



1. 통계 패키지(소프트웨어)

- 컴퓨터로 통계 및 계량분석이 가능하도록 계산 과정을 정리해 놓은 프로그램
- SAS(Statistical Analysis System)
- SPSS(Statistical Package for the Social Sciences)
- Stata(Statistics Data)
- WinRats-32(Regression Analysis for Time Series)
- EViews(Econometric Views)
- Limdep(Limited Dependent model) 등

2. 프로그래밍 언어

- 컴퓨터에 명령을 내리는 데 필요한 '컴퓨터의 언어'
- Basic, Cobol, Fortran
- C, C++
- GAUSS(Matrix programming language), Matlab, S-plus
- R, Python 등



3. Python

- Python은 네덜란드 프로그래머인 Guido and Rossum이 1989년에 개발한 범용 프로그래밍 언어이며 무료로 제공되고 있음
- Python은 데이터 분석 외에도 웹 개발, 게임 개발, 멀티미디어, 보안 등 대부분의 애플리케이션 도메인 에서 사용되는 범용 언어임
- Python은 가장 많이 사용되는 10대 프로그래밍 언어 중의 하나이며 과학 컴퓨팅을 위한 주요 도구이 어서 데이터 과학(data science) 분야에서 사용이 빠른 속도로 확대되고 있음
- Python은 구문이 읽고 기억하기 쉬워서 머신 러닝 및 인공 지능 등 데이터 과학, 천문학, 기상학, 화학, 생물학 등 자연과학, 경제학을 비롯한 사회과학에서 많이 사용되고 있는 추세임
- Python은 많은 Python package(라이브러리 또는 모듈이라고 함)의 지원을 받고 있음
 - 데이터 통합 및 데이터 분석용 판다스(pandas) 패키지
 - 머신 러닝을 구축하기 위한 scikit-learn 패키지
 - 다차원 데이터의 수학적 연산을 위한 numpy 패키지
- 4. Spyder
 - 통합개발환경(Integrated Development Environment; IDE) : 소프트웨어 개발 과정에서 필요한 코딩 (coding), 디버깅(debugging), 컴파일(compile)의 과정을 하나로 패키지화한 소프트웨어
 - 코딩 : 프로그래밍 언어를 이용하여 구체적인 컴퓨터 프로그램을 만드는 기술
 - 디버깅 : 코드상의 오류를 찾아내어 수정하는 과정
 - 컴파일 : 컴퓨터가 처리한 언어를 사람이 읽을 수 있는 언어나 그림으로 변환하는 프로그램
 - Spyder는 Python의 통합개발환경 중 하나임

1. Anaconda 설치

- Python 프로그램을 작성하는 방법은 다양하지만 시작 단계에서는 기본 Python 언어 외에 주요 라이 브러리가 포함되어 있는 Python 배포판(Python distribution)을 설치하는 것이 좋음
- Python 배포판 중에서 가장 많이 사용되는 것이 anaconda인데 그 이유는 anaconda를 설치하면 필 요한 주요 패키지를 한 번에 설치할 수 있기 때문임
- anaconda를 설치하기 위해서는 웹페이지(https://www.anaconda.com/)에 접속하여 다음과 같은 순 서로 진행함
 - 웹페이지 초기 화면의 우측 상단에 있는 Free Download를 클릭





• Free Download를 클릭하면 나타나는 다음의 화면에서 우측 하단에 있는 Skip registration을 누름

Distribution

Free Download*

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, Al Assistant, Learning and more.

- Easily search and install thousands of data science, machine learning, and AI packages
- Manage packages and environments from a desktop application or work from the command line
- Deploy across hardware and software platforms
- Distribution installation on Windows, MacOS, or Linux

*Use of Anaconda's Offerings at an organization of more than 200 employees requires a Business or Enterprise license. See Pricing





• Skip registration을 클릭하면 나타나는 다음의 화면에서 Windows, Mac 및 Linux용 Anaconda가 있으 므로 운영체제별 선택 메뉴에서 사용자에게 적합한 Anaconda를 다운받아 설치하면 됨



For installation assistance, refer to Troubleshooting.

