
· 강사 신청  
· 강사 신청 / 관리  
· 인프라 예약  
· 학생 관리  
· 인프라 관리

교육실습 학생  
· 실습신청  
· 실습신청 관리  
· 분석설습 관리  
· 인력양성 웰메뉴얼

**실습신청**

총 > 교육실습 인프라 > 실습신청

상태: 실습명: 종양대 검색

※ 사전에 강사와 협의된 학생만 인프라 사용 신청이 가능 합니다.

번호	강사명	소속	실습명	사용일	강의인원수	신청
34	허진경	한국소프트웨어산업 협회	(동아대학교) 빅데이터 분석 프로젝트	2016-12-22 ~2016-12-30	30	<span>+ 신청</span>
33	윤효근	엑셀	하둡 실습	2016-12-22 ~2016-12-30	40	<span>+ 신청</span>
32	윤효근	엑셀	플라밍고 실습	2016-12-22 ~2016-12-30	40	<span>+ 신청</span>
31	맹윤호	연세대학교 정보대학원	중앙대학교 Python 겨울방학 특강을 위한 인프라 대여(2)	2016-12-22 ~2017-01-13	30	<span>+ 신청</span>
30	맹윤호	연세대학교 정보대학원	중앙대학교 Python 겨울방학 특강을 위한 인프라 대여(1)	2016-12-22 ~2017-01-13	30	<span>+ 신청</span>

# Running on Linux Server (실습)

실습 서버 사용 요청이 완료된 이후에, VM 가동

## 실습신청 관리

The screenshot shows a web-based application for managing experiment requests. At the top, there are search filters for '상태' (Status) set to '전체' (All), '실습명' (Experiment Name) set to an empty field, and a search button. Below is a table with columns: 번호 (Number), 강사명 (Instructor Name), 소속 (Affiliation), 실습명 (Experiment Name), 사용기간 (Usage Period), 상태 (Status), and 실습신청 (Experiment Application). A row for '테스트2' (Test2) is selected, highlighted with a red border. A red arrow points from the '서버보기' (View Server) button in the table to a detailed modal window.

**선택 VM 정보 팝업**

VM명	edu_교육실습 예약_2_2014-12-29 11:14:37
호스트서버	edu2
실습명	교육실습 예약
사용기간	2014-12-29 ~ 2014-12-31
서버사양	CPU: 2 Core Memory: 4 GB Memory Disk: 70 GB Disk
SSH 접속 정보	IP: 210.114.91.91 Port: 15179
원격 접속 정보	210.114.91.91 Port: 15179
아이디	eduuser
비밀번호	hello.edu
설정	서버종료

Buttons at the bottom of the modal: [▶ 시작] (Start), [■ 종료] (Stop), and [X 닫기] (Close).

