파이썬 활용 (#1/5)

2023. 02. 20

초빙교수 김현용
충북대학교 산업인공지능연구센터

CHUNGBUK NATIONAL UNIVERSITY

INDUSTRIAL AI RESEARCH CENTER

강의 일정

INDUSTRIAL AI RESEARCH CENTER

	1일차	2일차	3일차	4일차	5일차
주제	파이참과 엑셀파일	데이터 분석	웹 자동화	컴퓨터 비전	AI와 딥러닝
세부 내용	• **PyCharm** • 가상환경 • [실습] script 작성 • 파일 경로 • 파일 입출력 • [실습] 파일 입출력 • **OpenPyXL** - 엑셀파일 입출력 - 셀 편집 - 차트 그리기 • [실습] 셀서식 지정 • [실습] 차트 그리기	• 데이터분석 • **NumPy** • 회귀분석 • [실습] 선형회귀 • **Pandas** • [실습] 데이터프레임 다루기 • **Matplotlib** • [실습] 시각화 • [실습] 회귀분석	• **웹스크래핑** • [실습] 기온 가져오기 • **Open API** (파파고) • **Selenium** • [실습] 구글 검색 • **PyAutoGui** • [실습] 구글 검색	• 컴퓨터 비전 개요 • **OpenCV** - 영상 입출력 - 영상 데이터 조작 • Matplotlib 영상표시 • OpenCV - 그리기 함수 - 동영상 입출력 • OpenCV - 히스토그램 - 이진화 - 에지 검출	• 인공신경망 • 딥러닝 • **PyTorch** • 객체검출 • **YOLOv8-객체검출** • [실습] 객체검출 • YOLOv8-객체분할 • chatGPT

Python 설치 (#1/2)

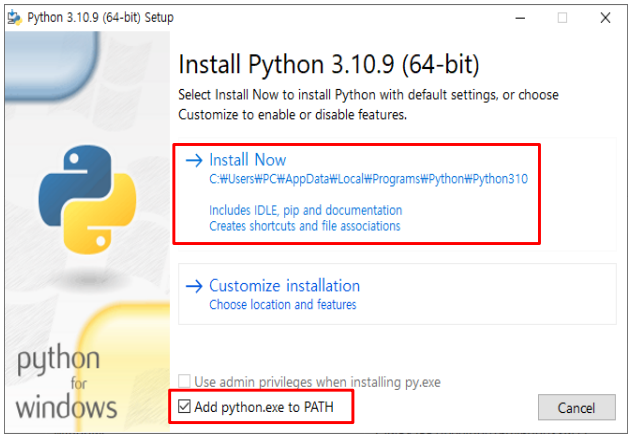
INDUSTRIAL AI RESEARCH CENTER

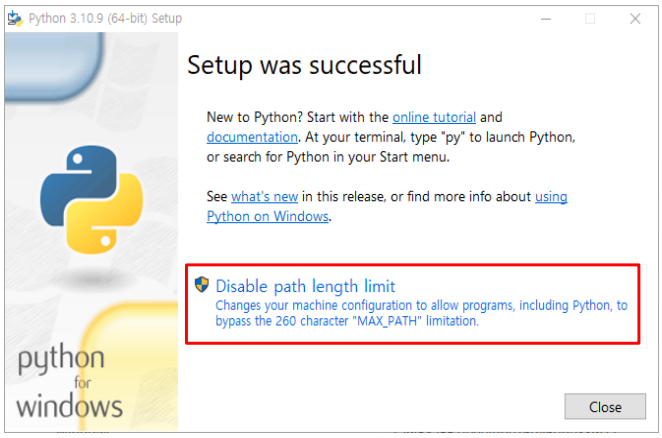
▣ 파이썬 설치 – www.python.org

- 프로그램 다운로드

Windows installer (32-bit)	Windows
Windows installer (64-bit)	Windows

- 프로그램 설치 : 파이썬 폴더를 경로에 포함 체크





Python 설치 (#2/2)

INDUSTRIAL AI RESEARCH CENTER

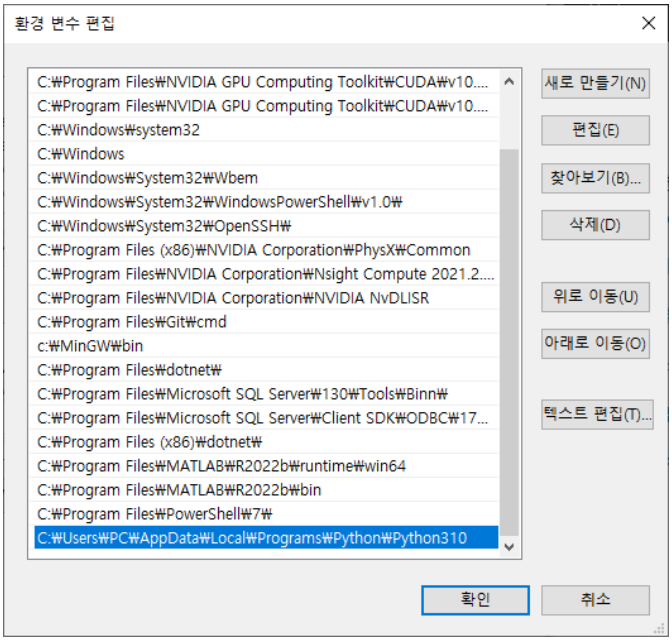
▣ 명령 프롬프트에서 경로 추가 확인

- Windows + R [실행] > cmd 또는 Window시작 > Windows 시스템 > 명령 프롬프트
- 현재경로 보기 : path
- 일시적인 경로 추가 : set path =%path%;추가경로;
- 영구적인 경로 추가 : setx path "%path%;추가경로"
- 파이썬 버전 확인 : python –version 또는
  파이썬에서 print(sys.version)

▣ **환경변수 편집에서 경로 추가 확인**

- 검색 > 고급 시스템 설정 보기 > 환경변수 > Path
- 찾아보기로 경로 추가
- 디폴트 경로:

C:\Users\PC\AppData\Local\Programs\Python\Python310



PyCharm 설치 (#1/3)

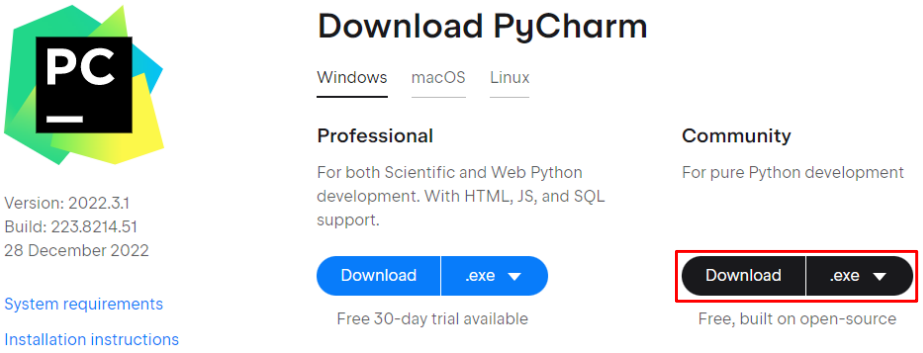
INDUSTRIAL AI RESEARCH CENTER

▣ **통합개발환경 (IDE, integrated development environment)**

- **코드 편집기(editor) + 실행기(compiler/interpreter) + 디버거(debugger)**
- 가독성 : Keyword, 변수, 문자열 등에 따라 색상으로 쉽게 구별 가능
- 강력하고 효율적인 편집기능 : 지능적인 코드 완성, 즉석 오류 검사, 화면 분리
- Debugger : break point 설정, 한 문장씩 수행, 실행 중 변수 값 조회
- 다양한 기능 제공 : 모듈/패키지 설치, 다양한 작업모드(터미널, 파이썬 콘솔) 지원

▣ PyCharm 다운로드 – www.jetbrains.com/pycharm/download

- JetBrains社에서 제작
- python에 특화된 가장 완성도 높은 IDE
- 유료/무료 버전



PyCharm 설치 (#2/3)

INDUSTRIAL AI RESEARCH CENTER

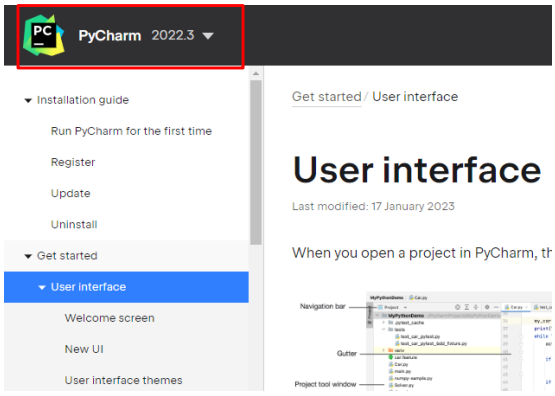
▣ 에디션 선택

- 학교 메일이 있으면 프로페셔널 에디션도 무료로 사용 가능 (연구용)