Download Distribution by choosing the proper installer for your machine.



Anaconda Installers



Windows

Python 3.12



Python 3.12

64-Bit (Intel chip) Graphical Installer (734.7M)



Python 3.12

业 64-Bit (x86) Installer (1007.9M)

2. Spyder 시작하기

- Ănaconda를 설치하면 Anaconda3 (64-bit) 앱이 나타나고(좌측 그림) 아래에 Anaconda가 제공해 주 는 데이터 분석에 필요한 도구를 나타내는 아이콘이 있는데 Spyder는 그 중의 하나임
- Spyder 아이콘을 클릭하면 우측 그림과 같이 Script editor 창, Console 창, Help/Variable Explorer/Files/Plots/Code Analysis 창 등 3개의 창이 나타남
 - Script editor 창 : 프로그램 Script를 작성하고 편집

명령어에 커서를 두고 F5 또는 F9을 눌러 실행

- Console 창 : 명령어를 실행한 결과를 출력
- Help/Variable Explorer/Files/Plots 창
 - Help : 도움말을 확인
 - Variable Explorer : 변수명 및 변수내용을 확인
 - Files : 파일 목록
 - Plots : 그래프 출력Spyder는 Python의 통합개발환경 중 하나임



▲ 제조대학교 Ⅲ. 명령어 실행방법

1. 직접 명령문

- Script editor 창에서 사칙연산을 위한 숫자를 직접 입력하여 한 줄 씩 실행할 경우 해당 줄에서 F9를 눌러 실행하면 Py를 계산기처럼 사용할 수 있음
- 한편, 몇 줄의 명령어를 입력한 후 한 줄 씩 실행할 경우 해당 줄에서 F9를 눌러 실행하고,
 여러 줄을 동시에 실행할 경우 해당 줄을 선택한 후 F9 눌러서 실행하며,
 입력한 명령어 모두를 한 번에 실행할 경우 F5를 눌러 실행 실행
- Console 창의 환영 메시지나 출력되어 있는 내용을 Ctrl+L을 쳐서 지운 후 (example 1-1.py)와 같이 명령어를 입력하여 F5를 누르면 (example1-1.py)의 실행 결과를 보여줌

example1-1.py (직접 명령문 1)	example1-1.py의 실행결과
2**3	runfile('C:/BOOK/PyBasics/PyDE/code/example1-1.py', wdir='C:/BOOK/PyBasics/PyDE/code')
2*3	8
3/3	6
3+3	1.0
3-3	6
print(2**3)	0
print(2*3)	
print(3/3)	
print(3+3)	
print(3-3)	



 Ctrl+L을 실행하여 console 창의 내용을 지운 후 (example1-2.py)와 같이 명령어를 입력하여 F5를 누 르면 (example1-2.py)의 실행 결과를 보여줌

example1-2.py (직접 명령문 2)

print(round(2**.5, 5)) # 2의 제곱근을 계산하고 소수점 5자리까지 인쇄

print(round(2*3, 5)) # 2의 3승을 계산하고 소수점 5자리까지 인쇄

print(round(2/3, 5)) # 2÷3을 계산하고 소수점 5자리까지 인쇄

print(round(3+3, 5)) # 3+3을 계산하고 소수점 5자리까지 인쇄

print(round(3-3, 5)) # 3-3을 계산하고 소수점 5자리까지 인쇄

example1-2.py의 실행결과
runfile('C:/BOOK/PyBasics/PyDE/code/example1-2.py', wdir='C:/BOOK/PyBasics/PyDE/code')
1.41421
6
0.66667
6
0

