

KOREAN Re

코리안리 재보험 주식회사



Pyhton

1. 파이썬 입문(10hr)

2025. 05

디지털러닝
부성순 bakpak@empas.com



파이썬 프로그램의 이해

4. 파이썬 프로그램의 라이브러리(패키지, 모듈)

| 파이썬 프로그램 특징 | 모듈: 자주 사용하는 코드나 유용한 코드를 논리적으로 묶어서 관리하고 사용할수 있도록 하는것임. 000.py가 하나의 모듈이 됨 |
|--|--|
| <ul style="list-style-type: none">✓ 인공지능관련 다양한 모듈 제공✓ 공개소프트웨어(무료)✓ 다양한 무료 모듈 (전세계인이 개발공유)✓ 타언어에 비해 접근성과 사용성이 용이하다. (문법이 쉬움) | <div><div><ul style="list-style-type: none">✓ 수치해석: NumPy, SciPy, SymPy✓ 데이터탐색: Pandas, MDP, Orange✓ 시계열/회귀분석: Statsmodels, Filterpy, Hmmlern✓ 분류인식: Scikit-Learn✓ 고속계산: Teano, Tensorflow</div><div>모듈 / 라이브러리/ 패키지</div></div> <div><ul style="list-style-type: none">✓ 베이지안 모형: PyMC3,✓ 딥러닝: Keras, Lasagne, Blocks✓ 영상신호처리: Pillow, Scikit-image✓ 문서처리: NLTK, Gensim✓ 음향신호처리: PyAudio-Analysiis, LibRosa✓ 확률적 그래프 모형: LibPGM, Pgmpy</div> |



구글에서 검색해 봅니다.

- [1] 파이썬 doc
- [2] 파이썬 라이브러리 종류
- [3] 파이썬 시각화 라이브러리
- [4] 파이썬 이미지 처리 라이브러리
- [5] 파이썬 데이터분석 라이브러리
- [6] 파이썬 인공지능 라이브러리
- [7] 파이썬 머신러닝 라이브러리



구글에서 검색해 봅니다.

- [1] 파이썬 matplotlib
- [2] 파이썬 seaborn
- [3] 파이썬 pandas
- [4] 파이썬 pillow



구글에서 검색해 봅니다.

- [1] filetype:pdf matplotlib
- [2] filetype:pdf pandas
- [3] filetype:pdf numpy



구글에서 검색해 봅니다.

- [1] 파이썬 GPT
<https://wikidocs.net/book/11839>

- [2] gpt 데이터분석 파이썬
<https://wikidocs.net/215183>

- [3] 랭체인 pdf 요약,



ChatGPT에서 검색해 봅니다.

- [1] 파이썬 에디터
- [2] 아나콘다와 미니콘다
- [3] 구글코랩
- [4] 파이썬 쉬움
- [5] 파이썬 어려움

Module 1.

파이썬 에디터 설정

1. 파이썬 에디터 비교 및 설치.pdf 의 **14**페이지부터

→ 파이썬 IDLE(순수파이썬프로그램을 의미함)

→ 주피터노트북 (입력작업과 출력작업을 구역별로 나누어서 할수 있는 에디터, 확장자는 .ipynb임)

별도의 라이브러리로서 pip install jupyter notebook으로 설치해야 사용할수 있음.

→ 구글코랩 (구글 드라이브에서 제공하는 클라우드형 주피터노트북 에디터)

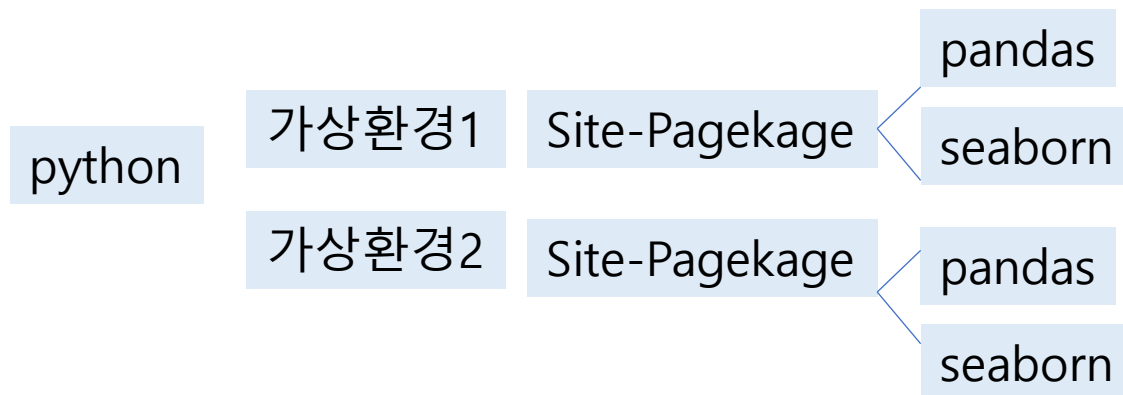
보편적인 라이브러리를 내장해서 제공함으로 라이브러리에 대한 설치 부담감이 적음.

단 자료를 연결할때 내 자료를 클라우드에 업로드해야 하기때문에, 업로드 공간에 대한 추가구매필요함

.파이썬은 라이브러리로 진행되는 프로그램으로 각 라이브러리마다 서로 호환이 안될수 있음.