	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

▣ 매뉴얼

- https://www.jetbrains.com/help/pycharm

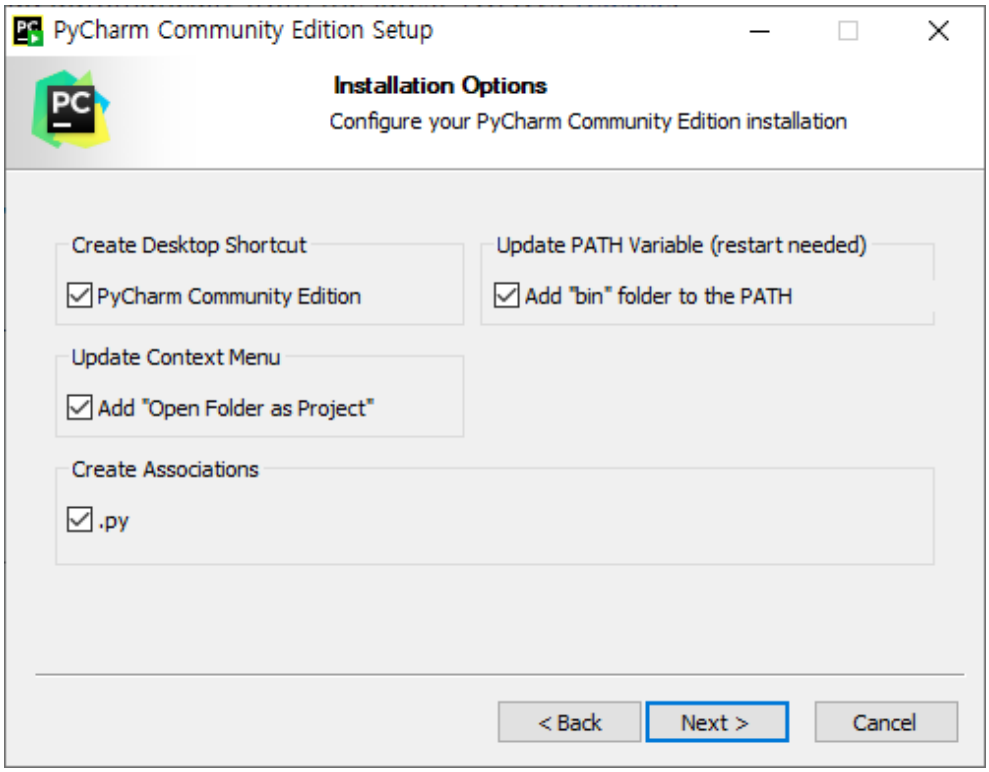


PyCharm 설치 (#3/3)

INDUSTRIAL AI RESEARCH CENTER

▣ PyCharm 설치 (2022.3.1)

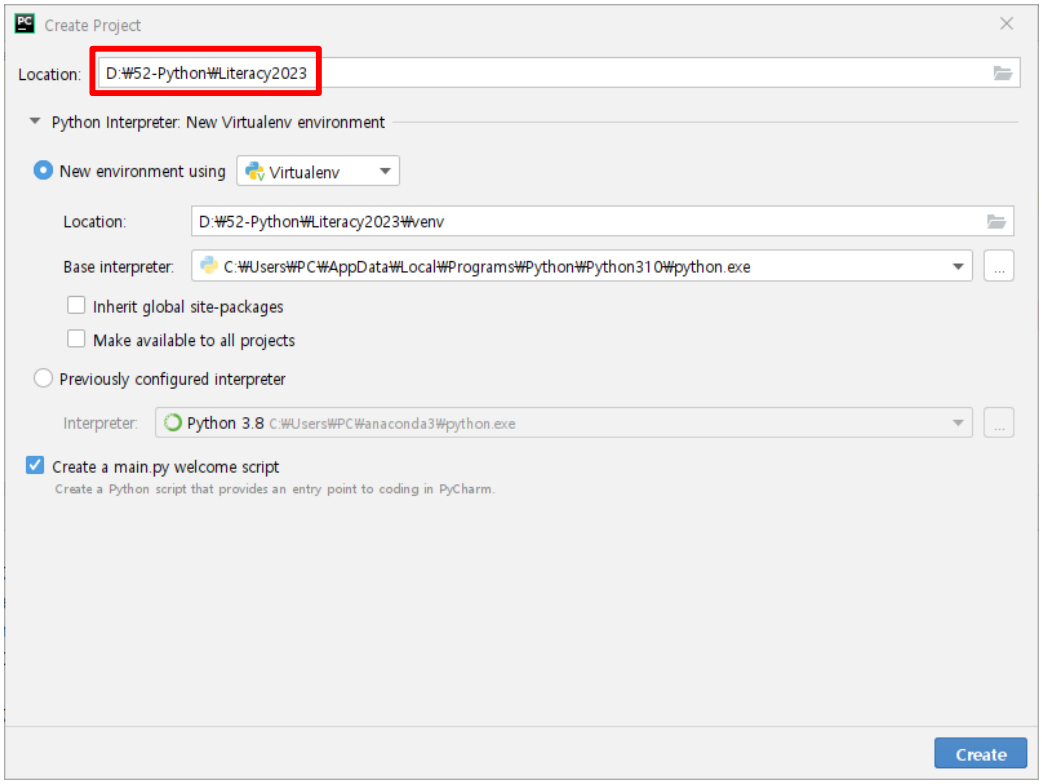
- 설치 옵션 모두 체크



PyCharm 사용법 (#1/6) - 프로젝트 생성

INDUSTRIAL AI RESEARCH CENTER

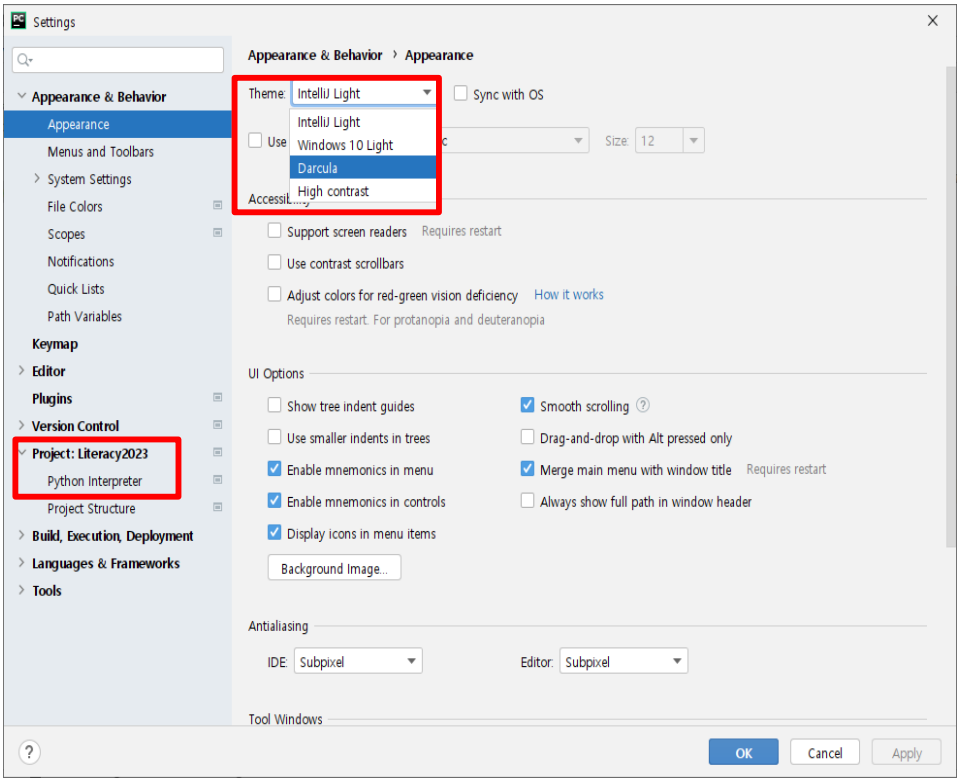
▣ 설치 후 > 새 프로젝트 만들기 : 자동적으로 가상환경을 구축함



PyCharm 사용법 (#2/6) - 환경설정

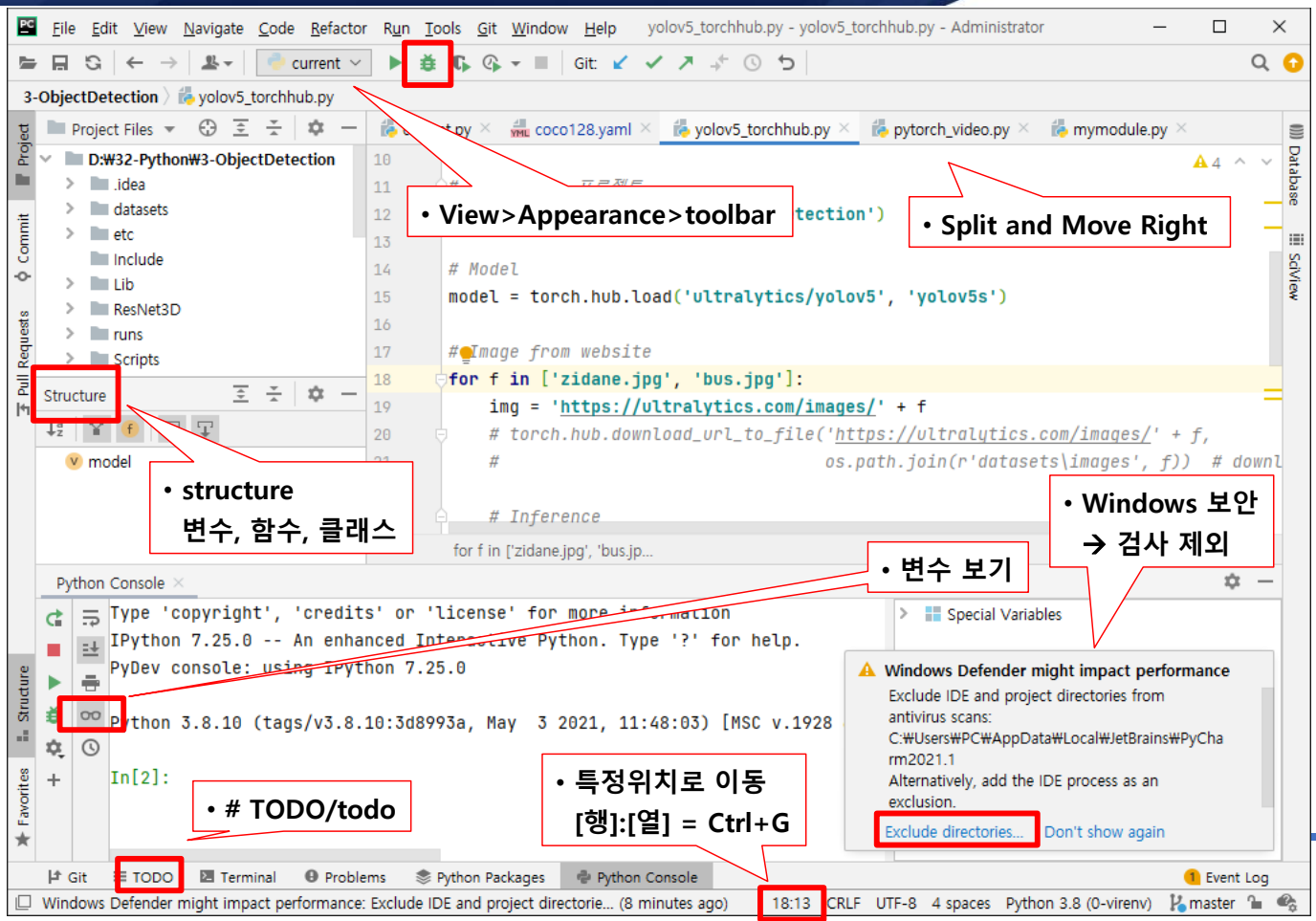
INDUSTRIAL AI RESEARCH CENTER

- File > Settings (Ctrl+Alt+S)
  - Appearance
    - Theme: Darcula
  - Keymap
    - Execute Selection (Alt+A)
    - Auto-Indent Lines (Ctrl+I)
  - Editor>General
    - Ctrl+마우스휠로 폰트크기 변경
  - **Plugins (한영전환)**
    - **Korean Language Pack**
  - Project
    - 가상환경
    - 모듈 설치/삭제/업그레이드



