

파이참 소개와 설치

Pycharm 소개

• 파이참 소개

- PyCharm은 JetBrains사에서 제작 및 배포하는 유료/무료 파이썬 IDE
- Professional 버전은 유료
 - 학생이라면 학생 인증을 하고 무료로 사용
 - 학생용 메일 필요 / *.ac.kr
- Professional / Community / educational

• <https://www.jetbrains.com/pycharm/>

- 페이지 하단에 버전 비교표


	PyCharm Professional Edition	PyCharm Community Edition
Intelligent Python editor	✓	✓
Graphical debugger and test runner	✓	✓
Navigation and Refactorings	✓	✓
Code inspections	✓	✓
VCS support	✓	✓
Scientific tools	✓	
Web development	✓	
Python web frameworks	✓	
Python Profiler	✓	
Remote development capabilities	✓	
Database & SQL support	✓	

Download .exe

Free trial

Download .exe

Free, built on open-source



PyCharm Educational Edition

Have you ever wanted to learn programming with Python? Or maybe you're using Python to teach programming? Then we have the perfect tool for you.

[CHECK IT OUT!](#)

JetBrains(젯브레인즈) 사 소개

- 체코 회사

- 체코 프라하에 설립된 소프트웨어, 프로그래밍 언어, 그리고 솔루션을 개발 및 공급하는 기업
- IDE를 만드는 기업으로 유명
 - R&D 센터는 러시아 상트페테르부르크에 위치

- 요즘 소프트웨어 시장의 트렌드가 된 구독 모델을 일찍이 정착시킨 기업

- 구독권
 - 모든 IDE 제품이 1개월, 또는 1년 단위로 구독권을 끊어서 사용하는 방식이고 1년치 구독을 하면 첫 1개월차 시점에 출시됐던 버전을 영구적으로 제공한다. (대신 업데이트는 없다)
- 영리한 비즈니스 전략
 - 구독형 모델로 지속적인 수입원을 만들면서도 정기 결제를 부담스러워 하는 유저들까지 붙잡아 두는 전략
- 학생 무료 정책
 - 학생들에게는 거의 퍼 주다시피 할 만큼 모든 툴을 무료로 제공하기 때문에, 학교 다닐 때 JetBrains 제품의 맛을 본 사람은 나중에도 JetBrains 제품을 고집하는 경우가 많다.

JetBrains(젯브레인즈) 사의 고객 한국 개발자

- 2022년, 한국이 전년대비 유료 고객 증가율 1위(59%)를 달성
 - 전체 유료 고객 수 기준으로는 세계 8위
- 특이하게 오픈 소스 소프트웨어 개발용으로 사용할 경우에는 무료
 - 대신 유료판을 따로 낸다던가 유료 기술지원을 하면 안 되는 등 제한 조건이 좀 빡빡한 편

pycharm 설치

• 다운로드

- <https://www.jetbrains.com/>
- 2022년 4월 현재
 - **pycharm-community-2022.1.exe**

• 버전

- Professionals
- Community

Download PyCharm

Windows macOS Linux

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

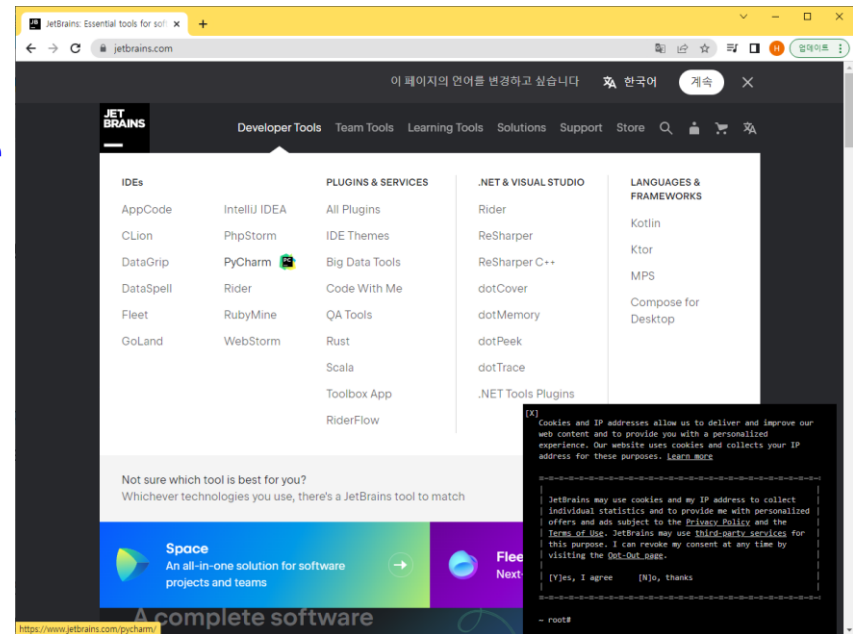
Free 30-day trial available

Community

For pure Python development

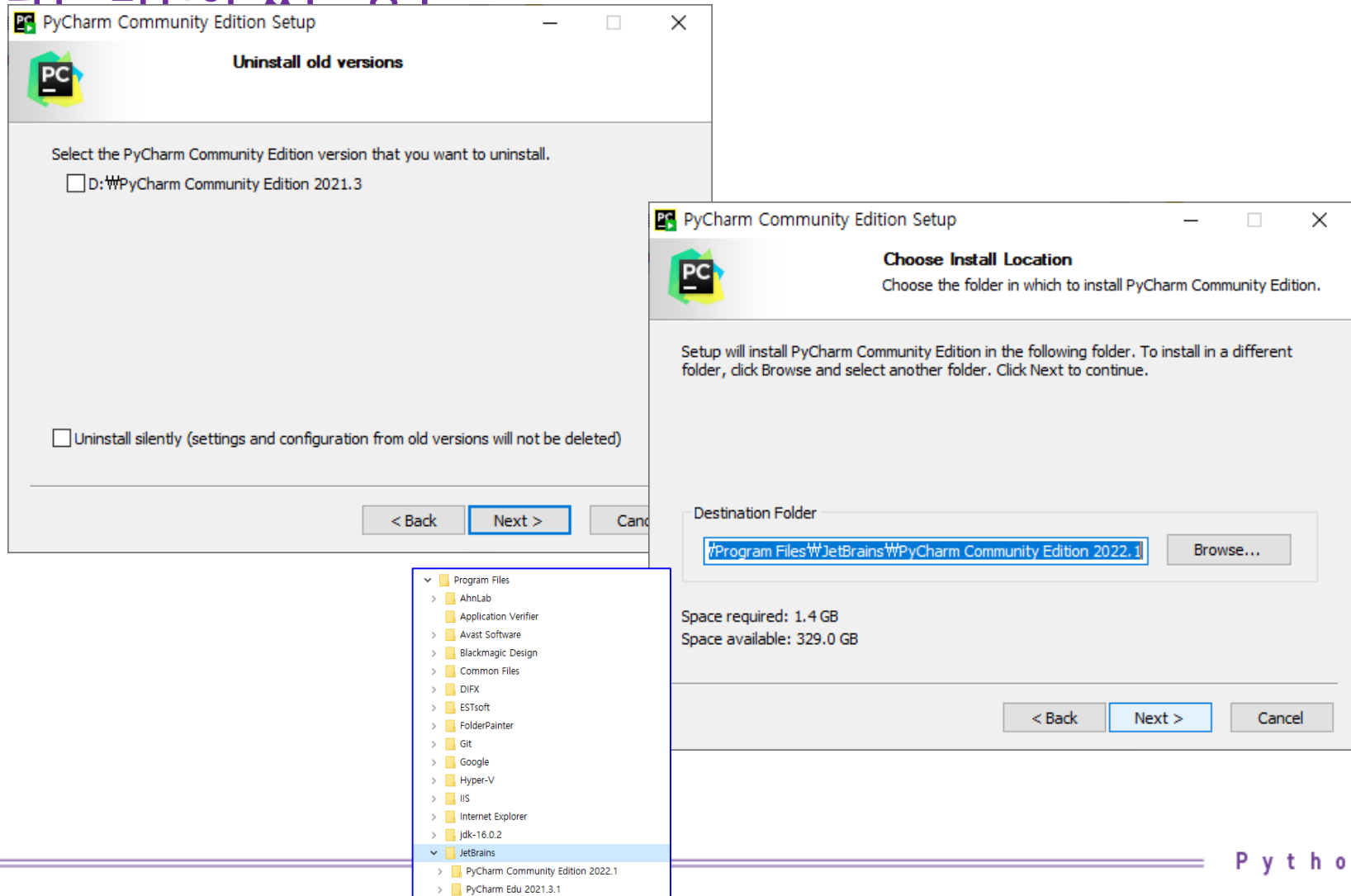
Download

Free, built on open-source

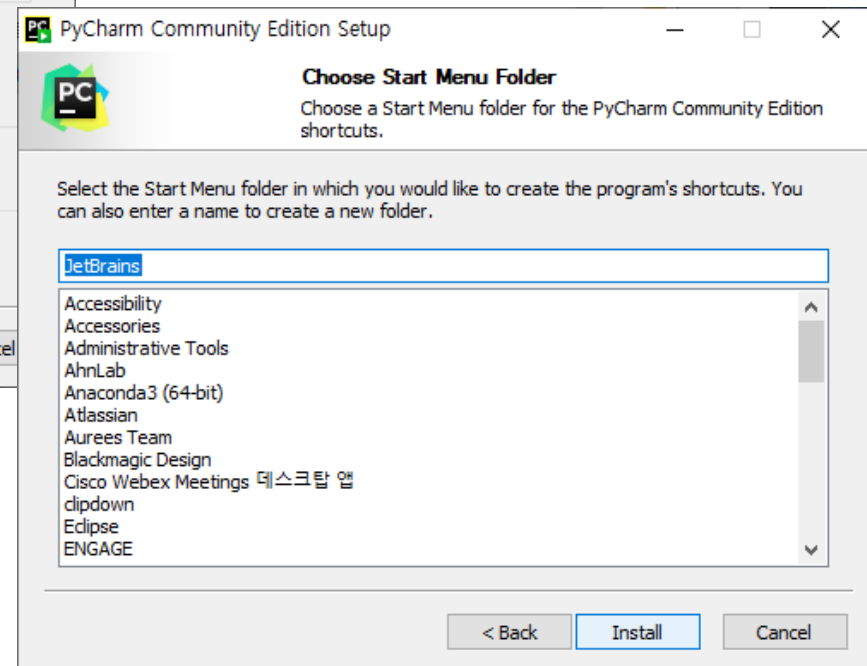
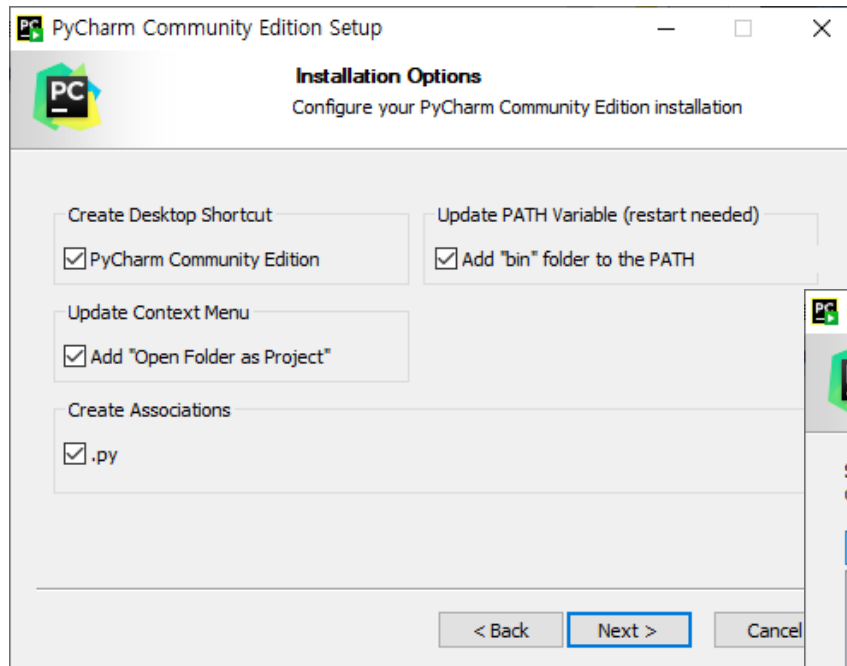


