

# NCS\_파이썬 노트정리\_545문제\_김승순

2021년 1월 3일 일요일 오전 10:51

## 20.11.24 파이썬 강의 Day1

데이터 분석가가 가져야할 기본 소양중 하나 = 데이터 웹크롤링 = 데이터 수집력

1. 빅데이터 수집을 위한 웹스크롤링을 할 수 있는 능력키우기
2. SQL처럼 파이썬으로 데이터를 검색하고 분석할 수 있는 능력 키우기
3. 딥러닝 개발자(연구원)가 되기위한 기본 코딩능력 갖추기

### 파이썬 설치

아나콘다 최신버전으로 설치

↓

파이썬 기본 프로그램 + 여러가지 유용한 패키지들

spyder 툴로 파이썬 코드를 연습할 예정!

파이썬으로 할 수 있는 것들?

1. 인공지능 --->머신러닝--->머신러닝의 신경망 ---> 딥러닝

### 001 대화식 모드로 프로그래밍하기

파이썬을 실행하는 방법 2가지

1. 대화식 모드  
-> 한 라인 한 라인씩 실행하는 것 (거의 쓸일 없음)

대화식 모드 실습!

1. anaconda prompt 을 연다
2. pythone 라고 입력 -> 접속됨

```
>>> a=1
>>> b=2
>>> a+b
3
```

요렇게 한 줄씩 입력!

2. 배치 모드

-> 여러개의 스크립트로 작성해서 한번에 실행하는 것

스파이더에서 아래와 같이 스크립트를 적어주고 전체 선택후 ctrl+enter 로 실행하는 것

```
a=1
b=2
print(a+b)
```

3이 출력됨.

문제 1. 아래의 스크립트를 배치모드로 실행하시오.

```
for i in [ 1, 2, 3, 4, 5, 6, 7]:
    print (i)
```

## 002 텍스트 에디터로 프로그래밍하기

spyder 와 같은 텍스트 창에서 프로그램 코드를 작성하는 .  
spyder 프로그램에서 ctrl + enter 키로 코드를 실행하거나 F5를 누르면 실행된다.

문제2. 아래와 같은 결과가 나오게 spyder 에서 프로그래밍 하시오.

```
2
3
4
```

답

```
for i in [ 2, 3, 4 ]:
    print (i)
```

## 003 변수명 만들기

어떤 값을 임시로 저장하는 변수의 이름을 만드는 방법과 규칙을 배울 것!

e.g)  
a = 1

e.g2)  
  
a=1  
b=2  
print (a+b)

설명: a라는 변수에 숫자 1을 할당한다.

변수이름 생성시 주의사항!

1. 변수 이름에는 다음 문자만 사용할 수 있다.
  - 소문자( a~z )
  - 대문자( A~Z)
  - 숫자 (0~9)
  - 언더스코어(\_)
2. 변수 이름은 숫자로 시작할 수 없다. ( 문자로 시작해야함)
3. 예약어를 사용 할 수 없다. ( 파이썬에서 이미 사용되고 있는 단어를 예약어라고 함)  
e.g) True, False, class, is, return

문제 3. 파이썬의 예약어가 무엇인지 확인하시오.

```
import keyword
print (keyword.kwlist)
```

결과

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

문제 4. 위의 예약어를 변수로 사용하면 에러가 나는지 직접 테스트 하시오!

```
and = 1
```

결과

SyntaxError: invalid syntax

#### 004 변수에 값 대입하기

다양한 값을 변수에 대입하는 방법을 배울 것.

파이썬은 변수에 값을 대입할때 '=' (assignment) 기호를 사용한다.

e.g)

```
a=1
b='scott'
print(b)
```

다른 프로그램 언어에서는 아래와 같이 좀 더 복잡한 코드로 작성해야 한다.

```
b varchar2(10) := 'scott';
```

기본적으로 파이썬의 코드는 심플함을 철학으로 삼고 있음

파이썬의 기본 철학 확인하는 방법

```
import this
```

```
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.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
```

파이썬은 c/c++, java 와는 달리 변수를 선언할 때 숫자형 자료인지 문자형 자료인지 그 자료형을 명시하지 않아도 된다.

PL/SQL          vs          파이썬

```
a number(10) := 2;          a = 2
```

문제 5. 변수 자료형(문자형 or 숫자형) 을 확인하시오.

```
b='scott'
print(b)
print (type(b) )
```

출력결과

```
scott
<class 'str'>
str 은 문자형 자료를 뜻한다.
```

문제 6. 숫자형변수의 자료형을 확인하시오.

```
a=2
print(a)
print ( type (a) )
```

출력결과

```
<class 'int'>
```

int는 숫자형 자료를 뜻한다.

#### 005 주석처리하기(#)

프로그램에서 주석 부분은 인터프리터에 의해 무시되는 텍스트의 한 부분이다.

코드를 설명하거나 나중에 어떤 문제를고치 위해 표시하는 등 다양한 목적으로 주석을 사용할 수 있다.

코드를 작성할때 주석을 잘 작성해두면 차후에 코드를 다시 보거나 타인이 코드를 검토할 때 매우 중요한 보조 활용이 될 수 있다.

그래서 주석을 항상 달아주는 습관 가지는 것이 좋다.

e.g)

##### 1. 한줄 주석

a = 1 # 변수 a 에 숫자 1을 할당 합니다.

print(a) # a 의 결과를 출력 합니다.