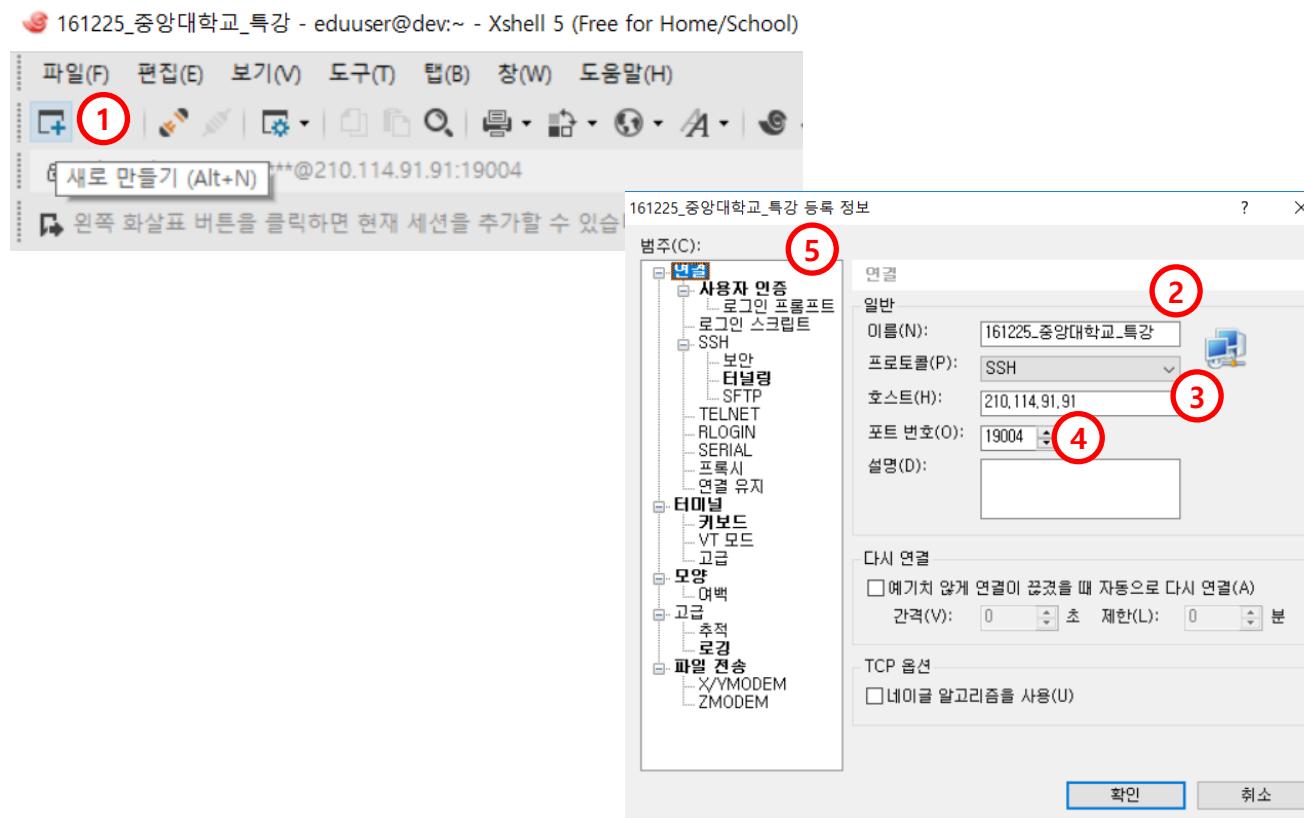
### Description

- ① [서버 보기] 클릭
- ② [실습 VM 정보] 창 확인
- ③ [시작] 버튼 클릭
- ④ 정상적으로 VM이 시작되면 IP와 Port 정보를 확인 가능
- ⑤ 아이디와 비밀번호 확인

# Running on Linux Server (실습)

## Xshell을 통한 접속 정보 등록 이후 Xftp를 활용한 코드 파일 업로드

- ✓ Xshell, Xftp 다운로드 <http://www.netsarang.co.kr/download/main.html>
- ✓ Xshell 설정하기



