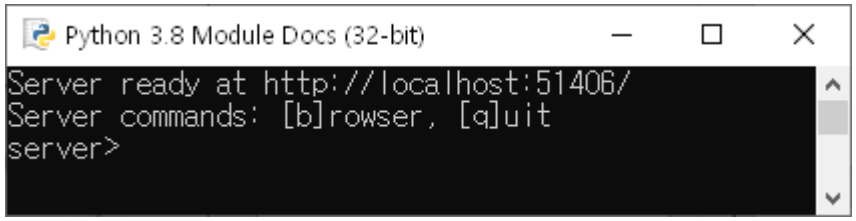
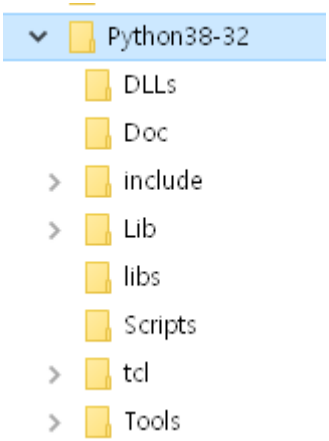
Module Docs
설치 모듈 문서

서버 프로그램, 브라우저에서 조회

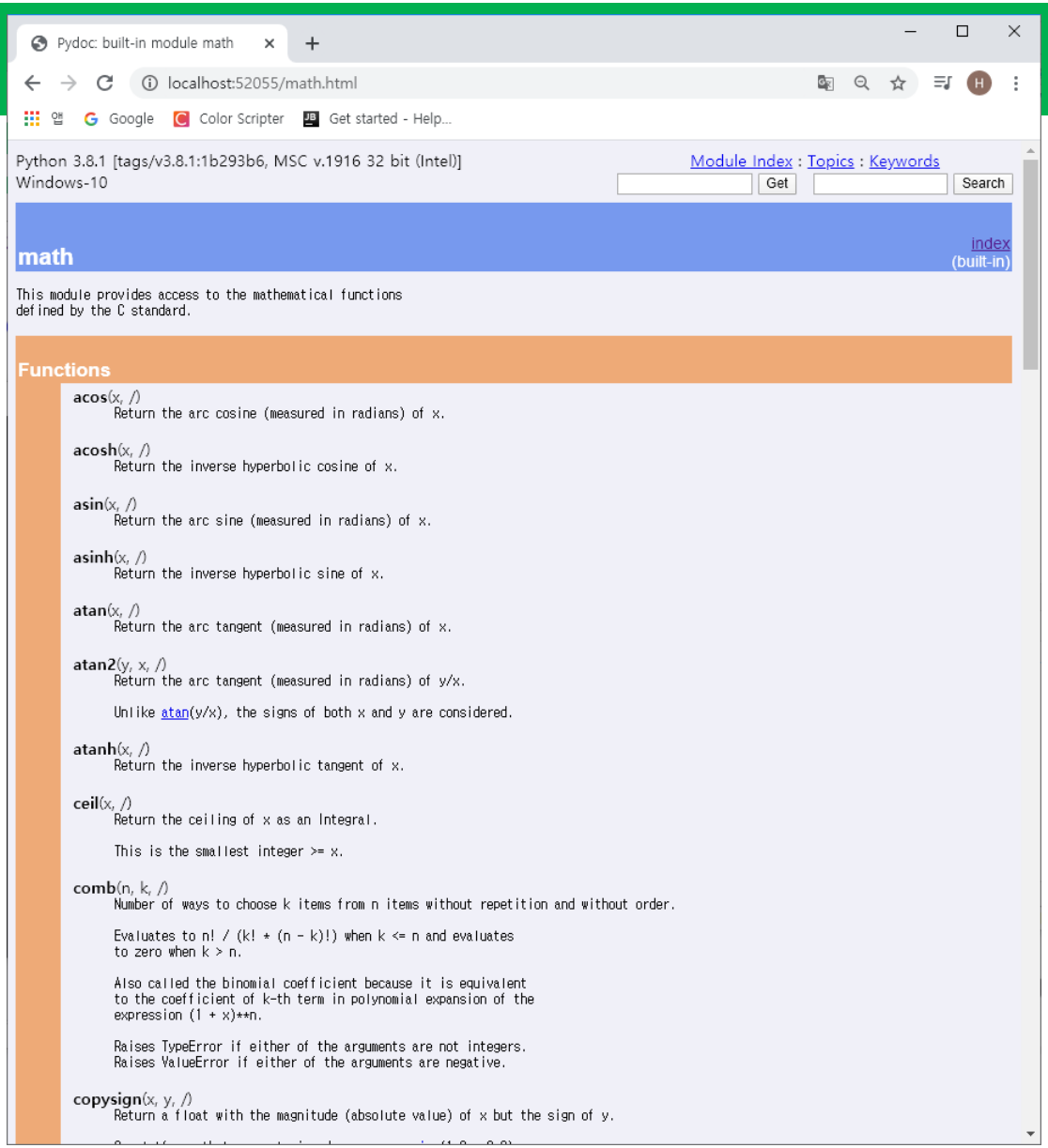


- **Built-in Modules**
  - 빌트인 모듈
- **D:₩Python38-32₩DLLs**
- **D:₩Python38-32₩lib**
  - 표준 모듈과 패키지
- **D:₩Python38-32₩lib₩site-packages**
  - 설치된 외부 모듈