설치과정 1/3

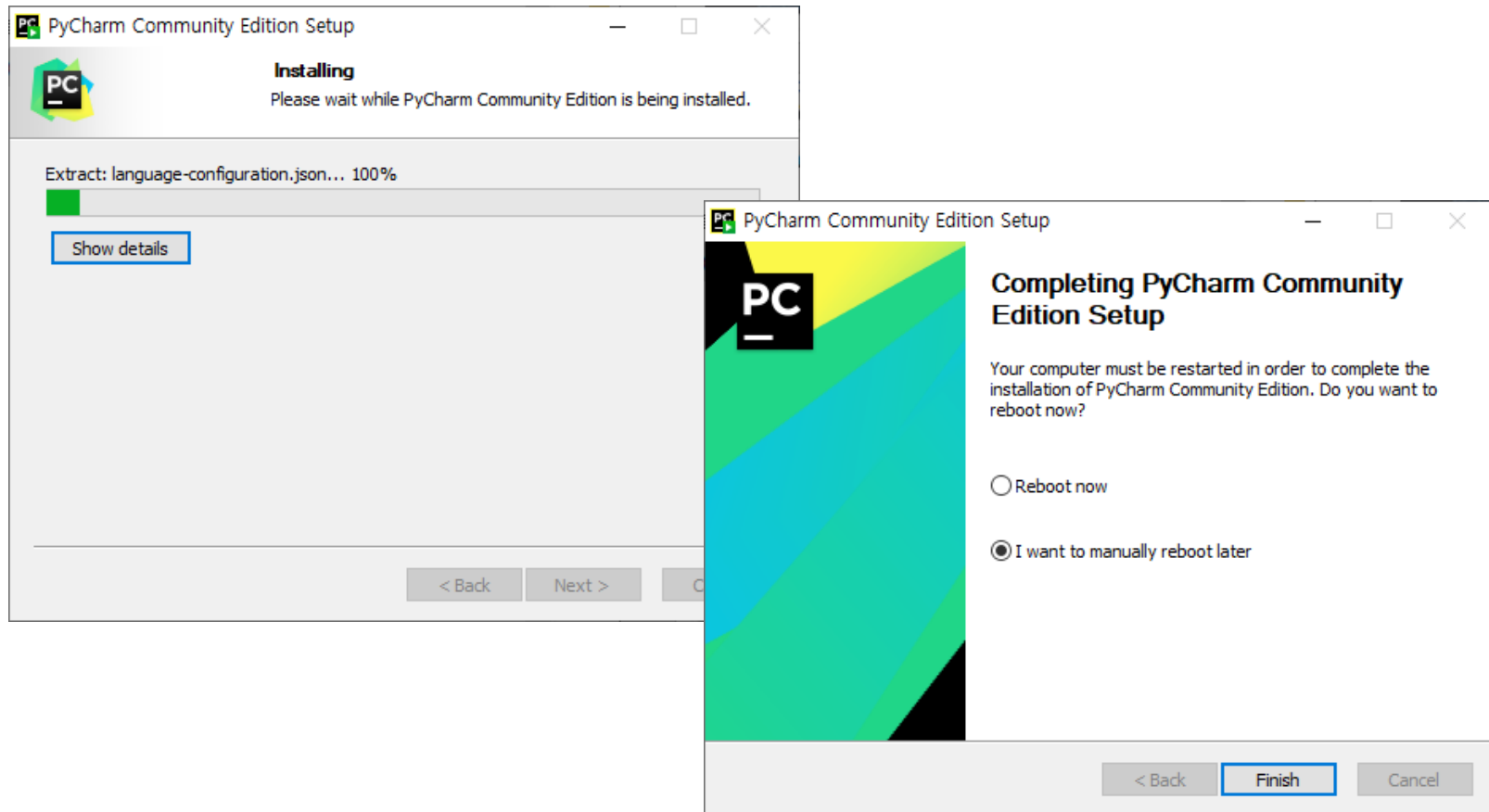
• 다른 버전이 있는 경우



설치과정 2/3

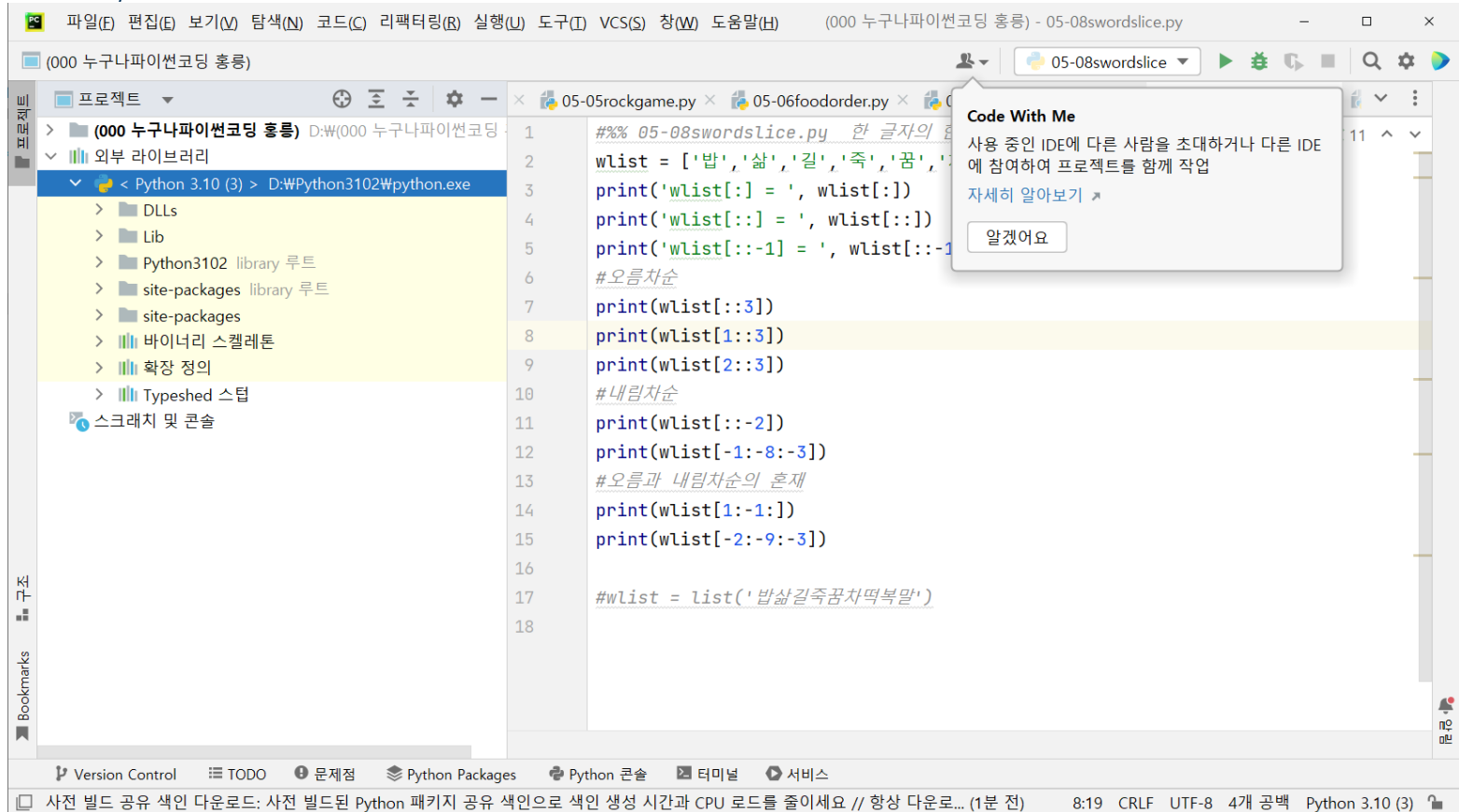


설치과정 3/3



실행

- 이전 버전의 설정으로 지정
 - 화면, 한글



새로운 기능 문서

- <https://www.jetbrains.com/ko-kr/pycharm/whatsnew/>

PyCharm 2022.1의 새로운 기능

사용자 지정 패키지 저장소에 대한 인증 지원, TypedDict에 대한 향상된 지원 및 새로운 Docker용 서비스 UI

이번 릴리스에서는 PyCharm의 전반적인 사용자 경험을 개선하고 생산성을 높일 수 있도록 일부 기능을 다듬는 데 중점을 두었습니다. 이전에 구현된 일부 기능을 마무리하고 특정 워크플로를 더 간단하게 만드는 데 우선순위를 두었습니다.

Downloads

IDE

사용자 지정 패키지 저장소에 액세스하도록 기본 HTTP 인증을 구성하고 수동 설치를 위해 터미널로 전환하지 않고도 PyCharm을 통해 종속성을 쉽게 관리할 수 있습니다. Python Packages(Python 패키지) 도구 창으로 이동하여 톨니바퀴 아이콘을 클릭한 다음, 대화상자 창에서 더하기 기호를 클릭하고 저장소 URL을 추가한 후, Basic HTTP(기본 HTTP) 옵션을 선택하여 필수 자격 증명을 입력합니다.

