



파이썬 프로그램 기본

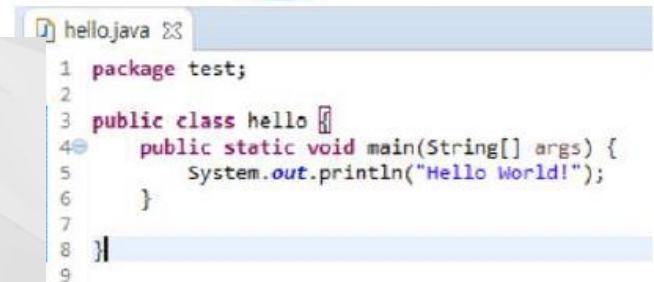
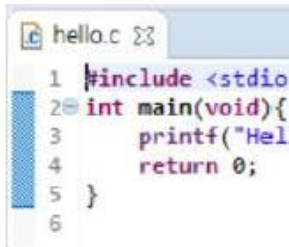
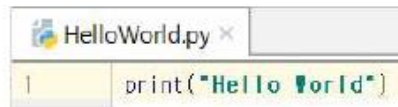
Part1. 파이썬 프로그램 설치 및 실행

학습목표

- 파이썬이 어떤 프로그램이며 사용되는 분야에 대하여 이해할 수 있습니다.
- 파이썬을 실행하기 위한 기본 및 가상환경을 설정할 수 있습니다.
- 파이썬 에디터 프로그램 중 주피터노트북을 설치하고 실행할 수 있습니다.

■ 파이썬이란?

컴퓨터 코팅 프로그램 중 하나

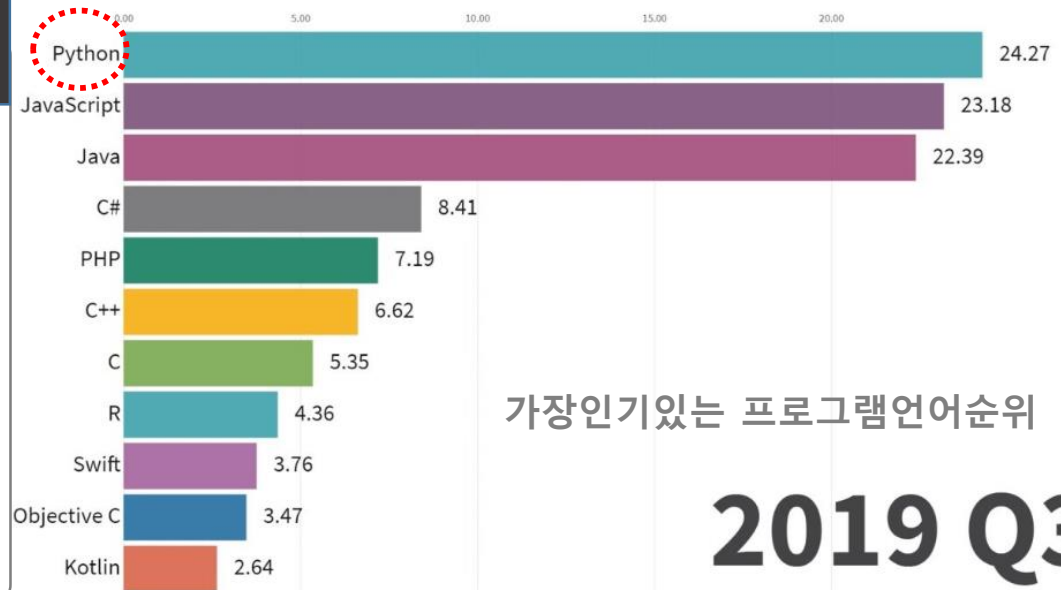




[Part1] 1. 파이썬 프로그램의 이해

파이썬 프로그램 특징

- ✓ 인공지능관련
다양한 모듈 제공
- ✓ 공개소프트웨어(무료)
- ✓ 다양한 무료 모듈
(전세계인이 개발공유)
- ✓ 타언어에 비해 접근성과 사용성이
용이하다.
(문법이 쉬움)



가장인기있는 프로그래밍언어순위

2019 Q3



1. 파이썬 프로그램의 이해

- 1990년 암스테르담의 귀도 반 로섬(Guido Van Rossum)이 개발한 인터프리터 언어
- 파이썬이라는 이름을 자신이 좋아하는 코미디 쇼인 "몬티 파이썬의 날아다니는 서커스 (Monty Python's Flying Circus)"에서 따왔다고 함
- 사전적 의미는 고대 신화에 나오는 파르나소스 산의 동굴에 살던 큰 뱀을 뜻하며, 아폴로 신이 델파이에서 파이썬을 퇴치했다는 이야기가 전해지고 있다.
- 대부분의 파이썬 책 표지와 아이콘이 뱀 모양으로 그려져 있는 이유가 여기에 있다.



파이썬 시리즈





1. 파이썬 프로그램의 이해

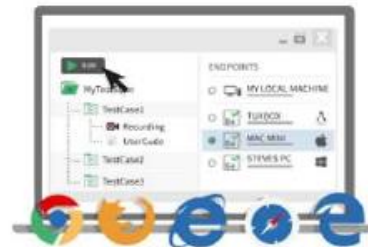
■ 파이썬 사용범위



웹 개발



머신 러닝



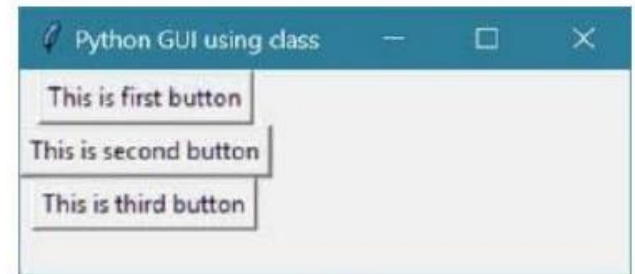
웹 테스트 자동화



데이터 분석



게임 개발



GUI 개발



1. 파이썬 프로그램의 이해

■ 데이터 분석

Data Analysis

Data Science

→ 비정형(unstructured data) 혹은 정형(structured) 데이터를 정제, 준비, 분석하는 활동



N드라이브



Google Drive

Pandas



NumPy

데이터 분류, 정제

데이터 시각화

matplotlib

데이터 분석



“기존 통계 분석”의 영역은 현황 파악과 진단만 하게 되는 것으로 샘플 데이터를 갖고 모수를 측정하는 방식으로 보통 진행됨.

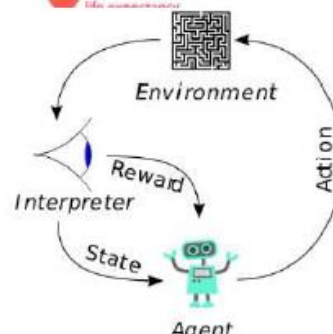
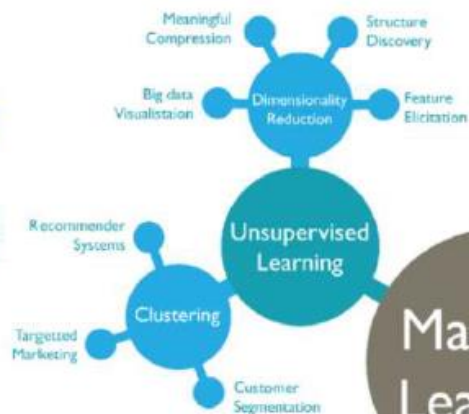
“빅데이터 분석”은 예측분석과 최적화분석 쪽에 중점을 두는 것



1. 파이썬 프로그램의 이해

- 딥러닝/머신러닝/인공지능

Machine Learning





1. 파이썬 프로그램의 이해

파이썬 장점

- 문법이 쉽고 순서가 영어 구문과 유사하여 빠르게 배울 수 있다.
- 풍부한 라이브러리로 개발 생산성이 매우 높다.
- 멀티패러다임 프로그래밍 언어이다.
(절차적 언어 지원, 객체지향, 함수형 프로그래밍 언어)
- 다양한 플랫폼에서 사용가능
- 쓰레딩 대신 단일 이벤트 루프를 사용해 소수 유닛에서 작업하는 비동기식 코드 작성에 유리, 메모리 자동 관리
- 다른 언어로 작성된 모듈을 연결하는 Glue Language로 이용
(CPython, Jython, IronPython, Pypy)
- 웹, 수치연산, 데이터베이스 프로그래밍 등 다양한 분야 개발 가능