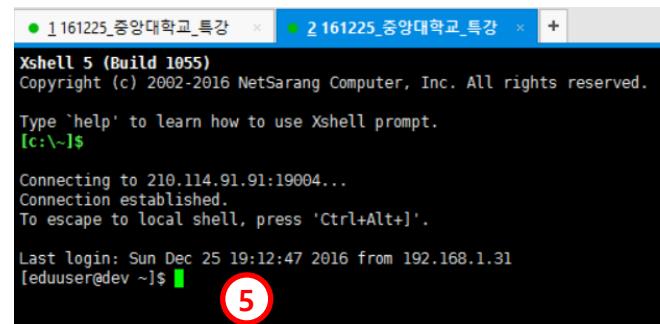
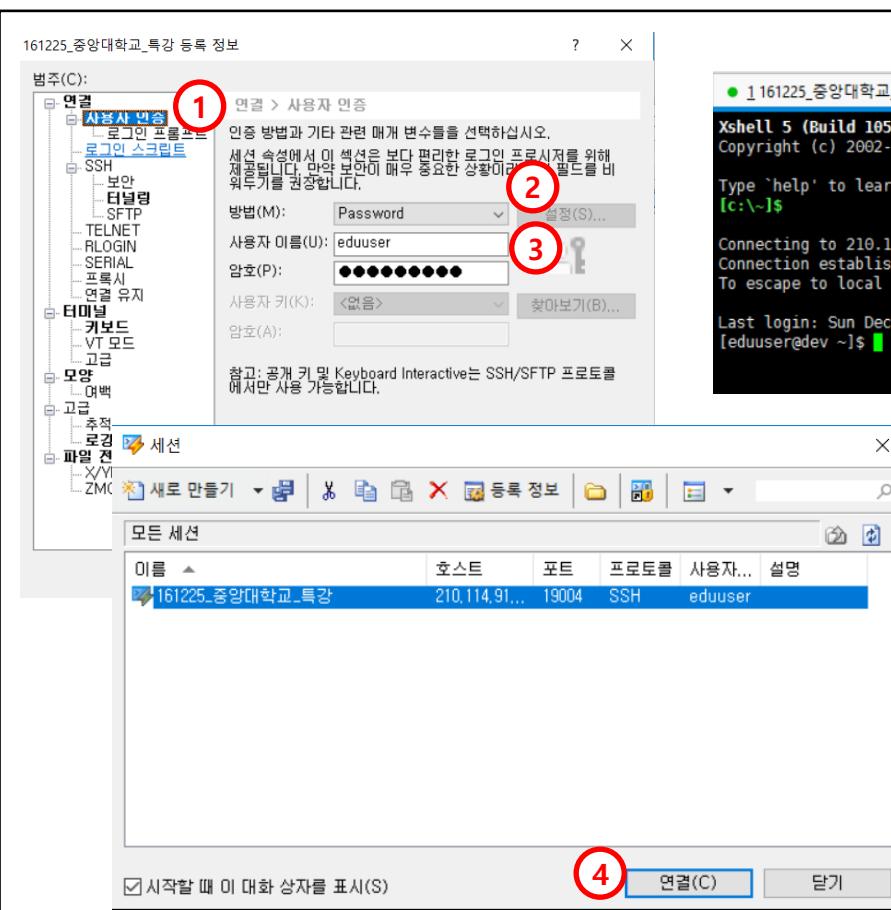
### Description

- ① Xshell 설치 이후에 [새로만들기]
- ② [연결] 탭에 원하는 [이름] 입력
- ③ 리눅스 서버의 Host IP입력
- ④ 포트번호 입력
- ⑤ 이후 [사용자 인증] 탭으로 이동

# Running on Linux Server (실습)

## Xshell을 통한 접속 정보 등록 이후 Xftp를 활용한 코드 파일 업로드

- ✓ Xshell, Xftp 다운로드 <http://www.netsarang.co.kr/download/main.html>
- ✓ Xshell 설정하기



### Description

- ➊ [사용자 인증] 탭
- ➋ 인증방법 [Password] 선택
- ➌ 사용자 이름에 할당받은 계정명과 암호 입력
- ➍ 세션 프로필 작성이 완료되면 [연결]
- ➎ 정상적으로 접속된 화면