##### 2. 여러줄 주석

""" 아래의 프로그램은 a 변수에 1을 할당해서 프린트하는 프로그램 입니다."""

```
a=1
print(a)
```

#### 006 자료형 개념 배우기

자료형이란 프로그래밍을 할 때 쓰이는 숫자, 문자열등의 자료형태로 사용되는 모든 것을 뜻함.

파이썬에서 자주 다루게 되는 자료형은 5가지!

1) 숫자형 자료형 : 숫자를 표현하는 자료형

e.g) a=1

2) 문자형 자료형 : 문자를 표현하는 자료형

e.g) b='scott'

3) 리스트 자료형 : [ ] 대괄호 안에 임의의 객체를 순서있게 나열한 자료형

e.g) d= [ 1, 2, 3 ]

print (d) 하면 1,2,3 이 출력

print (d [0] ) # : d 리스트에서 첫번째 요소를 프린트 하라! 는 뜻

파이썬은 시작을 0부터함. (자료의 순서 순번)

4) 튜플 자료형

5) 사전 자료형

문제 7. 아래의 리스트에서 문자 k를 출력하시오.

```
mmm=['a','b','c','d','e','k','m','n','z']
    0  1  2  3  4  5
print(mmm[5])
```

문제 8. 동전을 던져서 앞면이 나오는지 뒷면이 나오는지 확인하시오

```
import random
```

random 이라는 모듈을 이 코드에서 쓰겠다 는 뜻

SQL의 dbms.random 처럼 특정 목적으로 만든 파이썬 코드

모듈 : 특정 목적을 위해 만든 파이썬 코드의 집합.

```
coin = ['앞면','뒷면']
```

print (random.choice(coin)) -> random 모듈안의 choice 라는 함수를 이용해라~!

문제 9. 위에서는 동전을 던졌는데 이번에는 주사위를 던져서 주사위의 눈이 뭐가 나오는지 확인하는 파이썬 코드를 작성하시오!

```
import random
```

```
dice=['1','2','3','4','5','6']
```

```
print(random.choice(dice))
```

007 자료형 출력 개념 배우기(print)

print 함수를 이용하여

```
a=200
b='i love python'
c= ['a','b','c']
print(a)
print(b)
print(c)
```

결과

```
200
i love python
['a', 'b', 'c']
```

008 들여쓰기 개념 배우기

파이썬에는 실행코드 부분을 묶어주는 소괄호 '( )' 가 없음! 대신 들여쓰기로 괄호를 대신함  
파이썬은 다른 프로그래밍 언어와 달리 if, for, while 등과 같은 제어문, 루프문의 실행코드 부분을 구분해주는 괄호가 없음.

대신! 들여쓰기로 괄호를 구분함!

파이썬에서 제어문이나 함수이름, 클래스이름 뒤에 콜론(:) 으로 제어문, 함수이름, 클래스이름의 끝을 표시하며 콜론(:) 다음에 실행코드를 작성하는데, 이때 들여쓰기를 해야함

e.g 1) 실행코드를 다음 라인에 작성했을 경우

```
listdata = ['a','b','c'] -> listdata 라는 변수로 리스트를 생성해 a,b,c 라는 값을 listdata에 할당한 것
if 'a' in listdata: -> listdata 리스트 변수에 'a' 라는 요소가 있다면 아래의 print ('a 가 listdata에 있습니다.') 라는 명령어
1칸
2칸
3칸
4칸
(들여쓰기) print ('a가 listdata에 있습니다.')
```

e.g 2) 실행코드를 한 라인에 작성한 경우

```
listdata = ['a','b','c']
```

```
if 'a' in listdata: ('a'가 listdata에 있습니다.)
```

콜론 (:) 으로 if문의 끝을 알린다.

문제 10. 점심시간에 했던 주사위의 눈이 6개가 있는 변수를 만들고 if문을 써서 '주사위의 눈에 숫자 5가 있습니다.' 라는 메시지가 출력되게 하시오.

```
dice =[1,2,3,4,5,6]
if 5 in dice: print('주사위의 눈에 숫자 5가 있습니다.')
```

or

```
dice =[1,2,3,4,5,6]
if 5 in dice:
```

```
    print('주사위의 눈에 숫자 5가 있습니다.')
```

문제 11. 위의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 위와 동일한 메시지가 출력되게 하시오.

```
a= int(input('숫자를 입력하세요~'))
dice =[1,2,3,4,5,6]
if a in dice: print('주사위의 눈에 숫자',a,'가 있습니다.')
```

하면 우측 console 창에 숫자를 입력하라는 메시지가 뜬.  
거기에 숫자 입력후 엔터치면 값이 출력됨.

'int' 는 데이터 유형을 변경해주는 명령어

현재 숫자를 입력하세요~ 가 문자형 데이터이기 때문에 아래에 dice의 데이터들은 숫자이므로  
매칭이 되지 않을 테니, 숫자형으로 인식되어 조건 입력시 dice 에서 매칭되는 자료를 찾을 수 있도록 해주는 것.

dice 에 없는 숫자를 넣으면 아무 것도 출력되지 않음.

## 009 if문 개념 배우기 ① (if~else)

어떤 조건을 참과 거짓으로 판단할 때 if문을 사용한다.

참과 거짓을 구분하여 코드를 실행하면 if ~ else 를 사용하면 된다.

코드를 작성하다 보면 조건에 따라 수행하는 일을 달리해야 하는 경우가 생김.