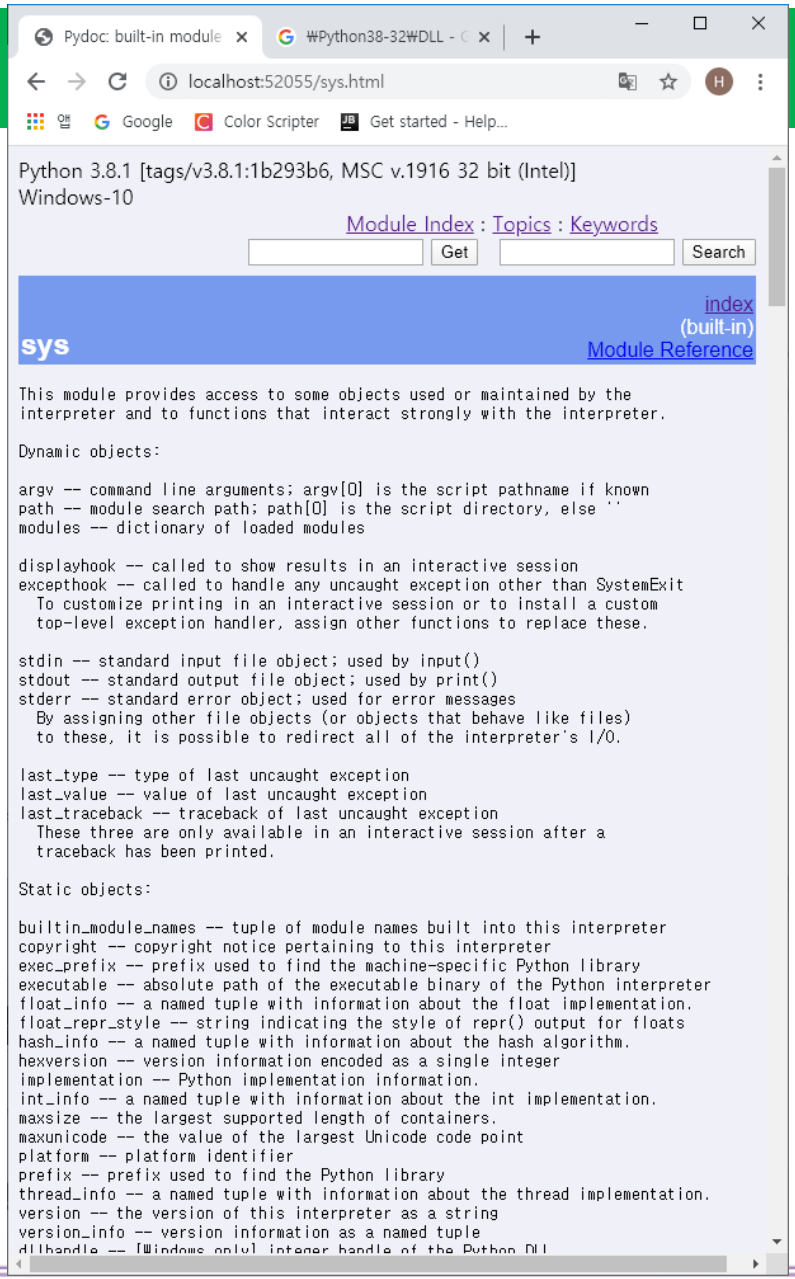
주요 빌트인 모듈

- **math**



주요 빌트인 모듈

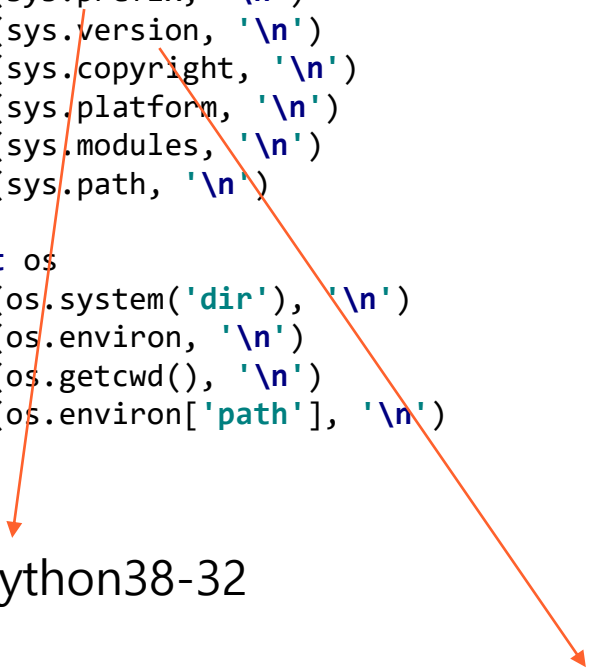
- **sys**


Pydoc: built-in module
localhost:52055/sys.html
Google
Color Scripter
Get started - Help...
Python 3.8.1 [tags/v3.8.1:1b293b6, MSC v.1916 32 bit (Intel)]
Windows-10
Module Index : Topics : Keywords
Get
Search
index
(built-in)
sys
Module Reference
This module provides access to some objects used or maintained by the
interpreter and to functions that interact strongly with the interpreter.
Dynamic objects:
argv -- command line arguments; argv[0] is the script pathname if known
path -- module search path; path[0] is the script directory, else ''
modules -- dictionary of loaded modules
displayhook -- called to show results in an interactive session
excepthook -- called to handle any uncaught exception other than SystemExit
To customize printing in an interactive session or to install a custom
top-level exception handler, assign other functions to replace these.
stdin -- standard input file object; used by input()
stdout -- standard output file object; used by print()
stderr -- standard error object; used for error messages
By assigning other file objects (or objects that behave like files)
to these, it is possible to redirect all of the interpreter's I/O.
last_type -- type of last uncaught exception
last_value -- value of last uncaught exception
last_traceback -- traceback of last uncaught exception
These three are only available in an interactive session after a
traceback has been printed.
Static objects:
builtin_module_names -- tuple of module names built into this interpreter
copyright -- copyright notice pertaining to this interpreter
exec_prefix -- prefix used to find the machine-specific Python library
executable -- absolute path of the executable binary of the Python interpreter
float_info -- a named tuple with information about the float implementation.
float_repr_style -- string indicating the style of repr() output for floats
hash_info -- a named tuple with information about the hash algorithm.
hexversion -- version information encoded as a single integer
implementation -- Python implementation information.
int_info -- a named tuple with information about the int implementation.
maxsize -- the largest supported length of containers.
maxunicode -- the value of the largest Unicode code point
platform -- platform identifier
prefix -- prefix used to find the Python library
thread_info -- a named tuple with information about the thread implementation.
version -- the version of this interpreter as a string
version_info -- version information as a named tuple

모듈 sys os 관련 코드

```
import sys
print(sys.prefix, '\n')
print(sys.version, '\n')
print(sys.copyright, '\n')
print(sys.platform, '\n')
print(sys.modules, '\n')
print(sys.path, '\n')

import os
print(os.system('dir'), '\n')
print(os.environ, '\n')
print(os.getcwd(), '\n')
print(os.environ['path'], '\n')
```

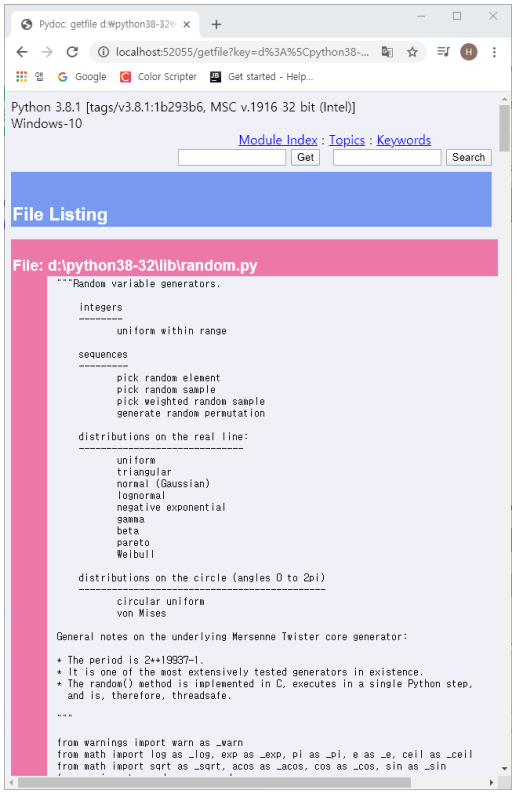


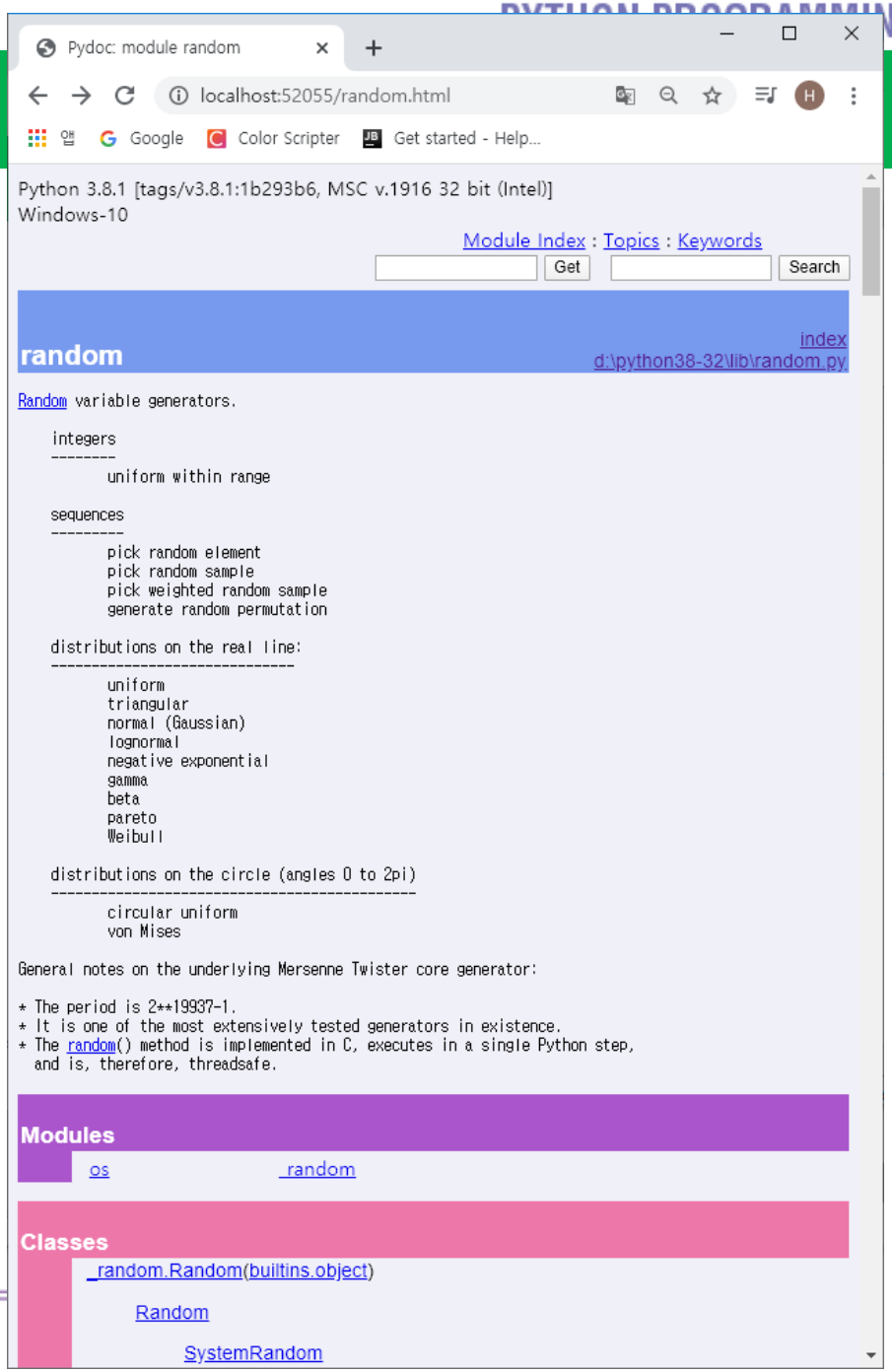
D:₩Python38-32

3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)]