실제 프로젝트 개발에서는 라이브러리 세트를 잘 맞추어야 하기 때문에 프로젝트별로 작업공간 (가상환경)을 계속 만들면서 작업해야함.

.가상환경 구축하는 방법도 여러가지임. 가장 쉬운 방법으로 안내함.



1

미니콘다다운로드

전체 이미지 동영상 쇼핑 뉴스 짧은 동영상 웹

anaconda.com
https://www.anaconda.com/docs/getting-started/mi...

Miniconda - Anaconda

For the latest Miniconda installers for Python 3.13, go to the Miniconda website. The latest installers are on the same page as the Anaconda Distribution

Installing Miniconda Miniconda release notes System Requirements

2

Apply now.

Pricing

Download >

화면상단

Provide email to download Distribution

Email Address:
bakpak@empas.com

☒ Agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent [here](#) at any time.

By continuing, I agree to Anaconda's [Privacy Policy](#) and [Terms of Service](#).

Submit >

3

Miniconda Installers

Download

For installation assistance, refer to [troubleshooting](#).

Windows

Mac

다운로드 프로그램 설치



Miniconda3-latest-Windows-x86_64.exe

4

Miniconda3 py313_25.3.1-1 (64-bit) Setup
Select Installation Type
Please select the type of installation you would like to perform for Miniconda3 py313_25.3.1-1 (64-bit).

Install for:

☒ Just Me (recommended)
☐ All Users (requires admin privileges)

계정이 한글일때는 반드시 All users

Miniconda3 py313_25.3.1-1 (64-bit) Setup

ANACONDA 설치할때 path 체크

☒ Add Miniconda3 to my PATH environment variable
NOT recommended. This can lead to conflicts with other applications. Instead, use the Command Prompt and PowerShell menus added to the Windows Start Menu.
☐ Register Miniconda3 as my default Python 3.13
Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Miniconda3 as the primary Python 3.13 on the system.
☐ Clear the package cache upon completion
Recommended. Recovers some disk space without harming functionality.

Anaconda, Inc.

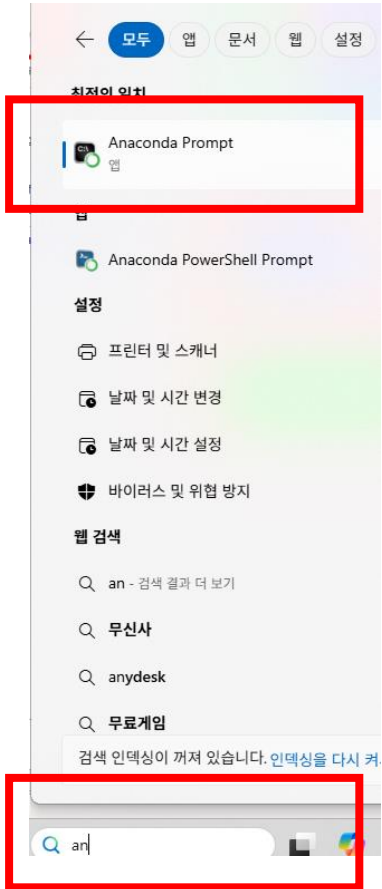
< Back Install Cancel

윈도우에서 cmd입력해서
나타나는 계정명확인

```
Microsoft Windows [Version 10.0.22H2]
(c) Microsoft Corporation. All rights reserved.
C:\Users\user>
```

가상환경 제작 (필수 아님)

1. 윈도우에서 아나콘다 실행



2. Python 실행해봄

```
Anaconda Prompt - python
(base) C:\Users\user>python
Python 3.13.2 | packaged by Anaconda, Inc. |
Type "help", "copyright", "credits" or "license()"
>>> |
```

3. exit() 로 파이썬 종료

```
(base) C:\Users\user>python
Python 3.13.2 | packaged by Anaconda, Inc. |
Type "help", "copyright", "credits" or "license()"
>>> KeyboardInterrupt
>>> exit()
```

4. 가상환경 제작: conda create -n 가상환경이름

```
(base) C:\Users\user>conda create -n nhi
```

conda 기본환경

가상환경 이름

5. 메시지 출력되면 y 눌러서 설치함.

6. 가상환경에 라이브러리를 설치하기 위해 가상환경으로 이동

conda activate 가상환경이름

```
Anaconda Prompt
(base) C:\Users\user>conda activate nhi
(nhi) C:\Users\user>
```

가상환경으로 변동

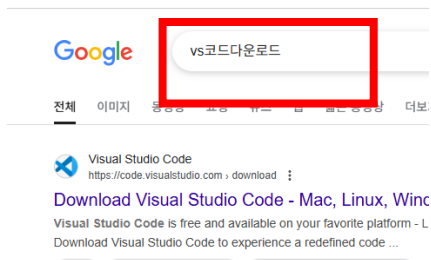
5. pip install 라이브러리명 (한번에 한개씩만 설치해도 가능함)

pip install pandas matplotlib seaborn

```
(nhi) C:\Users\user>pip install pandas matplotlib seaborn
Collecting pandas
```