1. 파이썬 프로그램의 이해

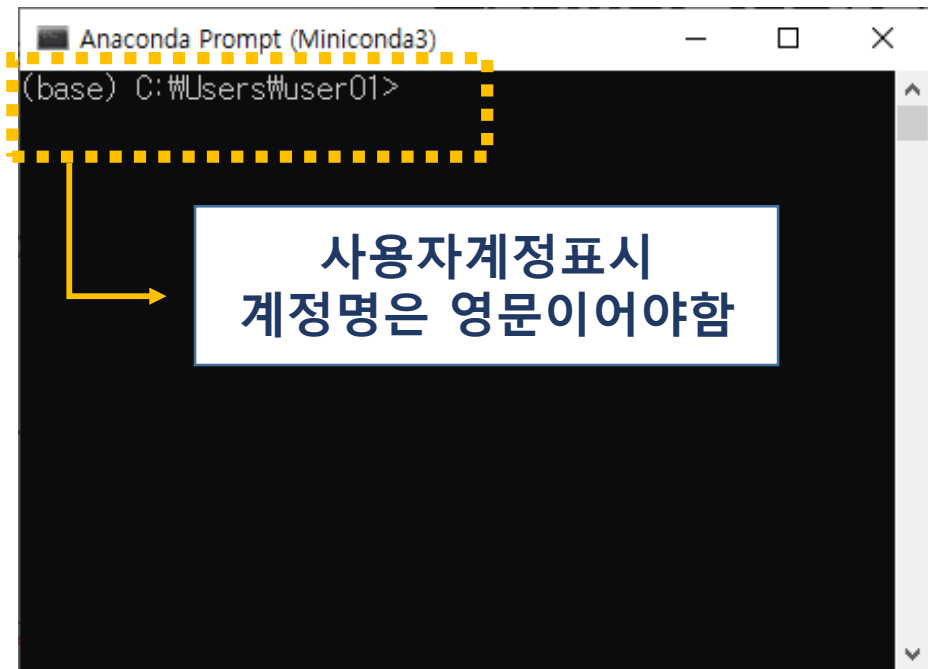
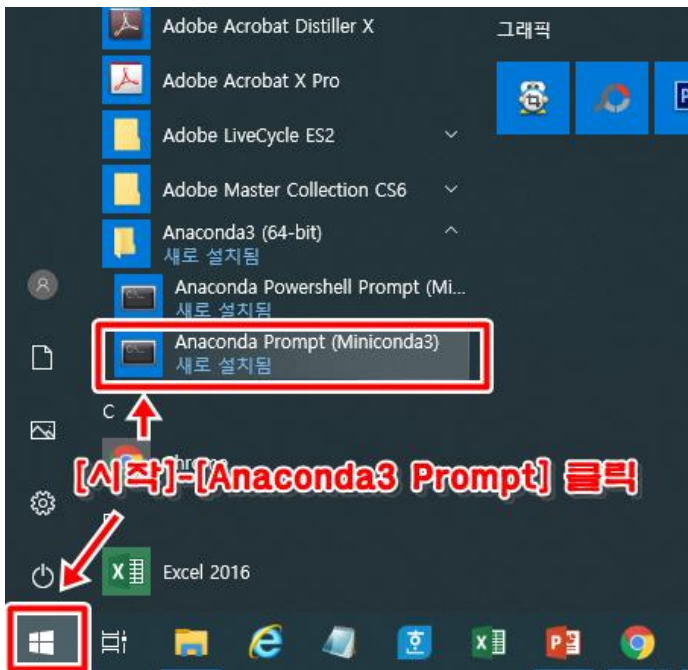
파이썬 단점

- 대단히 빠른 속도를 요구하거나 하드웨어를 직접 건드려야 하는 프로그램에는 부적합(C, C++ 이용)
- 2.x 버전과 3.x 버전이 호환되지 않는다.



2. 파이썬 가상환경 설정- 아나콘다 (미니콘다 실행)

- [시작] -> [anaconda prompt] 실행
- 프롬프트 화면 준비(cmd 화면임)





2. 파이썬 가상환경 설정- 명령프롬프트(cmd)에서 파이썬실행

- python으로 프로그램 실행후 >>> 표시가 나오면 명령을 입력함

```
Anaconda Prompt (Miniconda3)
(base) C:\Users\Wuser> python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 3+4
7
>>> mean(3+4)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'mean' is not defined
>>> exit()

(base) C:\Users\Wuser>
```



2. 파이썬 가상환경 설정- 가상환경 이름 정의

A

콘다프로그램 업데이트

```
(base) C:\Users\User> conda update conda  
Proceed ([y]/n)? y
```

B

가상환경 설정

가상환경: 작업별 공간임. 프로그램 다운로드(모듈) 및 저장등을 지정한 폴더에 모아놓는 공간임.

방1
이름: multi

방2
이름: Test

방3
이름: Pro1

방마다 물건을 채워야함.
(인터넷다운로드, 모듈)
그방에서만 사용할수 있음.
방은 폭파할수 있음.

```
(base) C:\Users\User> conda create -n multi python=3.9
```

사용자 방이름

C

생성된 가상환경 확인

```
(base) C:\Users\User> conda info --envs
```

D

생성된 가상환경 삭제

```
(base) C:\Users\User> conda remove --name multi --all
```



2. 파이썬 가상환경 설정- 가상환경 활성화

A

가상환경 활성화

(base) c:\User\User> **activate multi**

(nhi) C:\Users\User>



사용자 방이름



activate 방이름

B

가상환경 종료

(nhi) c:\User\User> **deactivate**

(base) c:\User\User>



기본 작업환경 세팅



참고 :
구글 자료 검색법

✓ 구글검색
'파이썬 가상환경'

✓ 구글에서 파일확장자로 검색
filetype:pdf 파이썬 가상환경



2. 파이썬 가상환경 설정- 가상환경에 모듈(파이썬에서 사용할 추가 패키지) 설치

A

모듈 설치

(nhi) C:\Users\user> **conda install jupyter**

activate 방이름

설치할 모듈이름(인터넷다운로드임)

```
Anaconda Prompt (Miniconda3) - conda install pandas

(base) C:\Users\WECOMS-003>activate nhi
(nhi) C:\Users\WECOMS-003>conda install pandas
Collecting package metadata (current_repodata.json): done
Solving environment: |

Proceed ([y]/n)? y
```

***참고:** 아나콘다 없이 파이썬만 설치되어 있을때
pip 최근자료로 업데이트 후 -> pip install 모듈명

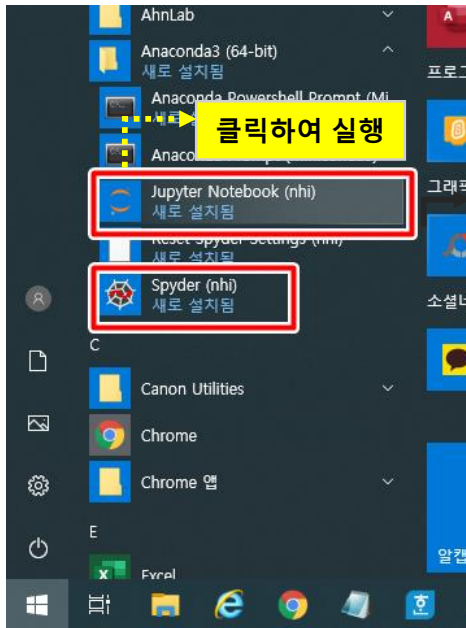
(nhi) C:\Users\user> **pip install pip --upgrade pip**

(nhi) C:\Users\user> **pip install jupyter**



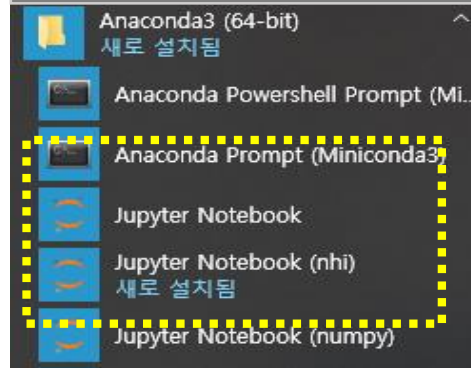
3. 파이썬 편집기 주피터 노트북 - 실행

- A. cmd 화면을 윈도우기반(GUI)으로 돌아옴
- B. 윈도우키-> 아나콘다-주피터 노트북 설치 확인
- C. 주피터노트북(nhi) 실행



참고: 구글검색 '파이썬 에디터'

참고: 가상환경 여러개 이고 각 가상환경안에서 주피터 다운로드 받았을때



참고: cmd 에서도 가능

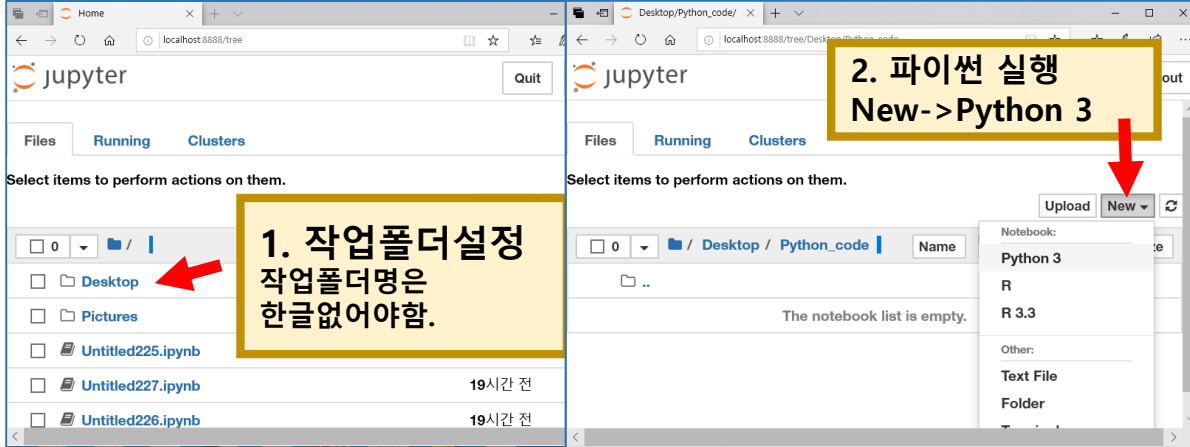
(nhi) C:\Users\User> **jupyter notebook**