주요 표준 모듈 random

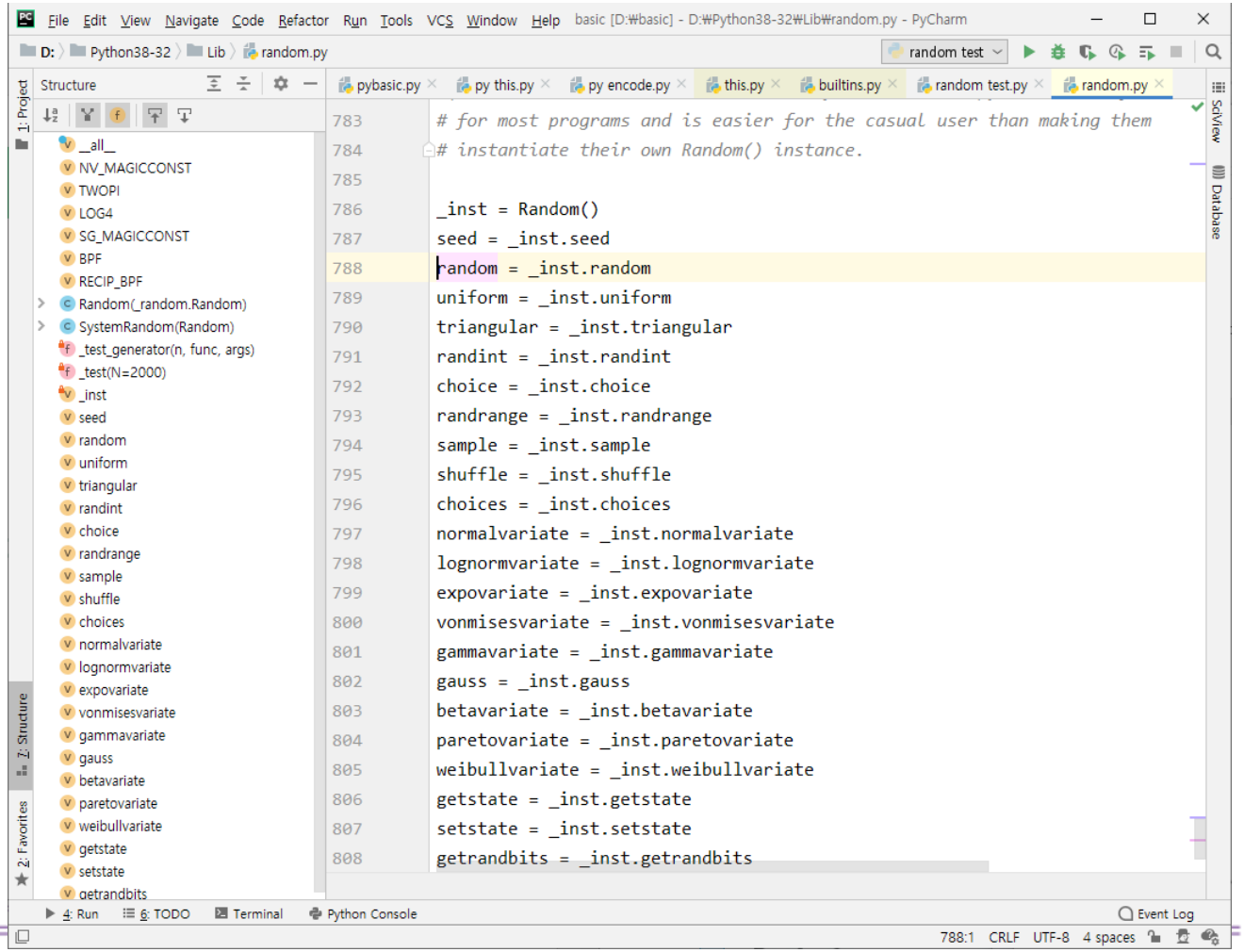
- **D:₩Python38-32₩lib**
  - random
    - **파일 random.py**





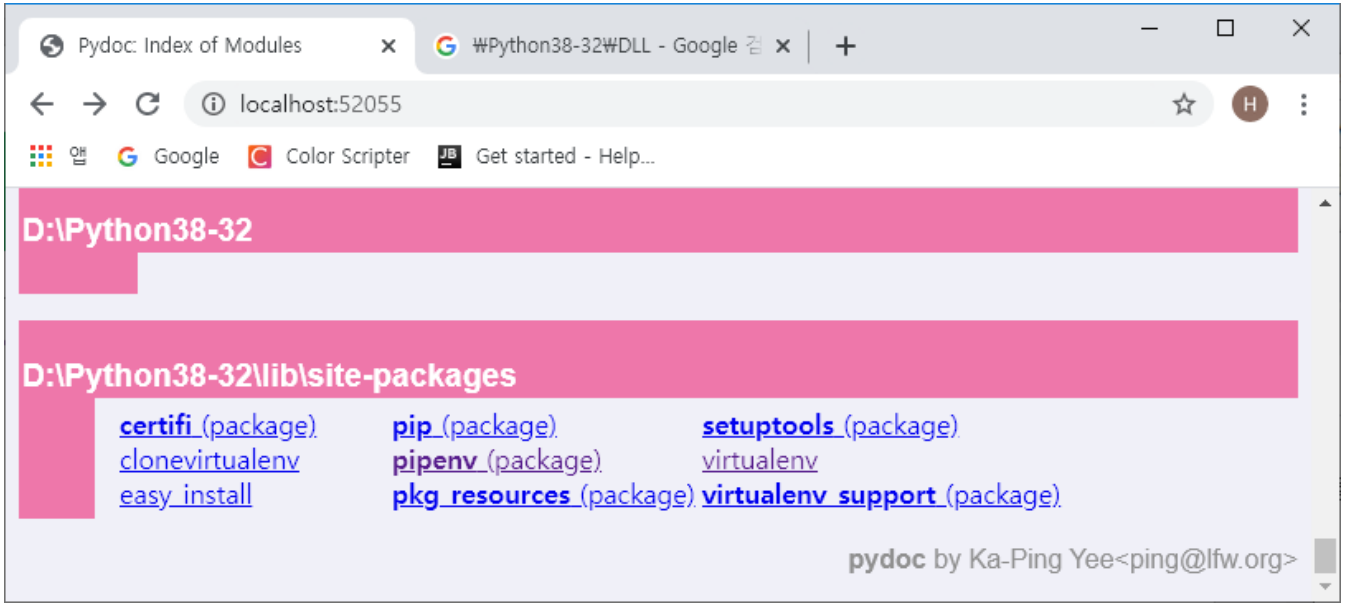
파일 random.py

- 파이참에서 직접 보기



설치된 외부 모듈

- **D:₩Python38-32₩lib₩site-packages**



Python
Cheat Sheet

Beginner's Python Cheat Sheet

Variables and Strings

Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.

Hello world

```
print("Hello world!")
```

Hello world with a variable

```
msg = "Hello world!"
print(msg)
```

Concatenation (combining strings)

```
first_name = 'albert'
last_name = 'einstein'
full_name = first_name + ' ' + last_name
print(full_name)
```

Lists

A list stores a series of items in a particular order. You access items using an index, or within a loop.

Make a list

```
bikes = ['trek', 'redline', 'giant']
```

Get the first item in a list

```
first_bike = bikes[0]
```

Get the last item in a list

```
last_bike = bikes[-1]
```

Looping through a list

```
for bike in bikes:
    print(bike)
```

Adding items to a list

```
bikes = []
bikes.append('trek')
bikes.append('redline')
bikes.append('giant')
```

Making numerical lists

```
squares = []
for x in range(1, 11):
    squares.append(x**2)
```

Lists (cont.)

List comprehensions

```
squares = [x**2 for x in range(1, 11)]
```

Slicing a list

```
finishers = ['sam', 'bob', 'ada', 'bea']
first_two = finishers[:2]
```

Copying a list

```
copy_of_bikes = bikes[:]
```

Tuples

Tuples are similar to lists, but the items in a tuple can't be modified.

Making a tuple

```
dimensions = (1920, 1080)
```

If statements

If statements are used to test for particular conditions and respond appropriately.

Conditional tests

```
equals            x == 42
not equal         x != 42
greater than      x > 42
  or equal to     x >= 42
less than         x < 42
  or equal to     x <= 42
```

Conditional test with lists

```
'trek' in bikes
'surly' not in bikes
```

Assigning boolean values

```
game_active = True
can_edit = False
```

A simple if test

```
if age >= 18:
    print("You can vote!")
```

If-elif-else statements

```
if age < 4:
    ticket_price = 0
elif age < 18:
    ticket_price = 10
else:
    ticket_price = 15
```

Dictionaries

Dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.

A simple dictionary

```
alien = {'color': 'green', 'points': 5}
```

Accessing a value

```
print("The alien's color is " + alien['color'])
```

Adding a new key-value pair

```
alien['x_position'] = 0
```

Looping through all key-value pairs

```
fav_numbers = {'eric': 17, 'ever': 4}
for name, number in fav_numbers.items():
    print(name + ' loves ' + str(number))
```

Looping through all keys

```
fav_numbers = {'eric': 17, 'ever': 4}
for name in fav_numbers.keys():
    print(name + ' loves a number')
```

Looping through all the values

```
fav_numbers = {'eric': 17, 'ever': 4}
for number in fav_numbers.values():
    print(str(number) + ' is a favorite')
```

User input

Your programs can prompt the user for input. All input is stored as a string.