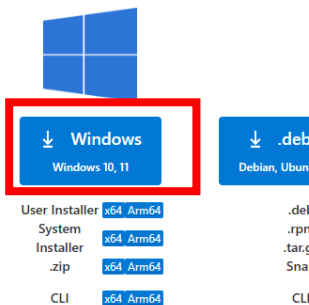
Vs코드 다운로드 및 추가프로그램 설치

1. vs코드 다운로드 및 설치

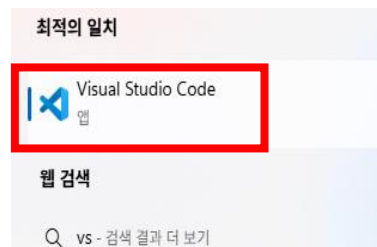


Download

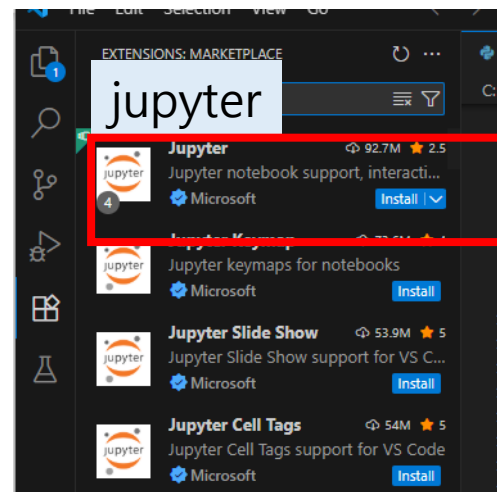
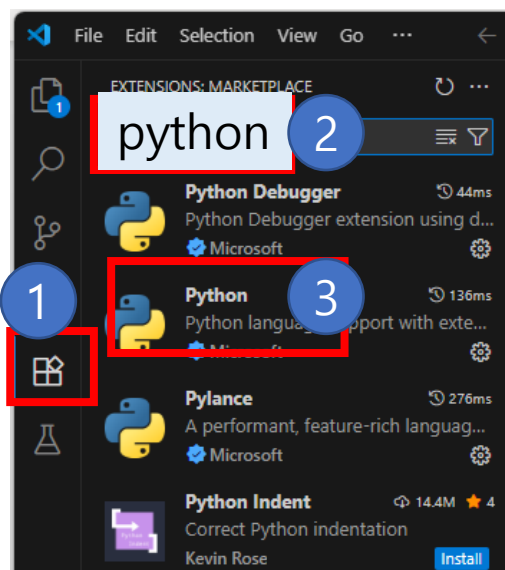
Free and built on open source