activate nhi

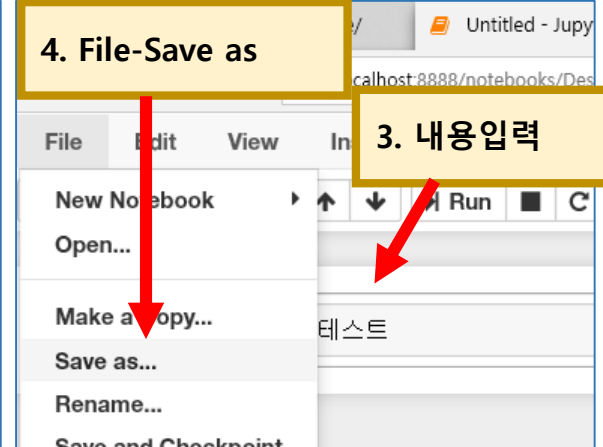


3. 파이썬 편집기 주피터 노트북 - 작업경로 설정, 주피터노트북에서 파이썬 연결



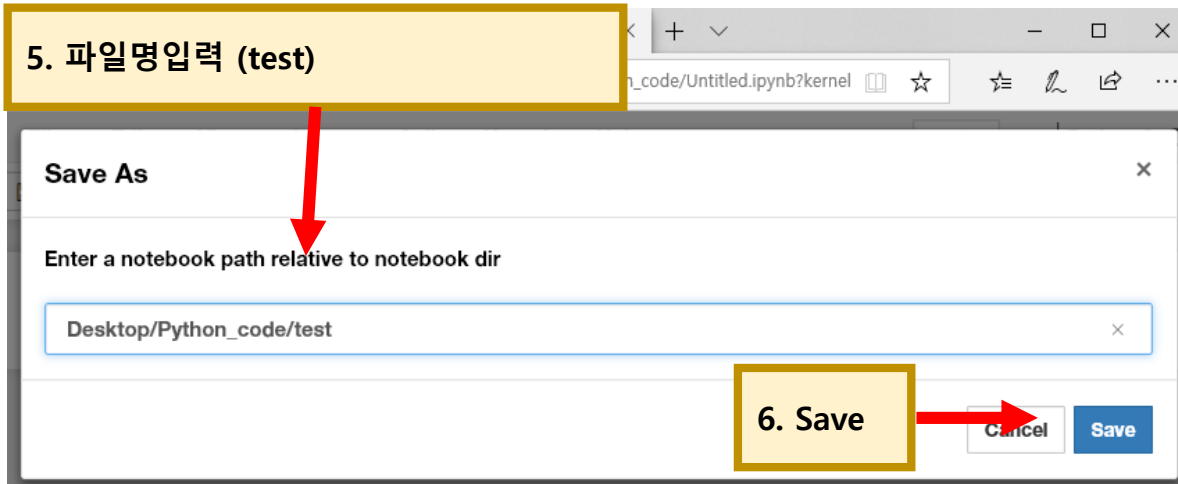
1. 작업폴더설정
작업폴더명은
한글없어야함.

2. 파이썬 실행
New->Python 3



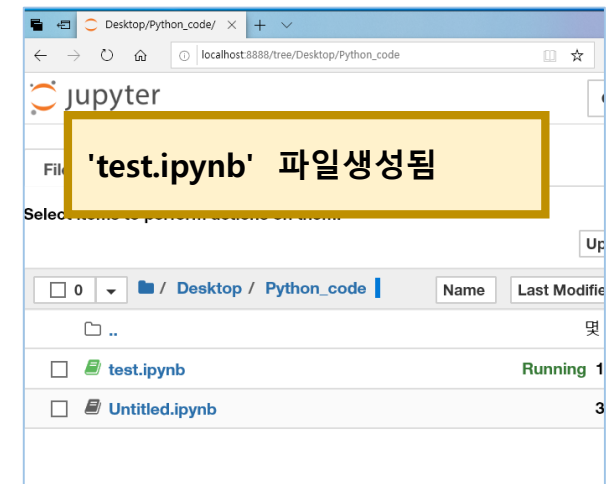
3. 내용입력

4. File-Save as



5. 파일명입력 (test)

6. Save



'test.ipynb' 파일생성됨



3. 파이썬 편집기 주피터 노트북 - 샘플코드 입력 및 실행

- 화면크기 조정: Ctrl키 누르고 마우스 가운데 휠 올리고 내리고

The screenshot shows the Jupyter Notebook interface with several annotations:

- 추가** (Add): Points to the '+' button in the toolbar.
- 삭제** (Delete): Points to the 'x' button in the toolbar.
- 이동** (Move): Points to the up and down arrow buttons in the toolbar.
- Ctrl+Enter임** (Ctrl+Enter): Points to the 'Run' button in the toolbar.

Below the toolbar, two code cells are shown:

```
In [7]: 1 print("Hellow World")
Hellow World
```

```
In [ ]: 1 a=3
2 b=4
3 print(a+b)
```

Annotations for the code cells:

- 1. 입력후 Shift+Enter** (코드 실행후아래에 명령프롬프트추가됨): Points to the end of the first code cell.
- 2. 자동추가됨** (3. 내용입력후 Ctrl+Enter): Points to the end of the second code cell.

※ Alt+Enter

(코드 실행후 아래에 명령 프롬프트 강제 추가됨)



[Part2] 4. 파이썬 실습 - 기본문법

A

세미콜론

- 많은 프로그래밍 언어들은 구문이 끝날 때 ;(세미콜론)을 붙여야 합니다.
- 하지만 파이썬은 세미콜론을 붙이지 않습니다.
- 보통 한 줄에 여러 구문을 사용할 때 세미콜론으로 구분해줍니다.

```
In [9]: print('Hello'); print('1234')
```

```
Hello  
1234
```

B

주석문

- 파이썬에서 사람만 알아볼 수 있도록 작성하는 부분을 주석(comment)라고 합니다.
- 즉, 주석은 파이썬 인터프리터가 처리하지 않으므로 프로그램의 실행에는 영향을 주지 않습니다.

```
In [7]: # Hello, world! 출력  
print('Hello, world!')
```

```
Hello, world!
```

```
In [8]: a = 1 + 2 # 더하기  
print('Hello, world!') #printf('1234567890')
```

```
Hello, world!
```



[Part2] 4. 파이썬 실습 - 기본문법

C

들여쓰기

- 들여쓰기는 코드를 읽기 쉽도록 일정한 간격을 띄워서 작성하는 방법입니다.
- 특히 파이썬은 들여쓰기 자체가 문법입니다.

```
In [ ]: print('Hello')  
        print('1234')    # 들여쓰기 문법 예러
```

```
In [ ]: if a == 10:  
        print('10입니다.')    # 들여쓰기 문법 예러
```

```
In [ ]: if a == 10:  
        print('10입니다.')
```



[Part2] 4. 파이썬 실습 - 기본문법

D

코드 블록

- 코드 블록은 특정한 동작을 위해서 코드가 모여 있는 상태를 뜻하며 파이썬은 들여쓰기를 기준으로 코드 블록을 구성합니다.
- 단, 같은 블록은 들여쓰기 칸 수가 같아야 하고, 공백과 탭 문자를 섞어 쓰면 안 됩니다.

```
In [ ]: if a == 10:
        print('10')
        print('입니다.')
```

```
if a == 10:
    print('10')
    print('입니다.')
```

공백 2칸
공백 4칸



```
if a == 10:
    print('10')
    print('입니다.')
```

공백 4칸
탭 1칸



```
if a == 10:
    print('10')
    print('입니다.')
```

공백 4칸
공백 4칸





[Part2] 4. 파이썬 실습 - 변수

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.



예) 비주얼베이직언어에서는

Dim Age as Byte

==> Age변수는 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



[Part1] 3. 파이썬 편집기 주피터 노트북 - 주피터노트북 간단실습

파일명

Part1_Python과 Jupyter 기본.html

[목표] 파이썬 기본 및 주피터 노트북 이해

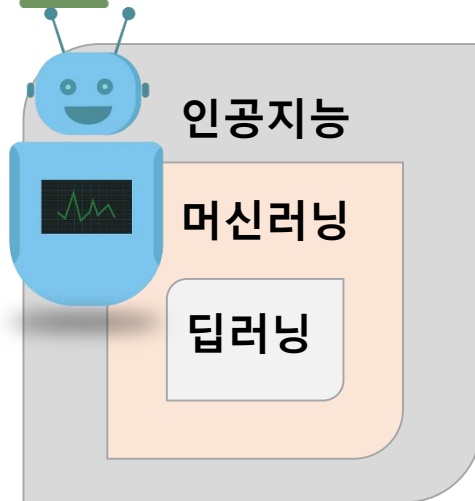
Part2. 파이썬 모듈과 데이터 형 이해

학습효과

- 프로그램의 Raw 파일인 데이터의 중요성에 대해 이해할 수 있습니다.
- 파이썬 변수와 Numpy 모듈을 실습하여 모듈과 모듈에 속한 함수 사용법에 대해 이해할 수 있습니다.