Package Repository

Name: Custom_repository

Repository URL: https://packages.jetbrains.team/pypi/p/py/space-pypi-test

Authorization: None Basic HTTP

Login: Valeria.Letusheva

Password:

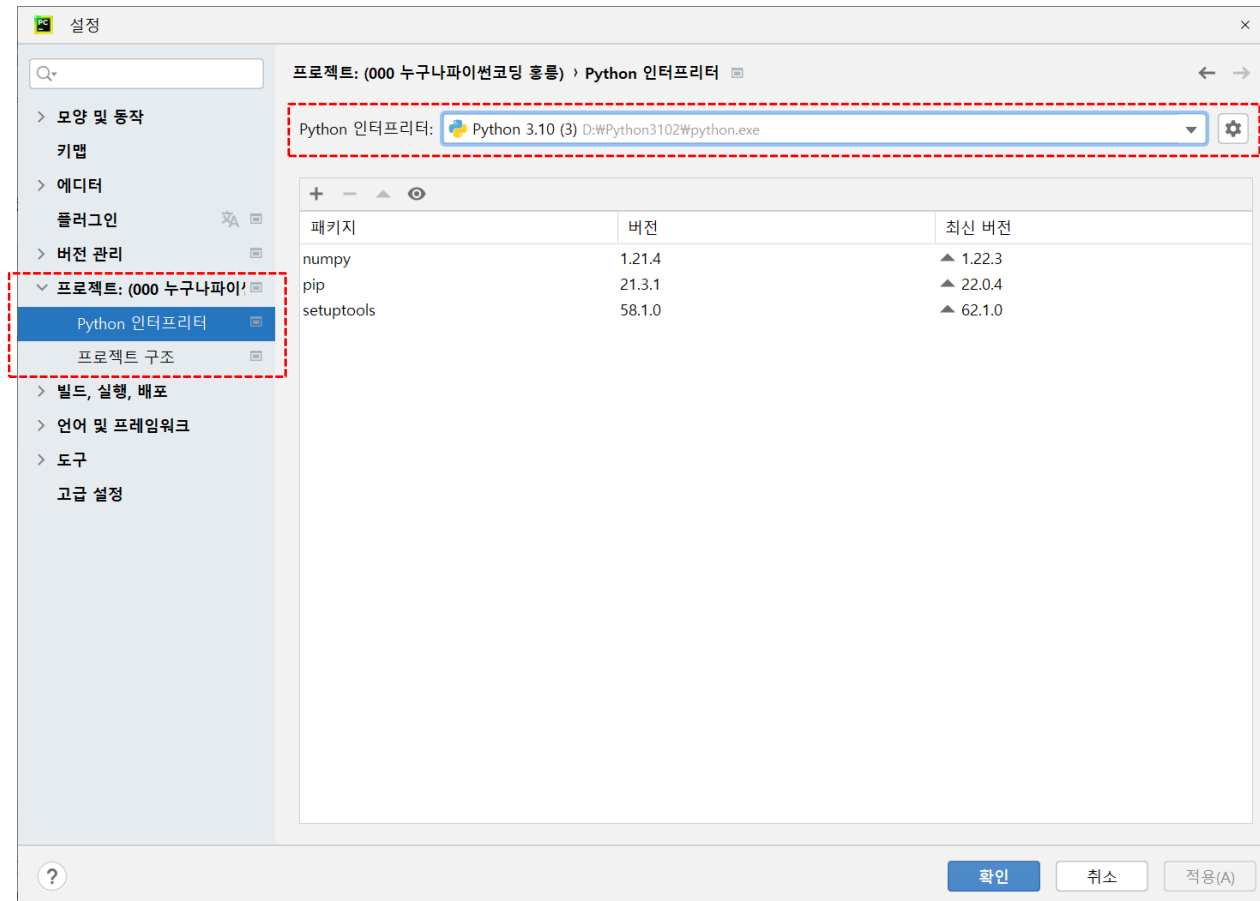
Quiz

- 다음 파이참에 대한 설명 중 잘못된 것은?
 - 홈페이지 jetbrains.com에서 내려 받는다.
 - 파이썬 언어로 프로그램을 개발할 수 있는 IDE이다.
 - 커뮤니티 버전은 무료이다.
 - 파이참만으로 개발환경을 설정할 수 있다.

개발환경 설정

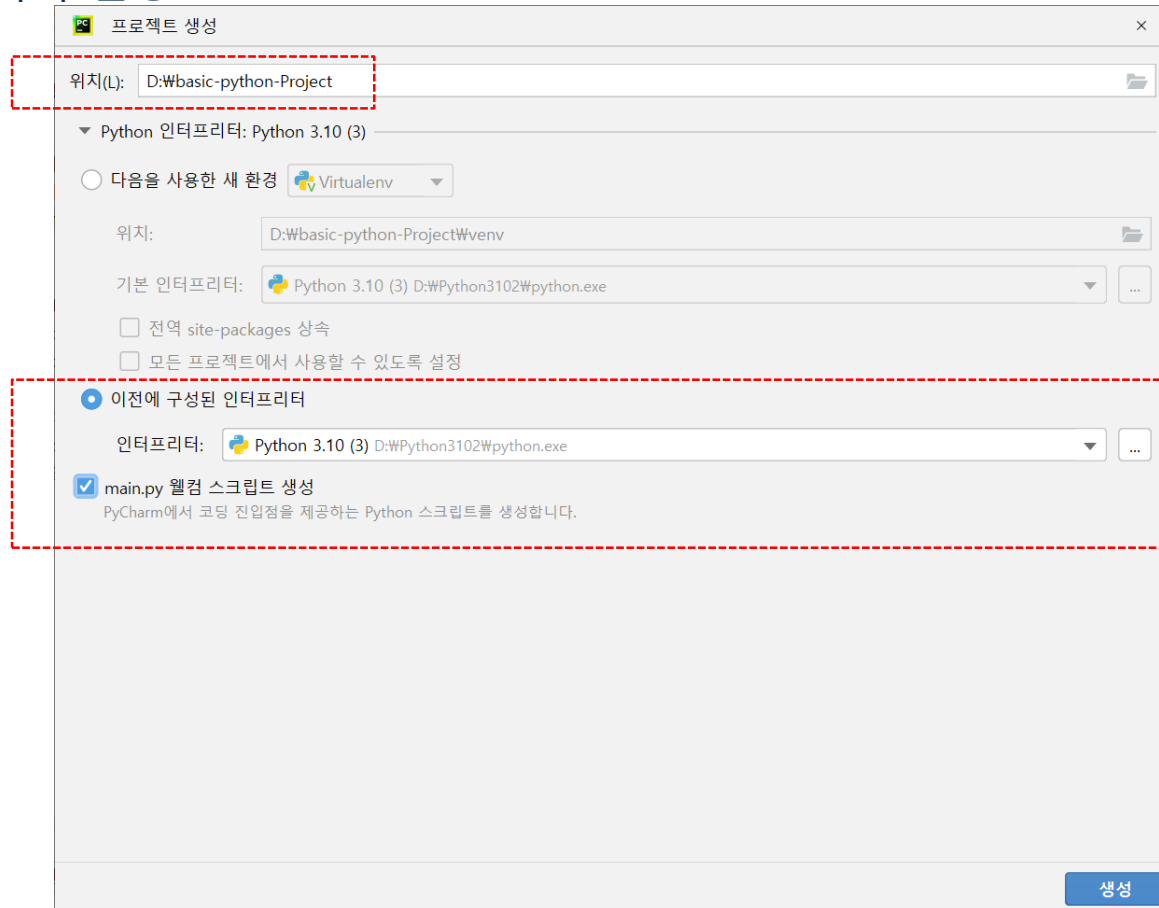
파이썬 인터프리터 설정: 2022.01 버전

- 메뉴 파일 | 설정
 - 프로젝트 ...