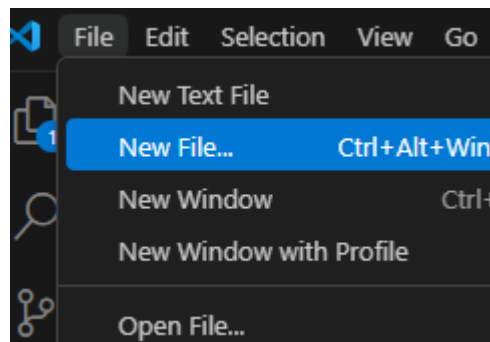
2. 실행



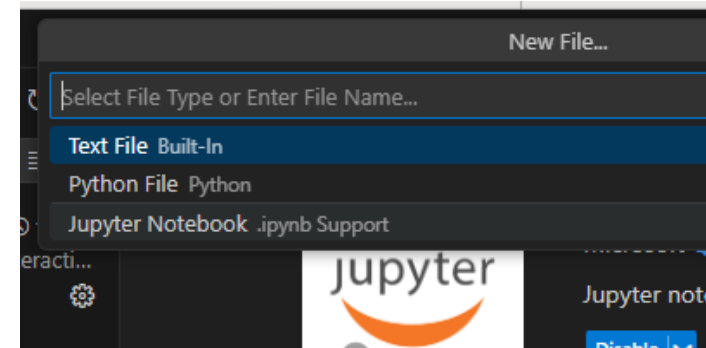
3. 필요한 프로그램 추가 설치



4. 실행

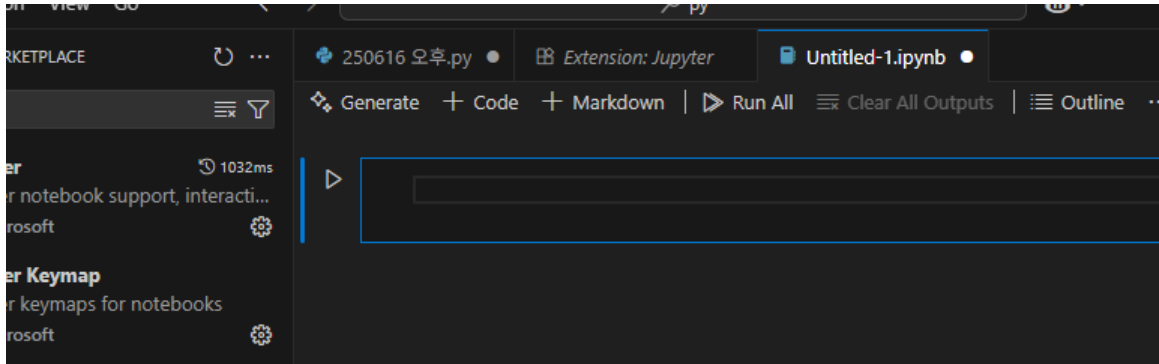


5. Py는 python file로 .ipynb는 주피터노트북으로

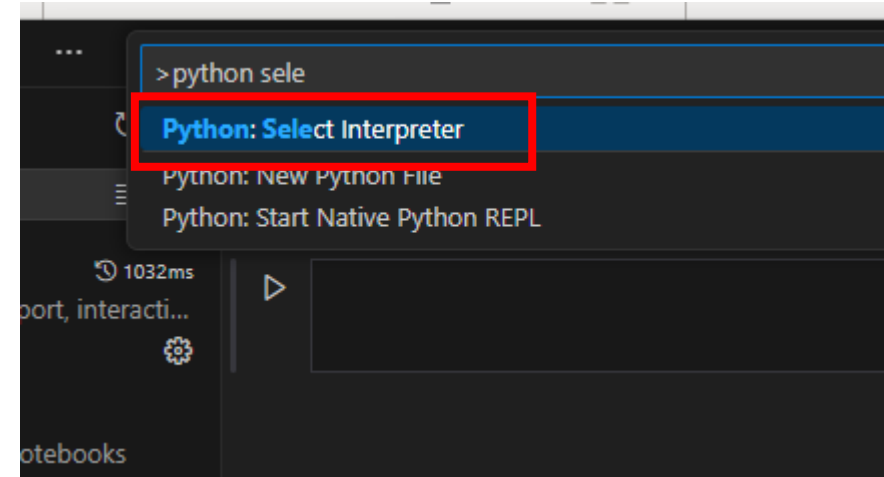


1. file-openFolder 로 작업 폴더를 지정하고 작업하면 좋음

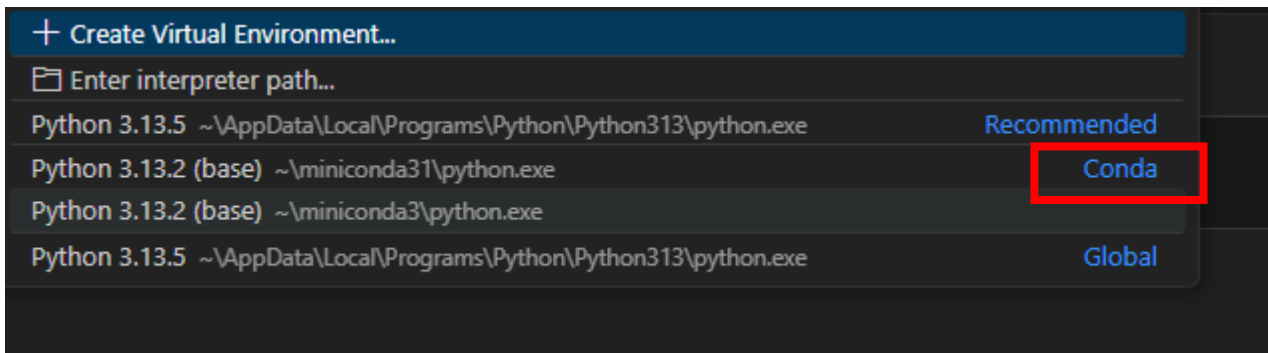
2. 작업화면 py든 ipynb는 에서 ctrl+shift+p 로 가상환경 선택하고 작업해야함.



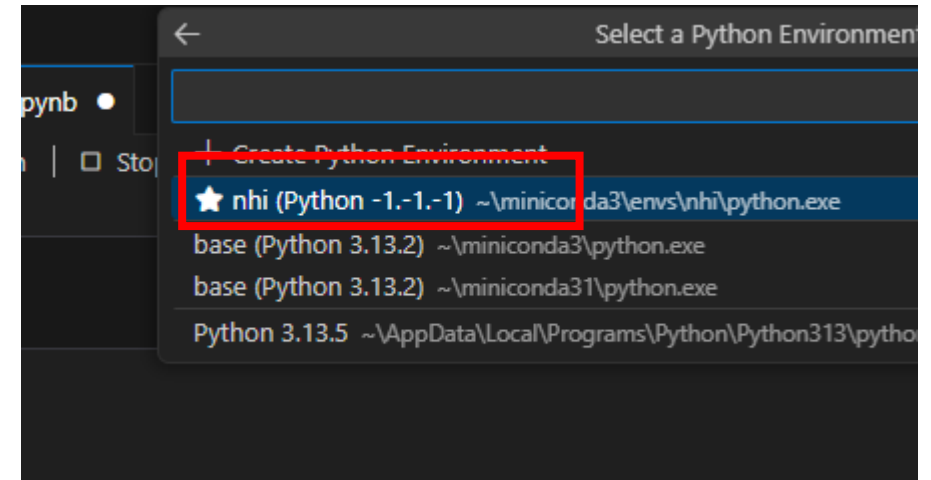
3. Python select interpreter 선택해서



4. 콘다환경 선택해서



5. 가상환경 선택함.