참고: 인공지능가능한 파이썬 프로그램으로 작업가능한 영역



1 머신러닝 기존의 데이터 마이닝과 다른점임(통계학적 코딩없음)

데이터를 분석, 스스로 학습하여 입력하지 않은 정보도 판단, 결정함

- 머신러닝이 적용될 적절한 분야 / 지식모델을 교육할 양질의 데이터 제공
- 막대한 양의 데이터에 알고리즘을 적용해 특정 패턴이 드러나면, 이런 패턴이 머신러닝 기반 애플리케이션을 훈련시키는 것임(시스템을 가르치는 충분한 데이터 필요)
- ✓ 최소 10만건 이상 데이터 필요

2 딥러닝

컴퓨터가 사람처럼 생각하고 배울 수 있도록 스스로 학습함

3 인공지능

인간의 학습능력을 프로그래밍하여 컴퓨터가 인간의 지능적 행동을 모방하게 함



[Part2] 1. 파이썬 데이터 분석 예

■ 데이터 마이닝

대용량 데이터로부터 패턴인식, 인공지능, 고급 통계분석 기법 등을 이용하여 숨겨져 있는 데이터간의 상호 관련성 및 유용한 정보를 추출하는 기술

■ 텍스트 마이닝 (워드클라우드)

텍스트 기반의 데이터로부터 새로운 정보를 발견할 수 있도록 정보 검색, 추출, 체계화, 분석을 모두 포함하는 Text-processing 기술 및 처리 과정. 텍스트 내에 존재하는 단어의 등장횟수 등을 평가하여 문서간의 유사성을 수치화 하는 텍스트 데이터를 분석하여 유사 문서 분류 및 문서 내 정보 추출과 같은 결과 산출



■ 오피니언 마이닝 (평판분석:Sentiment Analysis)

소셜미디어등의정형/비정형텍스트의긍정(Positive), 부정(Negative), 중립(Neutral)의선호도를판별하는기술로 특정서비스및상품에대한시장규모 예측,소비자의반응,입소문분석 (ViralAnalysis)등에활용되고 있다.

정확한오피니언마이닝을위해서는전문가에의한선호도를나타내는 표현/단어자원의축적이 필요하다.

40자 영화평

- 트랜스포머 최고~! 트랜스포머 머리와 마음이 즐거웠다. 아주 좋았어
- 트랜스포머 복잡했지만 괜찮았어요
- 트랜스포머 재미있었어요
- 트랜스포머 한마디로 대박
- 트랜스포머 스토리도 좋고 눈을 즐겁게 해주었다
- 트랜스포머 시험끝나고 봤는데 정말 죽여요 짱이에요~~~
- 트랜스포머 끝이허무하다.
- 트랜스포머 최고다.
- 트랜스포머 볼거리만으로 내용을 커버하기에는 약했다.



[Part2] 2. 데이터의 이해

예:간식 추천 시스템

정동이의 다음 일주일 간식을 보고 정동이에게 일요일레 추천해주고자 하는 간식은?

✓ 정동이의 일주일 간식

월	화	수	목	금	토	일
사과	초콜릿	바나나	토마토	토마토	수박	??

(1) 피자 (2) 브로콜리 (3) 맥주





[Part2] 2. 데이터의 이해

데이터를 대표하는 이름(필드별, 변수명)

데이터

이름	나이	성별	좋아하는음료	지병	몸무게	추천간식
이름1	20	1	콜라	고혈압	95	
이름2	30	0	물		80	
이름3	49세	0	물	고혈압	45	

- ▶ 숫자변수: 연속형 데이터, 이산데이터 (성별, 몸무게)
- ▶ 문자변수: 명목형 데이터 (값을 묶어서 개수를 파악하는 의미가 있을때)
이름, 나이, 성별, 좋아하는 음료, 지병

모든 프로그램에서는 변수명(필드명)을 갖고 모든 작업을 진행하며 데이터중 문자, 숫자가 섞여있을때는 문자변수로 취급함. 빈칸은 프로그램에서 NA로 나오며 계산될때 Error있을수 있을수 있으므로 프로그램에 필수자료라면 반드시 빈칸 없어야함.

결과에 영향을 미치는 값(필드, 변수): 설명변수, 독립변수, 특징값(Feature)

결과값(필드, 변수) 예: 추천간식 종속변수, 라벨

x값들에 따른 y값의 변화(회귀) 또는 분류를 자동계산하는것이 인공지능임(AI)



[Part2] 2. 데이터의 이해

예: 쇼핑몰 고객분석

- 쇼핑몰 사이트의 사용자 클릭 스트림을 통해 실시간 개인화
- 대용량 이메일 서버의 스팸 탐지 및 필터링
- 위치 정보 기반 광고 서비스
- 사용자 및 시스템 이벤트를 이용한 실시간 보안 감시
- 시스템 정보 수집을 통한 장비 고장 예측
- 카드 결제시 각종 상황에 맞는 이벤트 제시
- 실시간 차량 추적 및 위치 정보 수집을 이용한 도로 교통 상황 파악
- 사용자의 액션 수집을 이용한 이상 행위 탐지

Tips!

【 데이터 품질 오류의 범위 】

데이터 품질 오류란, 데이터의 부정확 등으로 인해 발생하는 오류를 의미하며 보다 넓은 의미로는 데이터 활용 시 발생하는 접속 지연, 검색 불편 등 데이터를 활용하는 사용자 측면의 불편 및 불만사항 까지도 포함할 수 있음



[Part2] 3. 데이터 종류

처리해야하는 데이터 종류에 따라서 프로그램별 효과가 다르며 파이썬에서도 설치 모듈이 다름

예: 마이크로칩안의 숫자값처리는 c언어 /네트워크 시스템,웹관리는 자바
파이썬은 모든분야 가능하며 숫자전문은 numpy, DB형은 pandas, 차트는 matplotlib

정형데이터

정형 데이터는 즉시 통계적 분석에 사용될 수 있을만한 형태로 정리되고 가공된 데이터

예) 엑셀 파일, 즉 각 지역별 인구통계 데이터, 의학 실험 결과에 대한 데이터 등이 행과 열에 맞게 정리된 자료

비정형데이터

정형화 되지 않은 raw data 들을 의미

예) 동영상, 사진, SNS에 적혀있는 텍스트 등

반정형 데이터

정형과 비정형 데이터 사이에 있는 자료

비정형 데이터만큼 정리가 가공이 안되어있지는 않지만, 일반적인 통계분석에 바로 사용할 수 있을만큼 정제되어있지는 않은 데이터

예) 들어 신문 기사나 서적의 텍스트 등



[Part2] 3. 데이터 종류

처리해야하는 데이터 종류에 따라서 프로그램별 효과가 다르며 파이썬에서도 설치 모듈이 다름

예: 마이크로칩안의 숫자값처리는 c언어 /네트워크 시스템,웹관리는 자바
파이썬은 모든분야 가능하며 숫자전문은 numpy, DB형은 pandas, 차트는 matplotlib

양적 데이터

양적 데이터는 길이, 무게와 같이 양적인 수치 등을 제어 측정할 수 있는 데이터로 하나 하나 셀 수 있는 이산적 데이터
예) 자녀수, 불량갯수 등

어떤 구간 안에 모든 값을 가지는 연속형 데이터(길이, 무게)로 나눌 수 있다.

양적 데이터는 질적 데이터보다 정리하는 방법이 다양하고, 이산적 데이터 보다는 연속적 데이터의 정리 절차가 더 복잡

질적 데이터

관찰 대상을 몇 개의 범주로 나누어 사람이나 개체들이 어떤 범주에 속하는 가를 나눌 수 있는 데이터
예)성별, 지지정당, 교육수준 등

질적 데이터들의 각 계급 빈도가 차지하는 비율을 상대적 빈도(relative frequency) 라 하며
상대적 빈도는 한 계급에 속하는 빈도가 전체 관찰 수에 비한 비중

101호 전문직	201호 사업	301호 공무원	401호 회사원
102호 회사원	202호 공무원	302호 회사원	402호 회사원
103호 회사원	203호 기타	303호 전문직	403호 사업
104호 사업	204호 회사원	304호 기타	404호 공무원
105호 전문직	205호 공무원	305호 사업	405호 회사원
106호 기타	206호 사업	306호 기타	406호 기타

직업(X)	빈도수(f)
전문직	3
회사원	7
사업	5
기타	5
공무원	4
합계	24

직업	상대적 빈도	백분율
전문직	0.125 (3/24)	12.5%
회사원	0.292 (7/24)	29.2%
사업	0.208 (5/24)	20.8%
기타	0.208 (5/24)	20.8%
공무원	0.167 (4/24)	16.7%
합계	1	100%



[Part2] 4. 파이썬 실습 - 모듈이해

파이썬(Python) 모듈(Module)이란?

자주 사용되는 코드나 유용한 코드를 논리적으로 묶어서 관리하고 사용할 수 있도록 하는 것입니다.