PyCharm 사용법 (#3/6) - 화면 구성

INDUSTRIAL AI RESEARCH CENTER

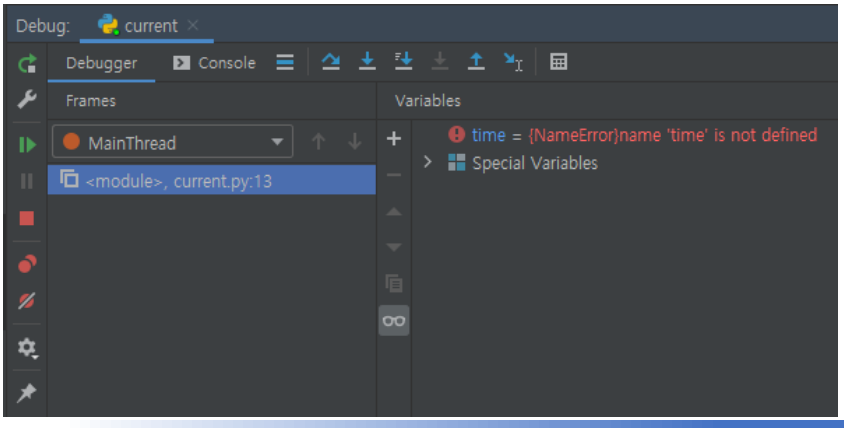


PyCharm 사용법 (#4/6) - 디버깅(debugging)

INDUSTRIAL AI RESEARCH CENTER

▣ **Break point (중단점) +**  → **Debug 창**

- Variables창 : 현재 시점의 변수값 확인
- 실행 아이콘 : Rerun(처음~B/P), **Resume (F9)** (현 B/P~다음 B/P), Pause, Stop(디버깅 중지)
- 단계별 디버깅
  - Show Execution Point – 현재 실행중인 코드로 이동
  - **Step Over (F8)** – 현재 코드를 실행하고 다음 줄로 이동 (함수도 하나의 명령으로 간주)
  - Step Into – 현재 시점에 호출하는 함수로 이동
  - Step Out – 코드를 실행하여 현재 함수를 벗어나는 부분에서 멈춤
  - Evaluate expression – 실행 중 변수값 변경 등
- Console



CHUNGBUK NATIONAL UNIVERSITY

PyCharm 사용법 (#5/6) - Help>Keymap Reference

INDUSTRIAL AI RESEARCH CENTER

PyCharm

Find any action inside the IDE	Ctrl + Shift + A

CREATE AND EDIT

Show intention actions	Alt + Enter
Basic code completion	Ctrl + Space
Smart code completion	Ctrl + Shift + Space
Type name completion	Ctrl + Alt + Space
Complete statement	Ctrl + Shift + Enter
Parameter information / context info	Ctrl + P / Alt + Q
Quick definition	Ctrl + Shift + I
Quick / external documentation	Ctrl + Q / Shift + F1
Generate code	Alt + Insert
Override / implement members	Ctrl + O / Ctrl + I
Surround with...	Ctrl + Alt + T
Comment with line comment	Ctrl + /
Extend / shrink selection	Ctrl + W / Ctrl + Shift + W
Optimize imports	Ctrl + Alt + O
Auto-indent lines	Ctrl + Alt + I
Cut / Copy / Paste	Ctrl + X / Ctrl + C / Ctrl + V
Copy document path	Ctrl + Shift + C
Paste from clipboard history	Ctrl + Shift + V
Duplicate current line or selection	Ctrl + D
Move line up / down	Ctrl + Shift + Up / Down
Delete line at caret	Ctrl + Y
Join / split line	Ctrl + Shift + J / Ctrl + Enter
Start new line	Shift + Enter
Toggle case	Ctrl + Shift + U
Expand / collapse code block	Ctrl + NumPad + / -
Expand / collapse all	Ctrl + Shift + NumPad + / -
Save all	Ctrl + S

VERSION CONTROL

VCS operations popup...	Alt + `
Commit	Ctrl + K
Update project	Ctrl + T
Recent changes	Alt + Shift + C
Revert	Ctrl + Alt + Z
Push...	Ctrl + Shift + K
Next / previous change	Ctrl + Alt + Shift + Down / Up

MASTER YOUR IDE

Find action...	Ctrl + Shift + A
Open a tool window	Alt + [0-9]
Synchronize	Ctrl + Alt + Y
Quick switch scheme...	Ctrl + `
Settings...	Ctrl + Alt + S
Jump to source / navigation bar	F4 / Alt + Home
Jump to last tool window	F12
Hide active / all tool windows	Shift + Esc / Ctrl + Shift + F12
Go to next / previous editor tab	Alt + Right / Alt + Left
Go to editor (from a tool window)	Esc
Close active tab / window	Ctrl + Shift + F4 / Ctrl + F4

FIND EVERYTHING

Search everywhere	Double Shift
Find / replace	Ctrl + F / R
Find in path / Replace in path	Ctrl + Shift + F / R
Next / previous occurence	F3 / Shift + F3
Find word at caret	Ctrl + F3
Go to class / file	Ctrl + N / Ctrl + Shift + N
Go to file member	Ctrl + F12
Go to symbol	Ctrl + Alt + Shift + N

NAVIGATE FROM SYMBOLS

Declaration	Ctrl + B
Type declaration (JavaScript only)	Ctrl + Shift + B
Super method	Ctrl + U
Implementation(s)	Ctrl + Alt + B
Find usages / Find usages in file	Alt + F7 / Ctrl + F7
Highlight usages in file	Ctrl + Shift + F7
Show usages	Ctrl + Alt + F7

REFACTOR AND CLEAN UP

Refactor this...	Ctrl + Alt + Shift + T
Copy... / Move...	F5 / F6
Safe delete...	Alt + Delete
Rename...	Shift + F6
Change signature...	Ctrl + F6
Inline...	Ctrl + Alt + N
Extract method	Ctrl + Alt + M
Introduce variable / parameter	Ctrl + Alt + V / P
Introduce field / constant	Ctrl + Alt + F / C
Reformat code	Ctrl + Alt + L

ANALYZE AND EXPLORE

Show error description	Ctrl + F1
Next / previous highlighted error	F2 / Shift + F2
Run inspection by name...	Ctrl + Alt + Shift + I
Type / call hierarchy	Ctrl + H / Ctrl + Alt + H

NAVIGATE IN CONTEXT

Select in...	Alt + F1
Recently viewed / Recent locations	Ctrl + E / Ctrl + Shift + E
Last edit location	Ctrl + Shift + Back
Navigate back / forward	Ctrl + Alt + Left / Right
Go to previous / next method	Alt + Up / Down
Go to line / column...	Ctrl + G
Go to code block end / start	Ctrl +] / [
Add to favorites	Alt + Shift + F
Toggle bookmark	F11
Toggle bookmark with mnemonic	Ctrl + F11
Go to numbered bookmark	Ctrl + [0-9]
Show bookmarks	Shift + F11

BUILD, RUN, AND DEBUG

Run context configuration	Ctrl + Shift + F10
Run / debug selected configuration	Alt + Shift + F10 / F9
Run / debug current configuration	Shift + F10 / F9
Step over / into	F8 / F7
Smart step into	Shift + F7
Step out	Shift + F8
Run to cursor / Force run to cursor	Alt + F9 / Ctrl + Alt + F9
Show execution point	Alt + F10
Evaluate expression...	Alt + F8
Stop	Ctrl + F2
Stop background processes...	Ctrl + Shift + F2
Resume program	F9
Toggle line breakpoint	Ctrl + F8
Toggle temporary line breakpoint	Ctrl + Alt + Shift + F8
Edit / view breakpoint	Ctrl + Shift + F8

jetbrains.com/pycharm
jetbrains.com/help/pycharm
blog.jetbrains.com/pycharm
@pycharm

6597

JET BRAINS

PyCharm 사용법 (#6/6) - 단축키

INDUSTRIAL AI RESEARCH CENTER

▣ 환경설정 (Ctrl+Alt+S) = File > Settings

▣ 실행

- **현재창 실행 (Ctrl+Shift+F10)**
- 선택후 실행 (Shift+Alt+F10)
- 선택창 실행 (Shift+F10)
- **선택영역/커서라인 실행 (Shift+Alt-E → Alt+A)**

▣ 편집

- 선택영역/커서라인 4칸 이동 (Tab)
- 선택영역/커서라인 -4칸 이동 (Shift+Tab)
- 한 줄(#) 주석/해제 (Ctrl+/)
- 한 줄 삭제 (Ctrl+Y)
- 선택영역/커서라인 잘라내기/복사 (Ctrl+X/C)
- 선택영역/커서라인 붙여넣기 (Ctrl+V)
- 선택영역/커서라인 복제 (Ctrl+D)
- 자동 들여쓰기 (Ctrl+Alt+I → Ctrl+I)
- 다음 줄로 이동 (Shift+Enter)