Module 2

파이썬 변수 & 자료형

2. 파이썬 변수.pdf

3. 파이썬 자료형.pdf

파이썬 변수 (파이썬기초 및 데이터분석 및 시각화.pdf 17페이지)

1. 변수

▶ 변수의 이름 작성 규칙

- 변수명에는 문자, 숫자, 언더바(_)를 포함할 수 있으며 숫자는 가장 처음에 등장할 수 없음.
- 파이썬에선 대소문자를 구분하기함.
- 파이썬 지정 단어(예약어)

A 파이썬 변수 특징

A3=20

변수명 자료(값)

숫자, 문자형을 변수에 미리 설정해야 동작하는 일부 프로그램과 달리 파이썬은 실행 시점에 변수의 type(형)이 정해지기 때문에 동적 프로토타이핑 언어라고 말한다.

※ 비주얼메이직언어에서는

Dim Age as Byte

==> Age 변수는 0~255까지 숫자만 입력가능한 변수임을 미리 설정(선언)함

==> Age="20세" 로 입력하면 에러발생

==> 프로그램 개발후 변수의 성격은 변경못할수 있음.

초기개발단계에서 잘 확립해야함.

B 변수명작성규칙

- 영문자(대, 소문자 구분), 숫자, 언더바(_)를 사용할 수 있다.
- 첫 자리에는 숫자를 사용할 수 없다.
- 파이썬 키워드는 변수 명으로 사용할 수 없다.

잘못된 변수명: A-3, A 3, 3A, A.3, if

C 같은 변수명 사용시 마지막에 값이 기록됨

```
In [1]: 1 a+b

-----
NameError
all last)
<ipython-input-1-ca730b97bf8a> in <modu
----> 1 a+b

NameError: name 'a' is not defined
```

GPT에게
질문하기



| | |
|----|-------------------------|
| 01 | 파이썬에서 변수란 |
| 02 | 파이썬 예약어 |
| 03 | 파이썬 예약어 확인하기 |
| 04 | name 'a' is not defined |
| 05 | 파이썬 a-1=3 |

파이썬 변수 (파이썬기초 및 데이터분석 및 시각화.pdf 20페이지)

*변수명으로 사용할수 없는 리스트

```
import keyword #예약어 (if같은)  
print(keyword.kwlist)
```

```
#내장함수 (print등)  
import builtins  
print(dir(builtins))
```

PDF에 없는
내용

예약되어 있는 단어를 변수로 사용하면 안됨, 예약되어 있는 단어는 초록색으로 표시
아래의 내용을 입력하고 shift+Enter로 노트창을 추가하면서 실행해봅니다.

```
: 1 print(3)  
3
```

print는 화면에 값을 뿌려주는
출력구문

```
: 1 print=4  
:  
: 1 print(3)
```

변수=값 일때 왼쪽의
변수명이 초록색으로 나오면
안됨

```
TypeError                                Traceback (most recent call last)  
<ipython-input-5-ce36b08be018> in <module>  
----> 1 print(3)  
  
TypeError: 'int' object is not callable
```

예약어 print를 변수로
사용해서 생기는 에러

```
: 1 print  
:  
: 4
```

print하면 변수로 인지해서
print에 할당된 4가 출력됨

```
: 1 del print  
:  
: 1 print  
:  
: <function print>
```

변수를 삭제하고 다시 작업함.

파이썬 자료형

(GPT프롬프트) 파이썬 자료형을 작업해야하는 영역과 연결해줘

| 자료형 | 예시 | 주요 특징 | 활용되는 작업 영역 |  |
|--------------------------|--------------------------|---------------------|------------------------------|---|
| <code>int</code> (정수) | <code>10, -3</code> | 수학 계산, 반복문 제어 등에 사용 | 수치 분석, 반복제어, 알고리즘 | |
| <code>float</code> (실수) | <code>3.14, -0.5</code> | 소수점 연산 가능 | 통계 계산, 머신러닝 수치 처리 | |
| <code>bool</code> (불린) | <code>True, False</code> | 조건문, 필터링 | 조건 분기, 판다스 필터링 | |
| <code>str</code> (문자열) | <code>"hello"</code> | 텍스트 처리 가능 | 데이터 클렌징, 텍스트 마이닝, 웹 크롤링 | |
| <code>list</code> (리스트) | <code>[1, 2, 3]</code> | 순서 있음, 변경 가능 | 반복 처리, 데이터 그룹 | |
| <code>tuple</code> (튜플) | <code>(1, 2)</code> | 순서 있음, 변경 불가 | 함수 반환값, 좌표 저장 | |
| <code>set</code> (집합) | <code>{1, 2, 3}</code> | 중복 제거, 교집합/합집합 등 | 중복 제거, 데이터 비교 | |
| <code>dict</code> (딕셔너리) | <code>{'a': 1}</code> | 키-값 구조, 빠른 탐색 | JSON 처리, API 응답 저장, DB 조회 결과 | |
| <code>None</code> | <code>None</code> | 값이 없음 표현 | 값 초기화, 누락값 처리 | |

3-3. 문자 자료형

문자열(String)

문자열은 문자들의 모임, 문자를 나열한 것을 문자열이라고 함.

▶ 이스케이프 문자: 별도의 기능을 제공하는 특수한 문자

| 이스케이프 문자 | 설명 |
|----------|------------------|
| \n | 개행(Newline, 줄바꿈) |
| \t | 탭(Tab) |
| \0 | NULL 문자 |
| \\ | 문자 '\' |
| \' | 단일 인용부호(') |
| \" | 이중 인용부호(“) |

자료형에서 리스트난 numpy(배열)은 인덱싱과 슬라이싱이 가장 중요함.