2. 할당 명령문

- 할당 명령문은 특정한 데이터 또는 연산 결과를 새로운 문자열에 할당하여 하나의 객체를 정의하는 명 령문으로 작업 결과의 반환을 요구하지 않음
- 할당 명령문의 형태는 할당 연산자인 =를 사용
- 여기서 x = [1,2,3,4,5]는 1부터 5까지 5개의 수치형(numeric) 원소를 결합하여 길이 5인 벡터를 생성 한 후 식별문자 'x'에 할당하는 명령문임
- y = np.arange(1, 11, 2)는 np 패키지의 arange함수를 이용하여 1에서 11까지(단, 11은 포함하지 않음) 2의 간격으로 숫자 벡터를 생성함
- (example1-3.py)와 같이 명령어를 입력하여 F5를 누르면 (example1-3.py)의 실행 결과를 보여줌

example1-3.py (할당 명령문)	example1-3.py의 실행결과
import numpy as np # numpy package를 np로 불러옴	runfile('C:/BOOK/PyBasics/PyDE/code/example1-3.py', wdir='C:/BOOK/PyBasics/PyDE/code')
x = [1,2,3,4,5]	x = [1, 2, 3, 4, 5]
y = np.arange(1, 11, 2) # np에 있는 arange 함수를 이용	y = [1 3 5 7 9]
z = x + y	z = [2 5 8 11 14]
w = x - y	w = [0-1-2-3-4]
print("x = ",x)	
print("y = ",y)	
print("z = ",z)	
print("w =",w)	

3. 코드 입력 및 실행

- Source editor 창에서 프로그램 Source를 작성, 편집, 저장, 실행, 불러오기 등을 할 수 있음
 - 작성된 프로그램을 한 줄씩 실행하는 방법은 F9를 누름
 - 여러 줄 또는 모든 줄을 동시에 실행하는 방법은 여러 줄을 선택하고 F9를 누름
 - 모든 줄을 동시에 실행하는 방법은 F5를 누름

print("Q2 quantile of x : ", np.quantile(x, .50))
print("Q1 quantile of x : ", np.quantile(x, .25))
print("Q3 quantile of x : ", np.quantile(x, .75))

- (b1-ch1-1.py)와 같이 명령어를 입력하여 F5를 누르면 (b1-ch1-1.py) 실행 결과를 보여줌

b1-ch1-1.py	b1-ch1-1.py의 실행결과
x=[1,2,3,4,5,6,7,8,9,10]	runfile('F:/BOOK/pyBasics/pyStat/code/b1-ch1-1.py', wdir='F:/BOOK/pyBasics/pyStat/code')
print("x list is : ", x)	x list is : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
x.sort(reverse=True)	Sort with descending order of v list is $\cdot \begin{bmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$
print("Sort with descending order of x list is : ", x)	
mean_x=sum(x)/len(x)	Mean is : 5.5
print("Mean is :",mean_x)	Range of x : 9
range = max(x) - min(x)	Mean is : 5.5
print("Range of x : ", range)	Median is : 5.5
import statistics # statistics package를 이용	Variance of x : 9.1666666666666666666666666666666666666
m_x = statistics.mean(x)	Standard Deviation of x : 3.0276503540974917
me_x = statistics.median(x)	Q2 quantile of x : 5.5
<pre>var_x = statistics.variance(x)</pre>	Q1 quantile of x : 3.25
sd_x = statistics.stdev(x)	Q3 quantile of x : 7.75
print("Mean is :", m_x)	
print("Median is :", me_x)	
<pre>print("Variance of x : ", var_x)</pre>	
print("Standard Deviation of x : ", sd_x)	
import numpy as nn # numpy nackage륵 이용	

4. 프로그램 저장하기 및 불러오기

- 작성된 프로그램 Source를 저장하기 위해서는 File/Save As를 선택한 후 다음 그림과 같이 본인이 원
 하는 디렉터리에 파일이름(예, b1-ch1-1)을 입력하면 되는데 프로그램의 확장자는 py로 지정됨
- 저장된 파일을 불러오기 위해서는 File/Open File을 선택한 후 불러올 파일이 위치한 디렉터리에서 해 당 파일을 불러오면 됨

