
알고리즘 실습

180906 - 개발환경구축 및
기본문법

오늘의 목표

- 파이썬3 설치
 - 파이썬3 다운로드 & 설치
 - PyCharm 설치
- 파이썬 실습
 - 인터프리터 이용
- 파이썬 속제
 - unittest!

파이썬 설치: 파이썬3 다운로드 & 설치

Python 다운로드

- <https://www.python.org/downloads/>
 - Main 메뉴 - Downloads
- Download Python 3.7.0



Python2 vs. Python3

- Python2

- 과거, 그리고 현재
- 라이브러리가 많음
- 예전에 나온 OS에 내장
- Until 2020([PEP373](#))

- Python3

- 현재, 그리고 미래
- 큰 성능 개선: 특히 **Network**, **Unicode**
- 최신 OS에 내장
- 라이브러리가 적음(옮겨지는 중)

Python 3.7.0 다운로드

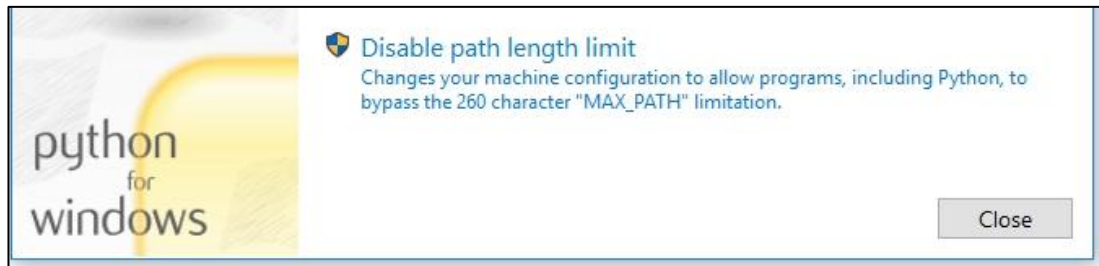
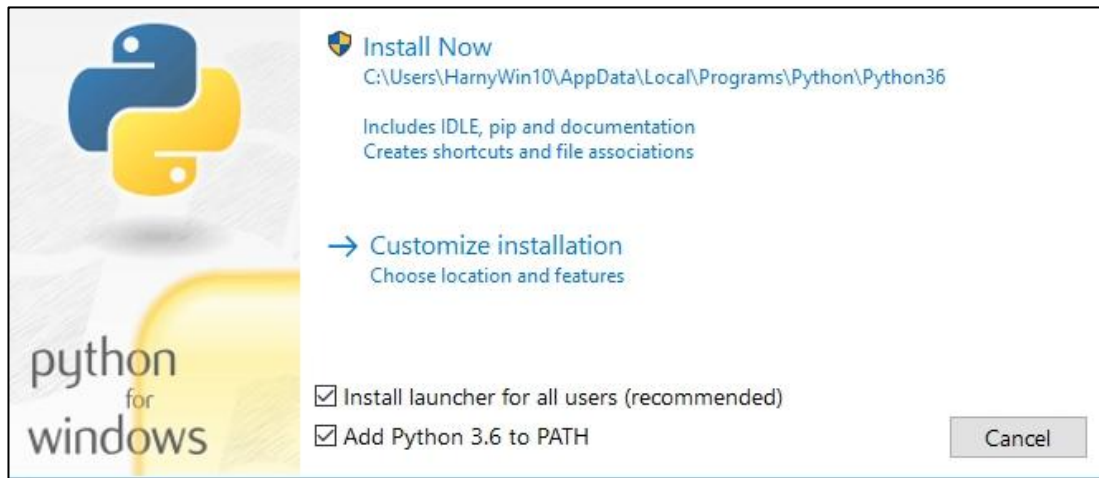
- 운영체제에 맞는 파일 다운로드: 대부분 Windows x86_64

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		2d0fc9f3a5940707590e07f03ecb08b9	22540566	SIG
XZ compressed source tarball	Source release		692b4fc3a2ba0d54d1495d4ead5b0b5c	16872064	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6dd08e7027d2a1b3a2c02cfacbe611ef	27511848	SIG
Windows help file	Windows		69082441d723060fb333dcda8815105e	7986690	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	708496ebbe9a730d19d5d288afd216f1	6926999	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	ad69fdacde90f2ce8286c279b11ca188	31392272	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	a055a1a0e938e74c712a1c495261ae6c	1312520	SIG
Windows x86 embeddable zip file	Windows		8dff09a1b19b7a7dcb915765328484cf	6320763	SIG
Windows x86 executable installer	Windows		3773db079c173bd6d8a631896c72a88f	30453192	SIG
Windows x86 web-based installer	Windows		f58f019335f39e0b45a0ae68027888d7	1287064	SIG

설치시 주의 사항!: 환경변수 체크

- 설치 전 Add Python 3.7 to PATH 체크!
 - 환경변수 추가
- 설치 완료후 Disable path length limit 클릭
 - 인자 길이 제한 해제



설치 후 재부팅

- PowerShell 또는 Command Prompt 에서 python 입력
- Python 3.7.0 이 출력되면 완료



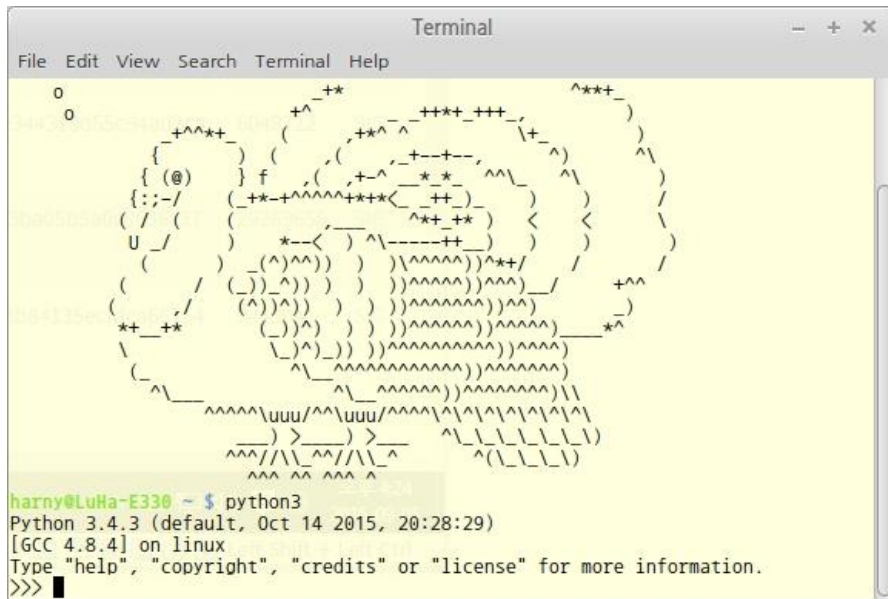
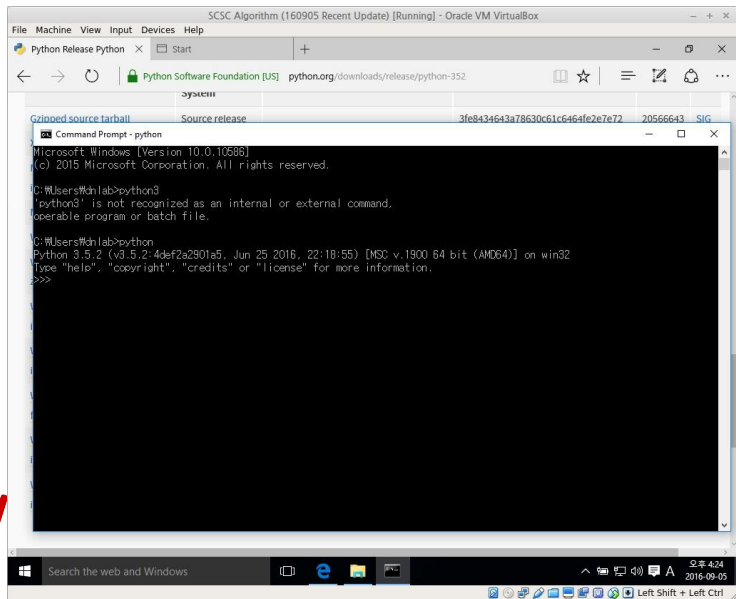
The image displays two screenshots of Windows command shells. The top screenshot is a Windows PowerShell window titled 'Windows PowerShell'. The prompt is 'PS C:\Users\HarnyWin10>' and the command 'python' has been entered. The output shows 'Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)] on win32' followed by instructions to type 'help', 'copyright', 'credits', or 'license' for more information. The prompt '>>>' is visible. The bottom screenshot is a Command Prompt window titled 'Command Prompt - python'. It shows the Windows version '10.0.15063' and copyright information for 2017 Microsoft Corporation. The prompt is 'C:\Users\HarnyWin10>' and the command 'python' has been entered. The output is identical to the PowerShell window, showing 'Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 18:41:36) [MSC v.1900 64 bit (AMD64)] on win32' and the same instructions. The prompt '>>>' is also visible.