조건이 참인지 거짓인지 검사를 하고, 참인 경우에는 a라는일을 하고 거짓인 경우에는 b라는 일을 해라  
라는 식으로 처리를 할 수 있음!

SQL때의 decode 문 처럼!

e.g)

if 조건:

실행코드1

else:

실행코드2

```
a=int( input('숫자를 입력하세요~'))
```

```
if a%2 == 0:
```

```
    print('짝수입니다.')
```

```
else:
```

```
print('홀수입니다.')
```

%는 SQL의 mod 처럼 나눈 나머지 값을 출력하는 파이썬 연산자 코드

파이썬의 이퀄은 '==' 두개를 쓴다!

그냥 이퀄 표시 한 개 '=' 를 쓰면 '할당 연산자' 임.

문제 12. 주사위의 눈을 담은 dice 변수를 만들고 숫자를 물어보게 해서 해당 숫자가 주사위의 눈 중에 있으면

'해당 숫자가 있습니다.' 라는 메시지가 출력되게 하고 없으면 '해당 숫자가 없습니다' 라는 메시지가 출력되게 하시오.

```
dice=[1,2,3,4,5,6]
a=int ( input('숫자를 입력하세요~'))
if a in dice:
    print('주사위의 눈에 숫자',a,'가(이) 있습니다.')
else:
    print('주사위의 눈에 숫자',a,'가(이) 없습니다.')
```

문제 13. 숫자 두개를 아래와 같이 각각 물어보게 하고 아래처럼 메시지가 출력되게 하시오.

첫번째 숫자를 입력하세요~2

두번째 숫자를 입력하세요~3

2는 3보다 작습니다.

첫번째 숫자를 입력하세요~3

두번째 숫자를 입력하세요~2

3은 2보다 큼니다.

```
a=int ( input('첫번째 숫자를 입력하세요~'))
b=int ( input('두번째 숫자를 입력하세요~'))
if a < b:
    print(a,'는',b,'보다 작습니다')
else:
    print(a,'는',b,'보다 큼니다')
```

콤마 (,) 로 각 자료를 구분하는 것.!

문자열은 ' ' 로 감싸고, 구분은 콤마 로 한 것

그러나! 이 경우는 같다. 를 구현할 수가 없음.

else 를 수행해버림.

010 if문 개념 배우기 ② (if~elif)

여러개의 조건을 순차적으로 체크하고 해당 조건이 참이면 특정 로직을 수행하고자 할때 if~elif 문을 사용한다.

e.g)

if 조건1:

```
    실행코드1
elif 조건2:
    실행코드2
else:
    실행코드3
```

e.g 2) 문제 13을 다시 수행하는데 같은 숫자 2개가 입력되면 서로 같습니다. 라는 메시지가 출력되게 하시오.

```
a=int ( input('첫번째 숫자를 입력하세요~'))
b=int ( input('두번째 숫자를 입력하세요~'))
if a < b:
    print(a,'는',b,'보다 작습니다')
elif a==b:
    print(a,'와',b,'는 서로 같습니다')
else:
    print(a,'는',b,'보다 큼니다')
```

선생님 답.

```
a=int ( input('첫번째 숫자를 입력하세요~'))
b=int ( input('두번째 숫자를 입력하세요~'))
if a < b:
    print(a,'는',b,'보다 작습니다')
elif a>b:
    print(a,'는',b,'보다 큼니다')
else:
    print(a,'는',b,'와 같습니다.')
```

elif 는 몇개든 계속해서 쓸 수 있다.

```
if 조건1:
    실행코드1
elif 조건2:
    실행코드2
elif 조건3:
    실행코드3
elif 조건4:
    실행코드4
elif 조건5:
    실행코드5
else:
    실행코드6
```

else 는 위의 조건 2가 조건 3이 아니라면! 이라는 뜻으로 사용

011 for문 개념 배우기 ① (for)

특정 코드를 반복적으로 수행하기 위해서는 반복문을 사용해 하는데 파이썬에서는 for 문이 반복문을 수행하기 위해 가장 많이 사용되는 문법이다.

e.g)

```
for 변수 in 범위:
    반복적으로 실행할 코드
```



## 범위

### 1. 리스트 범위인 경우

```
for i in [1,2,3]:  
    print(i)
```

### 2. 튜플 범위인 경우

```
for i in (1,2,3):  
    print(i)
```

### 3. range() 범위인 경우 (제일 자주 사용)

```
for i in range(10):  
    print(i)
```

시작점인 0부터 10개를 출력해 0~9 가 출력됨.

20을 입력하면 0부터 19까지 20개 출력

### 4. 사전형 범위인 경우 (어쩌다 한번씩 사용)

키 값

↓ ↓

```
m = {'i': '나는', 'am': '입니다', 'boy': '소년' }
```

```
for i in m:  
    print(i)
```

### 5. 문자형 범위인 경우

```
for i in 'I am a boy':  
    print(i)
```

for 루프(반복하다) 문의 range 사용법 정리

```
for i in range(6):  
    print(i)
```

0부터 5까지 출력한다.

```
for i in range(1,6):  
    print(i)
```

1부터 5까지 출력된다.

```
for i in range(1, 6, 2):  
    print(i)
```

1, 3, 5가 나옴.

즉 1~5사이를 출력하는데 숫자 2개씩 건너뛴 것.

```
for i in range(6, 1, -1):  
    print(i)
```

6부터 2까지 나옴.

문제 14. ★ 을 5개 출력하시오.