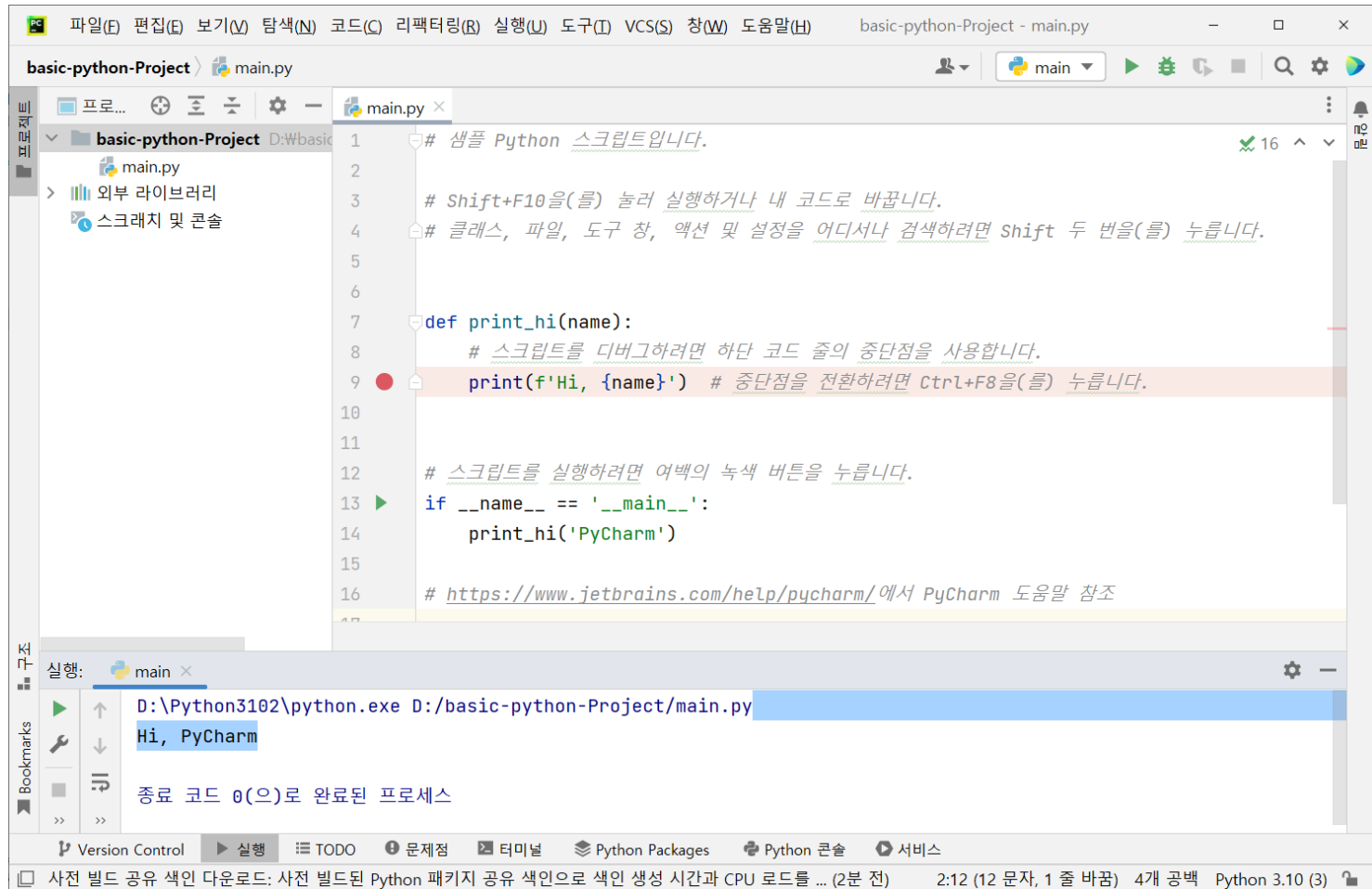
새 프로젝트 생성

- 메뉴 파일 | 새 프로젝트...
 - 인터프리터 설정



생성된 프로젝트

• 자동 생성된 파일 main.py 실행



새로운 파일 생성 후 실행

• 파일 if.py 작성

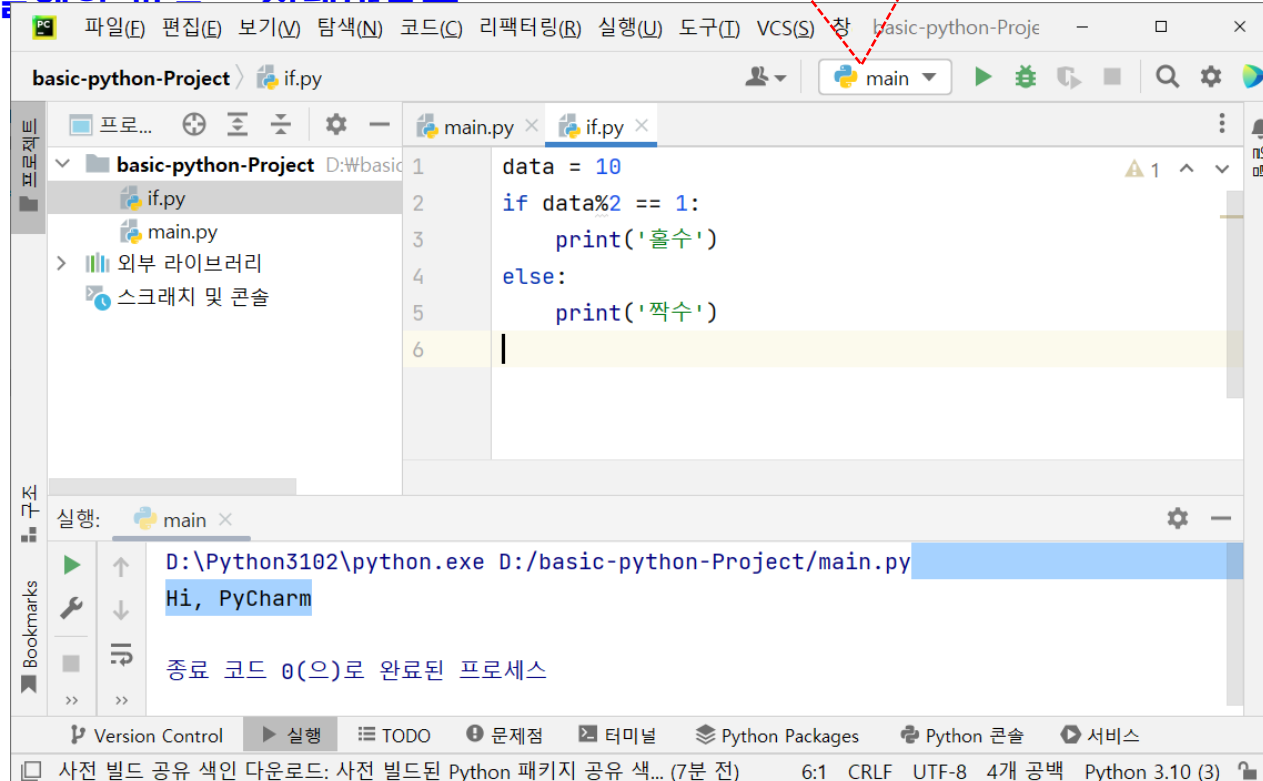
– Ctrl + shift + F10

• 현재 편집한 파일 실행

– Alt + shift + F10

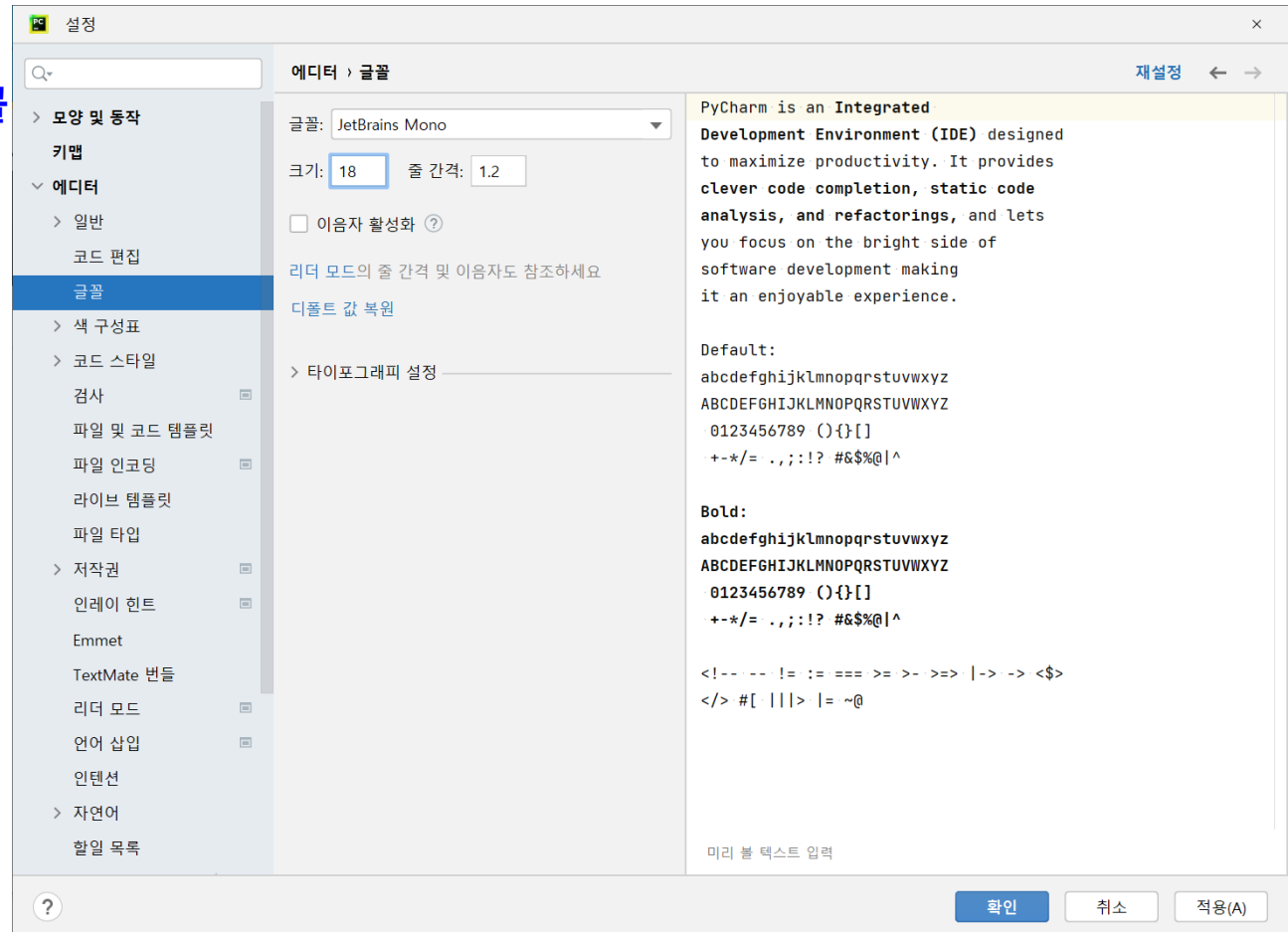
• 실행한 파일의 서태치드로

기본은 이전에 실행한 파일이 실행



편집기 폰트 설정

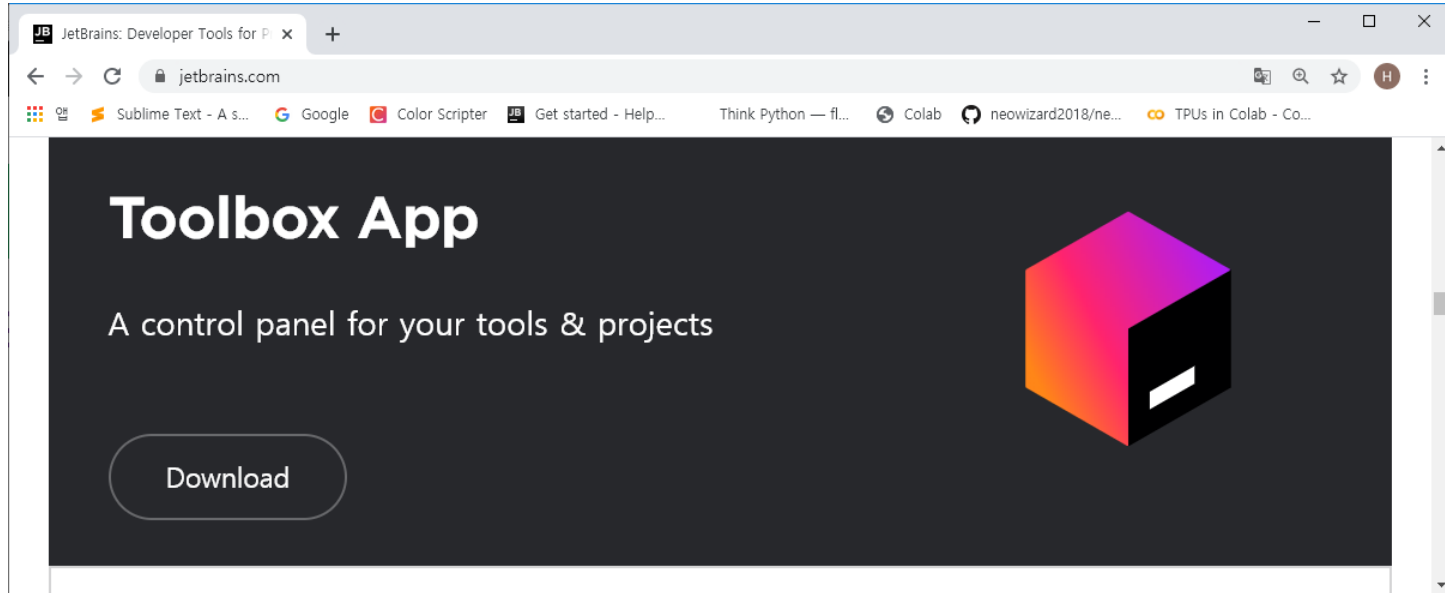
- 메뉴 파일 | 설정
 - Ctrl + alt + s
 - 에디터 | 글꼴