(선택)파이썬 설치: 환경변수 설정

환경변수(어려움/몰라도 됨)

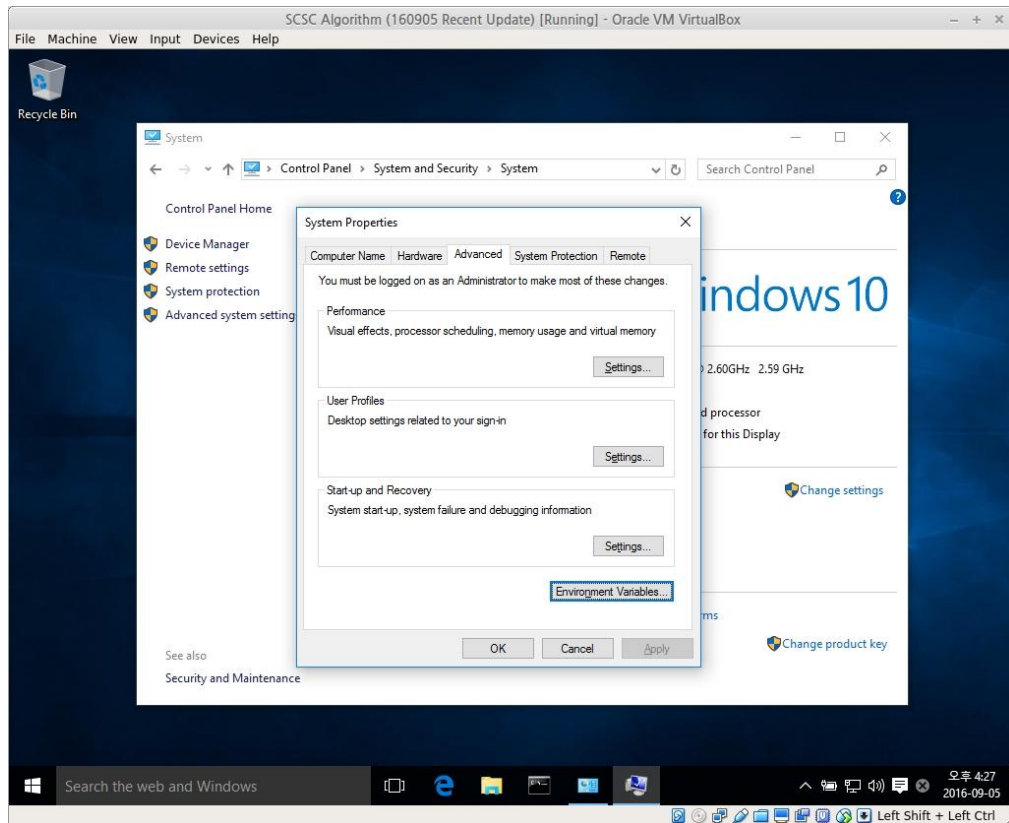
- 환경변수란?

- 컴퓨터의 동작에 영향을 미치는 '값'



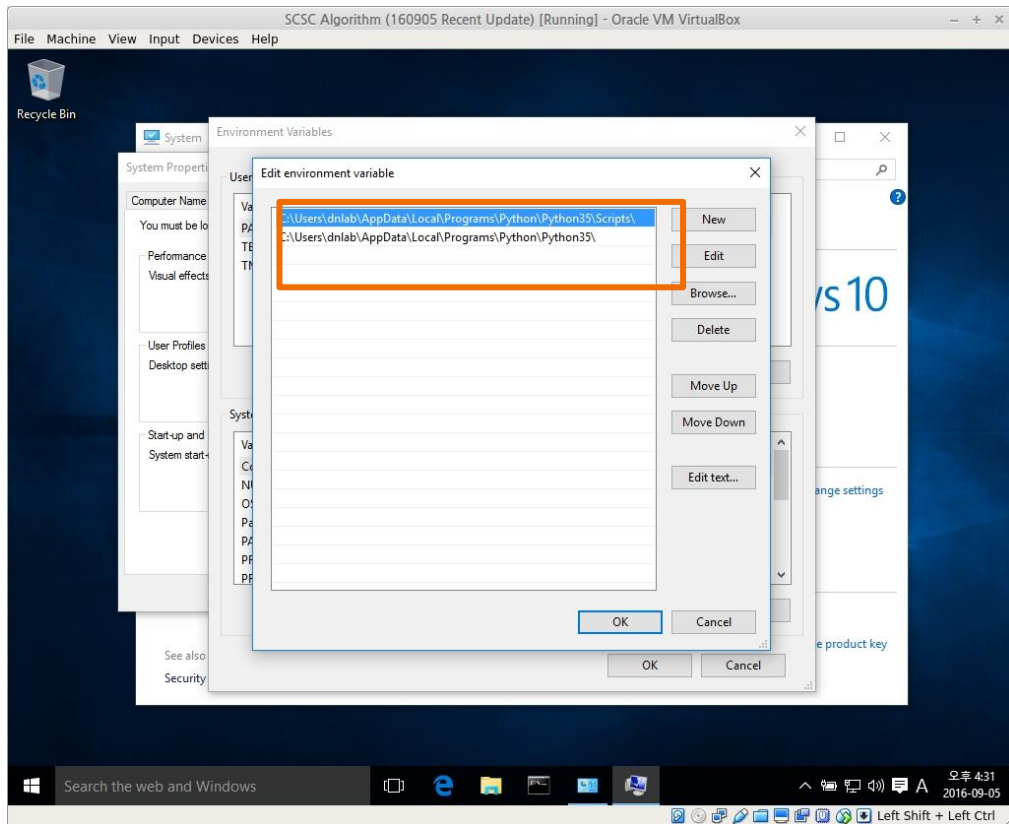
Python 3.7.0 환경변수 설정

- Control Panel - System and Security - System
 - 왼쪽 Advanced system settings - Environment Variables



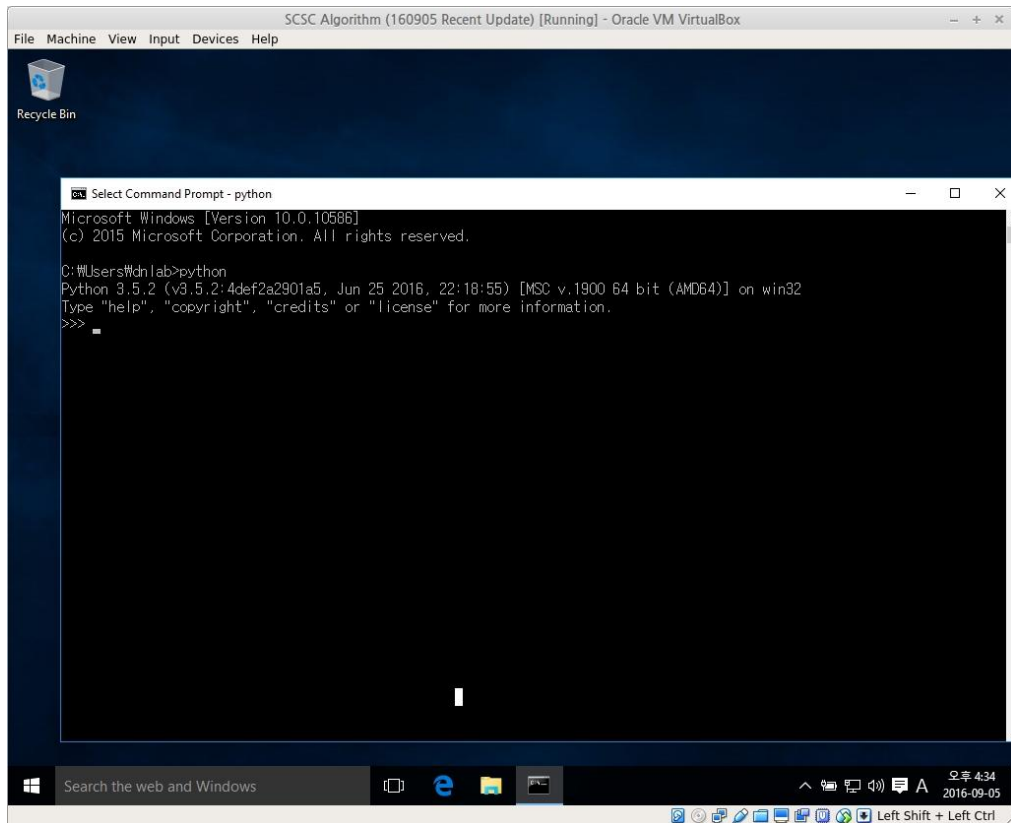
Python 3.7.0 환경변수 설정

- User variables for xxx
 - PATH 더블클릭
 - Python 3.7.0 설치 경로 입력
 - 설치 경로 내부 Scripts 입력



Python 3.7.0 환경변수 확인

- Command Prompt 실행
 - a. python 입력
 - b. python 3.7.0이 실행되면 성공!
 - c. 야호!