★★★★★ 처럼!

```
print('★'*5)
```

문제 15. 아래와 같은 결과가 출력되는 파이썬 코드를 작성하시오. (SQL 알고리즘 5번문제)

숫자를 입력하세요~5입력하면

```
★
★★
★★★
★★★★
★★★★★
```

가 나오도록! 3줄이 끝남!

```
a=int (input('숫자를 입력하세요~'))
for a in range(1,6):
    print('★'*a)
```

```
a=int (input('숫자를 입력하세요~'))
for a in range(1,6):
    print(a)
를 입력하면
```

와 같이 출력되는 것을 활용한 것.

```
즉
a가 1일때는 ★
a가 2일때는 ★★
a가 3일때는 ★★★
a가 4일때는 ★★★★
a가 5일때는 ★★★★★
```

로 수행되서 삼각형이 된 것

문제 16. 점심시간에는 주사위를 1번만 던졌는데, 이번에는 주사위를 10번을 던져서 출력되는 눈을 확인하시오.

```
import random

dice = (1,2,3,4,5,6)

for i in range(10):
    print(random.choice(dice))
```

for 뒤에 아무문자나 써도 되는데 보통 i를 많이 씀

임의의 문자 i지만 그것을 활용해서 아래와 같 출력할 수도 있음.

파이썬은 첫 숫자를 0으로 하기에 그냥 i 쓰면 '0번째는~' 부터시작됨

SQL에서 주사위굴릴때도 level 을 쓰지 않고 connect by 만 써도 수행되는 것처럼 여기서의 i도 level 같은 역할

i 말고 임의의 아무 문자를 써도 괜찮음.

혹은

```
import random

dice = (1,2,3,4,5,6)

for i in range(1,11): #1이상 11미만이란 뜻
    print(i,'번째는',random.choice(dice))
```

로 해도 됨.

문제 17. 동전을 10번 던지세요.

```
import random
coin=['앞면','뒷면']
for i in range(10):
    print(i+1,'번째는',random.choice(coin),'입니다.')
```

import random -> random 모듈 코드를 지금 현재 코드창에서 사용하겠다. 라는 뜻

필요한 대부분은 거의 다 코딩이 되어 있어서, 그냥 import 해오기만 하면 됨  
그래서 파이썬이 쉽다고 하는 것!

문제 18. 동전을 10번 던졌을때 앞면이 나오는 횟수가 어떻게 되는가?

```
import random

coin=['앞면','뒷면']
cnt=0 -> cnt 라는 변수에 0을 할당한다. (변수이름 임의 해도됨 숫자만 아니면 ok)
for i in range(1,11): -> 1부터 10까지 루프문을 10번 실행하라! 는 뜻
    result= random.choice(coin)
    if result == '앞면':
        cnt = cnt + 1 -> 할당 연산자 오른쪽부터 실행해서 그 결과를 왼쪽변수에 할당해 준다. 는 뜻.
print(cnt) -> for 문과 위치를 맞춰서 같은 라인에 print(cnt) 를 쓰면 for문이 다 완료되면 print(cnt) 가 수행됨.
```

어떻게 위의 명령문이 시행되는지를 시뮬레이션 해보자면

1 번째는 뒷면 입니다.  
2 번째는 뒷면 입니다.  
3 번째는 앞면 입니다. -> if문에 지정한 '앞면' 조건이 나옴. so, cnt=cnt+1 이 수행되서, cnt = 0+1 이 되어 cnt가 1  
4 번째는 뒷면 입니다.  
5 번째는 뒷면 입니다.  
6 번째는 앞면 입니다. -> cnt= cnt+1 -> cnt= 1 + 1 -> cnt 가 2입니다.  
7 번째는 뒷면 입니다.  
8 번째는 앞면 입니다. -> cnt= cnt+1 -> cnt= 1 + 1 -> cnt 가 3입니다.  
9 번째는 앞면 입니다. -> cnt= cnt+1 -> cnt= 1 + 1 -> cnt 가 4입니다. 로 최종 cnt는 4가 됨.  
10 번째는 뒷면 입니다.

```
import random

dice=['앞면','뒷면']
cnt=0
for i in range(1,11):
    result=random.choice(coin)
    if result == '앞면':
        cnt=cnt+1
print('앞면이 나온 횟수는 총',cnt,'입니다.')
```

문제 19. 주사위 100번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오.

```
import random

dice=[1,2,3,4,5,6]
cnt=0
for i in range (1,101): <- for문 작성. 마무리가 되었으니 콜론 찍어줘야함.
    result=random.choice(dice) <- result 라는 변수를 만들어줌.
    if result == 3: <- if문 작성. 마무리가 되었으니 콜론을 찍어줘야함
        cnt=cnt+1 <- 실행문
print('주사위는 3이 나온 횟수는 총',cnt,'입니다.')
```

문제 20. 주사위를 1000번 던지고 주사위의 눈 5가 나올 확률을 구하시오.

```
import random

dice=[1,2,3,4,5,6]
cnt=0
for i in range (1,1001):
    result=random.choice(dice)
    if result == 5:
        cnt=cnt+1
print('주사위를 1000번 던질때 5가 나올 확률은',cnt/1000,'%','입니다.')
```

1/6 근사치가 나오므로 정답!