▣ 도움말/검색/이동

- **Quick Documentation (Ctrl+Q)**
- Quick Definition (Ctrl+Shift+I)
- 웹브라우저 문서 열기 (Shift+F1)
- 링크 열기 (Ctrl+mouse)
- 함수 단위 이동 (Alt+↓,↑)
- 함수 시작/끝 이동 (Ctrl+{, })
- 편집 전/후 위치로 이동 (Ctrl+Alt+ ←/ →)
- 특정라인으로 이동 (Ctrl+G)

파이썬 코딩 스타일

INDUSTRIAL AI RESEARCH CENTER

▣ **PEP8** (Python enhancement proposal #8) **스타일 가이드**를 따르자

- **문법뿐만 아니라 가독성(readability)**을 중시하는 파이썬 철학에서 비롯된 파이썬 코드를 작성하는 스타일 가이드
- 들여쓰기(indentation) 4칸, 한 행은 최대 79자로 제한할 것
- 함수, 변수, 속성은 **lowercase_underscore (lower_snake_case)**
- 클래스와 예외는 **CapitalizedWord (UpperCamelCase)**, 상수는 **ALL_CAPS**
- 분류 접두사 SERVICE_TIMEOUT → **ERROR**_SERVICE_TIMEOUT
- 클래스의 인스턴스는 self, 클래스는 cls로 지정
- 최상위 함수나 클래스 정의는 두 개의 빈 줄로 분리
- import 문은 한 라인에 하나씩, 문서 맨 위에 작성하고, 표준, 써드파티, 사용자 순서로 작성

The Zen of Python

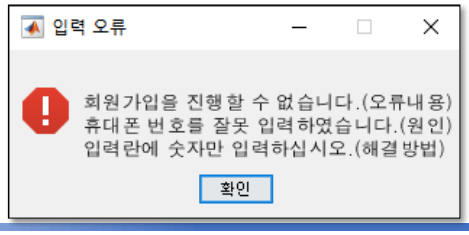
```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
```

▣ 주석

- **코드는 의미를, 주석은 의도를!** 예) `if key == 27:` *# ESC key를 입력한 경우*
- 모듈, 함수 머리에 docstring (""" """) → ~.__doc__로 접근

▣ 메시지 표시

- 개발자용 메시지와 사용자용 메시지의 구분
- **에러 메시지는 에러 내용/원인/해결방법 순으로 표시**



[실습] Script 작성 (#1/2)

INDUSTRIAL AI RESEARCH CENTER

```python
# 모듈 불러오기 (3가지) : built-in(내장) 함수는 필요없음
import sys                      # core module (표준모듈)
import numpy as np              # third-party module / aliasing(별명)
from myinfo import ID, PW       # user-defined module


def py_version():    # 함수, 변수명은 lowercase_underscore 형식
    """함수를 설명하는 Docstring
    다중 라인 주석처리, 이 때에도 들여쓰기(indentation)는 필수
    """
    print("파이썬 버전 : " + sys.version)


# 클래스, 함수 정의 앞뒤에는 2줄 띄움
class PyClass(): # 클래스명(명사)은 CapitalizedWord 형식
    """클래스에 대한 주석 (Docstring)"""
    def __init__(self, num=1):
        self.num = num
        print('클래스 객체 생성')

    def add(self):
        result = 0
        for n in range(1, self.num+1):
            result += n
        return result

    def multiply(self):
        # result = 1
        for n in range(1, self.num+1):
            result *= n
        return result
```

오류

▣ 스크립트 작성

- 파일 생성, 이름 바꾸기
- 파일 경로 가져오기
- 화면 분할
- 모듈

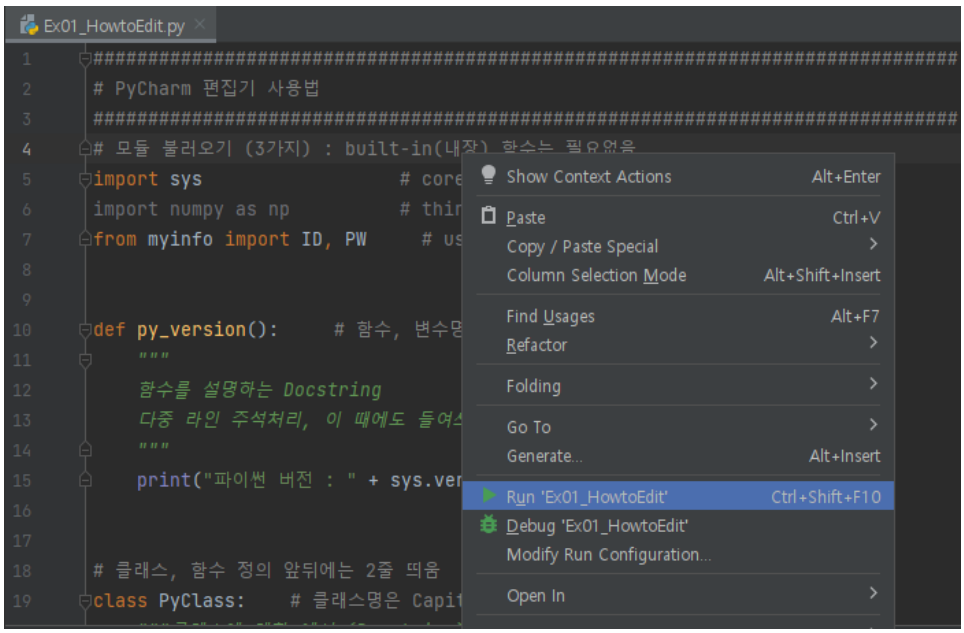
```python
if __name__ == "__main__":        # 실제 실행할 부분
    key = 'erp'
    print('id : ' + ID[key])      # 상수는 ALL_CAPS 형식
    print('password : ', PW[key])

    print(py_version.__doc__)     # Docstring 표시
    py_version()
    print(dir(sys))               # 모듈의 멤버 변수와 함수 조회

    print(PyClass.__doc__)        # Docstring 표시
    print(dir(PyClass))
    a = PyClass(10)
    # Python 2 버전
    print('1에서 %d까지의 합 : %s' % (a.num, format(a.add(),",")))
    # Python 3 버전
    print('1에서 {}까지의 곱 : {:,}'.format(a.num, a.multiply()))
```


[실습] Script 작성 (#2/2)

INDUSTRIAL AI RESEARCH CENTER

▣ 실행하기

- Script 실행
- Block/Line 단위 실행



▣ 디버깅

- 오류 역추적
- 디버깅 툴

```
Traceback (most recent call last):
  File "D:\52-Python\Literacy2023\Ex01_HowtoEdit.py", line 51, in <module>
    print('1에서 {}까지의 곱 : {:,}'.format(a.num, a.multiply()))      # Python 3버전
  File "D:\52-Python\Literacy2023\Ex01_HowtoEdit.py", line 34, in multiply
    result *= n
UnboundLocalError: local variable 'result' referenced before assignment
```

내장함수(built-in functions)

INDUSTRIAL AI RESEARCH CENTER

▣ builtins 모듈에 정의 → import 없이 사용 가능

1. abs()	**21. enumerate()**	**41. list()**	61. slice()
2. aiter()	22. eval()	42. locals()	62. sorted()
3. all()	23. exec()	43. map()	63. staticmethod()
4. any()	24. filter()	**44. max()**	**64. str()**
5. anext()	**25. float()**	45. memoryview()	**65. sum()**
6. ascii()	**26. format()**	**46. min()**	**66. super()**
7. bin()	27. frozenset()	**47. next()**	**67. tuple()**
8. bool()	28. getattr()	48. object()	**68. type()**
9. breakpoint()	29. globals()	49. oct()	69. vars()
10. bytearray()	30. hasattr()	**50. open()**	**70. zip()**
11. bytes()	31. hash()	**51. ord()**	
12. callable()	**32. help()**	52. pow()	
13. chr()	33. hex()	**53. print()**	
14. classmethod()	**34. id()**	54. property()	
15. compile()	35. input()	**55. range()**	
16. complex()	**36. int()**	56. repr()	
17. delattr()	37. isinstance()	57. reversed()	
18. dict()	38. issubclass()	**58. round()**	
19. dir()	**39. iter()**	59. set()	
20. divmod()	**40. len()**	60. setattr()	

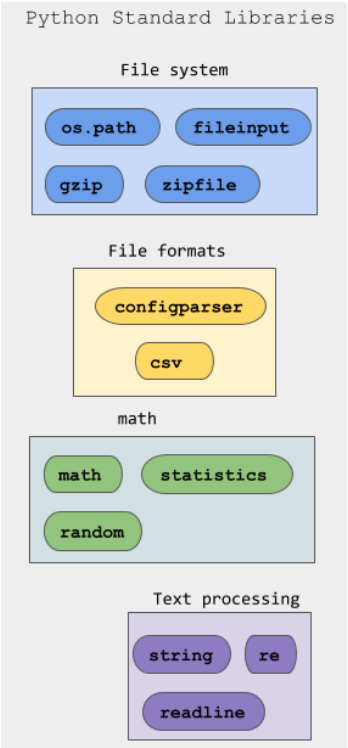
- 주요내장함수 참고자료 : https://wikidocs.net/32#abs

파이썬 표준 라이브러리(모듈)

INDUSTRIAL AI RESEARCH CENTER

▣ 파이썬과 함께 설치됨 → pip install 없이 **import** 하여 사용 가능
- The Python Standard Library (https://docs.python.org/3/library)