파이썬 실습: 인터프리터 이용

인터프리터(어려움/몰라도 됨)

- 인터프리터란?
 - 입력받은 즉시 실행하는 컴퓨터 프로그램
- 인터프리터형 언어
 - 소스코드(컴파일된 실행파일이 아님!)를 입력받아 맨 윗줄부터 실행
 - HTML, Javascript, Python, etc.

Standard Output 실습

- Hello World!를 출력해보자!
 - 소스코드는 어디?
 - 결과는 어디?

```
harny@LuHa-E330 ~ $ python3
Python 3.4.3 (default, Oct 14 2015, 20:28:29)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world!")
Hello world!
>>> □
```


Standard Input/Output 실습

- 이름을 받아 인사하는 프로그램을 만들어보자!
 - 변수?
 - 큰따옴표 / 작은따옴표 차이는?
 - '+'표시는 무엇을 의미하는가?

```
>>> name = input('Whats your name? ')
Whats your name? LuHa
>>> print('Hello', name)
Hello LuHa
>>> print('Hello, ' + name + '!')
Hello, LuHa!
>>> □
```

인터프리터의 한계

- 매번 소스코드를 입력해서 계산을 해야하는가?
- 소스코드를 작성해서 실행해보자!

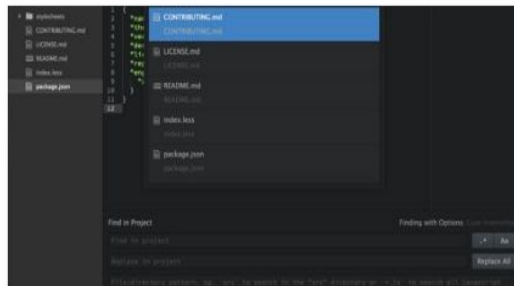
IDE 설치

Why we use IDE?

- 통합 개발 환경: Integrated Development Environment
 - 소스코드 작성, (컴파일), 실행, 디버깅(오류 해석) 등을 하나의 프로그램에서 수행
- 소스코드: 일정한 규칙(프로그래밍 문법)에 맞게 적어둔 텍스트 문서
 - 메모장으로도 할 수 있음
- 그럼에도 IDE를 사용하는 이유?

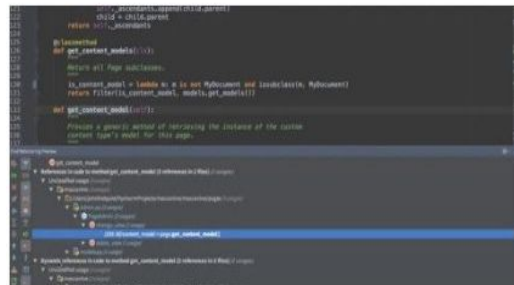
Python IDE

- 가장 많이 활용되는 IDE: Pycharm or Atom
- Pycharm: Python 전용 개발 환경 프로그램, 다양한 기능이 포함되어 있음
- Atom: 코딩을 위한 메모장, 모든 언어에서 활용 가능

[Get It Here](#)

775 👍 302 🗨

PyCharm ▾

[Get It Here](#)

160 👍 48 🗨

설치파일 다운로드

PyCharm

Tools Languages Resources Buy

PyCharm What's New Features Docs & Demos Buy [Download](#)

Download PyCharm

Windows macOS Linux

Professional
Full-featured IDE for Python & Web development
[Download](#)
Free trial

Community
Lightweight IDE for Python & Scientific development
[Download](#)
Free, open-source

Version: 2017.3.2
Build: 173.4127.16
Released: December 28, 2017

[System requirements](#)
[Installation Instructions](#)
[Previous versions](#)

유료, 30일 체험 무료

PyCharm

Tools Languages Resources Buy

PyCharm What's New Features Docs & Demos Buy [Download](#)

Thank you for downloading PyCharm!

Your download should start shortly. If it doesn't, please use [direct link](#).
Download and verify the file's [SHA-256 checksum](#).

Getting Started

Enter your email address to receive tips and tricks

☐ I agree to my email address being used by JetBrains to send me educational materials about the product I evaluate, including commercial communications, and to process my personal data for this purpose. I agree that JetBrains may send the data to [third-party services](#) for this purpose in accordance with [JetBrains Privacy Policy](#). I can revoke my consent at any time in [my profile](#). In addition, an unsubscribe link is included in each email.

[SUBSCRIBE](#)

New to PyCharm?

[Installation Instructions](#)

[First steps](#)

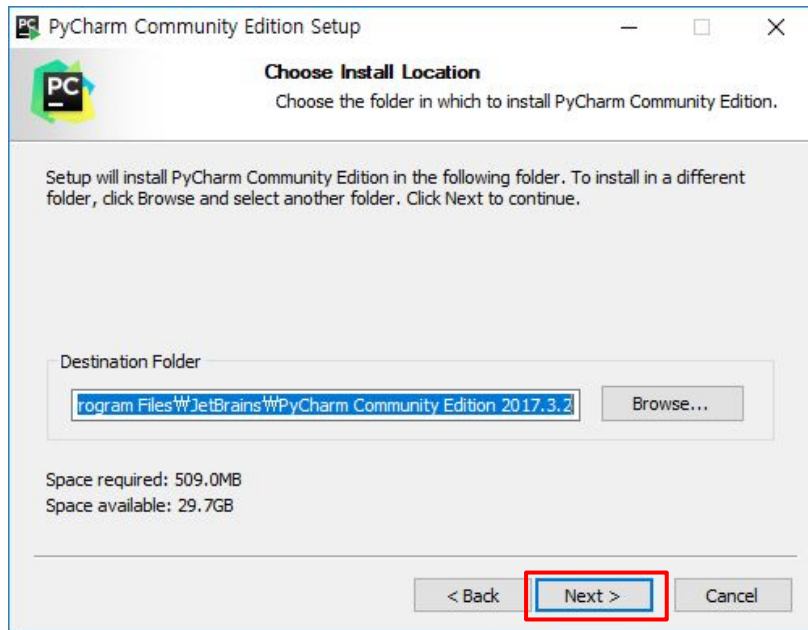
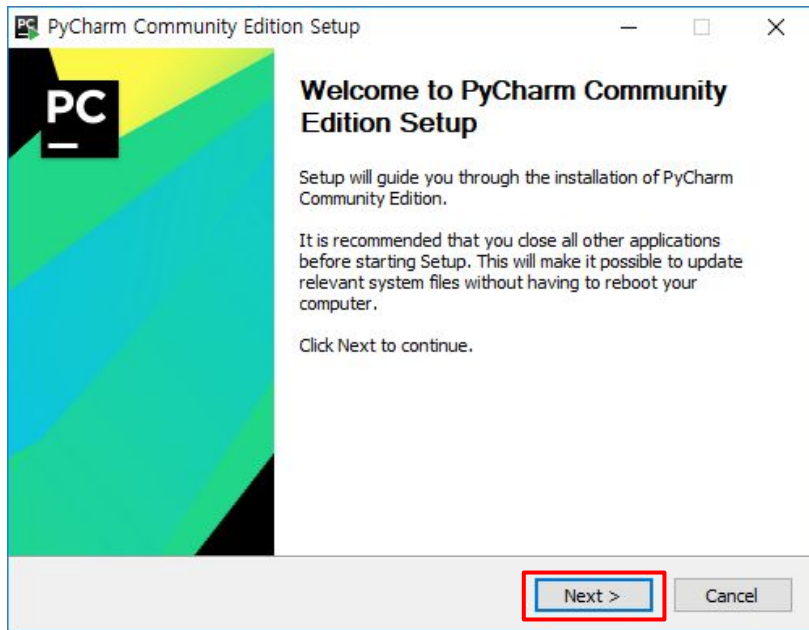
[Meet PyCharm](#)