문제 21. 동전을 10000번 던져서 앞면이 나올 확률을 출력하시오.

```
import random

coin=['앞면','뒷면']
cnt=0
for i in range(1,10001) :
    result=random.choice(coin)
    if result == '앞면':
        cnt=cnt+1
print('동전을 만번 던져서 앞면이 나올 확률은',cnt/10000,'%입니다.')
```

(오늘의 마지막 문제)

문제 22. SQL 알고리즘13번 문제를 파이썬으로 구현하시오!

주사위 하나와 동전 한개를 동시에 던져서 주사위의 눈은 5가 나오고  
동전은 앞면이 나올 확률은 어떻게 되는가 ?

힌트 and 를 사용하면 됨.

```
import random

coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]
cnt=0
for i in range(1,100001) :
```

```

result1=random.choice(coin)
result2=random.choice(dice)
if result1 == '앞면' and result2==5:
    cnt=cnt+1
print('동전과 주사위를 동시에 던져 앞면과 숫자 5가 나올 확률은',cnt/100000,'%입니다.')
```

#### 20.11.25 파이썬강의 Day2(문제23~문제40)

데이터 분석가 및 딥러닝 개발자가 되기 위해서 갖춰야할 기술

1. SQL
2. Python -> 단 데이터 분석에 초점을 두고 배울 것!

\*파이썬 200제 예제에서 파이썬 기본기를 다지기!

문제 23., (SQL 알고리즘 4번) for loop 문을 이용해서 구구단 2단을 출력하시오.

```

for i in range (1,10):
    print('2 x', i, '=',2*i)
```

문제 24. (SQL알고리즘 8번) for loop 문을 이용해서 구구단 전체를 출력하시오.

답

```

for i in range (1,10):
    print('2 x', i, '=',2*i)
for i in range (1,10):
    print('3 x', i, '=',3*i)
for i in range (1,10):
    print('4 x', i, '=',4*i)
for i in range (1,10):
    print('5 x', i, '=',5*i)
for i in range (1,10):
    print('6 x', i, '=',6*i)
for i in range (1,10):
    print('7 x', i, '=',7*i)
for i in range (1,10):
    print('8 x', i, '=',8*i)
for i in range (1,10):
    print('9 x', i, '=',9*i)
```

#### 011-1 중첩 for loop문

loop 문을 중첩시켜서 수행한다는 것.

loopa문 자체를 반복시킨다.

e.g)

```

for i in range (1,10) :
    print('2 x', i, '=',2*i)
```

위의 것은 구구단 2단을 출력하기 위한 문법.

위의 실행문을 유지

for j in range(1,10): # for 문 다음에는 항상 콜론 (:) 으로 마무리 해줘야 한다.

```

    for i in range (1,10) : # 세미콜론 뒤에는 탭 한번 or 4칸 띄우기
        print('2 x', i, '=',2*i)
```

위와 같이 하면 2단을 출력하는 for loop문이 9번 출력되어 2단이 9번 나온다.

위 처럼 2단이 9번 나오는데 아니라 2단부터 9단까지 순차적으로 출력해야 하므로

다음과 같이 작성해야 한다. (포인트는 print 문의 2가 계속 바뀌줘야 하는 것.)

```
for j in range(2,10): # j는 2부터 10미만 까지 출력이 된다.
    for i in range (1,10) : #i는 1부터 10미만 까지 출력이 된다.
        print(j, 'x', i, '=', j*i)
```

설명:

j=2 일때, i가 1부터 9까지 반복실행  
즉, 2x1 = 2\*1수행, 2x2= 2\*2 수행 ..... 2x9= 2\*9 수행 하여 2단 출력

j=3 일때, i가 1부터 9까지 반복실행  
j=4 일때, i가 1부터 9까지 반복실행  
.  
.  
.  
j=9 일때, i가 1부터 9까지 반복실행

문제 25. 주사위를 10번 던져서 주사위의 눈을 10개 출력하시오.

```
import random

dice=[1,2,3,4,5,6]
for i in range(1,11):
    print(random.choice(dice))
```

문제 26. 주사위를 10번 던졌을 때 짝수가 나오는 횟수를 출력하시오.

```
import random

dice=[1,2,3,4,5,6]
cnt=0
for i in range(1,11):
    result = random.choice(dice) #주사위의 눈이 result 변수에 할당된다.
    if result%2 == 0: # result 변수의 주사위 눈이 짝수이면
        cnt = cnt+1 #cnt 를 1씩 증가 시킨다.
print(cnt) #print의 위치를 for 문과 맞춰줘서 cnt에 담긴 최종결과를 출력한다.
```

문제 27. 위의 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 작업을 5번 반복해서 짝수가 나오는 횟수가 5개가 나오게 하시오. (반복 loop문사용)

```
import random

dice=[1,2,3,4,5,6]

for j in range(1,6):
    cnt=0
    for i in range(1,10):
        result = random.choice(dice)
        if result%2 == 0:
            cnt =cnt+1
    print(cnt)
```

문제 26의 실행문을 그대로 유지하면서 cnt 위에 for j in range 를 추가했다.

이때 for문이 끝나면 다음 줄 부터는 탭1칸 or 4칸 공간을 띄워야 하는데 일일이 모든 줄에 하려면 번거롭다.

그럴땐 밑의 실행문을 모두 드래그 한 뒤 tap 을 누르면 형식에 맞게 4칸씩 밀려난다. 만약 취소하고 싶다면, 드래그 된 상태에서 shift + tap 하면 된다.

설명: 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 for loop문을 5번 반복하기 위해서 그 위에 for loop문(for j in range~~) 을 쓰고 번 반복했다.