▣ 주요 구성
- 내장함수, 내장상수, 내장 자료형, 내장 예외
- Text Processing : string, re 등
- Data Types : datetime, array, copy, pprint 등
- Numeric and Mathematical Modules : math, random, statistics 등
- File and Directory Access : pathlib, os.path, glob, shutil
- Data Persistence : pickle, sqlite3
- Data Compression and Archiving : zlib, gzip, zipfile, tarfile
- File Formats : csv
- Generic Operating System Services : os, io, time, arqparse, getopt, logging
- Concurrent Execution : threading, multiprocessing, subprocess, shed, queue
- Networking and Interprocess Communication : asyncio, socket
- Internet Data Handling : json
- Structured Markup Processing Tools : html, xml
- Internet Protocols and Support : webbrowser, urllib, http, ftplib, smtplib
- Graphical User Interfaces with Tk : tkinter
- Python Runtime Services : sys

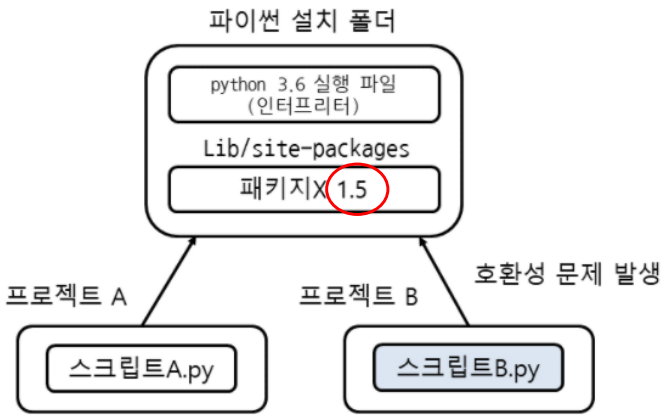


가상환경 (#1/4)

INDUSTRIAL AI RESEARCH CENTER

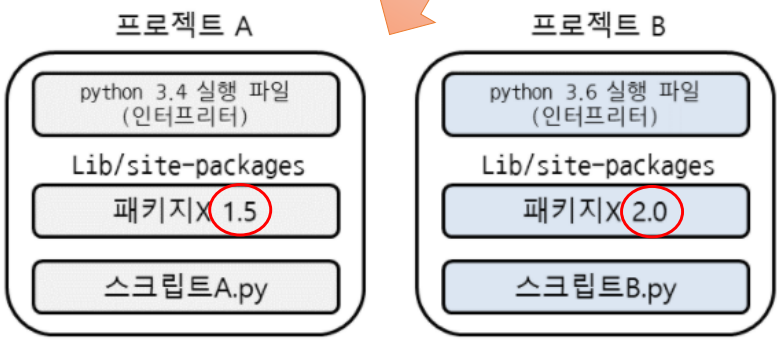
▣ **전역(global) 파이썬 환경**

- 프로젝트A는 패키지X 1.5를 사용해야 하고,
  프로젝트 B는 패키지X 2.0을 사용해야 하는 경우
  → 호환성(버전충돌) 문제 발생



▣ **가상환경(Virtual environment)**

- 독립된 실행환경을 만들어 줌



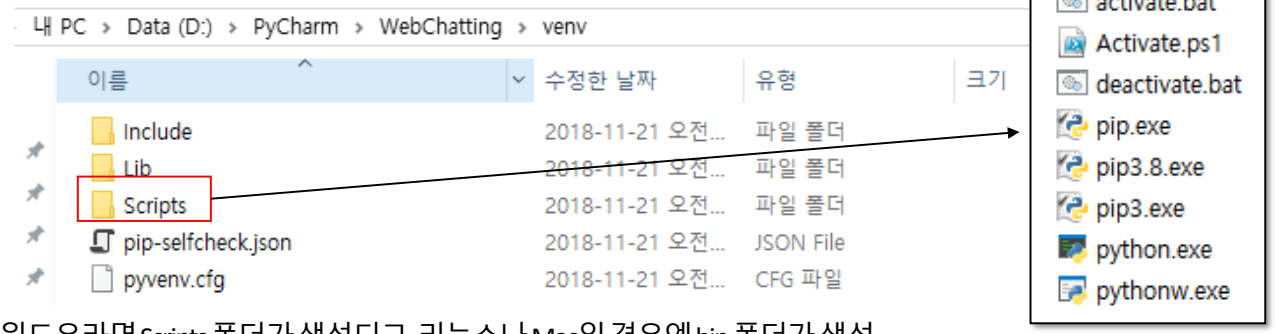
가상환경 (#2/4)

INDUSTRIAL AI RESEARCH CENTER

▣ 가상환경(Virtual environment) 만들기

- python –m venv 경로 : 해당 폴더가 없으면 자동으로 생성

```
D:₩PyCharm₩WebChatting>python -m venv D:₩PyCharm₩WebChatting₩venv
```



- 윈도우라면 Scripts 폴더가 생성되고, 리눅스나 Mac일 경우엔 bin 폴더가 생성

▣ 활성화 : Script 폴더의 activate.bat 실행

- pip로 모듈을 설치하면 Lib\site-packages 안에 패키지가 저장됨
- 경로/폴더명 등 변경시에는 activate.bat 파일을 수정해야 함

▣ 비활성화 : deactivate.bat 실행

가상환경 (#3/4) - 복사

INDUSTRIAL AI RESEARCH CENTER

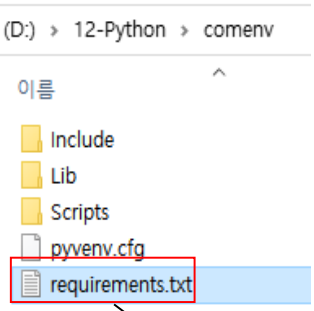
▣ 가상환경을 이식하기 위한 준비

- **pip list** : 설치된 패키지 목록보기

```
D:\12-Python\comenv>Scripts\activate

(comenv) D:\12-Python\comenv>pip list
Package                  Version
------------------------ ----------
altgraph                 0.17
beautifulsoup4           4.9.3
certifi                  2020.11.8
```

▣ 설치 패키지 목록/버전 저장

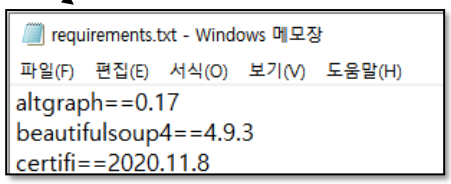
- **pip freeze > requirements.txt**



▣ target으로 가상환경 가져오기

- target 가상환경 활성화

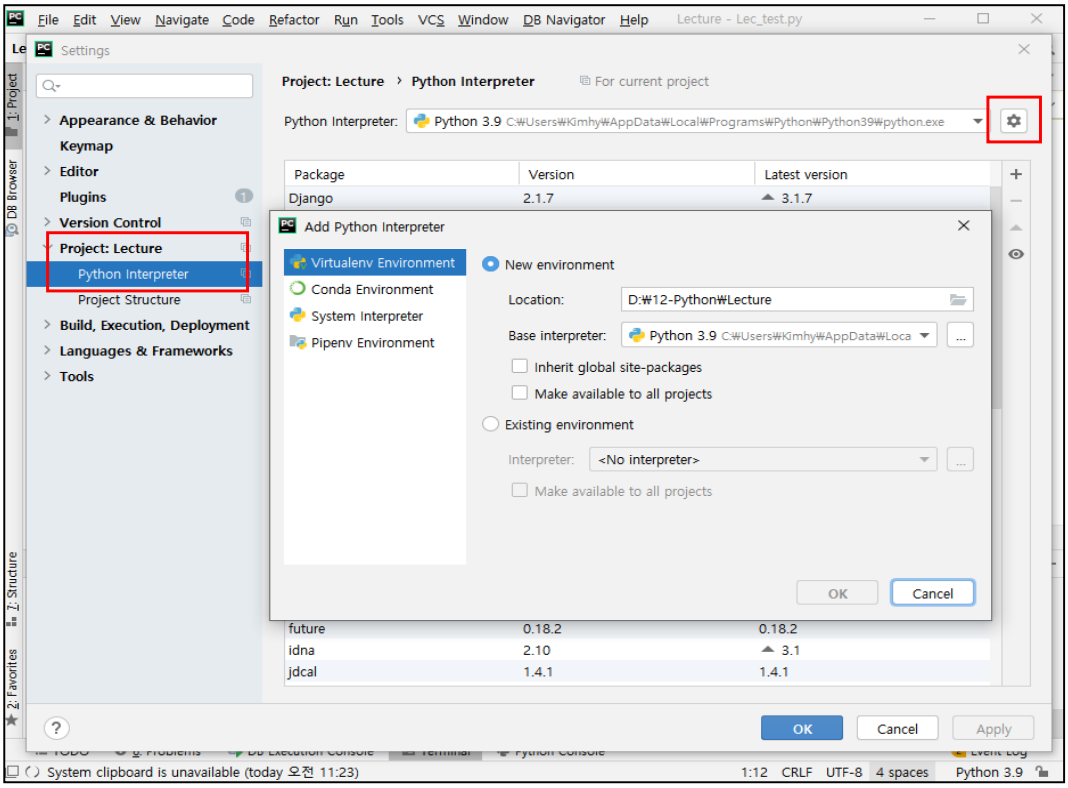
- **pip install –r requirements.txt**



가상환경 (#4/4)

INDUSTRIAL AI RESEARCH CENTER

▣ 파이참 가상환경 설정 : File>Settings>Project>Python Interpreter → 터미널 재실행



RPA(robotic process automation)

INDUSTRIAL AI RESEARCH CENTER

▣ RPA란?