[Videos and Webinars](#)

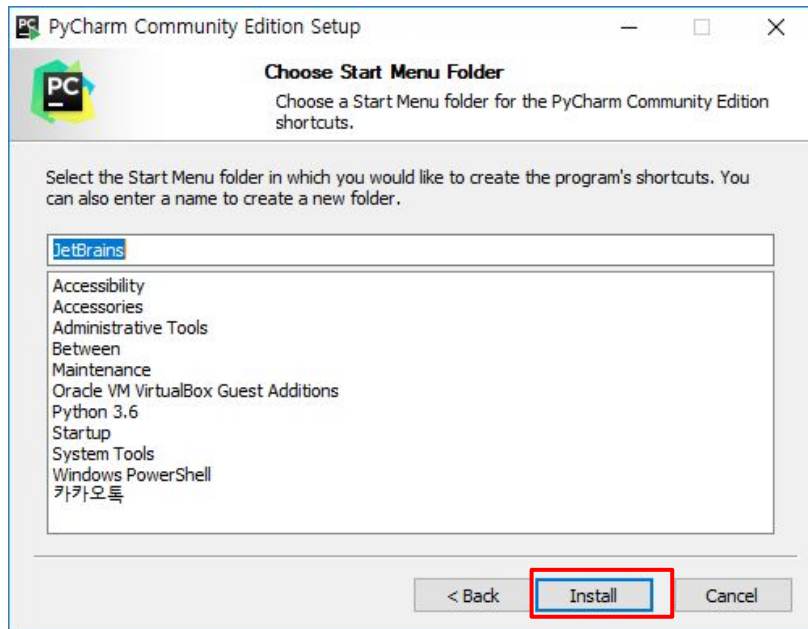
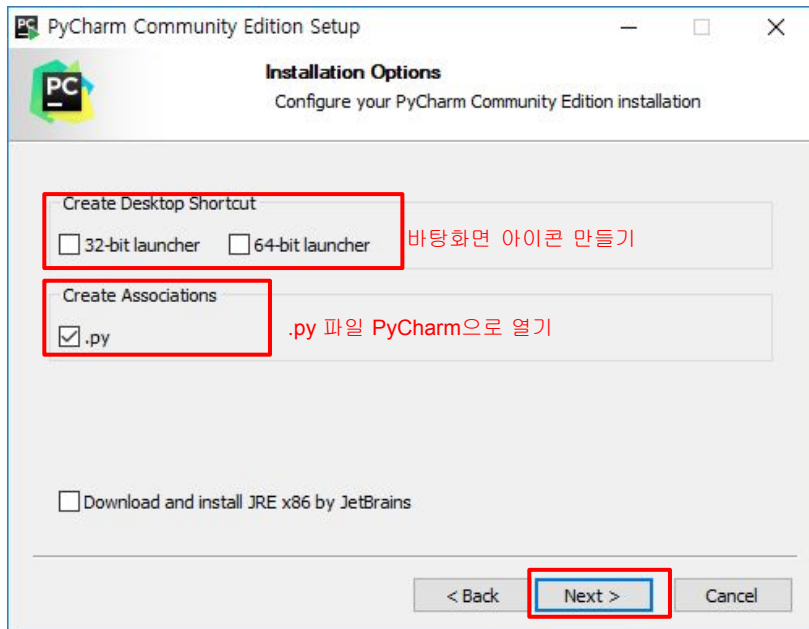
download-cf.jetbrains.com의 pycharm-community-2017.3.2.exe(188MB)를(을) 실행하거나 저장하시겠습니까?

실행(R) **저장(S)** 취소(C)

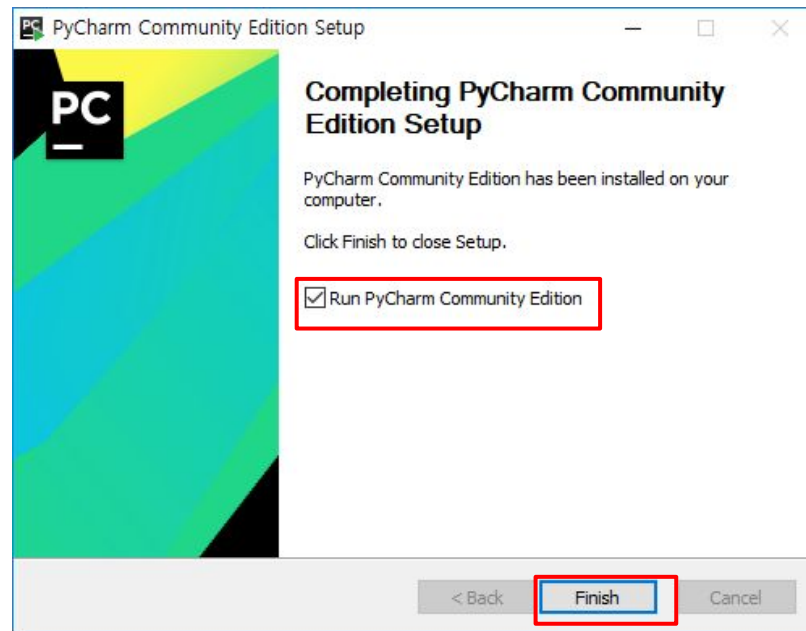
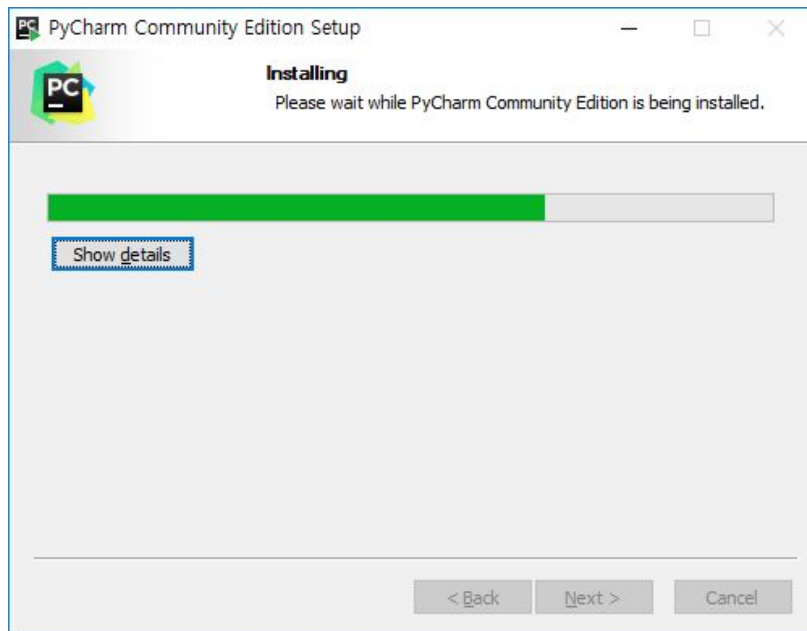
설치 파일 실행



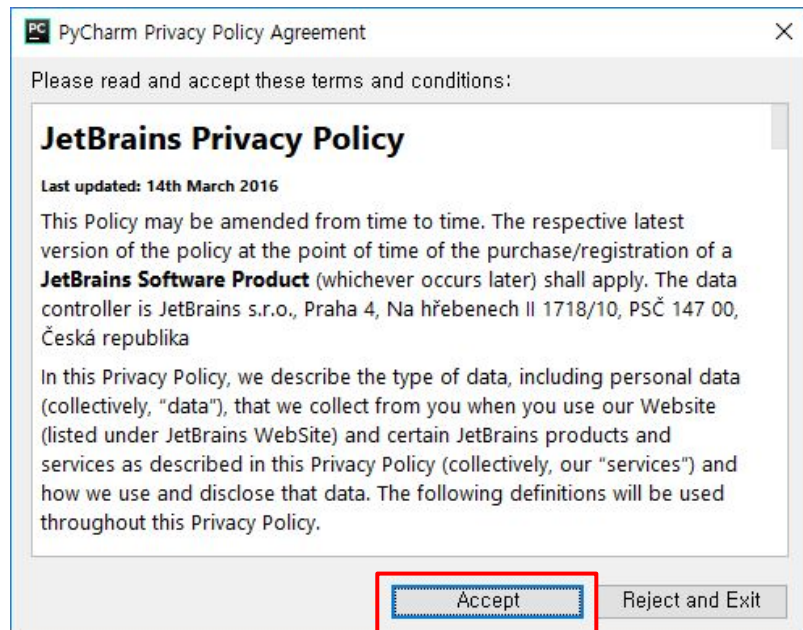
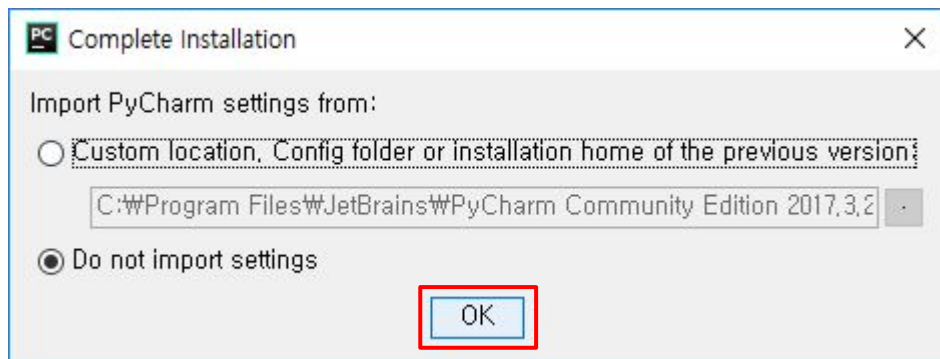
설치 파일 실행(Cont'd)