Prompting for a value

```
name = input("What's your name? ")
print("Hello, " + name + "!")
```

Prompting for numerical input

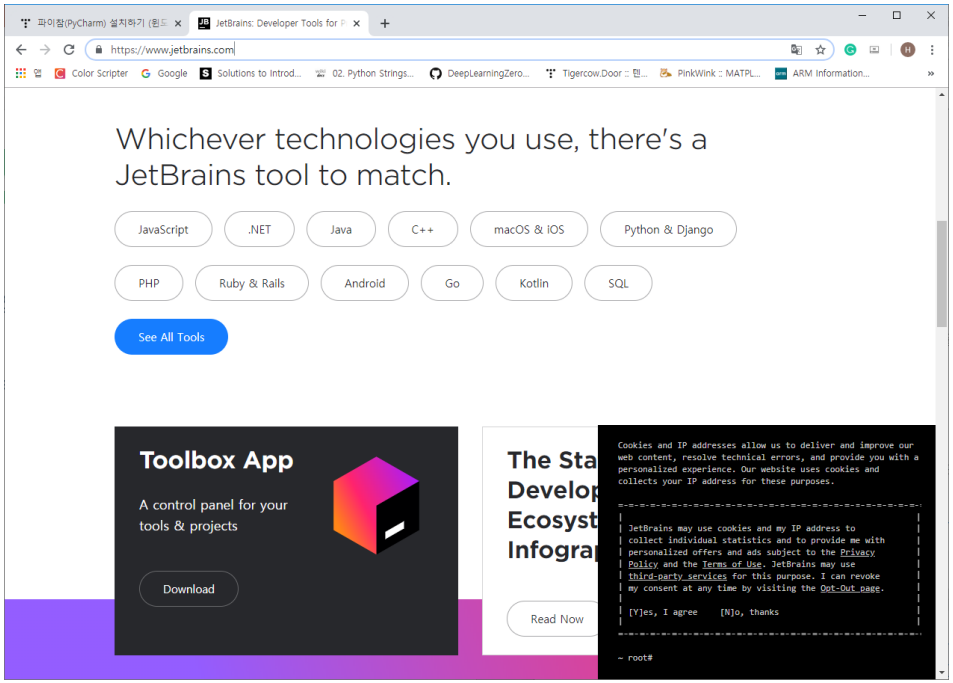
```
age = input("How old are you? ")
age = int(age)

pi = input("What's the value of pi? ")
pi = float(pi)
```

파이참 소개와 설치

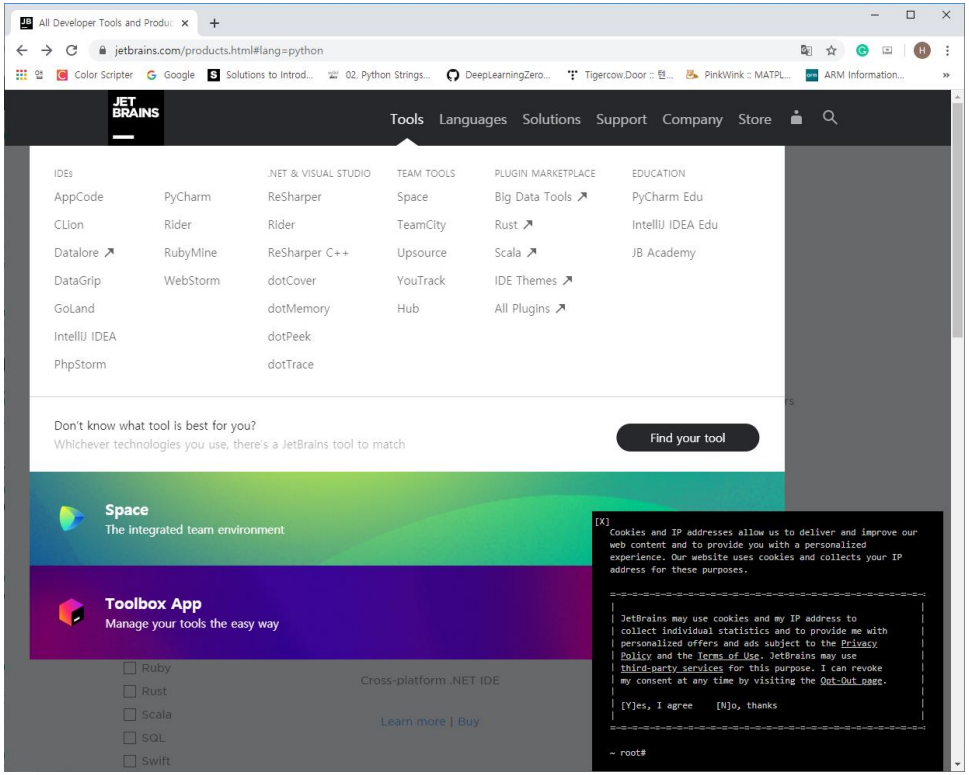
https://www.jetbrains.com/

- 파이참 소개
  - PyCharm은 Jetbrains 사에서 제작 및 배포하는 유료/무료 파이썬 IDE
  - Professional 버전은 유료
    - **학생이라면 학생 인증을 하고 무료로 사용**
      - **학생용 메일 필요 / *.ac.kr**
  - Professional / Community / education



최근 홈페이지

- **버전 별 차이**

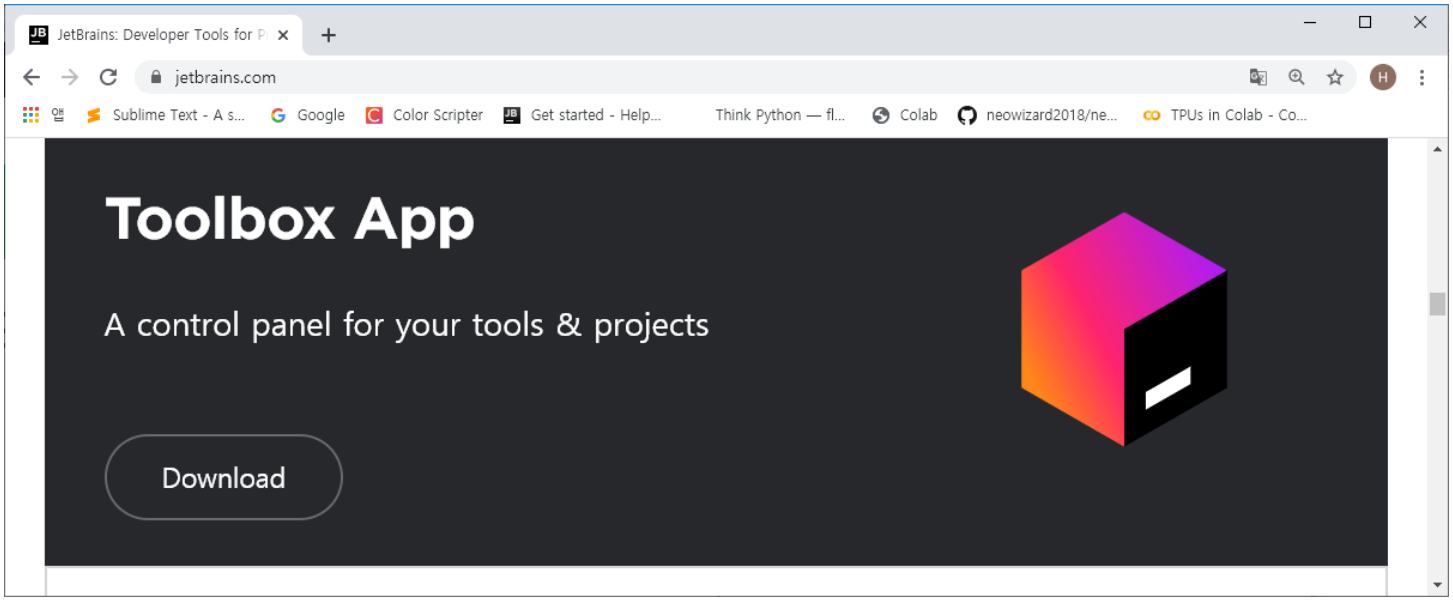


에디션 선택

	전문가	커뮤니티
지능적인 Python 에디터	✓	✓
그래픽 디버거 및 테스트 러너	✓	✓
탐색 및 리팩토링	✓	✓
코드 검사	✓	✓
VCS 지원	✓	✓
과학 도구	✓	✓
웹 개발	✓	
Python 웹 프레임워크	✓	
Python 프로파일러	✓	
원격 개발 기능	✓	
데이터베이스 및 SQL 지원	✓	
	무료 30일 체험	무료