- 사람이 컴퓨터로 하던 정형화되고 반복적인 업무를 프로그램으로 자동화하는 것
- (소프트웨어/디지털) 로봇 또는 봇 → **사무자동화**
- 맥킨지(McKinsey)의 조사에 따르면 60%의 직업에서 "**직무활동 중 최소 1/3 이상**을 자동화할 수 있다"

▣ RPA를 위한 파이썬 모듈 → 프로그래머가 아닌 일반인도 가능

분야	파이썬 모듈	비고
엑셀 자동화	**openpyxl, win32com**	Day 1
데이터 전처리, 분석	**numpy, pandas, matplotlib**	Day 2
데이터베이스	sqlite	-
Web Scraping, Crawling	**requests, BeautifulSoup**	Day 3
Web browser 자동화	**Selenium**	Day 3
이메일 자동화	smtplib, EmailMessage	-
보고서 생성 자동화	Jinja2	-
모니터 화면제어(키보드,마우스)	**pyautogui**	Day 3
자동화 통합관리	Robot Framework	-
작업 스케쥴링(주기적인 실행)	schtasks(win), crontab(linux)	-
OCR (문자/숫자 인식)	tesseract	-
윈도우(GUI) 프로그래밍	pyQt	-

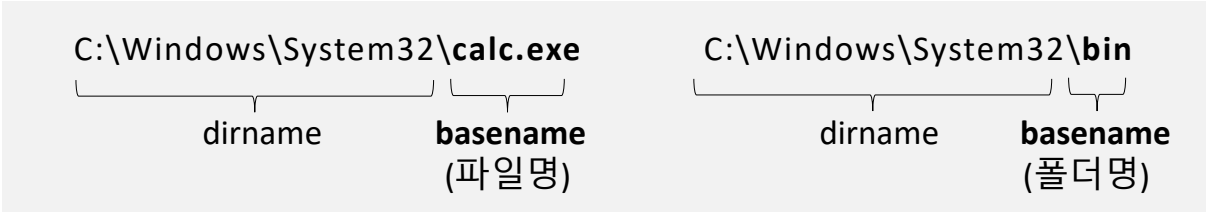
파일경로 다루기 (#1/2)

INDUSTRIAL AI RESEARCH CENTER

▣ 윈도우와 리눅스(맥)의 차이점

- 경로 구분자 : 윈도우에서는 '\', 리눅스에서는 '/' 사용
- 대소문자 구분 : 윈도우에서는 구분하지 않음 vs. 리눅스에서는 구분

▣ Base name은 맨마지막에 오는 것으로, 파일명일 수도 있고 폴더명일 수도 있다.



▣ os 표준모듈

- 현재경로 (.)
- 상위경로 (..)
- 디렉토리 만들기 (os.mkdir)
- 디렉토리 삭제 (os.rmdir)

파일경로 다루기 (#2/2)

INDUSTRIAL AI RESEARCH CENTER

▣ os 표준모듈

기능	사용예
현재 작업 폴더 얻기	**os.getcwd()**
디렉토리 변경/생성	os.chdir("C:\Tmp") / os.mkdir("C\Tmp")
특정 경로에 대해 절대 경로 얻기	os.path.abspath(".\Scripts")
경로 중 디렉토리명만 얻기	os.path.dirname("C:/Python35/pip.exe")
경로 중 파일명만 얻기	os.path.basename("C:/Python35/pip.exe")
경로 중 디렉토리명과 파일명을 나누어 얻기	dir, file = os.path.split("C:/Python35/pip.exe")
파일 각 경로를 나눠 리스트로 리턴하기 os.path.sep은 OS별 경로 분리자	"C:\Python35\pip.exe".split(os.path.sep) → ['C:', 'Python35', 'pip.exe']
경로를 병합하여 새 경로 생성	os.path.join('C:\Tmp', 'a') → "C:\Tmp\a"
디렉토리 안의 파일/서브디렉토리 리스트	os.listdir("C:\Python35")
파일/디렉토리가 존재하는지 체크	os.path.exists("C:\Python35")
디렉토리가 존재하는지 체크	os.path.isdir("C:\Python35")
파일 경로가 존재하는지 체크	os.path.isfile("C:\Python35\python.exe")
파일의 크기	os.path.getsize("C:\Python35\python.exe")

파일 입출력

INDUSTRIAL AI RESEARCH CENTER

▣ 1단계 : 파일 경로 만들기 (단축키: Ctrl+Shift+C)

- **os.getcwd()**

```
# 다양한 파일경로 설정방법
fname = os.path.join(os.getcwd(), 'hello.txt')
fname = 'D:\\52-Python\\Literacy2023\\hello.txt'
fname = r'D:\52-Python\Literacy2023\hello.txt'
fname = 'D:/52-Python/Literacy2023/hello.txt'
```

▣ 2단계 : 파일객체 생성

- 내장함수 **f = open(filepath, mode, [encoding='utf-8'])**

'r' : 읽기(default) – 파일이 없으면 에러 발생

'w' : 쓰기 - 새로운 파일 생성 or 기존 파일 덮어쓰기, 'w+' – 읽고 쓰기

'a' : 추가 - 새로운 파일 생성 or 기존 파일의 뒷부분에 추가 (파일포인터가 파일끝)

'rb', 'wb', 'ab' : 이진파일 – 용량이 적고, 고속

▣ 3단계 : 읽고/쓰기

- 작은 파일일 경우 → **f.readlines(), f.read()**로 전체를 한꺼번에 읽음
- 큰 파일은 메모리를 고려하여 한 라인씩 읽는 **f. readline()**을 권장함

▣ 4단계 : 파일객체 닫기 → **f.close()**

[실습] 파일 입출력

```
fname = os.path.join(os.getcwd(), '인사말.txt')  # 파일경로 설정 (C:\\Python\\hello.txt)

f = open(fname, 'rt',  encoding='utf-8')     # 파일객체 생성: r,w,a + t / b

# 파일 읽기
contents = f.read()          # 방법1 : 한꺼번에 읽기 - read(n) : n문자
print(contents)

lines = f.readlines()        # 방법2 : 모든 라인을 리스트로 반환
print(lines)
f.seek(0)                    # 파일포인터 이동 (byte 단위)
f.tell()                     # 파일포인터 위치 반환
for line in lines:           # 한 라인씩 순회하며 읽음
    print(line, end='')

while True:
    line = f.readline()      # 방법3 : 한 라인씩 읽기
    if not line:
        break
    print(line, end='')

f.close()                    # 파일객체를 닫음
```

```
# 컨텍스트를 이용한 편의문법 : 코딩이 줄고, 자동 close
fname = os.path.join(os.getcwd(), '인사말.txt')
with open(fname, 'r', encoding='utf-8') as f:
    contents = f.read()
print(contents)
```


엑셀파일 지원 모듈

▣ 엑셀파일 지원 파이썬 모듈

- xlswt
- **OpenPyXL**
- XlsxWriter
- PyExcelerate

http://openpyxl.readthedocs.org

▣ openpyxl 모듈

- 엑셀 프로그램이 설치되어 있지 않아도 사용 가능
- 대용량, 이미지 지원 등 엑셀의 모든 기능이 다 있다.
- 문서화가 잘되어 있고, 최근까지 활발하게 업데이트가 되고 있다.
- **pip install openpyxl**

```
import  openpyxl

openpyxl.load_workbook(fname)
```

▣ win32com 모듈 (MS COM 방식)

- 엑셀 프로그램이 설치되어 있어야 함
- 성능 우수, openpyxl에 비해 3배 정도 빠름

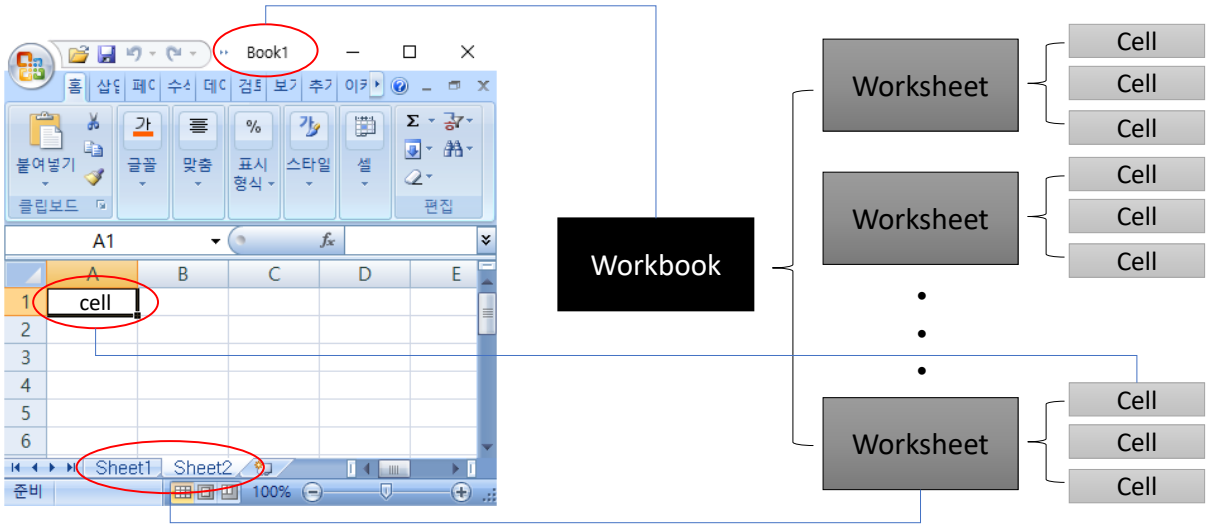
```
import win32com.client      # pip install pywin32

excel = win32com.client.Dispatch("Excel.Application")
excel.Visible = False
excel.Workbooks.Open(fname)
excel.Quit()
```

엑셀파일의 구성요소

INDUSTRIAL AI RESEARCH CENTER

▣ **Workbook(파일), Worksheet(시트), Cell(셀)**이라는 객체로 구성

- 엑셀 통합문서, 파일(*.xlsx)을 workbook이라고 함
- workbook은 하나 이상의 worksheet로 구성됨
- 현재 활성화되어 있는 워크시트를 active sheet라고 함
- Worksheet는 row(행)과 column(열) 구조의 cell로 구성되고, cell에 데이터가 있음



OpenPyXL - 엑셀파일 열기

INDUSTRIAL AI RESEARCH CENTER

▣ 엑셀문서(Workbook) 열기

▣ 시트(Worksheet) 접근

- 시트 목록을 리스트 반환
- Active Sheet 가져오기
- 특정 Sheet 가져오기

```
import openpyxl

wb = openpyxl.load_workbook('재고장.xlsx')

wb.sheetnames  # ['물류재고장', '입출고내역', '랙번호']
ws = wb['물류재고장']
ws = wb.worksheets[0]
ws = wb.active # 현재 활성화된 sheet를 가져옴

row = ws[2:5]
col = ws['A:C']
area = ws ['A1:C3' ]
```