딕셔너리 자료형은 json 형식과 동일하기 때문에 외부대량데이터 연동시 가장 많이 사용됨.
→ 단 구조가 많이 복잡한 경우가 많아 복잡한 딕셔너리 구조형을 학습하는 것이 좋음.

Numpy & 슬라이싱 인덱싱 이해

자료의 형식은 좌측에 있는 엑셀자료와 같음을 인지해야함.

작업1) 자료불러와 제목과 자료 분리

```
신규회원=[['지점','1월','2월','3월'],  
          ['A',1,2,5],  
          ['B',3,4,1],  
          ['C',4,1,1]]
```

```
header=신규회원[0]  
data=신규회원[1:]  
data
```

header

data

| | A | B | C | D |
|---|----|----|----|----|
| 1 | 지점 | 1월 | 2월 | 3월 |
| 2 | A | 1 | 2 | 5 |
| 3 | B | 3 | 4 | 1 |
| 4 | C | 4 | 1 | 1 |
| - | | | | |

작업2) 작업하기 편리하게 인원수만 분리함

```
import numpy as np  
data=np.array(data)  
인원수=data[:,1:].astype(np.uint8)  
인원수
```

data[:, 1:]

모든행

1열이후

| | A | B | C | D |
|---|----|----|----|----|
| 1 | 지점 | 1월 | 2월 | 3월 |
| 2 | A | 1 | 2 | 5 |
| 3 | B | 3 | 4 | 1 |
| 4 | C | 4 | 1 | 1 |
| - | | | | |

Numpy & 슬라이싱 인덱싱 이해

작업3) 인원수 자료중 1월에만 100을 더하고자 함. Numpy 자료임으로 '브로드 캐스트' 가능함

*브로드캐스트: 각 인덱스마다 +100을 하지 않고 행,열 단위로 한번에 작업하는 기능을 의미
넘파이 자료만 브로드캐스트 가능함

| | 1월 | 2월 | 3월 |
|---|----------|----------|----------|
| | 인원수[:,0] | 인원수[:,1] | 인원수[:,2] |
| | A | B | C |
| 1 | 1 | 2 | 5 |
| 2 | 3 | 4 | 1 |
| 3 | 4 | 1 | 1 |

```
#####  
# 1월에만 +100을 하고자할때  
#####  
인원수[:,0]+=100  
인원수  
0s  
[[101, 2, 5],  
 [103, 4, 1],  
 [104, 1, 1]], dtype=uint8)
```

Numpy & 슬라이싱 인덱싱 이해

작업4) 2월자료에만 *2.3 을 하고자 함. dtype변경을 하지 않으면 정수자료임으로 정수만 출력됨

*형은 맨처음 입력할때 0~255의 정수만 입력하였으므로 자동 설정된 것임

구글검색 '넘파이 형선언', '넘파이 형변환'

```
신규회원=[['지점','1월','2월','3월'],  
          ['A',1,2,5],  
          ['B',3,4,1],  
          ['C',4,1,1]]
```

```
1 #####  
2 # 2월에만 *2.3을 하고자할때  
3 # dtype을 변경해야함  
4 #####  
5 인원수=인원수.astype(np.float32)  
6 인원수[:,1]=인원수[:,1]*2.5  
7 인원수  
✓ 0.0s  
  
array([[101. , 12.5, 5. ],  
       [103. , 25. , 1. ],  
       [104. , 5. , 1. ]], dtype=float32)
```

```
1 #####  
2 # 2월에만 *2.3을 하고자할때  
3 # dtype이 int라 소수점이하 절사됨  
4 #####  
5 인원수[:,1]=인원수[:,1]*2.5  
6 인원수  
✓ 0.0s  
  
array([[101, 5, 5],  
       [103, 10, 1],  
       [104, 2, 1]], dtype=uint8)
```

정수형중 0~255까지 숫자

Numpy & 슬라이싱 인덱싱 이해

작업5) C지점에만 전체 -100을 하고자 할때

```
1 #####
2 # C지점에만 전체 -100을 하고자 할때
3 #####
4 인원수[2,:] = 인원수[2,:]-100
5 인원수
```

✓ 0.0s

```
array([[101. , 12.5,  5. ],
       [103. , 25. ,  1. ],
       [  4. , -95. , -99. ]], dtype=float32)
```

인원수[0, :] ← 1
인원수[1, :] ← 2
인원수[2, :] ← 3

| | A | B | C |
|---|---|---|---|
| 1 | 1 | 2 | 5 |
| 2 | 3 | 4 | 1 |
| 3 | 4 | 1 | 1 |

신규회원=[['지점','1월','2월','3월'],
 ['A',1,2,5],
 ['B',3,4,1],
 ['C',4,1,1]]