문제 27. 위의 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 작업을 5번 반복해서 짝수가 나오는 횟수가 5개가 나오게 하시오. (반복 loop문사용)

```
import random

dice=[1,2,3,4,5,6]

for j in range(1,6):
    cnt=0
    for i in range(1,10):
        result = random.choice(dice)
        if result%2 == 0:
            cnt =cnt+1
    print(cnt)
```

문제 26의 실행문을 그대로 유지하면서 cnt 위에 for j in range 를 추가했다.

이때 for문이 끝나면 다음 줄 부터는 탭1칸 or 4칸 공간을 띄워야 하는데 일일이 모든 줄에 하려면 번거롭다.

그럴땐 밑의 실행문을 모두 드래그 한 뒤 tap 을 누르면 형식에 맞게 4칸씩 밀려난다. 만약 취소하고 싶다면, 드레그 된 상태에서 shift + tap 하면 된다.

설명: 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 for loop문을 5번 반복하기 위해서 그 위에 for loop문(for j in range~~) 을 쓰고 번 반복했다.

문제 28. 동전을 100번 던져서 앞면이 나올 확률을 출력하시오.

```
import random

coin=['앞면','뒷면']
cnt=0
for i in range(1,101):
    result=random.choice(coin)
    if result == '앞면':
        cnt=cnt+1
    print('동전 100번 던질때 앞면이 나올 확률은',cnt/100,'%입니다.')
```

문제 29. 동전을 100번 던져서 앞면이 나올 확률을 구하는 작업을 50번 반복해서 확률을 50개 출력하시오.

```
import random

coin=['앞면','뒷면']

for j in range(1,51):
    cnt=0
    for i in range(1,101):
        result=random.choice(coin)
        if result == '앞면':
            cnt=cnt+1
    print('동전 100번 던질때 앞면이 나올 확률은',cnt/100,'%입니다.')
```

#print 문은 for j ~와 같은 라인이 아니라 for i~와 같은 라인에 와야 한다.

문제 30. 파이썬 리스트의 append 함수를 이용해서 위의 확률 50개를 리스트에 담으시오.

\* 리스트 append 함수의 기본 문법

a=[] # 비어있는 리스트 a 를 생성합니다.

a.append(7) #a 리스트에 숫자 7을 넣는다.  
print(a)

a.append(8) #a 리스트에 숫자 8을 넣는다.  
print(a)

답.

```
import random
```

```
coin=['앞면','뒷면']  
a=[] #비어있는 리스트 a 를 생성한다.  
for j in range(1,51):  
    cnt=0  
    for i in range(1,101):  
        result=random.choice(coin)  
        if result == '앞면':  
            cnt=cnt+1  
    print('동전 100번 던질때 앞면이 나올 확률은',cnt/100,'%입니다.')  
    a.append(cnt/100) #a 리스트에 확률을 입력한다.  
print(a) # a리스트에 담겨 있는 값들을 출력한다.
```

통계를 코드로 구현 게시판 -> 이항분포란 -> 링크 3번

<http://cafe.daum.net/oracleoracle/SfWb/50>

을 파이썬으로 구현하기!

###

#문제 31. 두개의 동전을 300번 던져서 둘다 앞면이 나오는 횟수를 구하시오.

```
import random
```

```
coin1=['앞면','뒷면']  
coin2=['앞면','뒷면']  
cnt=0  
for i in range(1,301):  
    result_a=random.choice(coin1)  
    result_b=random.choice(coin2)  
    if result_a == '앞면' and result_b == '앞면':  
        cnt=cnt+1  
print('동전 2개를 동시에 300번 던질때 둘다 앞면이 나오는 횟수는',cnt,'입니다.')
```

###

#문제 32. x 라는 리스트를 만들고 x 리스트에 문제 31에서 앞면이 나오는 횟수를  
#x 리스트에 담는데, 위의 둘다 앞면이 나오는 횟수를 출력하는 for loop문을 1000번  
#반복해서 리스트 1000개를 입력하시오.

```
import random
```

```
coin1=['앞면','뒷면']  
coin2=['앞면','뒷면']  
x=[]
```



```
for j in range (1,1001):
```

```
    for i in range(1,301):
        result_a=random.choice(coin1)
        result_b=random.choice(coin2)
        if result_a == '앞면' and result_b == '앞면':
            cnt=cnt+1
    x.append(cnt)
```

```
print(x)
```

```
###
```

#문제 33. 문제 32에서 만든 확률 변수 x의 평균, 분산 및 표준편차를 출력하시오!

```
import random
```

```
import numpy as np #numpy 모델을 이 코드에서 사용하겠다.
```

```
coin1 =['앞면', '뒷면']
```

```
coin2 =['앞면', '뒷면']
```

```
x = []
for i in range(1, 1001):
    cnt = 0
    for i in range(1, 301):
        result1 = random.choice(coin1)
        result2 = random.choice(coin2)
        if result1=='앞면' and result2=='앞면':
            cnt = cnt + 1
    x.append(cnt)
```

```
print(np.mean(x)) # 74.684
print(np.var(x)) # 55.366144
print(np.std(x)) # 7.440842962998211
```

```
###
```