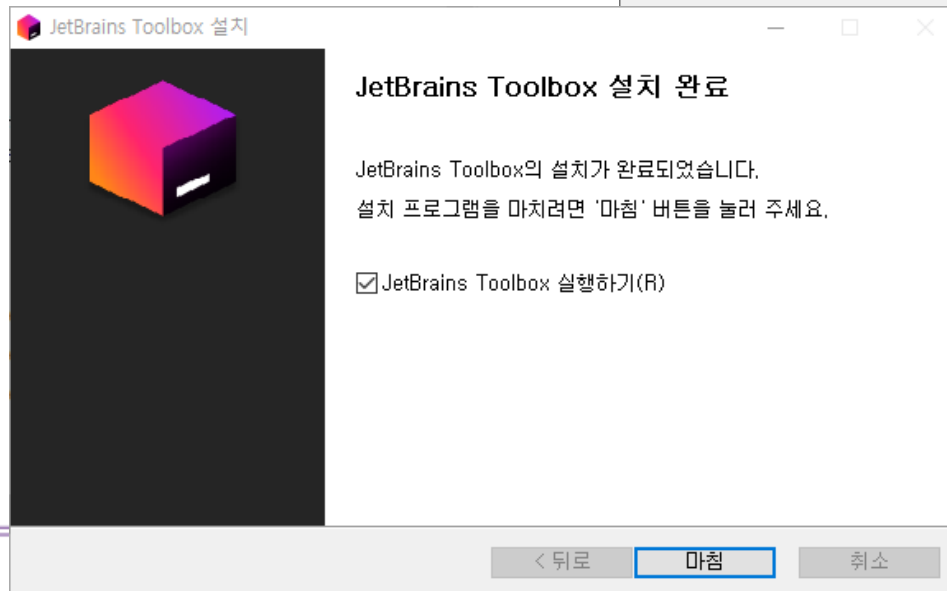
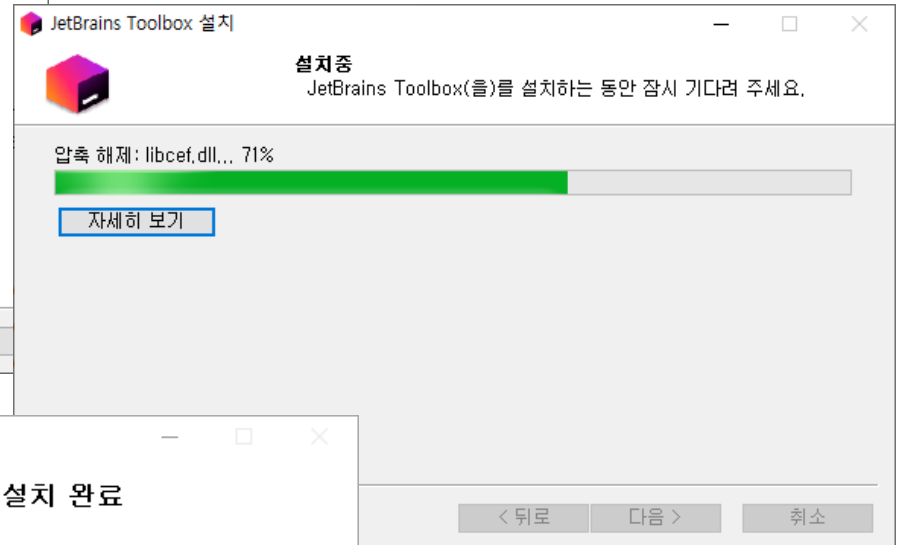
투박스 설치와 실행