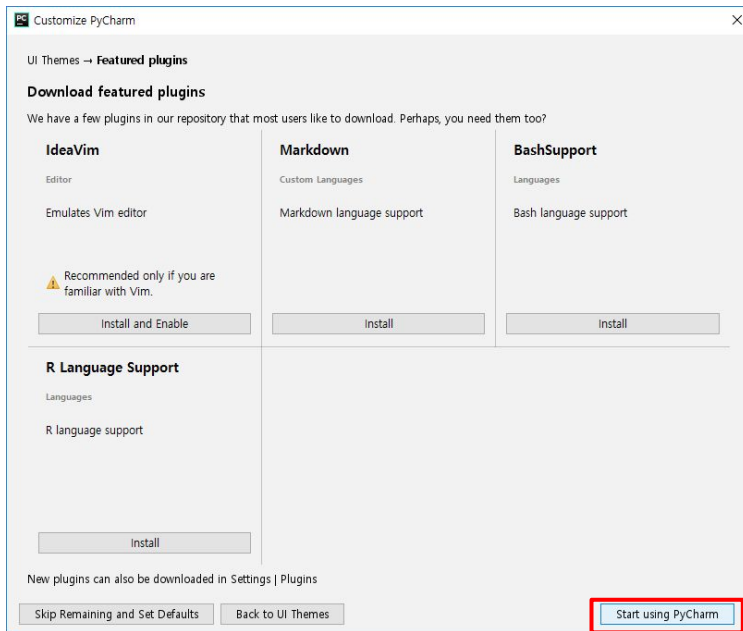
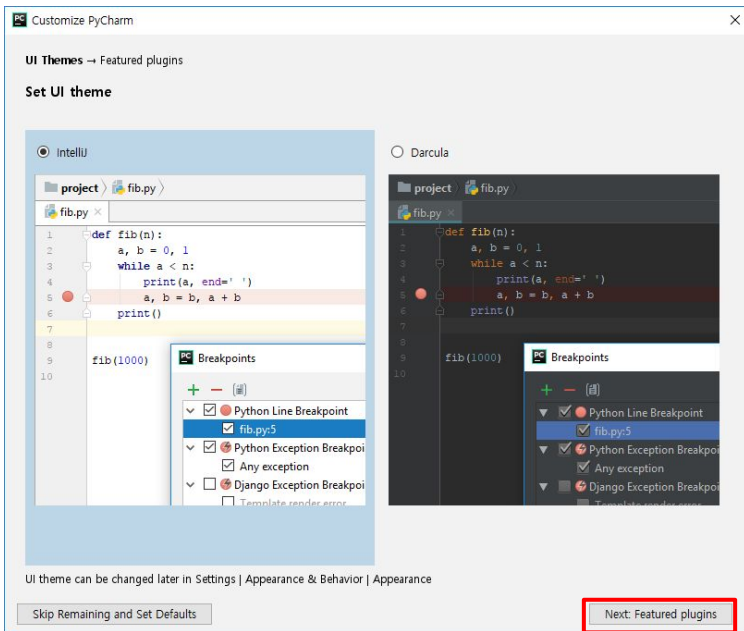
설치 파일 실행(Cont'd)



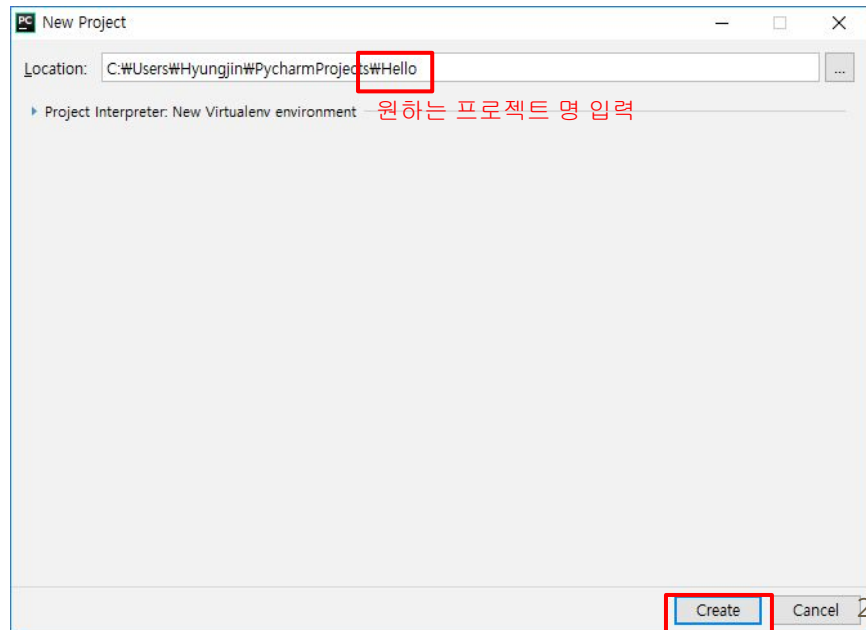
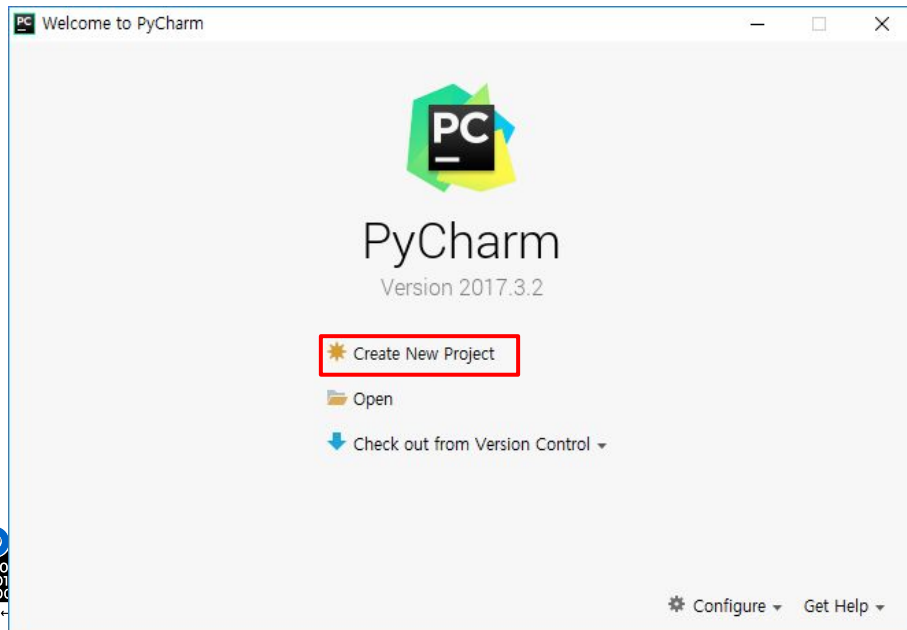
PyCharm 초기 설정하기