보통 하나의 파이썬 .py 파일이 하나의 모듈이 됩니다.

모듈 안에는 함수, 클래스, 혹은 변수들이 정의될 수 있으며, 실행 코드를 포함할 수도 있습니다.

```
In [59]: import numpy as np
print("a")    ### 1번줄이 잘 끝났는지
a

In [61]: ## 주피터메뉴의 Cell -> Cell Type ->

In [54]: np.mean(Age) #평균
Out[54]: 21

In [55]: np.sum(Age) #합
Out[55]: 126

In [56]: np.std(Age) #표준편차
Out[56]: 11.146748404803978

In [57]: np.min(Age) #최소
Out[57]: 21
```

- ✓ 수치해석:- NumPy, SciPy, SymPy
- ✓ 데이터탐색: Pandas, MDP, Orange
- ✓ 시계열/회귀분석: Statsmodels, Filterpy, Hmmlern
- ✓ 분류인식: Scikit-Learn
- ✓ 고속계산: Teano, Tensorflow
- ✓ 베이지안 모형: PyMC3,
- ✓ 딥러닝: Keras, Lasagne, Blocks
- ✓ 영상신호처리: Pillow, Scikit-image
- ✓ 문서처리: NLTK, Gensim
- ✓ 음향신호처리: PyAudio-Analysis, LibRosa
- ✓ 확률적 그래프 모형: LibPGM, Pgmpy

```
In [5]: from IPython.display import display

In [6]: import mglearn

-----
ModuleNotFoundError
<ipython-input-6-26a0454c3f6e> in <module>()
----> 1 import mglearn

ModuleNotFoundError: No module named 'mglearn'
```

에러나올시
라이브러리 설치



[Part2] 4. 파이썬 실습 - 자료형

참고: 점프 투 파이썬

<https://wikidocs.net> => 점프 투 파이썬 참조

02장 파이썬 프로그래밍의 기초, 자료형

02-1 숫자형

02-2 문자열 자료형

02-3 리스트 자료형

02-4 튜플 자료형

02-5 딕셔너리 자료형

02-6 집합 자료형

02-7 불 자료형

02-8 자료형의 값을 저장하는 공간, 변수

1. 수치자료형

- int(정수), float(실수), complex(복소수)

2. 불자료형

- bool(불)-(True or False)

3. 군집자료형

- str(문자열), list(리스트), tuple(튜플), set(집합, 셋), dict(사전, 딕셔너리)



[Part2] 4. 파이썬 실습 – 자료형(숫자형)

● 숫자형 : 숫자 형태로 이루어진 자료형

항목	사용 예
정수	123, -345, 0
실수	123.45, -1234.5, 3.4e10
8진수	0o34, 0o25
16진수	0x2A, 0xFF

Operator	Description	Example
+	더하기	$a + b = 30$
-	빼기	$a - b = -10$
*	곱하기	$a * b = 200$
/	나누기	$b / a = 2.0$
%	나머지	$b \% a = 0$
**	제곱	$a ** c = 1000$
//	몫	$a // c = 3$



[Part2] 4. 파이썬 실습 – 자료형(문자형)

- 문자형 : 문자열은 문자들의 모임, 문자를 나열한 것을 문자열이라고 함
- 인용부호로 단일(') 또는 이중(") 따옴표를 사용한다.
- [] [:] 을 사용하여 문자열의 하위 집합을 사용할 수 있다.
- index는 0 부터 전체 길이에서 -1 한 것 까지 이다.
- '+' 기호는 문자를 더하는 것이고 '*' 기호는 문자를 반복하는 것 이다.
- 인덱싱(Indexing)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	
p	y	t	h	o	n		i	s		v	e	r	y		p	...

[-16]	[-15]	[-14]	[-13]	[-12]	[-11]	[-10]	[-9]	[-8]	[-7]	[-6]	[-5]	[-4]	[-3]	[-2]	[-1]	
...	i	s		v	e	r	y		p	o	w	e	r	f	u	l



[Part2] 4. 파이썬 실습 – 자료형(문자형)

```
In [5]: str='Python is Very Powerful'
```

```
print(str)
print(str[0])
print(str[7:9])
print(str[10:])
print(str[-8:])
print(str[:-8])
print(str*2)
print(str+' Test')
```

Python is Very Powerful

P

is

Very Powerful

Powerful

Python is Very

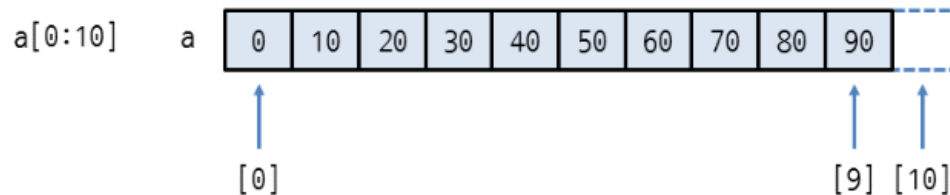
Python is Very PowerfulPython is Very Powerful

Python is Very Powerful Test



[Part2] 4. 파이썬 실습 – 자료형(List형)

- 리스트는 복합 데이터 유형에 가장 유용하다.
- 리스트 내의 항목은 쉼표(,)로 구분하고 ([]) 사각 괄호 안에 포함되어 있다.
 - 리스트명 = [요소1, 요소2, 요소3, ...]
- C의 배열과 비슷하며 리스트에 속하는 모든 항목이 서로 다른 데이터 유형이 될 수 있다.
- [] [:] 을 사용하여 리스트를 액세스 할 수 있다.
- 리스트의 index는 0 부터 전체 항목수에서 -1 한 것 까지 이다.
- '+' 기호는 연결 연산자, '*' 기호는 반복 연산자 이다.
- 인덱싱(Indexing)





[Part2] 4. 파이썬 실습 – 자료형(List형)

```
In [2]: lst = [1, 2, 3, 'a', 'abc', [44, 55] ]
```

```
print(lst[2])  
print(lst[0:3])  
print(lst[2:5])
```

3

[1, 2, 3]

[3, 'a', 'abc']

```
In [1]: lst=[1,2,3, 'a', 'abc', [44,55]]  
lst.append('kim')  
lst
```

```
Out[1]: [1, 2, 3, 'a', 'abc', [44, 55], 'kim']
```

```
In [3]: list( range(0, 10) )
```

```
Out[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```




[Part2] 4. 파이썬 실습 – 자료형(튜플형)

- 튜플은 리스트와 유사한 또 다른 시퀀스 데이터 형식이다.
- 튜플 내의 항목은 쉼표(,)로 구분하고 () 괄호 안에 포함되어 있다.
- c의 배열과 비슷하며 튜플에 속하는 모든 항목이 서로 다른 데이터 유형이 될 수 있다.
- [] [:] 을 사용하여 리스트를 액세스 할 수 있다.
- 튜플의 index는 0 부터 전체 항목수에서 -1 한 것 까지 이다.
- '+' 기호는 연결 연산자, '*' 기호는 반복 연산자 이다.
- 리스트와의 차이점은 **읽기 전용**이다.



[Part2] 4. 파이썬 실습 – 자료형(튜플형)

```
In [6]: tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
        tinytuple = (123, 'john')

        print (tuple)
        print (tuple[0])
        print (tuple[1:3])
        print (tuple[2:])
        print (tinytuple * 2)
        print (tuple + tinytuple)

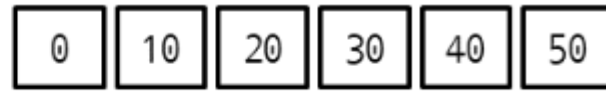
('abcd', 786, 2.23, 'john', 70.2)
abcd
(786, 2.23)
(2.23, 'john', 70.2)
(123, 'john', 123, 'john')
('abcd', 786, 2.23, 'john', 70.2, 123, 'john')
```



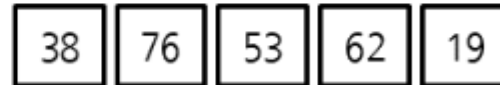
[Part2] 4. 파이썬 실습 – 자료형(스퀀시형)

- 파이썬에서는 리스트, 튜플, range, 문자열처럼 값이 연속적으로 이어진 자료형을 시퀀스 자료형(sequence types)라고 부릅니다.

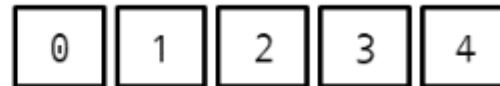
리스트: [0, 10, 20, 30, 40, 50]



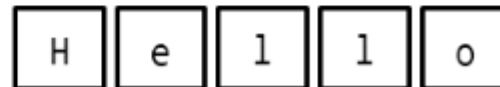
튜플: (38, 76, 53, 62, 19)



range: range(5)



문자열: 'Hello'





[Part2] 4. 파이썬 실습 – 자료형(스퀀시형)

● 특정 값이 있는지 확인하기

```
In [16]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]  
30 in a
```

```
Out[16]: True
```

```
In [17]: 'P' in 'Hello, Python'
```

```
Out[17]: True
```

```
In [18]: 30 not in a
```

```
Out[18]: False
```



[Part2] 4. 파이썬 실습 – 자료형(딕셔너리형)

- 파이썬에서는 연관된 값을 묶어서 저장하는 용도로 딕셔너리라는 자료형을 제공합니다.
- 즉, 사전(dictionary)에서 단어를 찾듯이 값을 가져올 수 있다고 하여 딕셔너리라고 부릅니다.