# Running on Linux Server (실습)

Ananconda 설치시, Python 실행에 필요한 다양한 Package들이 함께 설치되어 간편함

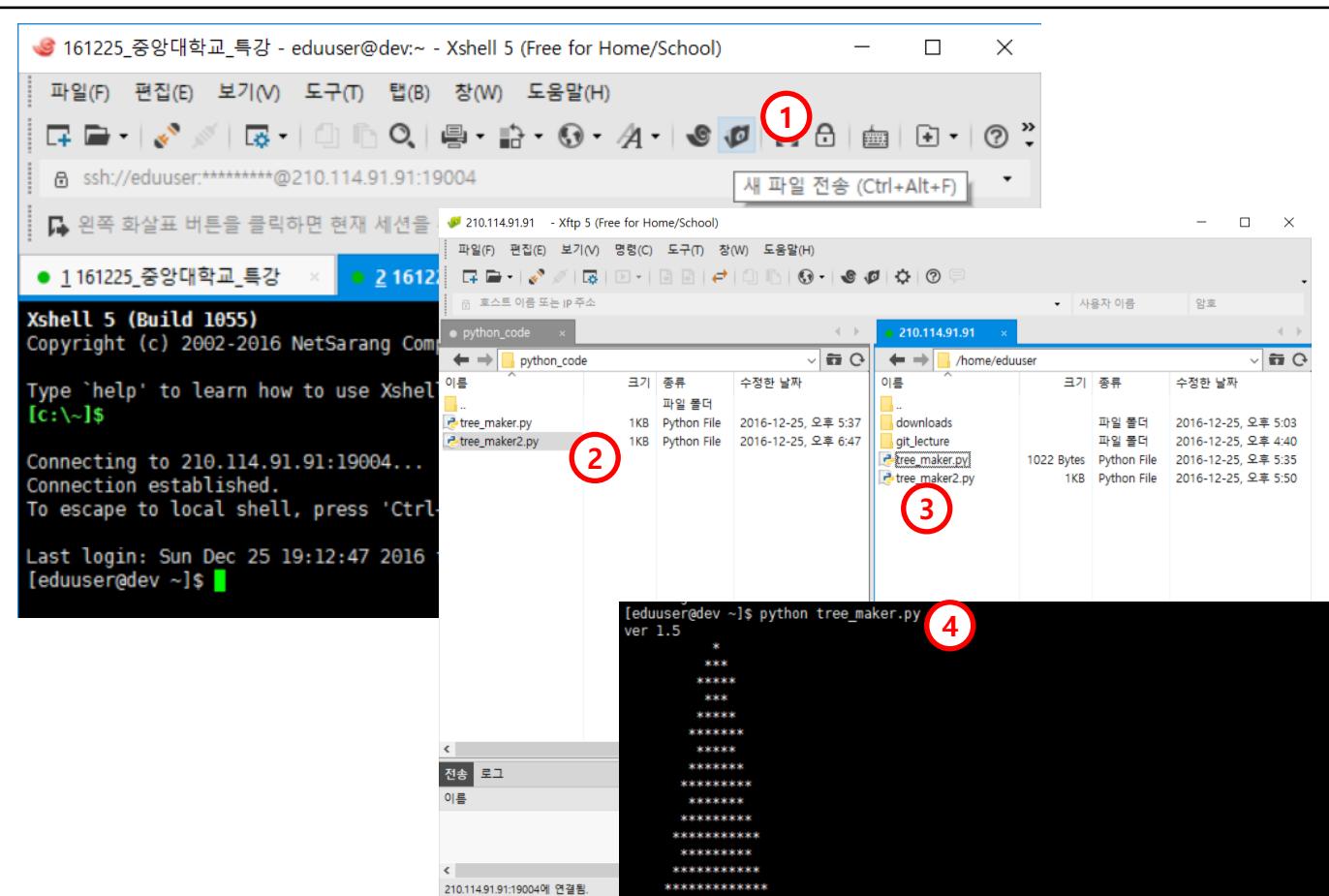
- ✓ Anaconda 다운로드 (2.7버전, 64비트용)
- ✓ \$ wget <https://repo.continuum.io/archive/Anaconda2-4.2.0-Linux-x86.sh>
- ✓ Anaconda 설치
- ✓ \$ bash Anaconda2-4.2.0-Linux-x86\_64.sh
- ✓ 설치 완료 후, 작동확인

```
[eduuser@dev ~]$ python
Python 2.6.6 (r266:84292, Nov 22 2013, 12:16:22)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print 'hello world'
hello world
>>> var1 = 'hello'
>>> var2 = 'world'
>>> print var1+var2
helloworld
>>> █
```

# Running on Linux Server (실습)

## Xshell을 통한 접속 정보 등록 이후 Xftp를 활용한 코드 파일 업로드

- ✓ Xshell, Xftp 다운로드 <http://www.netsarang.co.kr/download/main.html>
- ✓ Xshell 설정하기



### Description

- ① Xshell에서 [새 파일 전송] 클릭
- ② 작성한 소스코드 파일을 서버의 원하는 경로에 드래그앤 드랍을 통해 업로드
- ③ 업로드된 파일이 목록에 잘 나오는지 확인
- ④ Xshell의 Command line에 다음의 명령어 입력  
\$ python [실행할파일.py]  
[argument]

# 과제

---

# 과제

**Print 문을 활용해서 5단짜리 크리스마스 트리를 아래와 같이 리눅스 서버에서 실행시킬 것**

```
[c:\~]$ Connecting to 210.114.91.91:19004...
Connection established.
To escape to local shell, press 'Ctrl+A'

Last login: Sun Dec 25 19:14:35 2016 from 210.114.91.91
[eduuser@dev ~]$ python tree_maker2.py
ver 2.0
*
 ***
 *****
 ***
 *****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
 ****
```

**Google Style Guide 및 A Byte of Python 한글 요약 파일 목차 살펴보고 오기**

감사합니다

---