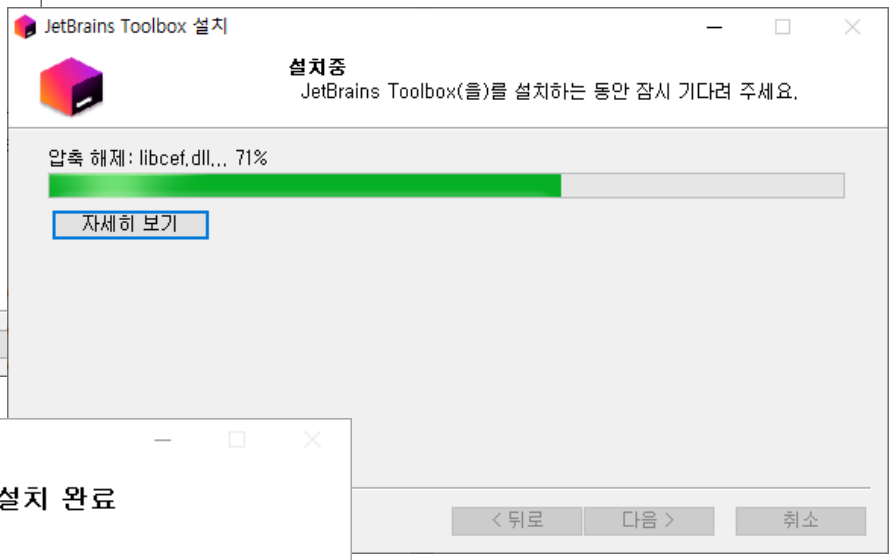
Toolbox

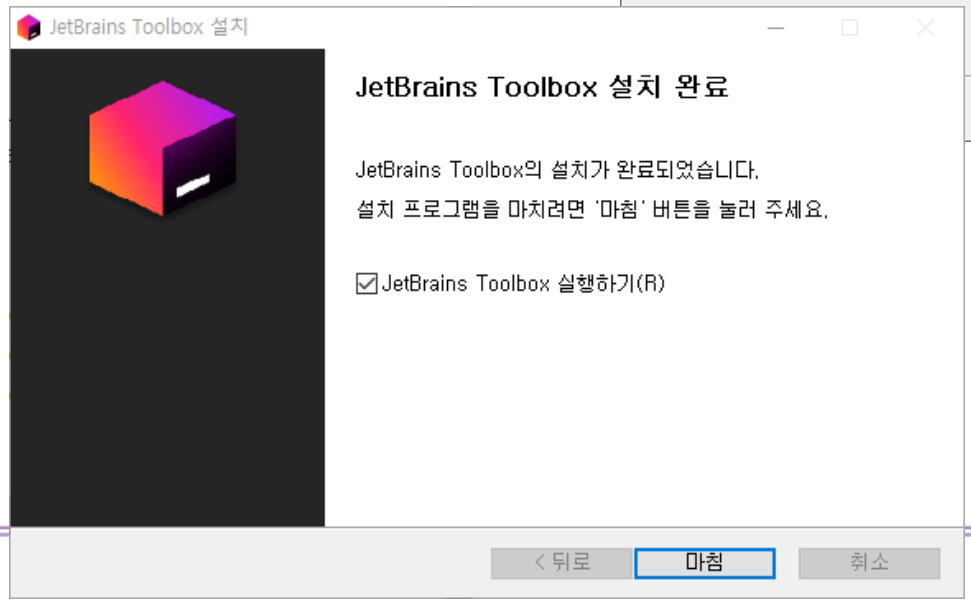
- 젯 브레인 사의 다양한 프로그램을 관리
  - 설치 프로그램 업그레이드
- 작성된 프로젝트 바로 열기



ToolBox 설치

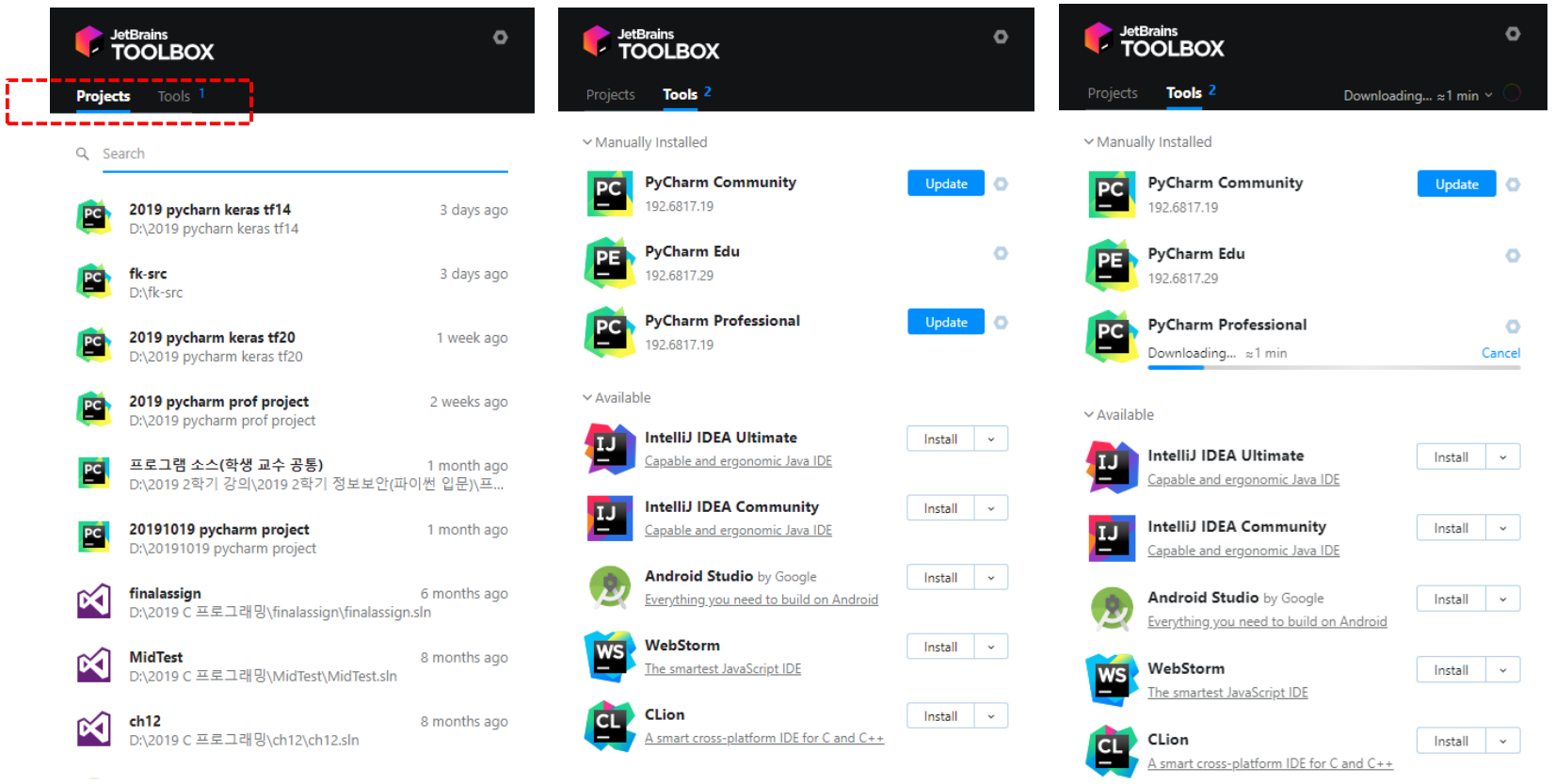

JetBrains Toolbox 설치
JetBrains Toolbox 설치를 시작합니다.
이 프로그램은 당신의 컴퓨터에 JetBrains Toolbox(을)를 설치할 것입니다.
설치를 시작하기 전 가능한 한 모든 프로그램을 종료하여 주시기 바랍니다. 이는 재부팅을 하지 않고서도 시스템 파일을 수정할 수 있게 해줍니다.
설치를 시작하시려면 '설치' 버튼을 눌러 주세요.
설치
취소


JetBrains Toolbox 설치
설치중
JetBrains Toolbox(을)를 설치하는 동안 잠시 기다려 주세요.
압축 해제: libcef.dll,,, 71%
자세히 보기
< 뒤로
다음 >
취소


JetBrains Toolbox 설치
JetBrains Toolbox 설치 완료
JetBrains Toolbox의 설치가 완료되었습니다.
설치 프로그램을 마치려면 '마침' 버튼을 눌러 주세요.
JetBrains Toolbox 실행하기(R)
< 뒤로
마침
취소