```
In [20]: lux = {'health': 490, 'health': 800, 'mana': 334, 'melee': 550, 'armor': 18.72}
```

```
In [22]: lux['health']    # 키가 중복되면 가장 뒤에 있는 값만 사용함
```

```
Out[22]: 800
```

```
In [23]: lux
```

```
Out[23]: {'health': 800, 'mana': 334, 'melee': 550, 'armor': 18.72}
```

● dict로 딕셔너리 만들기

- 딕셔너리 = dict(키1=값1, 키2=값2)
- 딕셔너리 = dict(zip([키1, 키2], [값1, 값2]))
- 딕셔너리 = dict([(키1, 값1), (키2, 값2)])
- 딕셔너리 = dict({키1: 값1, 키2: 값2})

[Part2] 4. 파이썬 실습 – 자료형(딕셔너리형)

```
In [24]: # 키=값 형식으로 딕셔너리를 만들  
lux1 = dict(health=490, mana=334, melee=550, armor=18.72)
```

```
In [25]: # zip 함수로 키 리스트와 값 리스트를 묶음  
lux2 = dict(zip(['health', 'mana', 'melee', 'armor'], [490, 334, 550, 18.72]))
```

```
In [26]: # (키, 값) 형식의 튜플로 딕셔너리를 만들  
lux3 = dict([('health', 490), ('mana', 334), ('melee', 550), ('armor', 18.72)])
```

```
In [27]: # dict 안에서 중괄호로 딕셔너리를 만들  
lux4 = dict({'health': 490, 'mana': 334, 'melee': 550, 'armor': 18.72})
```



[Part2] 4. 파이썬 실습 - 변수, 리스트, numpy 모듈

파일명

Part2_파이썬 변수 및 자료형.html

[목표] 프로그램의 변수 및 모듈 이해하기
구글링으로 필요한 자료 찾기

Part3. 파이썬 파일관리

학습효과

- 외부 csv자료를 판다스 모듈을 이용하여 파이썬으로 불러올 수 있습니다.
- 딕셔너리 구조의 판다스 자료의 인덱싱이 가능하며 정렬하여 자료를 나열할 수 있습니다.
- 시각화를 통해 자료의 분류를 확인할 수 있습니다.



[Part3] 1. 판다스 설치, 작업할 파일(csv등) 을 저장할 폴더 생성

	A	B	C	D	E	F
1	bun	name	kor	eng	bigo	
2	1	aa	30	35	No	
3	2	bb	50	81		
4	3	cc	24	45	No	
5	4	dd	89	37		
6	5	ee	36	48	No	
7	6	ff	97	67		
8	7	gg	78	84	Yes	
9	8	hh	86	88	Yes	
10						
11						

1행 제목이 판다스에서는
변수명으로 들어옴. (열 5개)

데이터 (행 8개)

값 없는 데이터: Null(NA) 데이터

[Part3] 1. 판다스 실습 - 컴퓨터에 있는 csv자료 관리

```
In [28]: import pandas as pd
```

→ conda install pandas

```
In [29]: df=pd.read_csv('c:/py_data/sample.csv')
```

→ c:/py_data 폴더가 있어야함.
그 폴더에 sample.csv 가 있어야함

```
In [30]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 8 entries, 0 to 7
```

→ 8개의 행, 엑셀에서는 2,3,4,5,6,7,8,9 행

```
Data columns (total 5 columns):
```

→ 5개의 열, 엑셀에서는 A,B,C,D,E

#	Column	Non-Null Count	Dtype
0	bun	8 non-null	int64
1	name	8 non-null	object
2	kor	8 non-null	int64
3	eng	8 non-null	int64
4	bigo	3 non-null	object

→ 8 non-null: 8개의 자료중 Null이 0개
3 non-null: 8개의 자료중 Null이 5개

```
dtypes: int64(3), object(2)
```

```
memory usage: 448.0+ bytes
```



[Part3] 1. 판다스 실습 - 컴퓨터에 있는 csv자료 관리

```
In [31]: 1 df.columns[0]
```

```
Out[31]: 'bun'
```

```
In [32]: 1 df.columns[0:3]
```

```
Out[32]: Index(['bun', 'name', 'kor'], dtype='object')
```

```
In [41]: 1 # 위로 2개자료
          2 df.head(2)
```

```
Out[41]:
```

	bun	name	kor	eng	bigo
0	1	a	1	2	N
1	2	b	2	3	Y

```
In [40]: 1 ##kor을 기준으로 작은값 먼저 나오기
          2 df.sort_values(by='kor')
```

```
Out[40]:
```

	bun	name	kor	eng	bigo
0	1	a	1	2	N
1	2	b	2	3	Y
2	3	c	2	3	Y
3	4	d	3	2	NaN
4	5	e	4	20	N
5	6	f	8	30	NaN

```
1 #kor값 기준으로 큰값 먼저 나오기
2 df.sort_values(by='kor', ascending=False)
```

	bun	name	kor	eng	bigo
5	6	f	8	30	NaN
4	5	e	4	20	N
3	4	d	3	2	NaN
1	2	b	2	3	Y
2	3	c	2	3	Y
0	1	a	1	2	N



[Part3] 1. 판다스 실습 - 컴퓨터에 있는 csv자료 관리

```
1 #df 자료 정보 (성격 및 기초통계)
2 df.describe() #개수, 평균, 표준편차, 최소, 사분위수, 최대값
```

	bun	kor	eng
count	6.000000	6.000000	6.000000
mean	3.500000	<u>3.333333</u>	<u>10.000000</u>
std	<u>1.870829</u>	<u>2.503331</u>	12.049896
min	<u>1.000000</u>	<u>1.000000</u>	<u>2.000000</u>
25%	<u>2.250000</u>	<u>2.000000</u>	<u>2.250000</u>
50%	3.500000	<u>2.500000</u>	3.000000
75%	4.750000	3.750000	15.750000
max	6.000000	8.000000	30.000000

```
In [52]: 1 print(df.count()) #r개수
```

```
bun    6
name    6
kor     6
eng     6
bigo    4
dtype: int64
```

```
In [53]: 1 print(df.mean()) #평균
```

```
bun    3.500000
kor    3.333333
eng   10.000000
dtype: float64
```

```
In [54]: 1 print(df.median()) #중앙값
```

```
bun    3.5
kor    2.5
eng    3.0
dtype: float64
```

```
In [59]: 1 df.var() #분산값이 클수록 너무크거나 작은값이 있음을 의미함
```

```
Out [59]: bun    3.500000
kor    6.266667
eng   145.200000
dtype: float64
```

```
In [55]: 1 df.std() #표준편차값이 클수록 너무크거나 작은값이 있음을 의미함
```

```
Out [55]: bun    1.870829
kor    2.503331
eng   12.049896
dtype: float64
```



[Part3] 1. 판다스 실습 - 컴퓨터에 있는 csv자료 관리

```
In [50]: 1 #상관계수 -1이 가까울수록 강한 음의 상관관계
          2 #상관계수 1이 가까울수록 강한 양의 상관관계
          3 df.corr()
```

Out[50]:

	bun	kor	eng
bun	<u>1.000000</u>	0.896803	0.842823
kor	0.896803	<u>1.000000</u>	0.928232
eng	0.842823	0.928232	<u>1.000000</u>

[미션]

c:/python_data 폴더의 sample값의 숫자 값을 바꾸어 저장하고 파이썬에서 평균, 분산, 상관계수 등을 확인하시오.

[미션]

sample.csv [E5]셀에 '재시험'을 입력하고 파이썬에서 자료를 불러오시오.



[Part3] 2. 판다스 실습 및 미션

모듈

```
import pandas as pd  
import matplotlib.pyplot as plt
```

파일명

Part3_pandas_서울시cctv자료분석.html

[목표]

- 엑셀자료 파이썬으로 가져와서 자료 인덱싱
- pandas를 이용한 자료관리
- 간단한 시각화

Part4. 판다스를 이용한 다중 파일 관리 및 분석(실무)

학습효과

- 판다스를 이용한 두개 이상의 파일을 읽어 들여 처리 하는 방법에 대하여 학습
- 형식이 일치하는 데이터에 대한 병합
- Index 값이 일치하는 데이터에 대한 병합
- 판다스를 이용한 데이터 검색 및 관리 기법

파일명

Part4_pandas_자료병합 및 서울시 데이터 분석.html

[목표]

- 두개의 엑셀 파일을 이용해 열단위 병합 실행
- 필드를 이용한 새로운 계산 필드 생성
- pandas 조건 검색 및 데이터 처리
- matplotlib을 이용한 데이터 분석

Part5. 반복문과 제어문

학습효과

- For문을 이용한 실행문 반복 실행 이해
- If문을 이용한 데이터 제어 방법 이해
- SKT 멤버십 월별 자료 병합 및 데이터 관리, 파일 저장 방법 학습

파일명

Part5_반복문과 분기문.html

[목표]

- 반복문(for, while), 분기문(if) 이해
- pandas를 이용한 다중파일 열 병합
- 대용량 데이터 처리(엑셀 처리 불가)
- pandas를 이용한 데이터 관리
- matplotlib을 이용한 데이터 시각화



[Part5] 1. 반복문(for)

➤ 구조1

for 변수 in 객체명:
반복할 실행문