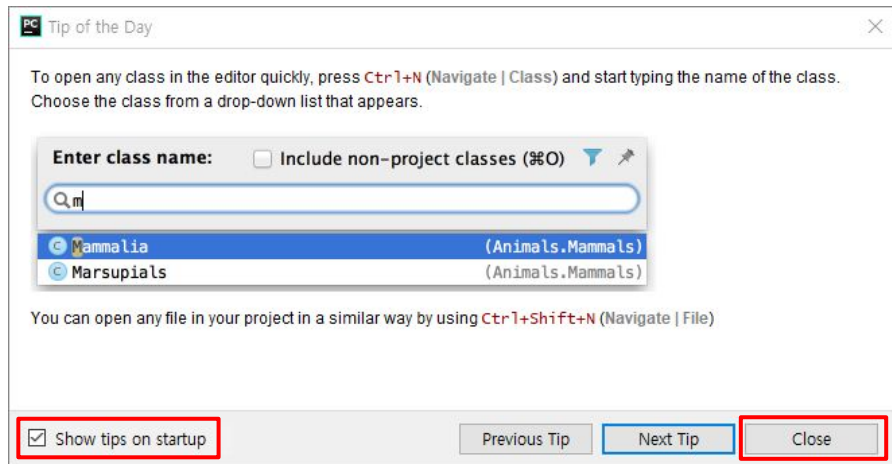
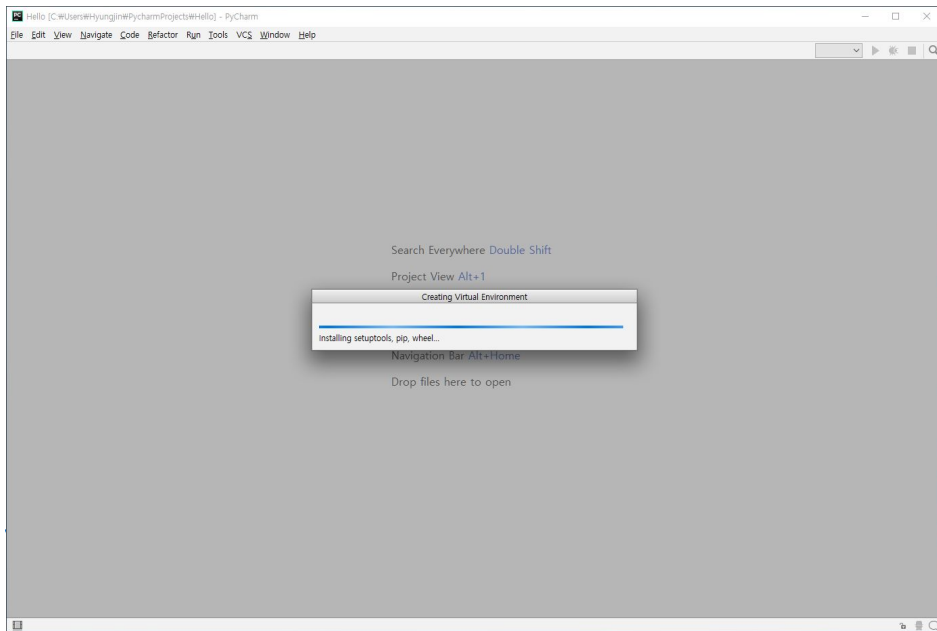
PyCharm 초기 설정하기(Cont'd)



프로젝트 생성하기

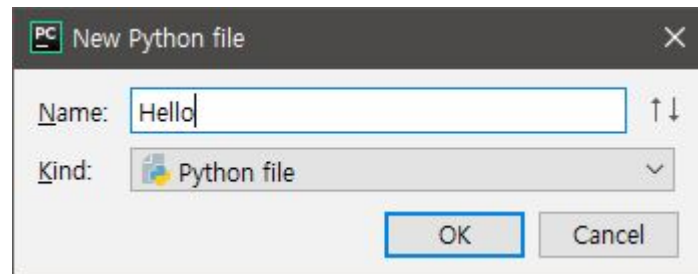
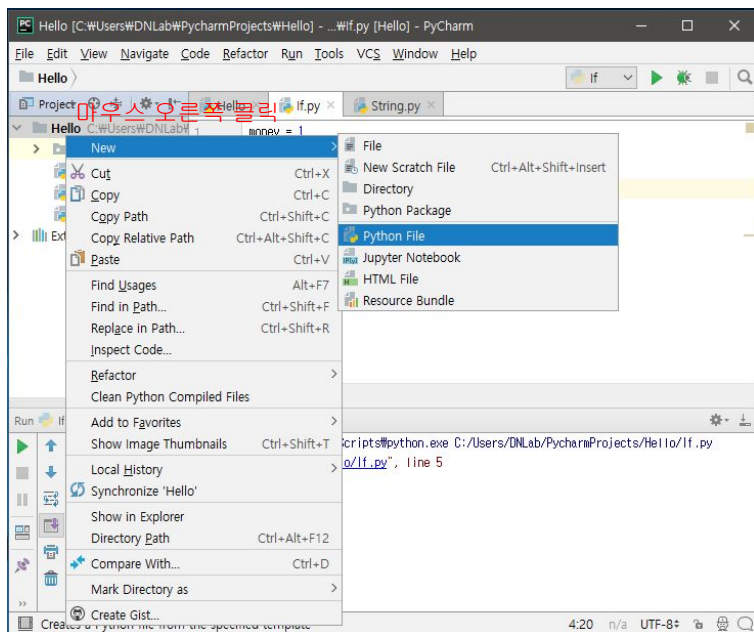


프로젝트 생성하기(Cont'd)

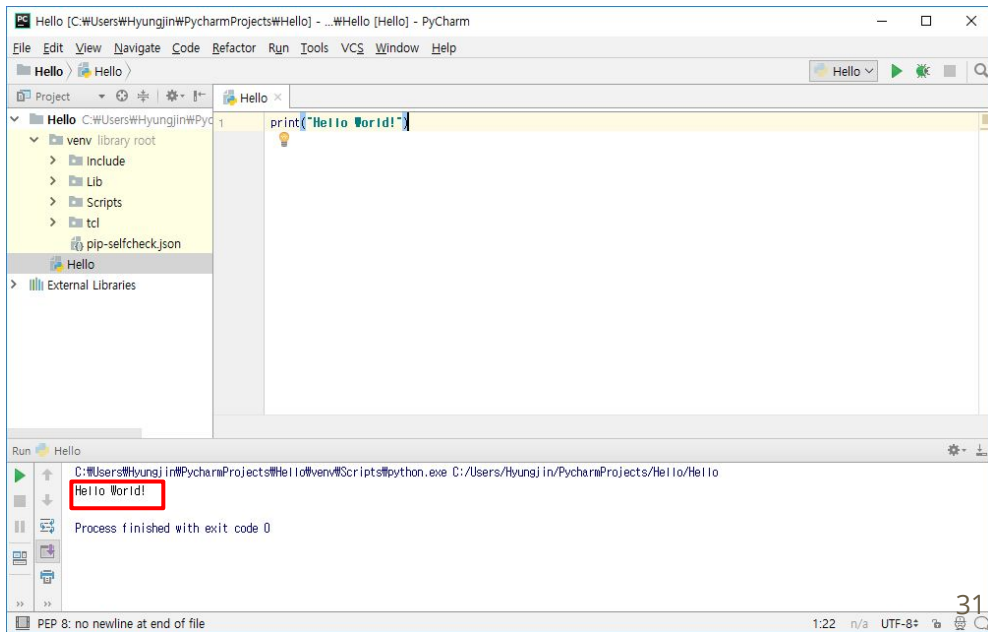
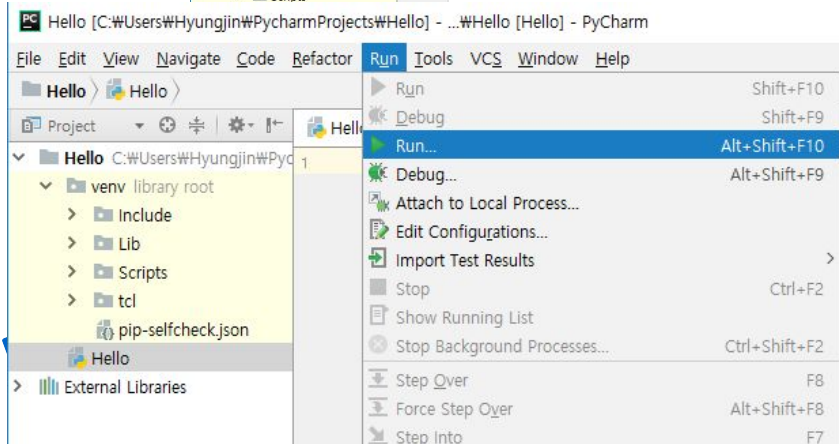
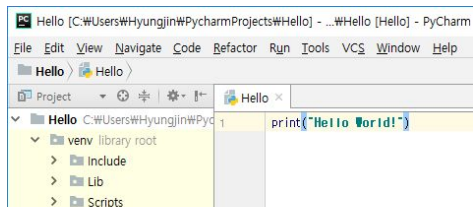


Pycharm 시작시 나오는 사용법 가이드
왼쪽 체크 박스를 없애면 다시 나오지 않는다.