▣ 셀 접근

```
ws['A1']          # 셀자체(객체)
ws['A1'].value    # 셀내용(값)

c = ws['A1']
c.value           # 셀의 값(내용) : '랙번호'
c.coordinate      # 셀의 좌표(위치) : A1

'Cell ' + c.coordinate + ' is ' + c.value
 → 'Cell A1 is 랙번호'
```

OpenPyXL - 데이터 접근

INDUSTRIAL AI RESEARCH CENTER

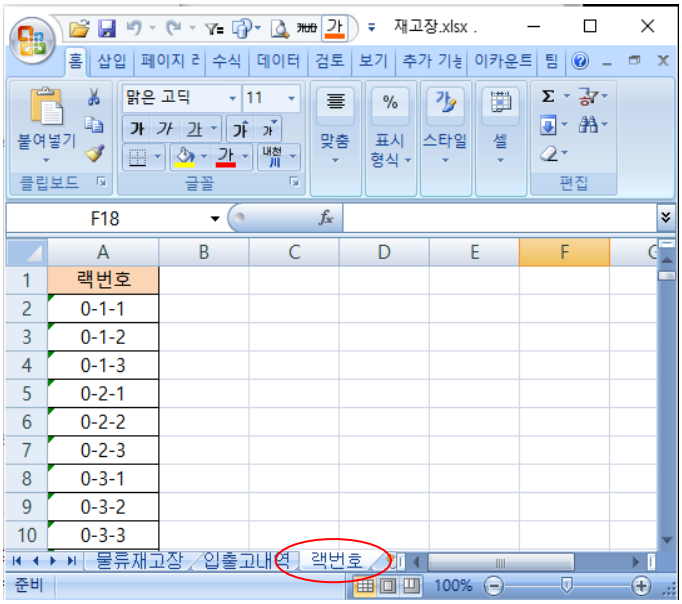
▣ cell()함수를 이용하여 index로 접근 → 루프를 돌릴 때 유리함

- 행수 : ws.max_row

- 열수 : ws.max_column

```
# Cell 함수로 Cell에 접근하기
ws.cell(row=1, column=1)

ws.cell(1, 1).value
ws.cell(1, 1).coordinate

for i in range(1, 8, 2):
    print(i, ws.cell(i, 2).value)
```

```
1 랙번호
3 0-1-2
5 0-2-1
7 0-2-3
```

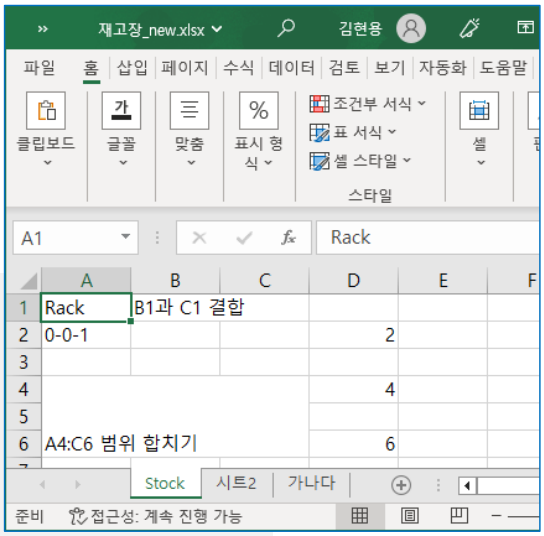


OpenPyXL - 엑셀파일 생성

INDUSTRIAL AI RESEARCH CENTER

▣ 새 문서 만들고 저장하기

- 워크북 생성 : 1개의 'Sheet'가 생성됨

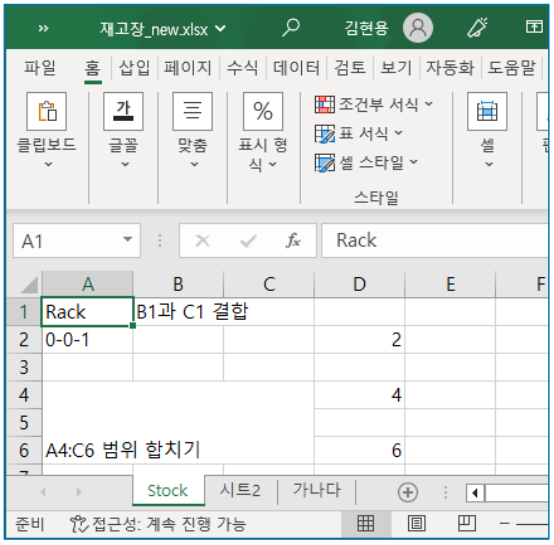



```
from openpyxl import Workbook

wb = Workbook()          # 새로운 엑셀파일 생성

wb.sheetnames            # default로 'Sheet' 시트 생성
ws = wb.active           # 현재 활성시트
ws.title = 'Stock'       # sheetname 변경

wb.create_sheet('가나다')          # 마지막 위치에 '가나다' 시트 추가
wb.create_sheet('시트2', 1)        # 해당위치 뒤에 '시트2' 시트 추가

ws['A1'].value = 'Rack'
# ws['A1'] = 'Rack'              # value 생략 가능
ws.cell(1,1).value = 'Rack'      # 인덱스가 (1,1)부터 가능, value 생략불가

wb.save('재고장_new.xlsx')       # 저장
```

OpenPyXL – 셀 편집

▣ merge_cells() 함수를 사용하여 셀 병합/해제

```
ws.merge_cells('B1:C1')
ws['B1'].value = 'B1과 C1 결합'
ws.merge_cells('A4:C6')
ws['A4'].value = 'A4:C6 범위 합치기'

ws.unmerge_cells('A4:C6')
```



▣ 수식(formulas) 사용하기

```
ws['D6'].value = '=SUM(D1:D5)'
```

▣ 행, 열 삽입/삭제/이동

```
ws.insert_rows(8)     # 8번째 행 삽입 : insert_rows(idx, amount)
ws.insert_rows(8, 5)  # 8번째 행에서 아래로 5열 추가
ws.insert_cols(2, 3)  # B번째 열에서 우측에 3열 추가

ws.delete_rows(8)     # 8번째 행 삭제
ws.delete_cols(2, 3)  # B번째 열에서 우측 3열 삭제

ws.move_range('B1:C11', rows=0, cols=1 )  # 행은 그대로, 열을 1칸 이동
```


OpenPyXL - 셀 데이터 불러오기

▣ 엑셀파일을 읽어와서 2차원 리스트에 할당하기

```
from openpyxl import load_workbook

wb = load_workbook( ' 재고장.xlsx ' )
#ws = wb.active
ws = wb[ ' 물류재고장' ]

data = []
for row in range(ws.max_row):
    line = []
    for col in range(ws.max_column):
        line.append( ws.cell(row+1, col+1).value )
    data.append(line)
```

	A	B	C	D	E	F	G
1	랙번호	부품코드	부품명	공정	재고량	주문량	생산일자
2	0-1-2	SV0185	헤라 로지-사틴크림 50ML 용기 내용기 증착	증착	9,494	5,000	2018-04-01
3	0-2-2	SC0086	한율 어린쑥수분진정크림 80ML(18한정 캡 외캡 무광코팅	코팅	4,475	4,000	2018-03-14
4	0-2-2	SI1017	한율 어린쑥수분진정크림 80ML(18한정 캡 내캡 사출	사출	3,718	5,000	2018-04-16
5	0-9-3	1940400015	PUSH-PHONE CAP 태평양-35φ 10각형(반투명)	완제품	7,000	2,000	2018-06-07
6	1-1-1	21051450018	kimchi keeper handle	완제품	160	500	2018-02-21
7	2-1-1	21051450017	HAPPY KIMCHI KEEPER PLAIN H	완제품	888	200	2018-05-30
8	1-1-1	7245842	아리따움 뽀오얀미소발효클렌징크림 200ml 환판	완제품	1,000	1,000	2018-04-04
9	1-2-1	7334281	중APC)려함빛극손상케어트200ML(가는모발)스크류캡	완제품	741	300	2017-04-20
10	1-1-2	7233467	미쟝센 버블컬러산화제 60ML 캡(신)	완제품	15,000	5,000	2018-05-21
11	1-1-3	7338317	설화수 탄력크림(견 5ML(15AD) 캡	완제품	20,000	10,000	2018-06-01
12	1-1-6	1940400013	PUSH-PHONE CAP태평양-35파이 (10각큰캡)	완제품	5,000	3,000	2018-05-10

물류재고장 입출고내역 랙번호

OpenPyXL - 셀서식

▣ 숫자 서식 지정하기 https://openpyxl.readthedocs.io/en/stable/styles.html#introduction

- cell의 **number_format** 속성 : 0 (zero 표시) # (zero 제거suppression)

▣ 폰트 설정 : **Font** 객체 - name, size, bold, italic, underline, strike(취소선), color

```
font = Font(name='Calibri',
            size=11,
            bold=False,
            italic=False,
            vertAlign=None,
            underline='none',
            strike=False,
            color='FF000000')
```

▣ 셀 채우기 : **PatternFill** 객체 - patternType, fgColor, bgColor, …

▣ 텍스트 정렬

- **Alignment** 객체 : horizontal, vertical, text_rotation, wrap_text, shrink_to_fit, indent

horizontal = general / left / center / right / fill / centerContinuous / justify(양쪽맞춤) / distributed(균등분할)

vertical = bottom / center / top / justify / distributed

▣ 테두리 지정하기

- **Side** 객체 : style='thin/medium/thick/dotted/dashDot/slantDashDot', color

- **Border** 객체 : left, right, top, bottom, diagonal, diagonal_direction, outline, vertical, horizontal

```
cell.number_format = '#,##0'          # 천 단위 쉼표로 자릿수 표시
cell.number_format = '#,##0.00'       # 소수점 이하 자릿수 표시
cell(i, 5).font = Font(bold=True)
cell.fill = PatternFill(patternType='solid', fgColor="AA8866")
cell.alignment = Alignment(horizontal='distributed')
side = Side(style='thick', color='00FF00')
cell.border = Border(left=side, right=side, top=side, bottom=)
```