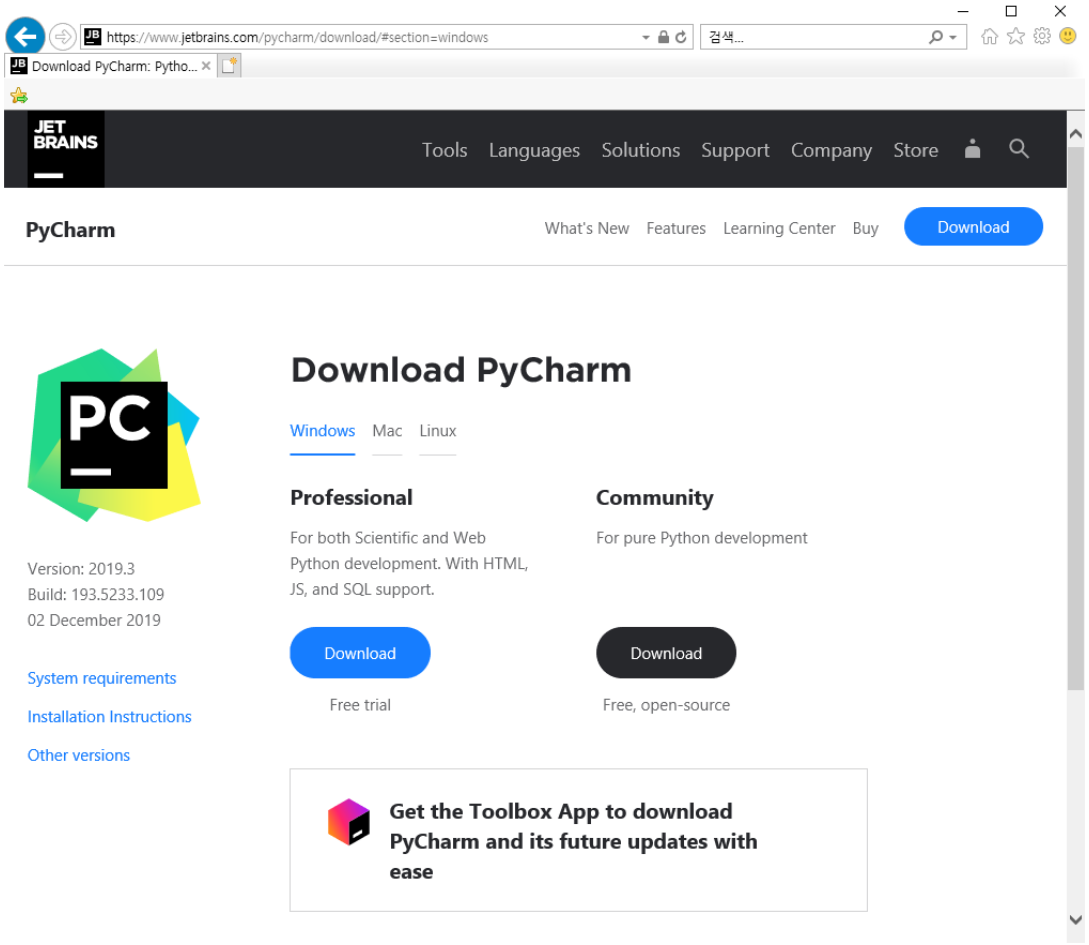
ToolBox 실행

- **작업 표시줄의 오른쪽 하단**
  - Projects: 프로젝트 바로 열기
  - Tools: 설치 도구 update

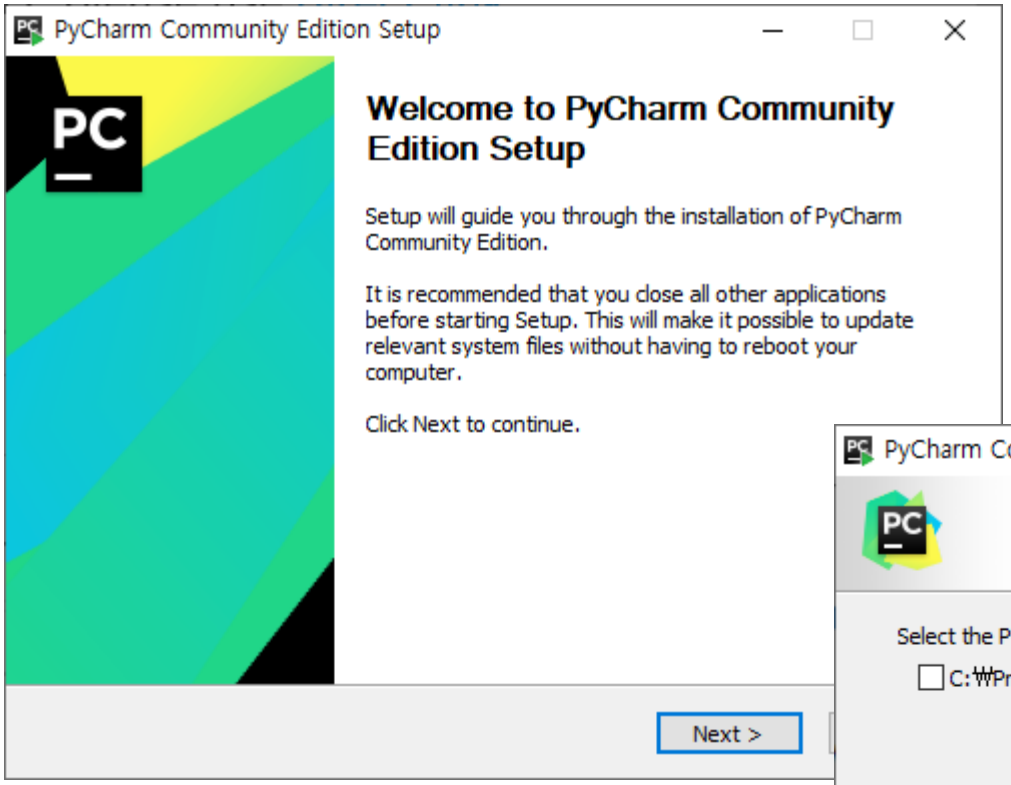


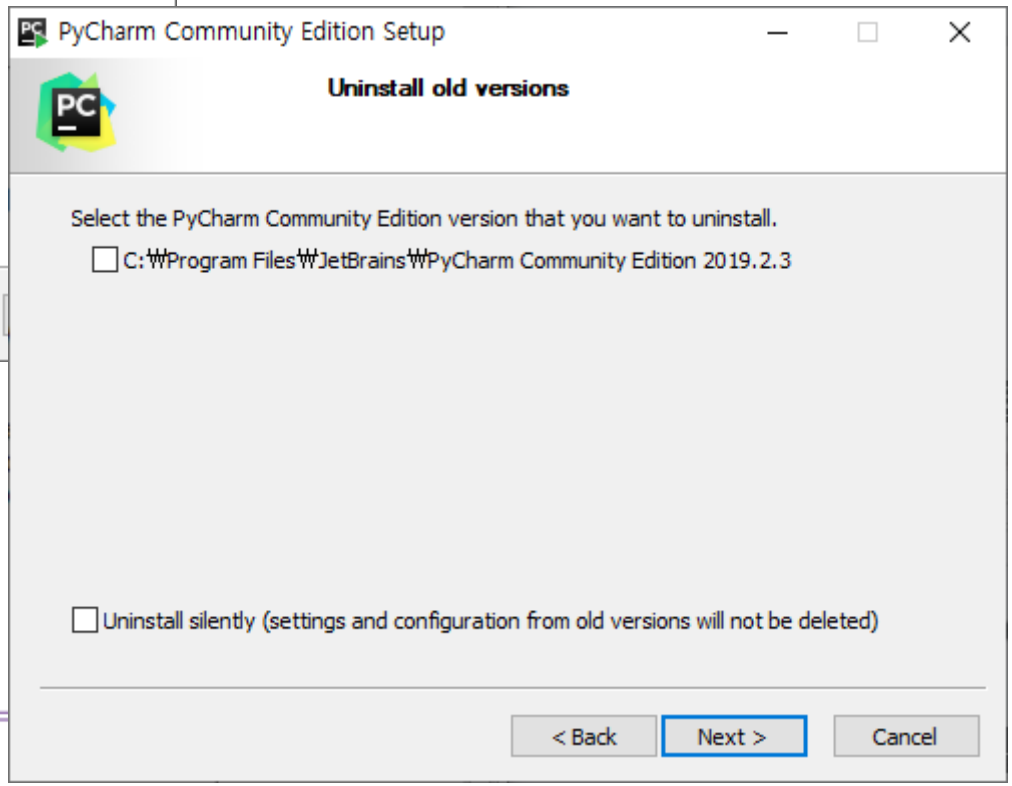
파이참 내려 받기

- 모두 사용 가능한 community 설치



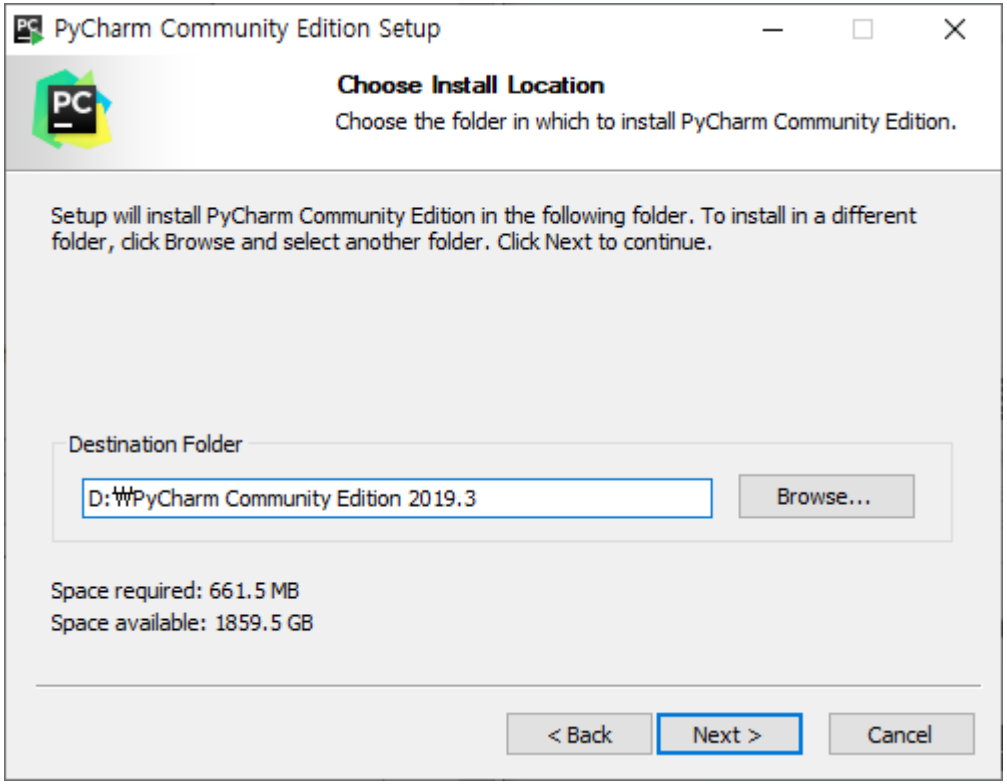
설치


PyCharm Community Edition Setup
Welcome to PyCharm Community Edition Setup
Setup will guide you through the installation of PyCharm Community Edition.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >


PyCharm Community Edition Setup
Uninstall old versions
Select the PyCharm Community Edition version that you want to uninstall.
C:₩Program Files₩JetBrains₩PyCharm Community Edition 2019.2.3
Uninstall silently (settings and configuration from old versions will not be deleted)
< Back
Next >
Cancel

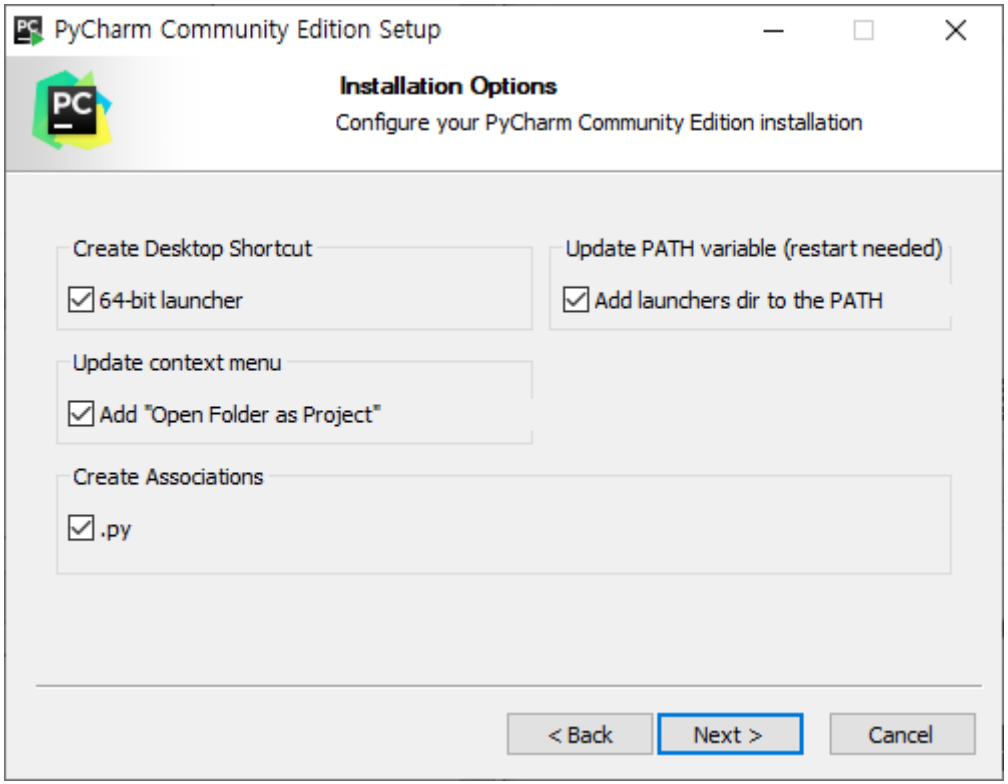
설치

- **기본 설치 폴더 수정**
  - C:₩Program Files₩JetBrains₩PyCharm Community Edition 2019.3