파일 생성하기

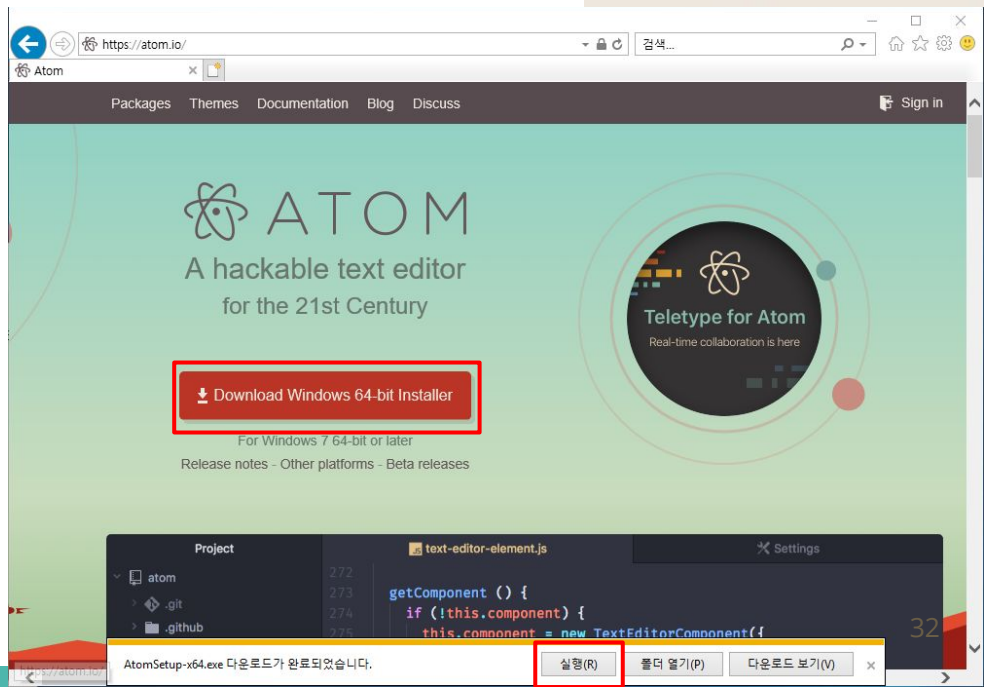
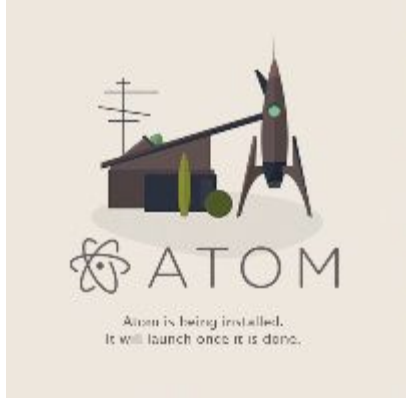


파일 실행하기



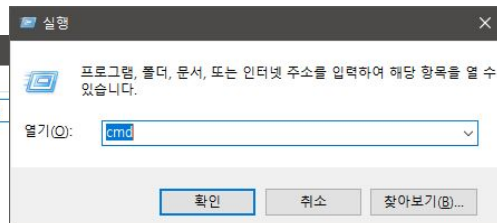
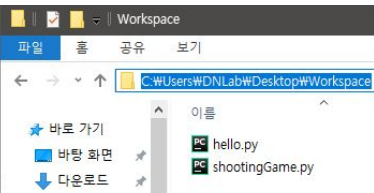
Atom 설치하기

- Atom 사이트 방문하기
 - <https://atom.io/>
- 메인 창에 있는 붉은 Download 버튼 클릭
 - 운영체제에 따라 자동으로 변환된다.
- 다운 받은 .exe 파일을 실행하면, 오른쪽 위 그림이 나오면서 별도의 조작없이 설치 후 실행된다.
- Atom 자체에는 실행 기능이 없으므로 Windows의 cmd 혹은 Mac이나 Linux의 terminal을 이용하여 컴파일러를 실행해야한다.



Window CMD를 통해 Python 실행하기

- 기본 CMD 명령어
 - cd : 현재 작업중인 폴더를 변경한다.
 - dir : 폴더 안에 무슨 파일이 있는지 보여준다.
- cd를 이용하여 실행할 .py 파일 경로로 이동하고, dir 명령어를 통해 파일을 확인한다.
- 'python 파일명.py'을 입력하면 실행



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\DNLab>cd Desktop\Workspace

C:\Users\DNLab\Desktop\Workspace>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 94B6-A12B

C:\Users\DNLab\Desktop\Workspace 디렉터리

2018-01-25 오후 10:29 <DIR> .
2018-01-25 오후 10:29 <DIR> ..
2018-01-16 오후 05:33 274 hello.py
2018-01-23 오후 11:05 10,259 shootingGame.py
                2개 파일              10,533 바이트
                2개 디렉터리 141,123,428,352 바이트 남음

C:\Users\DNLab\Desktop\Workspace>python hello.py
Hi
Hello. How are you?
I'm fine thank you. And you?
I'm fine too.

C:\Users\DNLab\Desktop\Workspace>
```

~~Python~~ 기본 문법

Python 3 Unittest

Unittest

- Test Driven Development(TDD)의 기본!
- 클래스, 메소드, 함수의 I/O가 정확한지 테스트
 - 정확하다면 이것들을 활용한 프로그램도 문제가 없을 것
- 왜 사용하는가?
 - ~~○ 요즘 개발 문화의 중심이니까! (Riotgames 개발자들이 강의할 때 강조하는 것!)~~
 - 비전공자의 학습에 굉장히 유용함

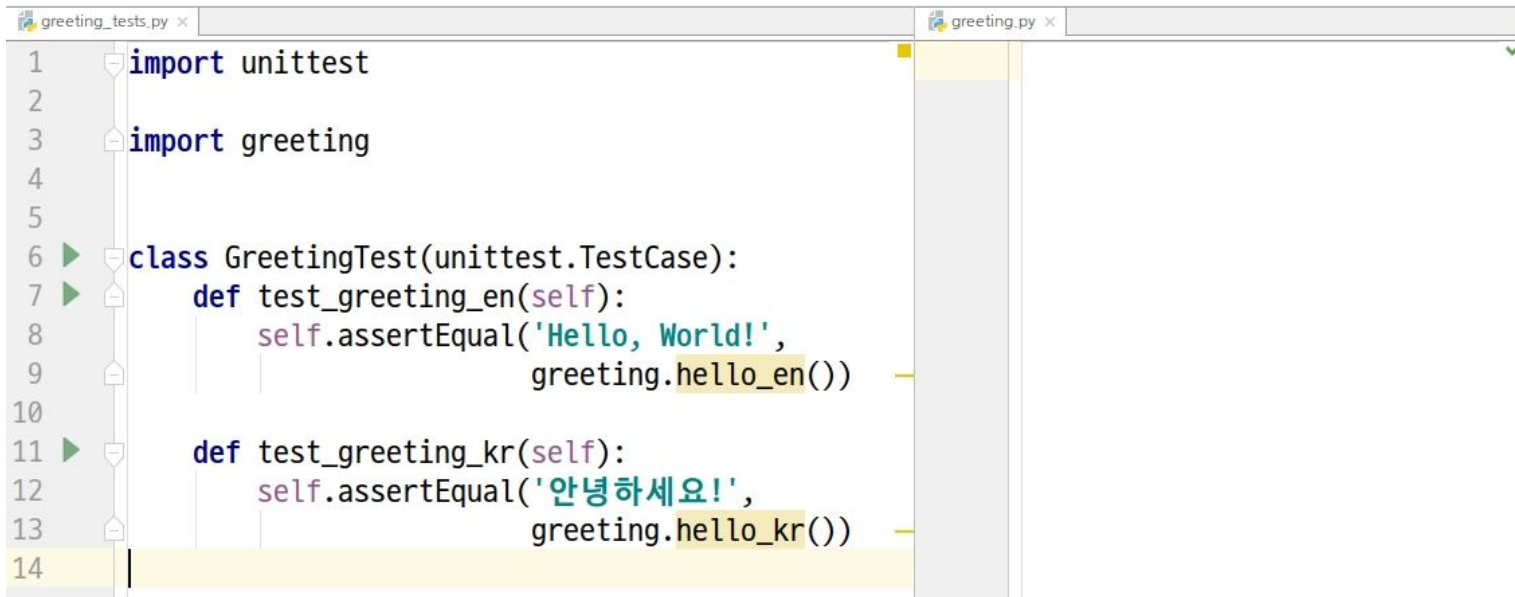
Unittest를 하기 위해 알아야 할 것

- Function(Class)
- source code import(include)
- 테스트 방법