```
In [ ]: Weekdays = [ 'Mon', 'Tue', 'Wed', 'Tur', 'Fir']  
        for day in Weekdays:  
            print(day)
```

➤ 구조2(Range 함수)

For 변수명 in Range(5):
반복할 실행문

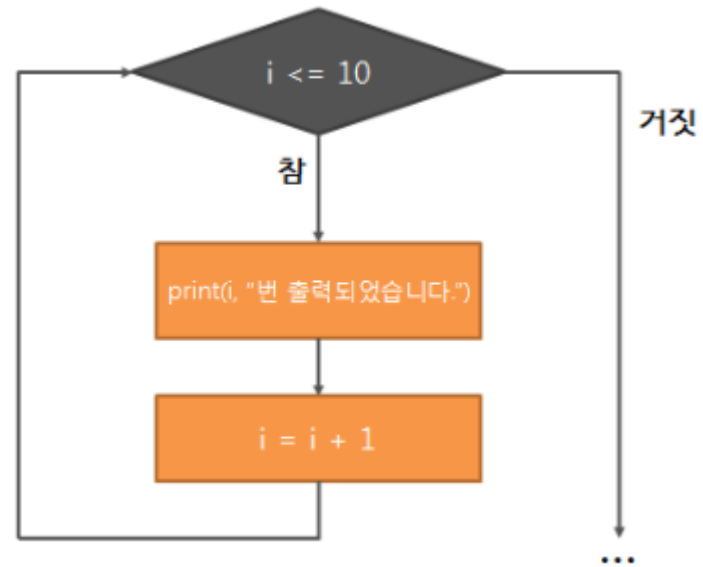
```
In [ ]: for i in range(5):  
        print( i )
```




[Part5] 2. 반복문(while)

➤ 구조

while 조건식:
반복 실행문



```
In [ ]: data=input('값 입력:')

while val != 'End':
    print(data)
    data=input('값 입력:')

print('종료되었습니다.')
```



[Part5] 2. 반복문(While)

➤ Break문 : 반복문 영역을 벗어남

```
In [ ]: i = 1
        while i <= 10:
            if i == 7:
                break
            print(i, "번 출력되었습니다.")
            i += 1
```

➤ Continue문: 아래의 문장은 실행하지 않고 반복문의 시작 지점으로 이동

```
In [ ]: i = 0
        while i <= 10:
            i += 1
            if i % 2 == 1: continue
            print(i, "번 출력되었습니다.")
```



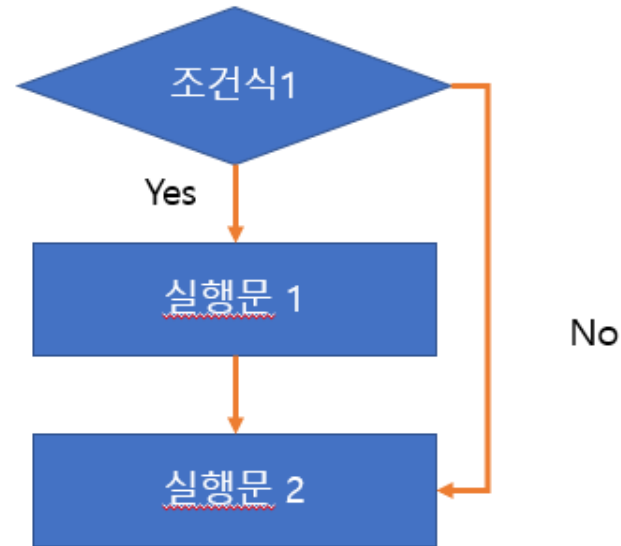
[Part5] 3. 조건문(if)

➤ 구조1

if 조건식:

실행문1

실행문2



```
In [1]: val=int(input('값 입력'))  
  
if val > 0:  
    print('0보다 큰 값입니다.')  
  
print('END')
```

```
값 입력20  
0보다 큰 값입니다.  
END
```



[Part5] 3. 조건문(if)

➤ 구조1

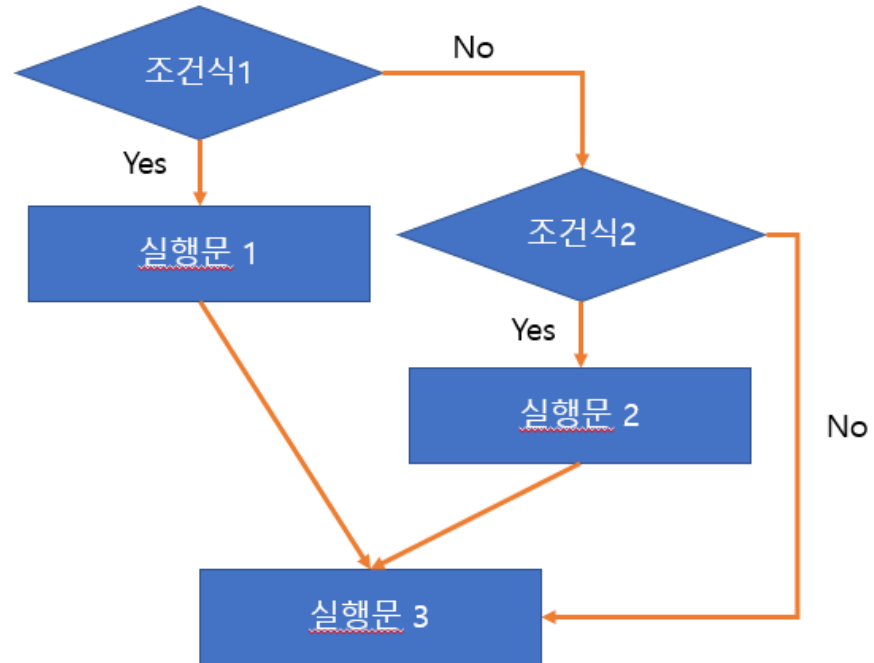
if 조건식1:

실행문1

elif 조건식2:

실행문2

실행문3



```
In [ ]: val=int(input('값 입력'))

if val > 0:
    print('0보다 큰 값입니다.')
elif val < 0:
    print('0보다 작은 값입니다.')

print('END')
```



[Part5] 3. 조건문(if)

➤ 구조1

if 조건식:

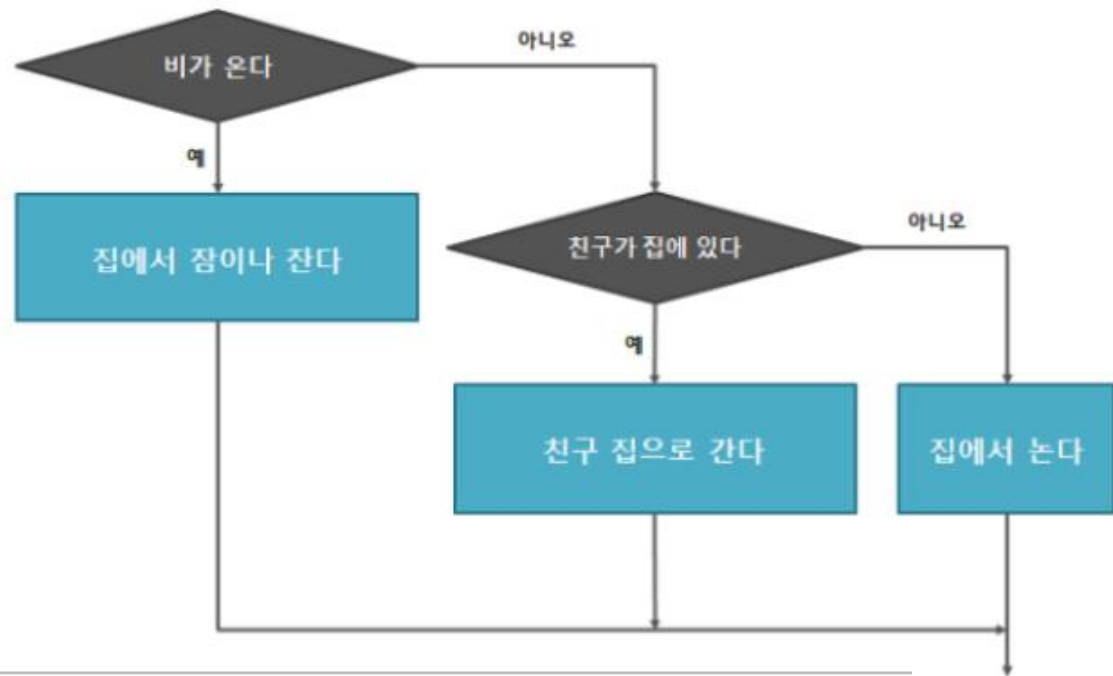
실행문1

elif 조건식:

실행문2

else:

실행문3



```
In [ ]: val=int(input('값 입력'))

if val > 0:
    print('0보다 큰 값입니다.')
elif val < 0:
    print('0보다 작은 값입니다.')
else:
    print('0과 같은 값입니다.')