#문제 34. 한개의 주사위를 360번 던져서 3의 배수의 눈이 나오는 횟수를 구하는 것을  
#1000번을 수행하여 x라는 변수에 입력하고 이 x의 평균, 분산, 표준편차를 출력하시오.

```
import random
import numpy as np
```

```
dice=[1,2,3,4,5,6]
```

```
x= []
for j in range (1,1001): #1000번 반복 수행하는 for loop문 작성
    cnt=0 #cnt 에 0을 할당
    for i in range(1,361): #360번 반복수행하는 for loop문 작성
        result=random.choice(dice) #주사위에서 눈 1개 출력해서 result에 담는다.
        if result%3 == 0: # 그 주사위의 눈을 3으로 나눈 나머지가 0이면
            cnt=cnt+1 #cnt 를 1씩 증가시키겠다.
    x.append(cnt) #비어 있는 리스트 x에 cnt 값(주사위 360번 던져 3의 배수가 나온 횟수)를 추가하겠다.
print(np.mean(x))
print(np.var(x))
print(np.std(x))
```

```
###
```

#012 for문 개념 배우기 ② (for~continue~break)

#for 반복문 내에서 continue 를 만나면 그 다음 반복 실행으로 넘어가며 break 를 만나면 for 반복문을 완전히 벗어나게 된다.

e.g)

for 변수 in 범위:

실행코드

continue # 이 부분을 그냥 지나치고 다음 반복문을 수행하라!

실행코드

break # for 반복문을 탈출

###

#문제 35. for문을 사용해서 숫자 1부터 10까지 출력하시오.

```
for i in range(1,11):  
    print(i)
```

###

#문제 36. 위의 결과에서 숫자 5만 출력하지 않고 나머지 숫자만 출력하시오.

```
for i in range(1,11):  
    if i == 5:  
        continue # 이 부분은 실행하지 말고 그냥 지나치고 다음 loop 실행문을 계속 실행해라.  
    print(i)
```

###

#문제 37. (SQL알고리즘 문제 2번) 1부터 10까지의 숫자를 출력하는데 짝수만 출력하시오. (=홀수는 출력하지 말것!)

```
for i in range(1,11):  
    if i%2 == 1: #i를 2로 나누었을때 1이 남으면  
        continue #지나쳐라  
    print(i)
```

###

#문제 38. (SQL알고리즘 문제 1번) 1부터 10까지의 합을 파이썬으로 구현하시오.

```
cnt=0  
for i in range(1,11):  
    cnt=cnt+i  
print(cnt)
```

#혹은

```
x=[ ]  
cnt=0  
for i in range(1,11):  
    x.append(i)  
print(sum(x))
```

###

#012 for문 개념 배우기 ② (for~continue~break) - for 문에서 사용하는 break 문

#e.g)

```
for i in range(1,11): #1부터 10까지 루프문을 수행하는데  
    print(i) #를 출력하다가  
    if i ==3: #i가 3이 되면  
        break #loop문을 종료시켜라.
```

###

#문제 39. 1부터 100까지 출력하는 for loop 문을 작성하는데 다음과 같이 숫자를 물어보게 해서  
# 입력된 숫자까지만 출력되게 하시오.

#숫자를 입력하세요~7

#1  
#2  
#3  
#4  
#5  
#6  
#7

```
a=int( input('숫자를 입력하세요~'))
for a in range(1,101):
    print(a)
    if a == 7:
        break
```

#or

```
a=int( input('숫자를 입력하세요~'))
for i in range(1,101):
    print(i)
    if i == a:
        break
```

###

#(오늘의 마지막 문제)

#문제 40. (SQL알고리즘 문제 19번.) 두 숫자를 각각 물어보게 하고 입력받은 두 숫자의 최대공약수를 출력하시오.

#Hint : 16%i == 0 and 24%i == 0

#Hint : print(max(16,24)) #둘 중 더 큰값이 나옴.

#Hint : print(min(16,24)) #둘 중 더 작은 값이 나옴.

#Hint : break를 써야함.

#첫번째 숫자를 입력하세요~16

#두번째 숫자를 입력하세요~24

#최대공약수는 8입니다.

```
a=int( input('첫번째 숫자를 입력하세요~'))
b=int( input('두번째 숫자를 입력하세요~'))
```

while

20.11.26 파이썬강의 Day3(문제41~문제85)

#013 for문 개념 배우기 ③ (for~else)

# for 반복문을 완전히 수행했을때만 실행하는 부분을 정의하려면 for ~ else 문을 사용해야 한다.

# for ~ else 에서 else 뒤에 실행되는 코드는 for 반복문을 성공적으로 수행해야지만 실행 된다.

# e.g)

```
for i in range(1,11):
    print(i)
else:
    print('Perfect')
```

1

```
2
3
4
5
6
7
8
9
10
Perfect
```

#break 가 중간에 들어가 중단되면 마지막 else절의 명령어는 실행되지 않는다.

# e.g)

```
for i in range(1,11):
    print(i)
    if i ==2:
        break
else:
    print('Perfect')
```

```
1
2
만 나온다.
```

###

#문제 41. 위의 예제 코드를 수정해서 숫자를 물어보게 하고 그 숫자를 입력하면 해당 숫자만큼 1번부터 출력되게 하시오.

```
a=int (input('숫자를 입력하시오~') )
```

```
for i in range (1,a+1):
    print(i)
```

###

#문제 42. 위의 입력한 숫자까지 다 출력하면 아래에 perfect 이란 단어가 출력되게 하시오.

```
a=int (input('숫자를 입력하시오~') )
```

```
for i in range (1,a+1):
    print(i)
else:
    print('Perfect')
```

###

#문제 43. 위의 코드를 수정해서 중단할 숫자 도 물어보게 해서 아래와 같이 실행되게 하시오.