Numpy & 슬라이싱 인덱싱 이해

작업6) 인덱싱 마무리

```
1 #####
2 # B지점의 3월자료를 100으로 변경하고자할때
3 #####
4 인원수[1,2]=100
5 인원수
✓ 0.0s
array([[101. , 12.5,  5. ],
       [103. , 25. , 100. ],
       [  4. , -95. , -99. ]], dtype=float32)
```

```
1 #####
2 # 2월자료를 모두 50으로 변경하고자 할때
3 #####
4 인원수[:,1]=50
5 인원수
✓ 0.0s
array([[101.,  50.,  5.],
       [103.,  50., 100.],
       [  4.,  50., -99.]], dtype=float32)
```

| | | 1월 | 2월 | 3월 |
|---|---|-----|-----|-----|
| | | A | B | C |
| A | 1 | 0,0 | 0,1 | 0,2 |
| B | 2 | 1,0 | 1,1 | 1,2 |
| C | 3 | 2,0 | 2,1 | 2,2 |

(참고) Numpy의 기능

작업7) Numpy 조건문

이 PT 자 보고난뒤에 별도로 'Numpy 조건문' 으로 검색하여 조건문을 인지합니다.
2개이상의 조건을 동시에 만족하는 And, 조건중 한 개만 만족하는 Or에 대한 문법도 확인합니다.

```
1 #####  
2 # 자료중 50이상인 값만 999로 변경하고자할때  
3 #####  
4 인원수[인원수[:, :]>50]=999  
5 인원수
```

✓ 0.0s

```
array([[999., 50., 5.],  
       [999., 50., 999.],  
       [ 4., 50., -99.]], dtype=float32)
```

```
1 #####  
2 # 위의 공식은 np.where 로 대체될수 있음  
3 #####  
4 인원수[np.where(인원수>50)]=999  
5  
6 인원수  
7
```

✓ 0.0s

```
array([[999., 50., 5.],  
       [999., 50., 999.],  
       [ 4., 50., -99.]], dtype=float32)
```

(참고) Numpy의 기능

작업8) Numpy를 이용하면 편리한 계산식 (브로드캐스트와 동시와 계산식도 제공됨)

브로드 캐스트 및 연산식(합,평균,최대,최소등) 제공함

np.mean→평균

```
1  ### 총인원수 몇명인지 알고 싶을때
2  ### 인원수당*인센티브한자료의 합
3  ### 평균등 계산
4  np.sum(인원수), np.sum(인원수*5000), np.mean(인원수), np.mean(인원수*5000)
```

✓ 0.0s

(3057.0, 15285000.0, 339.66666, 1698333.4)

(참고) Numpy의 기능

작업9) 월별인센티브를 넣어서 각 행에 곱셈 가능함.

```
1 #####3
2 ## 월별 인센티브 가 다를때
3 #####
4 월별인센티브=[20,
5 | | | 50,
6 | | | 15] #1월은 20, 2월은 50, 3월은 15일때
7 인원수, 월별인센티브, 인원수*월별인센티브
```

✓ 0.0s

```
(array([[999., 50., 5.],
       [999., 50., 999.],
       [ 4., 50., -99.]], dtype=float32),
 [20, 50, 15],
 array([[19980., 2500., 75.],
       [19980., 2500., 14985.],
       [ 80., 2500., -1485.])))
```

hstack: 열(세로로) 추가할때

```
1 #####
2 ### 4,5,6월자료가 추가될때
3 #####
4 import numpy as np
5 추가=[ [1,4,5],
6 |      [2,2,3],
7 |      [7,8,9] ]
8
9 인원수=np.hstack((인원수,추가))
10 인원수
```

✓ 0.0s

```
array([[999., 50., 5., 1., 4., 5.],
       [999., 50., 999., 2., 2., 3.],
       [ 4., 50., -99., 7., 8., 9.]])
```

array 끼리 추가(열단위, 행단위) 추가도 array의 shape이 일치해야함.

(참고) Numpy의 기능

array 끼리 추가(열단위, 행단위) 추가도 array의 shape이 일치해야함.

hstack: 열(세로로) 추가할때

```
1 #####
2 ### 4,5,6월자료가 추가될때
3 #####
4 import numpy as np
5 추가= [ [1,4,5],
6         [2,2,3],
7         [7,8,9] ]
8
9 인원수=np.hstack((인원수,추가))
10 인원수
```

✓ 0.0s

```
array([[999., 50., 5., 1., 4., 5.],
       [999., 50., 999., 2., 2., 3.],
       [ 4., 50., -99., 7., 8., 9.]])
```

vstack: 열(세로로) 추가할때

```
1 #####
2 ## E지점의 1월~6월자료를 추가하고자 할때
3 #####
4
5 E지점= [ [1,2,1,1,1,1] ]
6 np.vstack((인원수,E지점))
```

✓ 0.0s

```
array([[999., 50., 5., 1., 4., 5.],
       [999., 50., 999., 2., 2., 3.],
       [ 4., 50., -99., 7., 8., 9.],
       [ 1., 2., 1., 1., 1., 1.]])
```

(참고) Numpy의 기능 (머신러닝이나 인공지능 작업에 필수적인 개념임)