print('END')
```

Part6. 종합실습

학습효과

- 질병관리본부의 코로나 확진자 현황(서울)을 이용하여 각각의 데이터 분석 실시
- 입원 병원별 확진자 통계와 병원 위치 지도에 표시
- 확진자 현황별 시각화

모듈

```
import pandas as pd  
import matplotlib.pyplot as plt
```

파일명

Part6_코로나 확진자 현황 분석_서울.html

[목표]

- 엑셀자료 파이썬으로 가져와서 자료 인덱싱
- 분석 데이터를 이용한 시각화
- 지도 차트를 이용한 시각화

Part7. 사용자 정의 함수

학습효과

- 사용자 정의 함수 생성 방법 및 활용 방법 이해
- def 구조 이해 및 제작
- def 함수를 이용한 미니 프로젝트 진행
- 공공 API를 이용한 데이터 분석 및 관리

파일명

Part7_사용자 정의 함수와 공공데이터.html

[목표]

- 사용자 정의 함수 이해
- 사용자 정의 함수(def) 구조 및 생성
- 공공 API 이해 및 신청
- 공공API를 이용한 실시간 데이터 분석(미니 프로젝트)



[Part7] 1. 사용자 정의 함수

- 프로그래밍을 하다 보면 똑같은 내용을 반복해서 작성하고 있는 자신을 발견할 때가 종종 있다. 이때가 바로 함수가 필요한 때이다.
- 즉 반복되는 부분이 있을 경우 "반복적으로 사용되는 가치 있는 부분"을 한 문치로 묶어서 "어떤 입력 값을 주었을 때 어떤 결과 값을 돌려준다"라는 식의 함수로 작성하는 것이 현명하다.

➤ **def** 함수명():

<수행할 문장1>

<수행할 문장2>

.....

```
In [1]: def order():
        print('주문하실 음료를 알려주세요')
        drink = input()
        print(drink, '주문하셨습니다.')

        order()
```

끝에 콜론(:)을 빠트리지 않도록 주의
이 블록은
앞에서부터 네 칸씩
들여쓰기한다

여기에는 들여쓰기 하지 않는다

주문하실 음료를 알려주세요
콜라
콜라 주문하셨습니다.



[Part7] 1. 사용자 정의 함수

- 매개변수를 갖는 함수의 정의

➤ **def** 함수명(매개변수):
 <수행할 문장1>
 <수행할 문장2>


```
In [2]: def print_price(num_drink):                                # ❶ 매개변수(num_drink) 정의
        """음료의 잔 수(num_drink)를 전달받아,
        가격을 화면에 출력한다."""
        price_per_drink = 2500                                     # 한 잔 당 가격
        total_price = num_drink * price_per_drink                 # ❷ num_drink에 전달된 값 사용
        print('음료', num_drink, '잔:', total_price)              # ❸

        print_price(3)

음료 3 잔: 7500
```



[Part7] 1. 사용자 정의 함수

● 여러 개의 인자를 전달받는 함수

➤ **def** 함수명(변수1, 변수2):
 <수행할 문장1>
 <수행할 문장2>


```
In [3]: def print_order(drink, cake):           # ❶ 콤마(,)로 구분해 매개변수 여러 개 정의
        """음료(drink)와 케익을(cake)를 전달받아,
        주문 내용을 화면에 출력한다."""
        print('음료:', drink, '/', '케익:', cake)

        print_order('카페라테', '치즈케익')    # ❷ 함수에 여러 개의 인자를 전달하여 호출하기
        print_order('달근케익', '우유')        # ❸ 전달하는 인자의 순서에 주의하자!

        음료: 카페라테 / 케익: 치즈케익
        음료: 달근케익 / 케익: 우유
```



[Part7] 1. 사용자 정의 함수

- Return 문: 데이터를 반환하기(값을 반환하는 함수)

➤ def 함수이름:
 함수의 내용

 return 반환값

```
In [4]: def price(num_drink):  
        """음료의 잔 수(num_drink)을 전달받아,  
        가격을 반환한다."""  
        price_per_drink = 2500  
        total_price = num_drink * price_per_drink  
        return total_price  
  
        result = price(3)  
        print('가격:', result)  
  
가격: 7500
```

❶ 계산한 값 반환하기

❷ 함수 호출하고 반환값 저장하기



[Part7] 2. 공공데이터와 Open API

● 공공데이터(<https://data.go.kr>)

- 공공데이터포털은 행정안전부에서 운영하는 공공데이터 통합제공 시스템이다.
- 대한민국 정부가 보유한 다양한 공공데이터를 개방하여 누구나 편리하고 손쉽게 활용할 수 있게 하는 것이 목적입니다.

● 오픈 API

- 오픈API란 누구나 사용할 수 있도록 공개된 API를 말합니다.
- 데이터를 표준화하고 프로그래밍해 외부 소프트웨어 개발자나 사용자와 공유하는 프로그램입니다.





[Part7] 2. 공공데이터와 Open API

● 개발계정 신청

- 공공데이터 포털에 회원 가입
- 원하는 정보(데이터) 검색 후 [활용신청]

DATA

공공데이터포털
.GO.KR

데이터찾기

국가데이터

개발계정 신청

데이터신청

목록등록관리시스템

로그아웃

마이페이지

사이트맵

데이터활용

정보공유

이용안내

오픈API 상세

f

t

y

URL 복사

XML

청소년 자원봉사 프로그램 서비스 정보

1

개발계정을 먼저 신청 해 주세요.

* 개발계정이란?

오픈 API를 활용해 서비스를 개발할 때 필요한 계정으로 하루 평균 1,000건 규모의 트래픽을 제공합니다.

활용신청

여성가족부 청소년 활동진흥원에서 제공하는 청소년 활동 정보를 서비스하여 청소년들이 봉사활동 참여 기회를 확대할 수 있도록 한다.

0

관심

OpenAPI 정보

분류체계	사회복지 - 사회복지일반	제공기관	여성가족부
관리부서명	청소년활동진흥과	관리부서 전화번호	
보유근거		수집방법	
API 유형	REST	데이터포맷	XML



- 개발된 서비스를 이용자가 사용하기 위한 계정으로 하루 최대 10만건의 트래픽을 제공함.

1



[Part7] 2. 공공데이터와 Open API

● 오픈API 활용 승인

- 개발계정과 운영계정 신청이 모두 끝났다면, 오픈API 활용 신청에 대한 승인을 기다려야 함.
- 승인은 '자동승인'과 '심의승인' 두 가지 방법으로 진행

- 자동승인이란?

별도 확인 없이 승인신청과 동시에 자동으로 승인되는 경우를 말합니다. 활용 신청 건에 대해 제공기관 담당자의 별도 확인 없이 신속하게 승인 처리를 받을 수 있습니다.

- 심의승인이란?

신청 후 2~3일 후 결과 통보활용 신청 건에 대해 제공기관 담당자의 '승인'이 완료되어야 오픈API를 활용할 수 있습니다.

신청 후 보통 2~3일 정도 소요가 되며 승인 여부는 회원가입 시 등록한 이메일로 통보합니다.



[Part7] . 공공데이터와 Open API

● 오픈API 인증키 발급

- 승인이 완료됐다면 활용신청 건에 대해 1인당 하나의 인증키를 발급받을 수 있습니다.
- 프로그램에 데이터 제공 주소(URL)와 오픈API 인증키를 입력한 뒤 API를 호출하면 원하는 데이터를 손쉽게 받아 서비스 개발에 활용할 수 있습니다.

> 오픈 API 인증키와 URL을 기반으로 API 호출 가능

<http://data.ekape.or.kr/openapi-data/service/user/mtrace/breeding/cattleMove?cattleNo=410002042894485&ServiceKey=서비스키>

제공기관 : 축산물품질평가원
오픈API명 : 쇠고기이력정보서비스
상세기능명 : 소 이동정보

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <response>
  - <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
  </header>
  <notice/>
  <body>
    - <items>
      - <item>
        <cattleNo>410002042894485</cattleNo>
        <farmAddr>경상남도 산청군 삼장면</farmAddr>
        <farmerNm>정재진</farmerNm>
        <movePlace>전산등록</movePlace>
        <moveYmd>2009-06-10</moveYmd>
      </item>
      - <item>
        <cattleNo>410002042894485</cattleNo>
        <farmAddr>경상남도 진주시 명석면</farmAddr>
        <farmerNm>이재생</farmerNm>
        <movePlace>양수</movePlace>
        <moveYmd>2012-09-13</moveYmd>
      </item>
    </items>
  </body>
  <cattleNo>410002042894485</cattleNo>
```


Part7. 사용자 정의 함수

학습효과

- 사용자 정의 함수 생성 방법 및 활용 방법 이해
- def 구조 이해 및 제작
- def 함수를 이용한 미니 프로젝트 진행
- 공공 API를 이용한 데이터 분석 및 관리

파일명

Part7_사용자 정의 함수와 공공데이터.html

[목표]

- 사용자 정의 함수 이해
- 사용자 정의 함수(def) 구조 및 생성
- 공공 API 이해 및 신청
- 공공API를 이용한 실시간 데이터 분석(미니 프로젝트)