```
40 def test_parse_salmon2_times(self):  
41     times = self.parser.parse_salmon2_times()  
42     self.assertEqual([GROUND_TRUTH['start_time'],  
43                     GROUND_TRUTH['end_time']],  
44                     times)
```

실습: 테스트 코드 작성

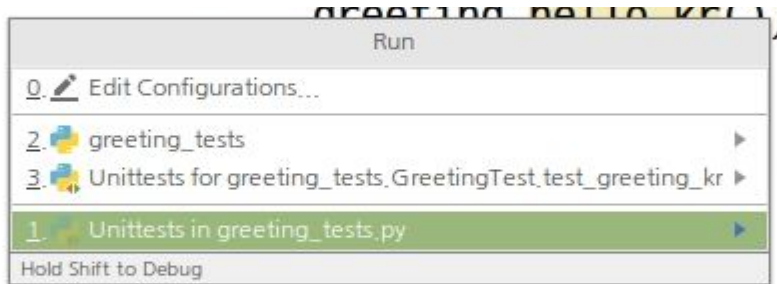
- 2개의 소스코드 생성
 - greeting_test.py / greeting.py



```
1 import unittest
2
3 import greeting
4
5
6 class GreetingTest(unittest.TestCase):
7     def test_greeting_en(self):
8         self.assertEqual('Hello, World!',
9                           greeting.hello_en())
10
11     def test_greeting_kr(self):
12         self.assertEqual('안녕하세요!',
13                           greeting.hello_kr())
14
```

실습: 테스트 실행

- 우리가 채워야할 소스코드(공백) 준비됨
- 테스트 코드 준비됨: 무언가 없다고 에러
- 해야할 일: 테스트 통과되게 만들기



Test Results	2 ms
greeting_tests	2 ms
GreetingTest	2 ms

AttributeError: module 'greeting' has no attribute 'hello_kr'

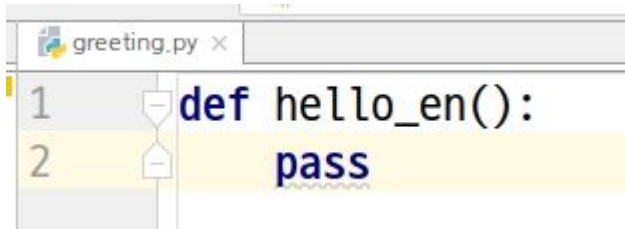
Ran 2 tests in 0.004s

FAILED (errors=2)

Process finished with exit code 1

Python3 Function

- def 키워드로 함수를 정의
- Python code block은 공백 4번으로 표시
 - Indentation은 python에서 굉장히 강한 문법 규칙이므로 꼭 지켜야함!
- 기대값과 결과값(이 경우엔 함수의 결과로 반환된 값)이 다르다며 Error
 - 아까와는 다른 Error



A screenshot of a code editor window titled 'greeting.py'. It shows two lines of Python code: line 1 is 'def hello_en():' and line 2 is 'pass'. The code is highlighted in yellow.

None != Hello, World!

Expected :Hello, World!

Actual :None

[<Click to see difference>](#)

Python3 Function return

- return: 함수의 결과를 반환하는 방법
- 그래도 테스트 통과하지 못 함!
 - 이유가 무엇일까?

```
greeting.py x
1 def hello_en():
2     return 'hello, world!'
3
```

hello, world! != Hello, World!

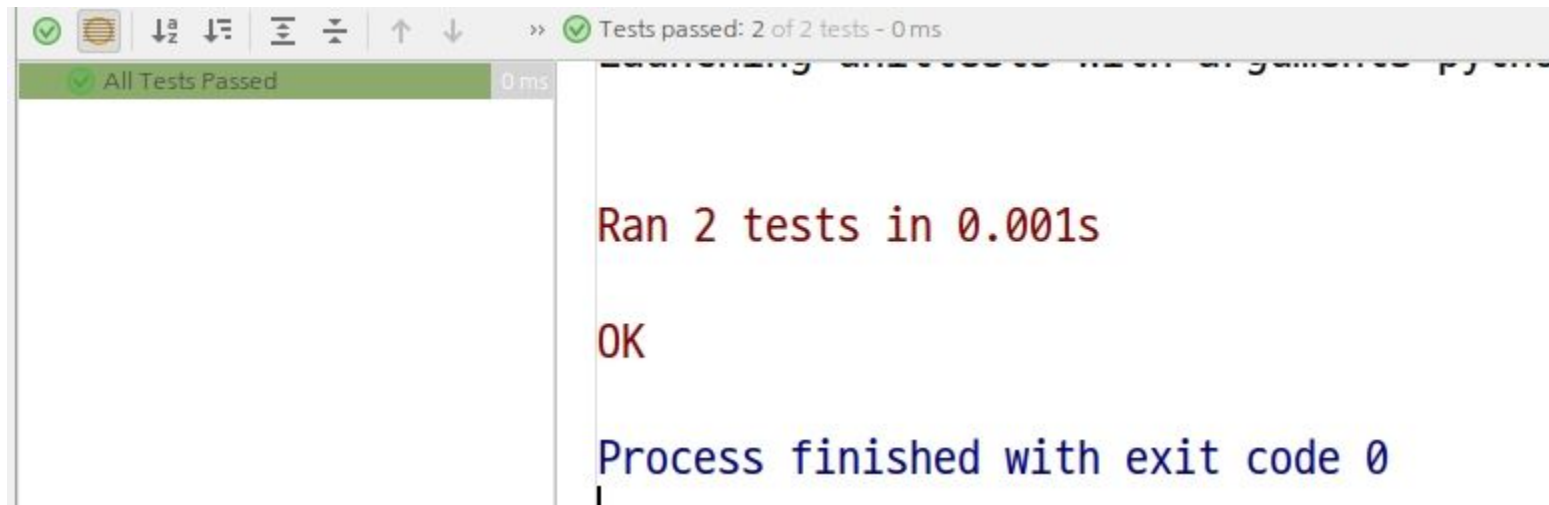
Expected :Hello, World!

Actual :hello, world!

[<Click to see difference>](#)

unittest 통과

- 이제 우리는 이 소스코드(greeting.py)가 정상적으로 동작하리라 기대할 수 있음



The screenshot shows a terminal window with a toolbar at the top. The toolbar includes icons for a checkmark, a list, and various navigation arrows. The status bar at the top right indicates "Tests passed: 2 of 2 tests - 0 ms". The main area of the terminal has a green header bar that says "All Tests Passed" followed by "0 ms". Below this, the output of the tests is displayed in a monospaced font:

```
Ran 2 tests in 0.001s

OK

Process finished with exit code 0
```

기타 유용한 정보

실습 숙제 제출

- 숙제 제출 기한: 2018. 09. 12. 18:00:00
 - 실습 전 날
- 파일 제목: AL_학번_이름_01
 - 파일 제목 다를 시 채점 안 합니다.

실습 숙제 제출할 것

- 2가지 파일을 제출
 - AL_학번_이름_숙제번호.zip
 - 소스코드
 - Pycharm을 사용했을 경우 Project 디렉터리에 .idea 같은 디렉터리는 제외
 - Jupyter + IPython을 사용했을 경우 'File - Download as' 에서 .py 다운로드 가능
 - AL_학번_이름_숙제번호.pdf
 - 보고서는 무조건 .pdf
 - .hwp, .doc 등 채점 안 함

실습 보고서에 들어가야 할 것

- 목표(할 일)
- 과제를 해결하는 방법
 - 알아야 할 것
- 과제를 해결한 방법
 - 주요 소스코드: 굳이 소스코드 전체를 붙일 필요는 없음
- 결과화면
 - 결과화면 설명(해석)
- 보고서는 기본적으로 '내가 숙제를 했음'을 보이는 것
 - 지나치게 대충 작성하면 의심하게 됨

출석부 및 실습 점수가 궁금하다면?

- 출석부 및 실습 채점표
 - 수업 시작 후 30분까지 지각, 이후 결석
 - 실습 딜레이 1일당: -2점

질문이 생기면?

- 이름: 문현수
- 전공: 통신및보안
- 과정: 석박사통합과정 8학기
- 연구실: 데이터네트워크연구실(공5633)
- 메일: munhyunsu@cs-cnu.org
- 알고리즘은 함께 해결해가는 과목이므로 과감하게 연락
- 이메일로 처리가 안 되는 급한일: 문자/전화 등