[실습] 엑셀파일 꾸미기 (#1/3)

▣ 셀서식 지정하기

C19

	A	B	C	D	E	F	G
1	랙번호	부품코드	부품명	공정	재고량	주문량	생산일자
2	0-1-2	SV0185	헤라 로지-사틴크림 50ML 용기 내용기 증착	증착	9,494	5,000	2018-04-01
3	0-2-2	SC0086	한율 어린쑥수분진정크림 80ML(18한정 캡 외캡 무광코팅	코팅	4,475	4,000	2018-03-14
4	0-2-2	SI1017	한율 어린쑥수분진정크림 80ML(18한정 캡 내캡 사출	사출	3,718	5,000	2018-04-16
5	0-9-3	1940400015	PUSH-PHONE CAP 태평양-35φ 10각형(반투명)	완제품	7,000	2,000	2018-06-07
6	1-1-1	21051450018	kimchi keeper handle	완제품	160	500	2018-02-21
7	2-1-1	21051450017	HAPPY KIMCHI KEEPER PLAIN H	완제품	888	200	2018-05-30

F8 1000

	A	B	C	D	E	F	G
1	랙번호	부품코드	부품명	공정	재고량	주문량	생산일자
2	0-1-2	SV0185	헤라 로지-사틴크림 50ML 용기 내용기 증착	증착	9,494	5,000	2018-04-01
3	0-2-2	SC0086	한율 어린쑥수분진정크림 80ML(18한정 캡 외캡 무광코팅	코팅	4,475	4,000	2018-03-14
4	0-2-2	SI1017	한율 어린쑥수분진정크림 80ML(18한정 캡 내캡 사출	사출	3,718	5,000	2018-04-16
5	0-9-3	1940400015	PUSH-PHONE CAP 태평양-35φ 10각형(반투명)	완제품	7,000	2,000	2018-06-07
6	1-1-1	21051450018	kimchi keeper handle	완제품	**160**	500	2018-02-21
7	2-1-1	21051450017	HAPPY KIMCHI KEEPER PLAIN H	완제품	**888**	200	2018-05-30

[실습] 엑셀파일 꾸미기 (#2/3)

INDUSTRIAL AI RESEARCH CENTER

```
import openpyxl
from openpyxl.styles import Alignment, PatternFill, Font, Border, Side

TITLE_CELL_COLOR = "AA8866“                              # 상수

wb = openpyxl.load_workbook('재고장.xlsx')
ws = wb['물류재고장']
# ws.append(['1-2-3', 'SI1234', '추가된 사출성형 품목', '사출', 6000, '2021-03-29'])

ws.freeze_panes = "C2“              # 틀고정 : ws.freeze_panes = 'A1' , None (고정없음)

col_widths = {'A':8, 'B':13, 'C':50, 'D':10, 'E':10, 'F':15}      # 열 크기 지정
for pos, width in col_widths.items():
    ws.column_dimensions[pos].width = width

# 행 높이 지정
for i in range(1, ws.max_row+1):                  # ws.row_dimensions[1].height = 30
    ws.row_dimensions[i].height = 20    #
    ws.cell(i, 5).number_format = '#,##0'
    if i != 1 and int(ws.cell(i, 5).value) < 1000:
        ws.cell(i, 5).font = Font(bold=True)
```


[실습] 엑셀파일 꾸미기 (#3/3)

INDUSTRIAL AI RESEARCH CENTER

```
# 폰트 지정
font_header = Font(name='맑은 고딕', size=12, bold=True, color='FFFFFF')
for cols in ws['A1':'F1']:
    for cell in cols:
        cell.fill = PatternFill(patternType='solid', fgColor=TITLE_CELL_COLOR)
        cell.alignment = Alignment(horizontal='distributed')
        cell.font = font_header

# 셀 테두리 지정
side = Side(style='thin', color='000000')
border = Border(left=side, right=side, top=side, bottom=side)
for row in ws:
    for cell in row:
        cell.border = border

# ws.column_dimensions['A'].hidden = True          # 열 숨기기
# # ws.column_dimensions['A'].hidden = False      # 숨긴 열 표시하기
# ws.row_dimensions[1].hidden = True              # 행 숨기기

wb.save('재고장_new.xlsx')
wb.close()
```

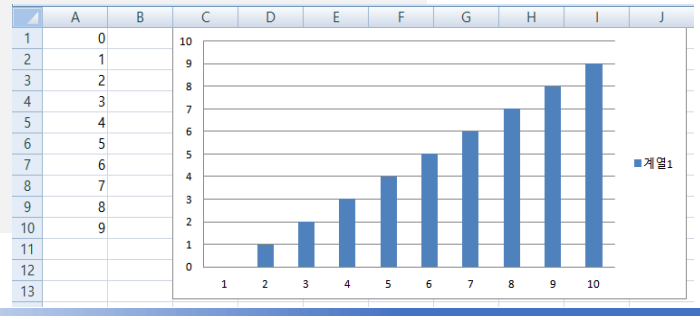
Openpyxl - 차트 그리기

INDUSTRIAL AI RESEARCH CENTER

▣ 차트 그리기 절차 (https://openpyxl.readthedocs.io/en/stable/charts/introduction.html)

- 셀의 직사각형 선택영역으로부터 Reference 객체를 만든다.
- Chart 객체를 만든다.
- 워크시트 개체에 Chart 객체를 추가한다.

```
from openpyxl import Workbook
from openpyxl.chart import BarChart, Reference

wb = Workbook()
ws = wb.active
for i in range(10):
    ws.append([i])

data = Reference(ws, min_col=1, min_row=1, max_col=1, max_row=10)
# labels = Reference(ws, 1, 1, 1, 10)
chart = BarChart()
chart.add_data(data)
# chart.set_catetories(labels)
ws.add_chart(chart, "C1")
wb.save("SampleChart.xlsx")
wb.close()
```



[실습] 엑셀 차트 그리기 (#1/2)

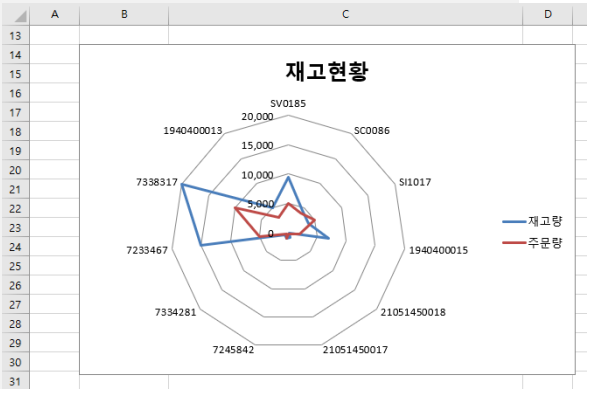
INDUSTRIAL AI RESEARCH CENTER

▣ 재고장.xlsx

	A	B	C	D	E	F	G
1	랙번호	부품코드	부품명	공정	재고량	주문량	생산일자
2	0-1-2	SV0185	헤라 로지-사틴크림 50ML 용기 내용기 증착	증착	9,494	5,000	2018-04-01
3	0-2-2	SC0086	한율 어린쑥수분진정크림 80ML(18한정 캡 외캡 무광코팅	코팅	4,475	4,000	2018-03-14
4	0-2-2	SI1017	한율 어린쑥수분진정크림 80ML(18한정 캡 내캡 사출	사출	3,718	5,000	2018-04-16
5	0-9-3	1940400015	PUSH-PHONE CAP 태평양-35φ 10각형(반투명)	완제품	7,000	2,000	2018-06-07
6	1-1-1	21051450018	kimchi keeper handle	완제품	160	500	2018-02-21
7	2-1-1	21051450017	HAPPY KIMCHI KEEPER PLAIN H	완제품	888	200	2018-05-30
8	1-1-1	7245842	아리따움 뽀오얀미소발효클렌징크림 200ml 환판	완제품	1,000	1,000	2018-04-04
9	1-2-1	7334281	중APC)려함빛극손상케어트200ML(가는모발)스크류캡	완제품	741	300	2017-04-20
10	1-1-2	7233467	미쟝센 버블컬러산화제 60ML 캡(신)	완제품	15,000	5,000	2018-05-21
11	1-1-3	7338317	설화수 탄력크림(견 5ML(15AD) 캡	완제품	20,000	10,000	2018-06-01
12	1-1-6	1940400013	PUSH-PHONE CAP태평양-35파이 (10각큰캡)	완제품	5,000	3,000	2018-05-10

```
import openpyxl
from openpyxl.chart import Reference, BarChart, LineChart, AreaChart, PieChart, RadarChart

wb = openpyxl.load_workbook('재고장.xlsx')
ws = wb['물류재고장']

# chart 데이터 참조범위 지정
data = Reference(ws, min_col=5, min_row=1, max_col=6, max_row=12)
labels = Reference(ws, 2, 2, 2, 12)              # Reference(ws, 5, 2, 5, 12)
```

```
# 차트 종류 지정
# chart = BarChart()          # 막대그래프
# chart.type = 'bar'          # 가로(bar), 세로(col) - 디폴트값
# chart = LineChart()         # 꺾은 선 그래프
# chart = AreaChart()         # 영역형 그래프
# chart = PieChart()          # 원형 차트
chart  = RadarChart()         # 방사형 차트

chart.grouping = 'stacked'  # 누적
# chart.overlap = 100       # for barchart: 100

chart.add_data(data, titles_from_data=True) # title...=True: 첫 셀값을 계열값(범례)로 사용
chart.set_categories(labels)

chart.title = '재고현황'          # 차트 제목
chart.x_axis.title = '랙번호'     # x축 제목
chart.y_axis.title = '재고량'     # y축 제목

chart.height = 10               # 차트 가로 크기
chart.width = 20                # 차트 세로 크기
ws.add_chart(chart, 'B14')      # chart를 B14에 삽입

wb.save('재고장_radarchart.xlsx')
wb.close()
```



감사합니다