작업10) 내적곱(np.dot)에 대한 이해 : 곱셈한뒤 그 곱셈의 값을 행단위로 더하는것

A,B,C 지점에 월별 인센티브를 더해서 1,2,3월의 총합을 구하고자 할때
계산식을 나누어 작업해도 되지만 (곱셈후, 행단위의 합계) np.dot 기능을 이용하여 한번에 작업가능함

| 인원수 | 각월별인센티브 | 인원수*각월별인센티브 | 내적곱결과 |
|--|--|--|--|
| 1월 2월 3월 | | | |
| $\begin{bmatrix} 3, & 4, & 5 \\ 2, & 3, & 4 \end{bmatrix}$ | $\begin{bmatrix} 100 \\ 200 \\ 300 \end{bmatrix}$ 1월에 곱할값 2월에 곱할값 3월에 곱할값 | $\begin{bmatrix} 3*100, & 4*200, & 5*200 \\ 2*100, & 3*200, & 4*300 \end{bmatrix}$ | $\begin{bmatrix} 2600 \\ 2000 \end{bmatrix}$ |
| 2행*3열 | 3행*1열 | | |

np.dot(a,b)가 되려면 반드시
a의 열과 b의 행값이 같아야함.
결과는 a의 행* b의 열임

```
1 a,b=np.array(a), np.array(b)
2 a*b, np.sum(a*b, axis=1), np.dot(a,b)
✓ 0.0s

(array([[ 300,  800, 1500],
        [ 200,  600, 1200]]),
 array([2600, 2000]),
 array([2600, 2000]))
```

결과물은
2행*1열임

(참고) Numpy의 기능 (머신러닝이나 인공지능 작업에 필수적인 개념임)

작업10) 내적곱(np.dot) 에 대한 이해 : 곱셈한뒤 그 곱셈의 값을 행단위로 더하는것

행열이 다르면 np.dot가 실행되지 않음.

```
1 a=[3,4,5],
2   [2,3,4]
3 b=[100,200,300]
4
5 print(np.shape(a), np.shape(b))
6 np.dot(a,b)
```

87] 0.0s

(2, 3) (1, 3)

ValueError Traceback (most recent call last)

Cell In[87], line 6

3 b=[100,200,300]

5 print(np.shape(a), np.shape(b))

----> 6 np.dot(a,b)

ValueError: shapes (2,3) and (1,3) not aligned: 3 (dim 1) != 1 (dim 0)

이때 reshape을 통해서 행열값을 변경하고 연산가능함.

```
1 a=[3,4,5],
2   [2,3,4]
3 b=[100,200,300]
4 c=np.reshape(b, (3,1))
5 print(np.shape(a), np.shape(b)), np.shape(c)
6 np.dot(a,c)
```

88] 0.0s

(2, 3) (1, 3)

array([[2600],
 [2000]])

Numpy는 이미지 전처리등에서 중요함.
이미지 처리와 관련한 자료는

이미지처리에 필요한 Numpy폴더에서 확인

Module 3

문자열 처리

4. 실무 문자열 처리 예제 모음.pdf

파이썬 문자열 처리

1 문자열 / 인덱싱과 슬라이싱

웹 페이지의 URL을 처리하거나 HTML 요소에서 텍스트를 추출할 때 문자열 자료형을 자주 사용합니다.

```
url = 'https://example.com'  
title = 'Example Domain'
```

인덱싱: url[0] / 슬라이싱 : url[2:5]

```
print(url[0])  
print(url[:3]) # 0,1,2 인덱싱 출력  
print(url[2:5]) # 2,3,4 출력  
print(url[-1]) # 마지막 자리 출력  
print(url[:-3]) # 0부터 마지막 3자리 전까지 출력
```



h
htt
tps
m
[https://example.](https://example.com)

2 문자열 연결

웹 크롤링을 할 때 문자열 연결은 URL을 구성하거나 데이터를 포매팅하는 데 자주 사용됩니다. 파이썬에서 문자열을 연결하는 방법에는 여러 가지가 있으며, 각 방법은 상황에 따라 적절하게 사용될 수 있습니다.

```
base_url = 'https://example.com/page'  
page_number = '1'  
full_url = base_url + page_number  
print(full_url) # Output: https://example.com/page1
```

```
base_url = 'https://example.com/page'  
for page_number in range(5):  
    full_url = base_url + str(page_number)  
    print(full_url)
```

출력물

<https://example.com/page0>
<https://example.com/page1>
<https://example.com/page2>
<https://example.com/page3>
<https://example.com/page4>

(구글검색) 점프투파이썬 정규표현식

<https://wikidocs.net/1669>

3

정규 표현식 (Regular Expressions)

정규 표현식은 문자열 검색과 치환을 효율적으로 수행할 수 있게 해줍니다. 특히 특정 패턴의 텍스트를 추출할 때 유용합니다.

```
import re
```

```
text = 'Contact us at contact@example.com'
```

```
email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
```

```
email = re.findall(email_pattern, text)
```

```
print(email)
```

GPT에게
질문하기



한글을 제외한 모든 문자 제거하는 정규표현식 코드

GPT에게
질문하기



파이썬 정규표현식 대표 패턴

Module 5

파이썬 함수 및 패키지 이해

점프투 파이썬의 함수 <https://wikidocs.net/23>

점프투 파이썬의 클래스 <https://wikidocs.net/27>

5. 파이썬 클래스, 인스턴스, 메서드, super, 오버라이딩의 이해.html

클래스는 게임 또는 인공지능망에서 아주 중요함.

파이썬에서 다른함수의 자료를 넘겨받는 기능을 **super**라고 함. 이 개념만 알면 됨.

Module 6

판다스 및 시각화 패키지

파이썬기초 및 데이터분석 및 시각화.pdf
42페이지부터~