Toolbox

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

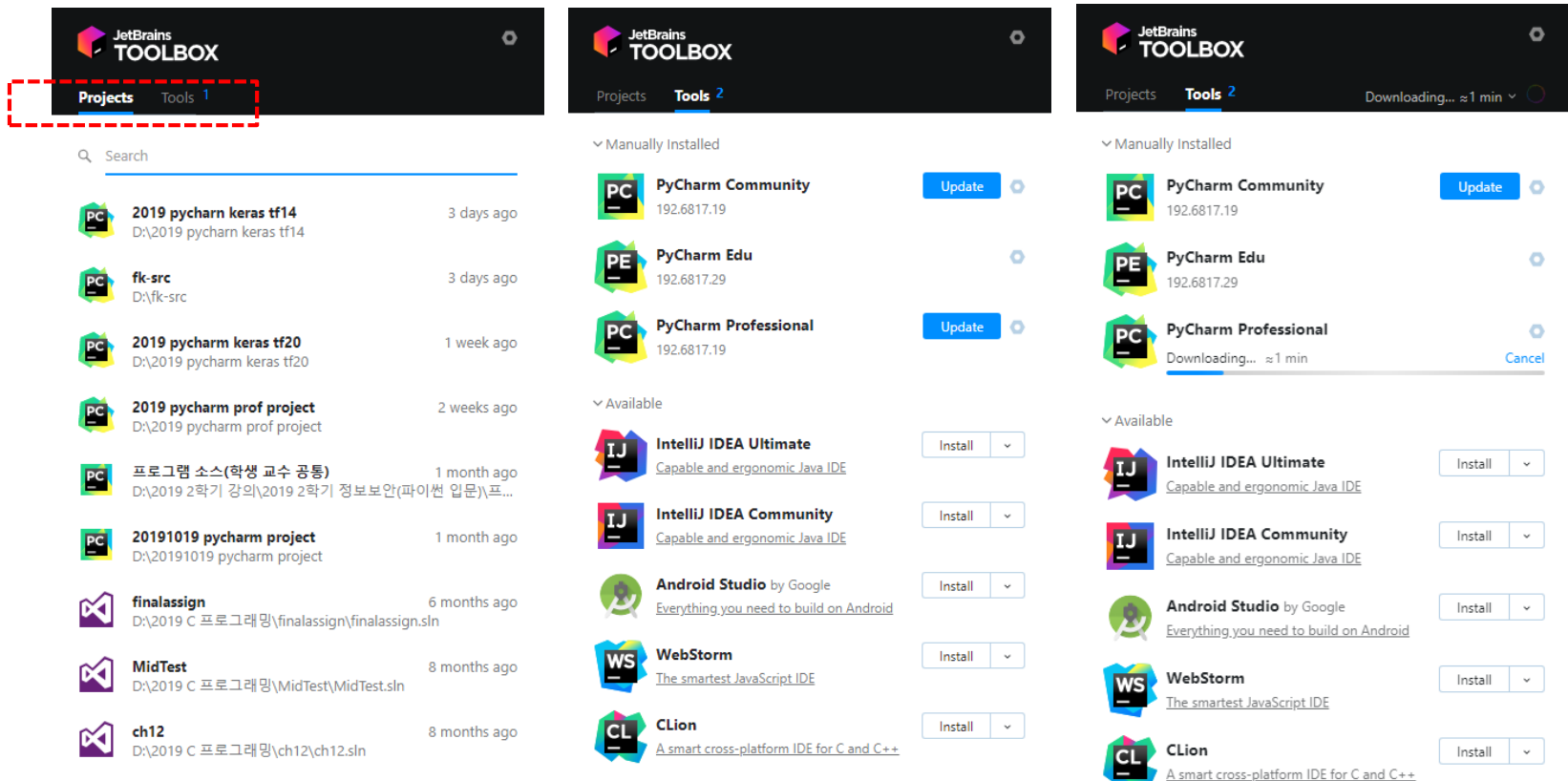


ToolBox 설치



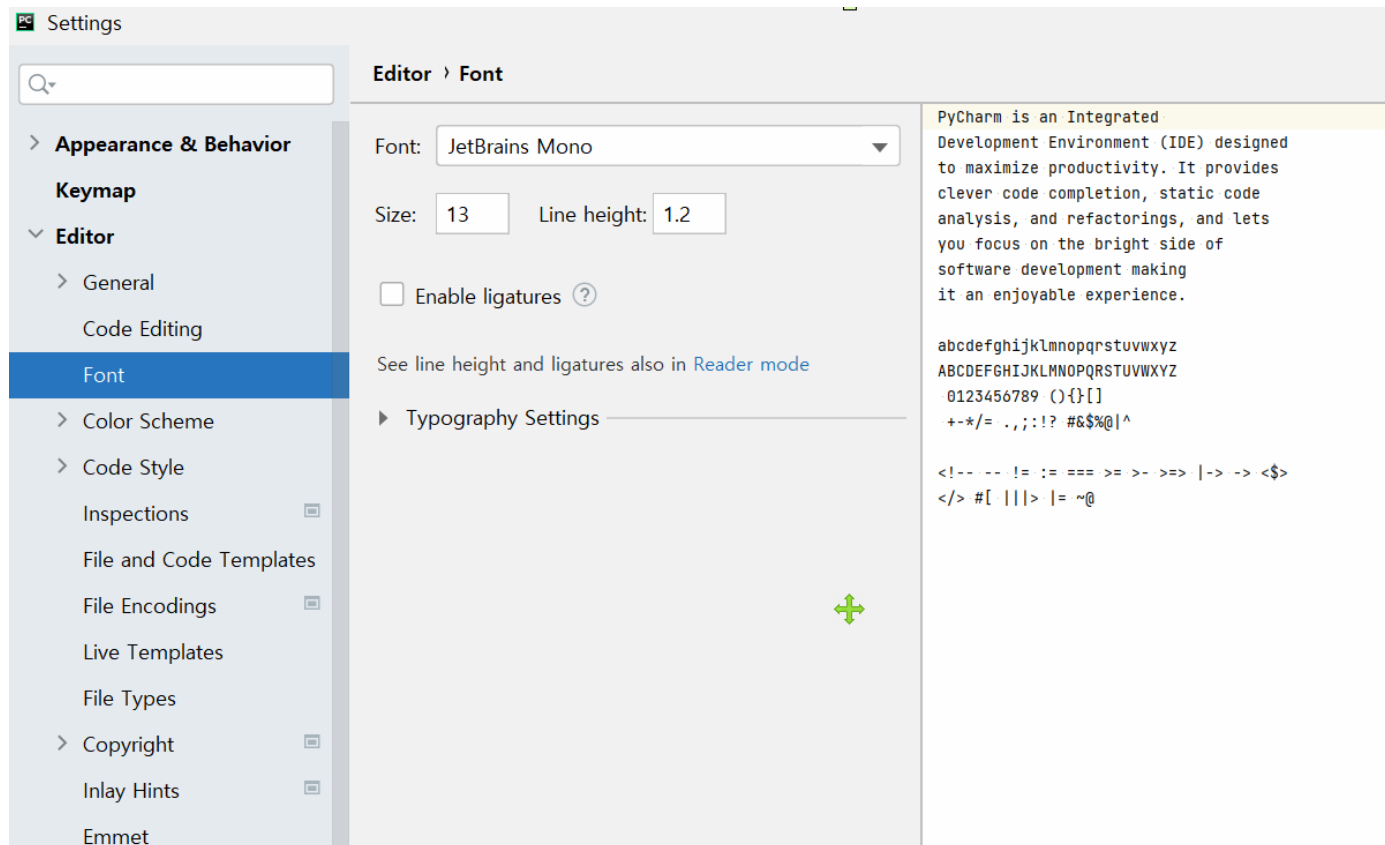
ToolBox 설치와 실행

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

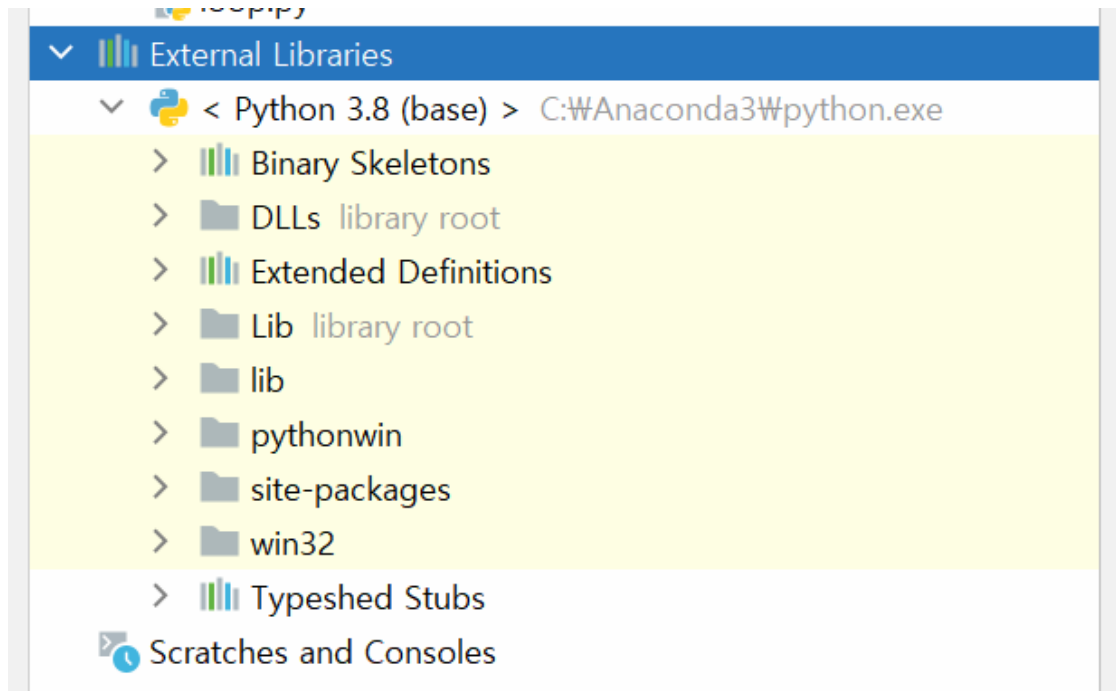


환경 설정

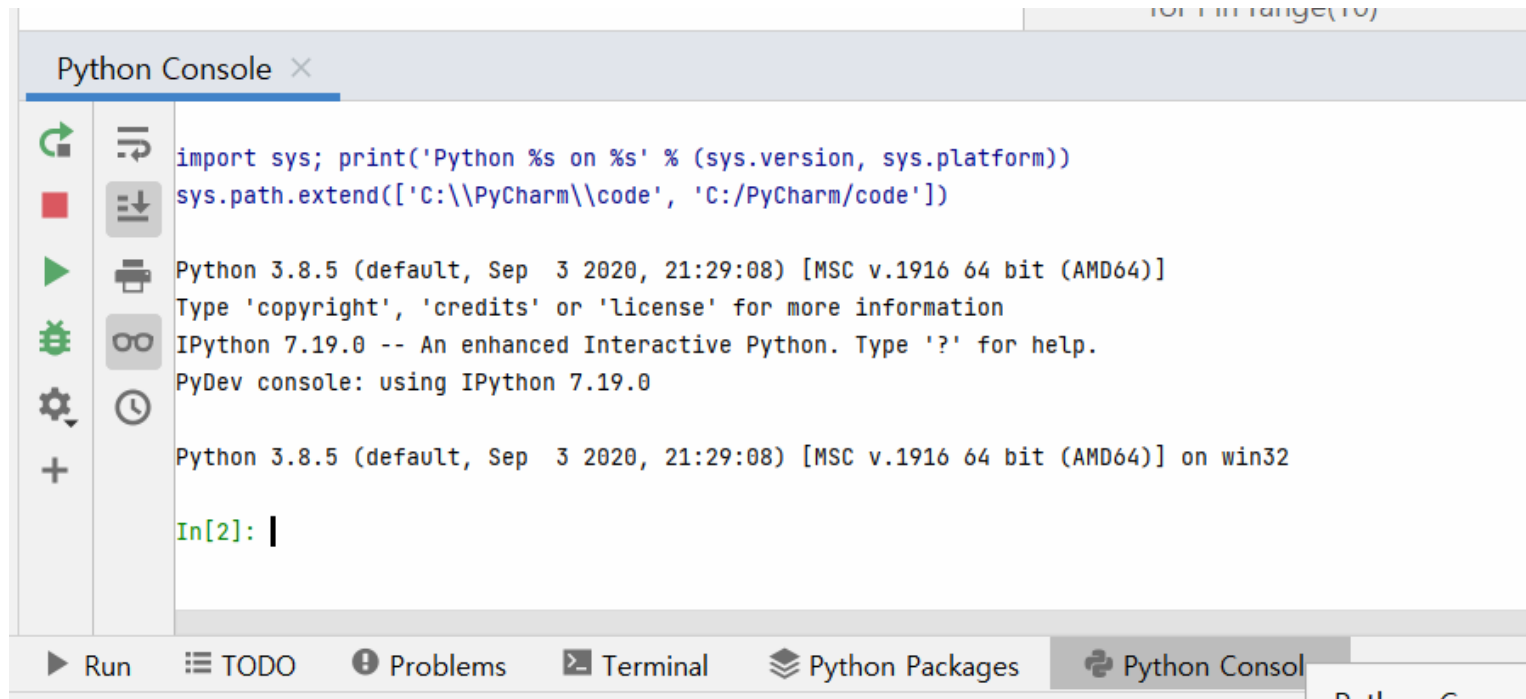
• 파일 | 설정



External Libraries



파이썬 콘솔



The screenshot shows the PyCharm Python Console window. The title bar is 'Python Console'. The console contains the following text:

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\PyCharm\\code', 'C:/PyCharm/code'])

Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.
PyDev console: using IPython 7.19.0

Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] on win32

In[2]: |
```

The bottom of the window shows a toolbar with icons for Run, TODO, Problems, Terminal, Python Packages, and Python Console. The Python Console icon is currently selected.

모듈 random

- `>>> import random as rd`
- `>>> rd.sample(range(10), 5)`
- `[3, 6, 2, 8, 4]`
- `>>> rd.sample(range(10), 5)`
- `[2, 1, 3, 5, 9]`
- `>>> a = rd.sample(range(10), 5)`
- `>>> a`
- `[6, 4, 5, 1, 7]`
- `>>> sorted(a)`
- `[1, 4, 5, 6, 7]`
- `>>> rd.random()`
- `0.6836451305607946`
- `>>> rd.randrange(1, 10)`
- `1`
- `>>> rd.randint(1, 2)`
- `1`
- `>>> rd.randint(1, 2)`
- `1`
- `>>>`

파이썬 코딩 연습

3-4-4.py

lee7py / Python-Programming Watch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security 0](#) [Insights](#)

Branch: master Python-Programming / Ch03 / 3-4-4.py / [Jump to](#) Find file Copy path

lee7py Add files via upload cb55c32 16 days ago

1 contributor

4 lines (4 sloc) 171 Bytes Raw Blame History   

```
1  ## Display a multiplication table for the numbers from 1 through 5.
2  for m in range(1, 6):
3      for n in range(1, 6):
4          print(m, "x", n, "=", m * n, "\t", end="")
```

3-4-6.py

lee7py / Python-Programming

Watch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security 0](#) [Insights](#)

Branch: master Python-Programming / Ch03 / 3-4-6.py / [Jump to](#)

[Find file](#) [Copy path](#)

lee7py Add files via upload cb55c32 16 days ago

1 contributor

6 lines (6 sloc) | 194 Bytes

Raw Blame History

```
1  ## Reverse the letters in a word.
2  word = input("Enter a word: ")
3  reversedWord = ""
4  for ch in word:
5      reversedWord = ch + reversedWord
6  print("The reversed word is " + reversedWord + ".")
```

3-4-9.py

lee7py / Python-Programming

Watch 1

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security 0

Insights

Branch: master

Python-Programming / Ch03 / 3-4-9.py / Jump to

Find file

Copy path

lee7py Add files via upload

cb55c32 16 days ago

1 contributor

11 lines (11 sloc) | 480 Bytes

Raw

Blame

History



```
1  ## Display the names of the 52 cards in a deck of cards.
2  ranks = ["2", "3", "4", "5", "6", "7", "8", "9", "10", "jack", "queen", "king", "ace"]
3  suits = ["spades", "hearts", "clubs", "diamonds"]
4  deckOfCards = [] # List to hold the names of the 52 cards in a deck.
5  # Use nested loops to fill the deckOfCards list.
6  for rank in ranks:
7      for suit in suits:
8          deckOfCards.append(rank + " of " + suit)
9  # Display the 52 cards.
10 for card in deckOfCards:
11     print(card)
```

Quiz

- 다음 파이참에 대한 설명 중 잘못된 것은?
 - 인터프리터를 아나콘다로 설정할 수 있다.
 - 프로젝트마다 다른 인터프리터를 설정할 수 없다.
 - Toolbox를 실행하면 화면에 Projects와 Tools 탭이 있다.
 - 파이참에서도 콘솔을 사용할 수 있다.

파이썬 문법 학습

https://realpython.com/python-dicts/

- 검색 realpython dict

The screenshot shows the Real Python website with the article 'Dictionaries in Python' by John Sturtz. The page features a navigation bar with links like 'Start Here', 'Learn Python', 'Store', and 'More'. A search bar and 'Join'/'Sign-In' buttons are also present. The main content area includes a large illustration of a robot and a person shaking hands, followed by the article title and author. A 'Table of Contents' section lists topics like 'Defining a Dictionary' and 'Accessing Dictionary Values'. On the right, there's a 'FREE Email Series' for 'Python Tricks' with a code snippet for merging dictionaries. Below that, 'All Tutorial Topics' are listed in a grid. At the bottom, a green banner says 'Improve Your Python'.

Real Python

Dictionaries in Python

by John Sturtz 21 Comments basics python

Tweet Share Email

Table of Contents

- Defining a Dictionary
- Accessing Dictionary Values
- Dictionary Keys vs. List Indices
- Building a Dictionary Incrementally
- Restrictions on Dictionary Keys
- Restrictions on Dictionary Values
- Operators and Built-in Functions
- Built-in Dictionary Methods

— FREE Email Series —

Python Tricks

```
1# How to merge two dicts
2# in Python 3.5+
3
4>>> x = {'a': 1, 'b': 2}
5>>> y = {'b': 3, 'c': 4}
6
7>>> z = {**x, **y}
8
9>>> z
10{'c': 4, 'a': 1, 'b': 3}
```

Email...

Get Python Tricks »

No spam. Unsubscribe any time.

All Tutorial Topics

advanced api basics best-practices
community databases data-science
devops django docker flask front-end
intermediate machine-learning python
testing tools web-dev web-scraping

Enhance Python with Podio

Improve Your Python

https://srv.realpython.net/click/23076429117/?c=52901525492&p=58946116052&r=33409

딕셔너리 학습

Dictionaries in Python - Real Python
realpython.com/python-dicts/

[Start Here](#)
[Learn Python](#)
[Store](#)
[More](#)

[Join](#)

👉 Stuck at home? Enjoy free courses, on us →

- If the key is already present in `d`, the corresponding value in `d` for that key is updated to the value from `<obj>`.

Here is an example showing two dictionaries merged together:

```

Python
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d2 = {'b': 200, 'd': 400}

>>> d1.update(d2)
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}

```

In this example, key `'b'` already exists in `d1`, so its value is updated to 200, the value for that key from `d2`. However, there is no key `'d'` in `d1`, so that key-value pair is added from `d2`.

`<obj>` may also be a sequence of key-value pairs, similar to when the `dict()` function is used to define a dictionary. For example, `<obj>` can be specified as a list of tuples:

```

Python
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d1.update([('b', 200), ('d', 400)])
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}

```

Or the values to merge can be specified as a list of keyword arguments:

```

Python

```

Table of Contents

- Defining a Dictionary
- Accessing Dictionary Values
- Dictionary Keys vs. List Indices
- Building a Dictionary Incrementally
- Restrictions on Dictionary Keys
- Restrictions on Dictionary Values
- Operators and Built-in Functions
- Built-in Dictionary Methods
- Conclusion

[Tweet](#)
[Share](#)
[Email](#)

[Recommended Video Course](#)
 Dictionaries in Python

High Quality Python Video Courses
[Watch Now](#)

Help

<https://srv.realpython.net/click/64754707389/?c=31941813441&p=58946116052&r=81003>

딕셔너리 종료

• 학습 후 퀴즈

Real Python

Start Here Learn Python Store More

Stuck at home? Enjoy free courses, on us →

would be best for a given situation.

Next you will learn about Python **sets**. The set is another composite data type, but it is quite different from either a list or dictionary.

Take the Quiz: Test your knowledge with our interactive “Python Dictionaries” quiz. Upon completion you will receive a score so you can track your learning progress over time:

[Take the Quiz »](#)

« Lists and Tuples in Python Dictionaries in Python Sets in Python »

Watch Now This tutorial has a related video course created by the Real Python team. Watch it together with the written tutorial to deepen your understanding: [Dictionaries in Python](#)

Table of Contents

- Defining a Dictionary
- Accessing Dictionary Values
- Dictionary Keys vs. List Indices
- Building a Dictionary Incrementally
- Restrictions on Dictionary Keys
- Restrictions on Dictionary Values
- Operators and Built-in Functions
- Built-in Dictionary Methods
- [Conclusion](#)

[Tweet](#) [Share](#) [Email](#)

Recommended Video Course

[Dictionaries in Python](#)

Python Tricks

Get a short & sweet **Python Trick** delivered to your inbox every couple of days. No spam ever. Unsubscribe any time. Curated by the Real Python team.

```
1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
```

[Help](#)

컴프리헨션:
내장, 내포, 추약, 해석

리스트, 셋, 사전

컴프리헨션(1)

• 국내 사이트

- 컴프리헨션 으로 검색
- <https://wikidocs.net/22805>
- <https://doorbw.tistory.com/174>
- <https://ddanggle.gitbooks.io/interpy-kr/ch15-comprehension.html>
- <https://dojang.io/mod/page/view.php?id=2285>

숫자 10개 생성
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

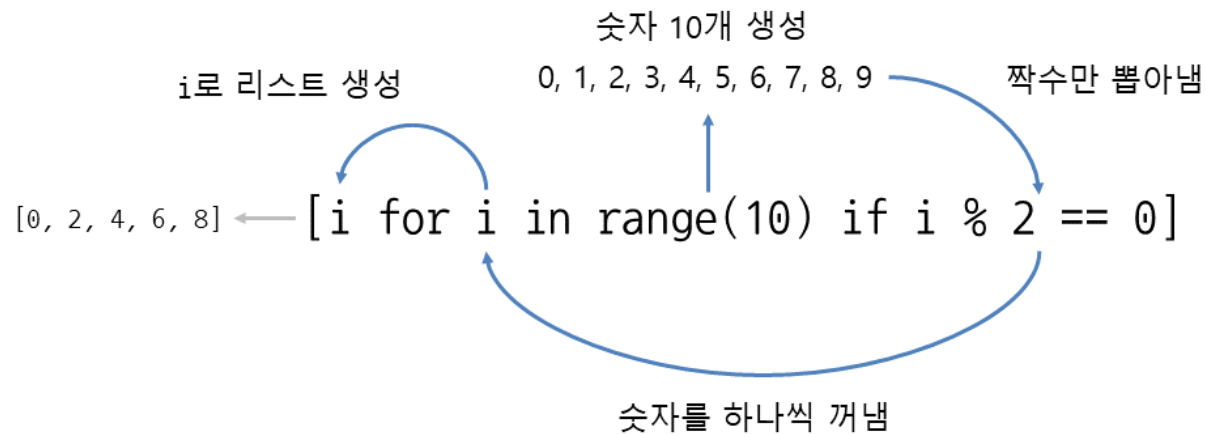
숫자를 하나씩
꺼냄

i로 리스트 생성

`[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]` ← `[i for i in range(10)]`

컴프리헨션(2)

- 조건식이 있는 내포



컴프리헨션(3)

- <https://realpython.com/list-comprehension-python/>



When to Use a List Comprehension in Python

by James Timmins · Nov 06, 2019 · 16 Comments · basics python

[Tweet](#) [Share](#) [Email](#)

Table of Contents

- [How to Create Lists in Python](#)
 - [Using for Loops](#)
 - [Using map\(\) Objects](#)
 - [Using List Comprehensions](#)
 - [Benefits of Using List Comprehensions](#)
- [How to Supercharge Your Comprehensions](#)
 - [Using Conditional Logic](#)
 - [Using Set and Dictionary Comprehensions](#)
 - [Using the Walrus Operator](#)
- [When Not to Use a List Comprehension in Python](#)
 - [Watch Out for Nested Comprehensions](#)
 - [Choose Generators for Large Datasets](#)
 - [Profile to Optimize Performance](#)
- [Conclusion](#)

내포의 특징

- 파이썬스러운(pythonic) 코딩 방식
- 효용성
 - 한 번 알아두면 쉽게 코딩
 - 속도는 반복보다 빠름
 - 내장 함수 `map()` 보다는 느림

Quiz

- 다음 파이썬의 사전과 컴프리헨션에 대한 설명 중 잘못된 것은?
 - 파이썬 사전은 키와 값의 쌍을 항목으로 하는 자료구조이다.
 - 3.6 버전 이후에 사전은 순서가 있는 자료구조이다.
 - 사전의 메소드 pop()은 인자가 하나도 없이 실행하면 오류가 발생한다.
 - 컴프리헨션으로는 리스트만을 생성할 수 있다.

파이썬 3.10의 확장 기능

- Match case

```
month = 1
match month:
    case 1:
        print("January")
    case 2:
        print("February")
    case _:
        print("None")

def http_error(status):
    match status:
        case 400:
            return "Bad request"
        case 401 | 403:
            return "Not allowed"
        case 404:
            return "Not found"
        case 405 | 406 | 407:
            return "I'm a teapot"
        case _:
            return "Something's wrong with the internet"

import random
lst = list(range(400, 408))
for _ in range(5):
    _ = random.choice(lst)
    print(_, http_error(_))
```

리스트나 튜플 매칭

```
def print_point(point):
    match point:
        case (0, 0):
            print("Origin")
        case (0, y):
            print(f"Y = {y}")
        case (x, 0):
            print(f"X = {x}")
        case (x, y):
            print(f"X = {x}, Y = {y}")
        case _:
            raise ValueError("Not a point")
```

```
print_point((0, 0)) # Origin
print_point((4, 0)) # X = 4
print_point([0, 2]) # Y = 2
```

unpacking이 가능하다면 어떤 꼴이든 상관없다.

```
print_point([2] + [4]) # X = 2, Y = 4
print_point([x for x in range(2)]) # Y = 1
print_point(("1", "2")) # X = 1, Y = 2
print_point([1 + 2, 300 + 20]) # X = 3, Y = 320
print_point("some thing".split()) # X = some, Y = thing
```

unpacking이 불가능하거나 어떤 case에도 걸리지 않으면 ValueError와 마주친다.

```
#print_point([1])
#print_point([1, 0, 3])
#print_point(2, 1) # 두 개의 인자가 들어가면 안 된다!
```

여러 항목의 매칭

```
classA = ["Alice", "Bob", "Cameron"]  
classB = ["Cheolsoo", "Yeonghee", "Minsu", "Heesu"]
```

```
while True:  
    name = input("Enter a name: ")
```

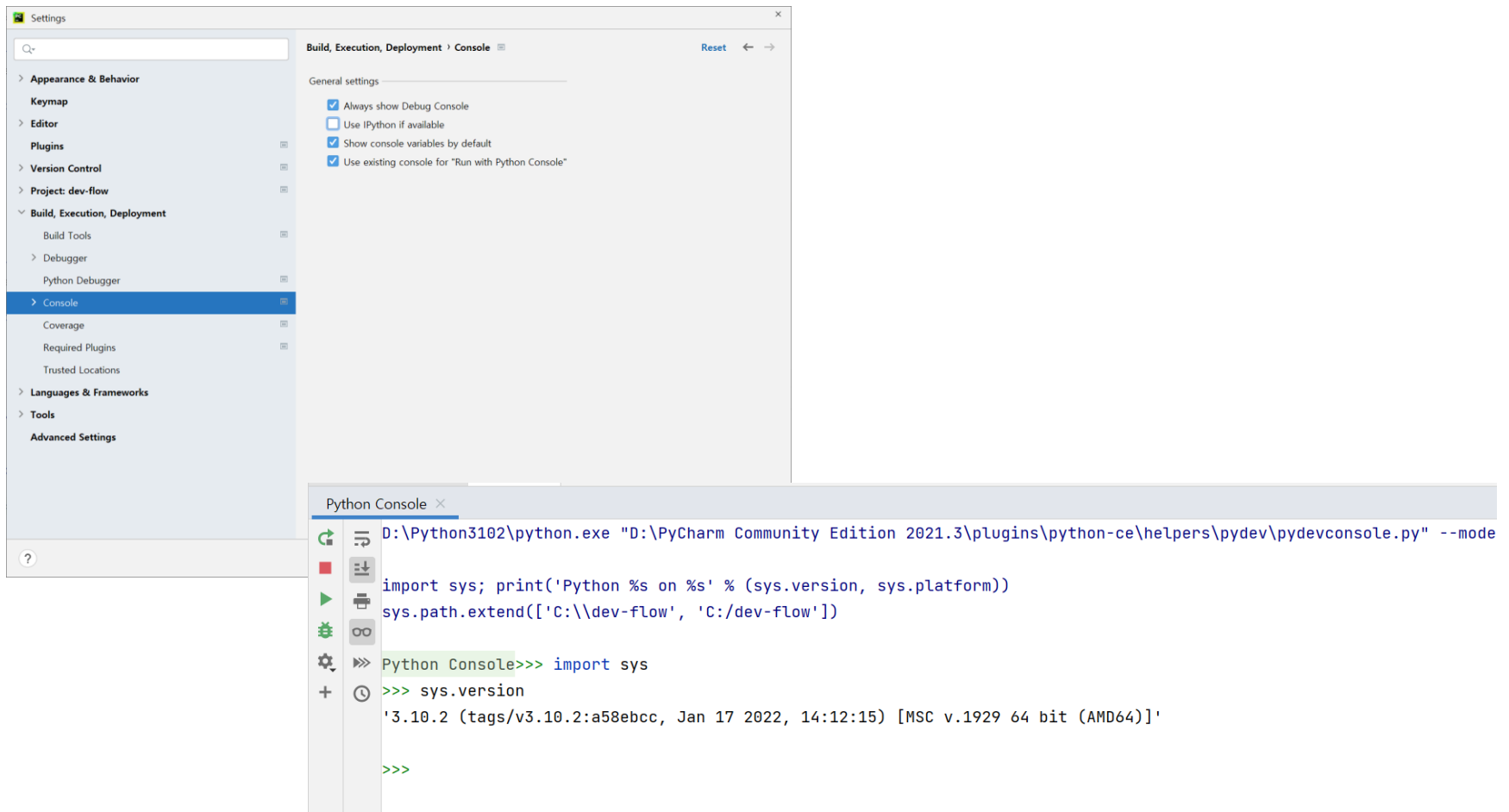
```
    match name:  
        case n if n in classA:  
            print(f"{n} is in classA")  
        case n if n in classB:  
            print(f"{n} is in classB")  
        case _:  
            print(f"{n} is unknown")  
            break
```