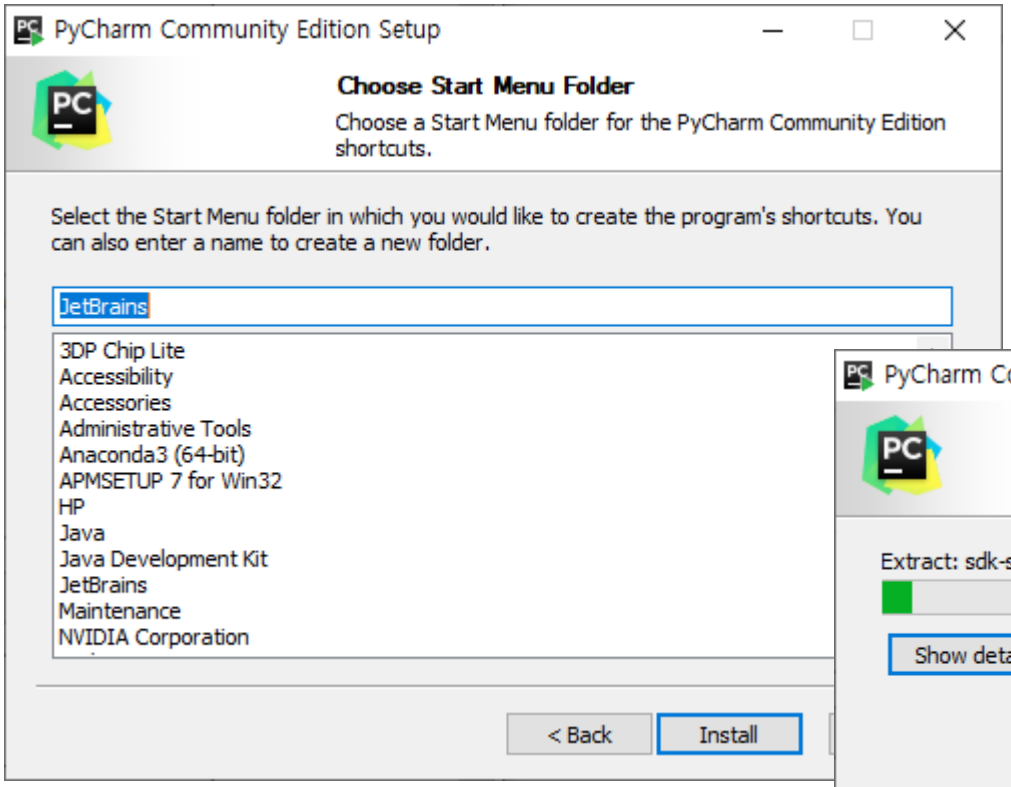


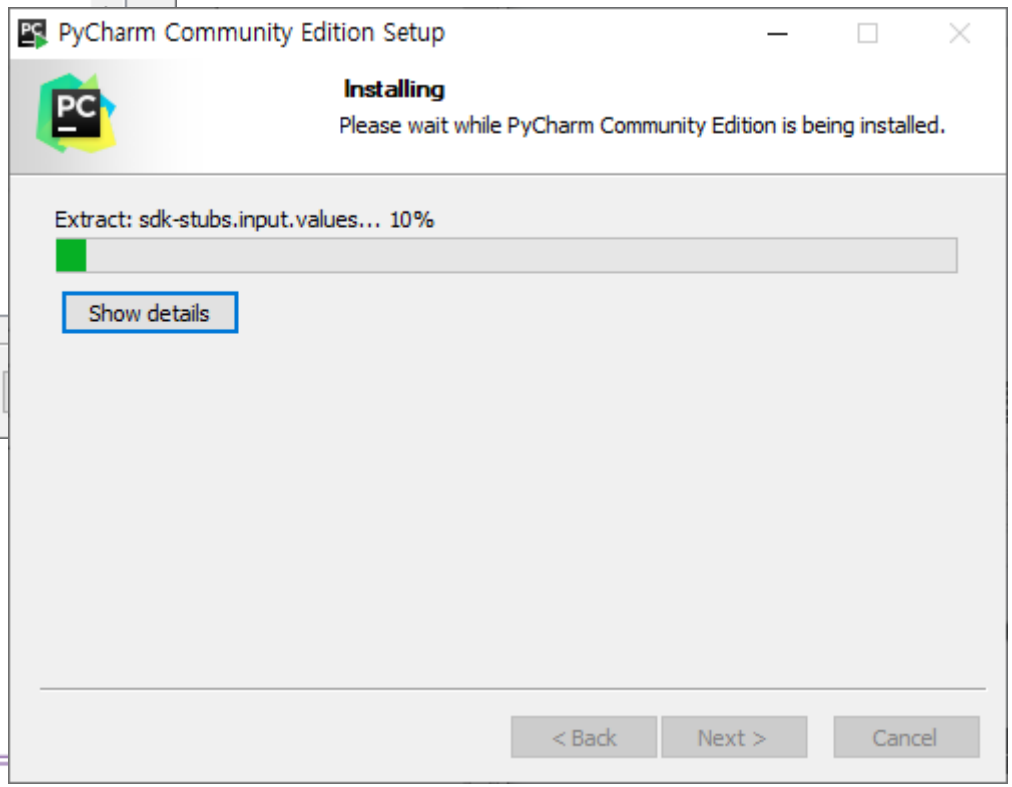
설치

- **모두 체크**

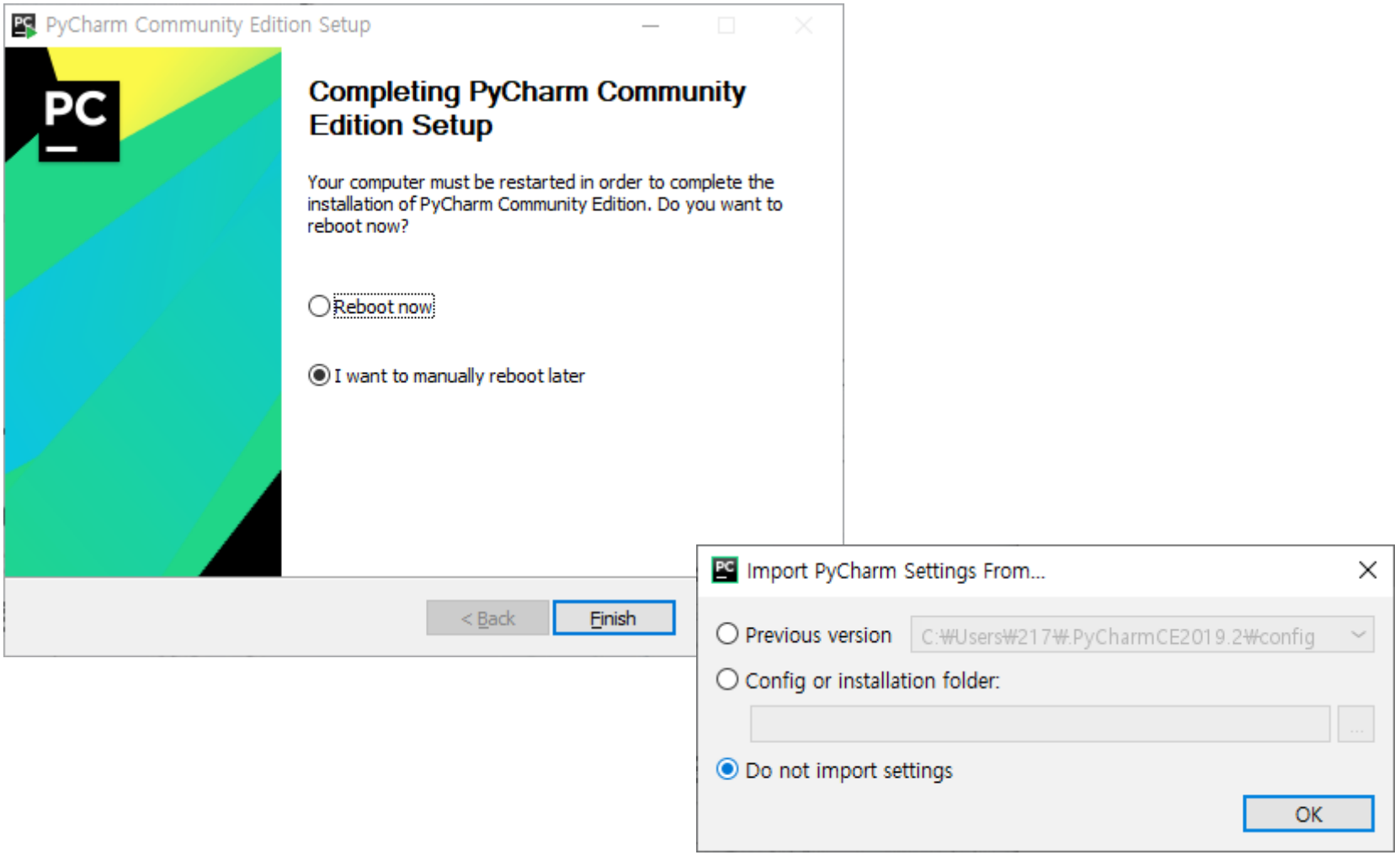

PyCharm Community Edition Setup
Installation Options
Configure your PyCharm Community Edition installation
Create Desktop Shortcut
64-bit launcher
Update PATH variable (restart needed)
Add launchers dir to the PATH
Update context menu
Add "Open Folder as Project"
Create Associations
.py
< Back
Next >
Cancel

설치


PyCharm Community Edition Setup
Choose Start Menu Folder
Choose a Start Menu folder for the PyCharm Community Edition shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
JetBrains
3DP Chip Lite
Accessibility
Accessories
Administrative Tools
Anaconda3 (64-bit)
APMSETUP 7 for Win32
HP
Java
Java Development Kit
JetBrains
Maintenance
NVIDIA Corporation
< Back
Install

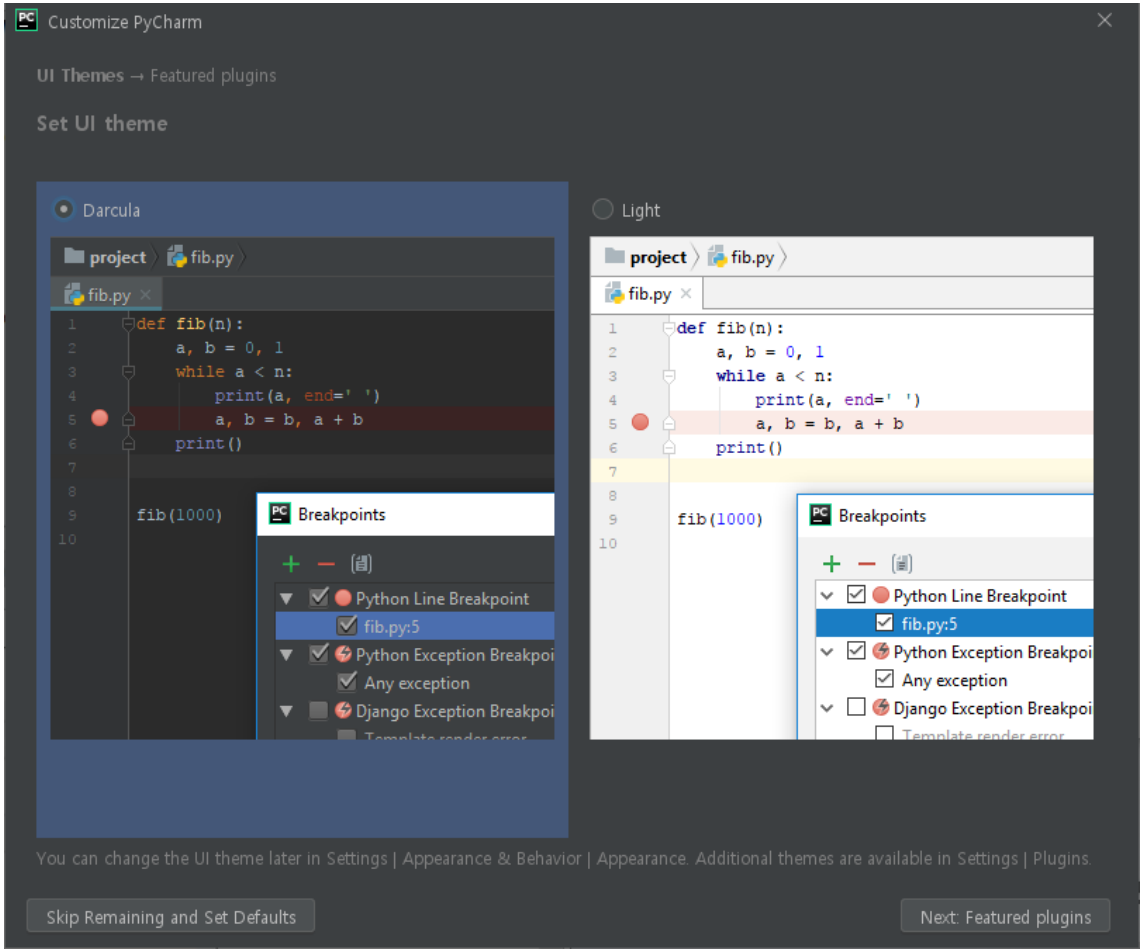

PyCharm Community Edition Setup
Installing
Please wait while PyCharm Community Edition is being installed.
Extract: sdk-stubs.input.values... 10%
Show details
< Back
Next >
Cancel

설치


PyCharm Community Edition Setup
Completing PyCharm Community Edition Setup
Your computer must be restarted in order to complete the installation of PyCharm Community Edition. Do you want to reboot now?
Reboot now
I want to manually reboot later
< Back
Finish
Import PyCharm Settings From...
Previous version
C:₩Users₩217₩.PyCharmCE2019.2₩config
Config or installation folder:
Do not import settings
OK

설치

- 원하는 색상 선택


Customize PyCharm
UI Themes → Featured plugins
Set UI theme
Darcula
Light
project
fib.py
def fib(n):
a, b = 0, 1
while a < n:
print(a, end=' ')
a, b = b, a + b
print()
fib(1000)
Breakpoints
Python Line Breakpoint
fib.py:5
Python Exception Breakpoi
Any exception
Django Exception Breakpoi
You can change the UI theme later in Settings | Appearance & Behavior | Appearance. Additional themes are available in Settings | Plugins.
Skip Remaining and Set Defaults
Next: Featured plugins

파이참 코딩 연습

convert.py

```
# convert.py
#     A program to convert Celsius temps to Fahrenheit
# by: Susan Computewell

def main():
    celsius = eval(input("What is the Celsius temperature? "))
    fahrenheit = 9/5 * celsius + 32
    print("The temperature is", fahrenheit, "degrees Fahrenheit.")

main()
```

futbal.py

```
# futval.py
#    A program to compute the value of an investment
#    carried 10 years into the future

def main():
    print("This program calculates the future value")
    print("of a 10-year investment.")

    principal = eval(input("Enter the initial principal: "))
    apr = eval(input("Enter the annual interest rate: "))

    for i in range(10):
        principal = principal * (1 + apr/100)

    print("The value in 10 years is:", principal)

main()
```