```
while True:  
    name = input("Enter a name: ")
```

```
    match name:  
        case "Alice" | "Bob" | "Cameron":  
            print(f"{name} is in classA")  
        case "Cheolsoo" | "Yeonghee" | "Minsu" | "Heesu":  
            print(f"{name} is in classB")  
        case _:  
            print(f"{name} is unknown")  
            break
```

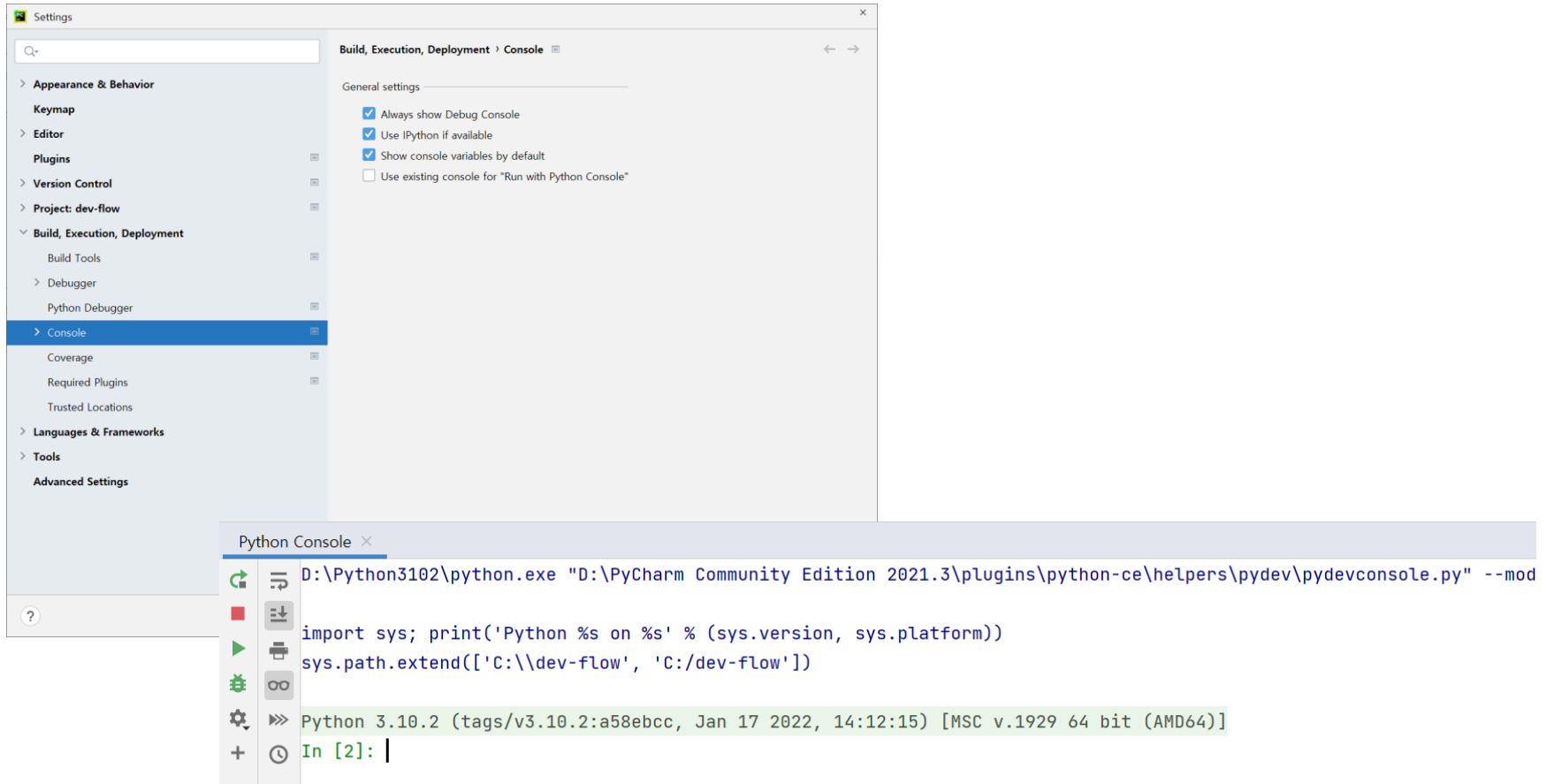
콘솔 지정 1

- 일반 python 콘솔



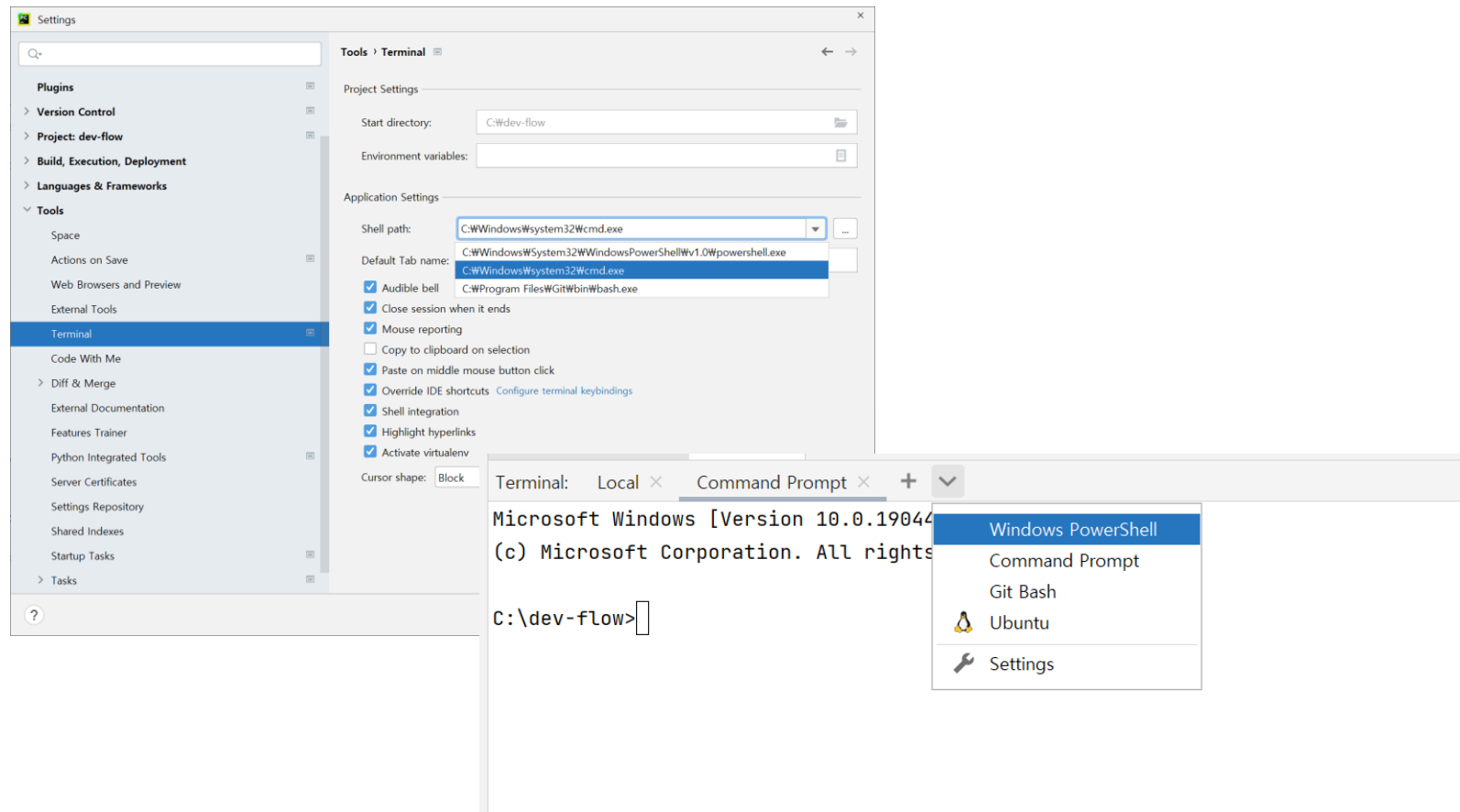
콘솔 지정 2

- ipython 콘솔



터미널 지정

- Cmd로 지정
 - 터미널에서 하단 화살표 리스트에서도 수정 가능



여러 터미널 실행

```
Terminal: Local × Command Prompt × Windows PowerShell × + ▼
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.


새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\dev-flow>
```

```
Terminal: Local × Command Prompt × Windows PowerShell × Git Bash × + ▼
PC@DESKTOP-482N0AB MINGW64 /c/dev-flow (main)
$
```


파이참 메뉴 및 기능 정리

- <https://greeksharifa.github.io/references/2019/02/07/PyCharm-usage/>


 Gorio Learning
 세상의 변화에 대해 관심이 많은 이들의 Tech Blog
Q

PyCharm 사용법(파이참 설치 및 사용법)

07 Feb 2019 | PyCharm usage

목차

- 설치
 - Settings
 - Sync Settings
 - File Encoding 설정
 - PyCharm 메모리 설정(Heap Memory)
- Project 창(`Alt + 1`)
 - 새 파일 생성
- 코드 실행(전체 또는 선택)
- 편집 창(코드 편집기)
 - .ipynb 파일 사용
 - 자동완성 기능
 - 빠른 선택, 코드 정리, 편집 동등 단축키
 - 찾기(및 바꾸기), (`Ctrl + F` | `Ctrl + H`)
 - 더 넓은 범위에서 찾기
 - 변수/함수 등이 사용된 위치 찾기
 - Refactor(이름 재지정)
 - Diff(Differences viewer for files)