#숫자를 입력하세요~10 -> 1~10부터 나와야함.

#중단할 숫자를 입력하세요~5 -> 1~5까지만 나오고 중단되어야 함. -> 이 경우 Perfect 가 나오면 안됨.

```
a=int (input('숫자를 입력하시오~') )
```

```
b=int (input('중단할 숫자를 입력하시오~') )
```

```
for i in range(1,a+1):
    print(i)
    if i ==b:
        break
else:
    print('Pefect')
```

###

#014 while문 개념 배우기(while~continue~break)

# for 루프문처럼 while 루프문도 같은 반복문 이다.

# for 루프문은 특정 범위에서 반복실행 하게 하는 반면

# while 루프문은 특정 조건에서 코드를 반복실행하게 한다.

```
# for루프문      vs      while 루프문
for i in range(10):      x=0
    print (i)            while x < 10:
                        print(x)
                        x = x + 1
```

#0~9 가 순서대로 출력됨.

#같은 결과를 구현하려 할때 for 문이 훨씬 간결하다보니 대체로 for문을 주로 사용함.

```
x=0
while x <10:
    print(x) # 이 실행문을 실행하려면 x<10이 라는 조건이 만족되어야 한다.
    x=x+1
```

# when 0<10 : 참

# when 1<10 : 참

#.

#.

#.

# when 10<10 : 거짓 (루프문의 실행문 실행하지 x)

###

#문제 44. 아래와 같이 숫자를 물어보게 하고, 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오.

#숫자를 입력하세요~5입력하면

#★

#★★

#★★★

#★★★★

#★★★★★

#가 나오도록! 3줄이 끝남!

# 단 for 문이 아닌 while 문을 사용하시오.

```
a=int (input('숫자를 입력하세요~'))
x=0
while x <a:
    x=x+1
    print(x*'★')
```

###

#문제 45. 아래와 같이 결과가 출력되게 하시오.

#숫자를 입력하세요~5입력하면

#★★★★★

#★★★★★

#★★★★

#★★★

#★★

a=int (input('숫자를 입력하세요~'))

```
while a>0:
    print('★'*a)
```

```
a= a-1
```

# a> 0 은 아래의 실행문이 실행되기 위한 참/거짓을 가르는 조건  
# 조건을 만족하면 아래의 명령문이 실행이 됨.

#처음에 5를 입력하면

```
#5> 0 이므로 참 -> '★'*5 를 수행함
#그 이후로
#4> 0 이므로 참 -> '★'*4 를 수행함
#3> 0 이므로 참 -> '★'*3 를 수행함
#2> 0 이므로 참 -> '★'*2 를 수행함
#1> 0 이므로 참 -> '★'*1 를 수행함
#0>0 은 거짓. 수행하지 않음.
```

```
###
```

#문제 46. 주사위 2개를 동시에 던져서 두개의 눈의 합을 구하는 실행문을 10000번 반복해서  
#그 10000개를 비어있는 리스트 x에 넣으시오.

```
import random
```

```
dice1=[1,2,3,4,5,6]
dice2=[1,2,3,4,5,6]
```

```
a=[ ]
x=0
while x <10000:
    result1 =random.choice(dice1)
    result2 =random.choice(dice2)
    a.append(result1+result2+result2)
    x = x+1
print(a)
```

# 리스트 a에 요소의 개수가 10000개인지 확인하고 싶다면?  
print( len(a)) #라고 입력하면 된다.

```
###
```

#문제 47. (SQL알고리즘 10번)위의 코드를 이용해서 두개의 주사위의 눈의 합이 10이 되는 확률을 구하시오.

```
import random
```

```
dice1=[1,2,3,4,5,6]
dice2=[1,2,3,4,5,6]
```

```
a=[ ]
cnt=0
while cnt <10000:
    result1 =random.choice(dice1)
    result2 =random.choice(dice2)
    if result1+result2 == 10: #두개의 주사위 눈의 합이 10이면
        a.append(result1+result2) # a 리스트에 추가해라.
    cnt = cnt+1 #cnt를 1씩 증가시킨다.
print( len(a)/10000) # len 함수로 리스트 a의 개수를 카운트하고 /모수인 10000으로 나누어주었다.
```

```
###
```

#015 None 개념 배우기

# none 은 아무것도 없다. (oracle의 null 같은 것) 는 의미의 상수이다.



```

for i in range(1,10001): #10000번 루프를 돌리는데!
    result1 = random.uniform(0,1) #0~1사이의 난수를 result1에 담는다.
    result2 = random.uniform(0,1) #0~1사이의 난수를 result2에 담는다.
    if (result1**2) + (result2**2) <=1: #result1의제곱 + #result2의 제곱 <= 1 이면 부채꼴안의 점인데,
        cnt = cnt+1 # 부채꼴 안에 점이 있으면 cnt 를 1증가 시킨다.
print(cnt/10000*4)

###

#017 실수형 자료 이해하기

# 실수는 소수로 나타낼 수 있는 유리수와 소수로 나타낼 수 없는 무리수로 구성된 집합이다.
# (무리수의 예 : 원주율, 오일러 자연상수 e)
# 파이썬은 실수를 지원하기 위해서 부동 소수형을 제공한다.

# * 부동 소수형의 특징
# 1. 8바이트만 이용해서 수를 표현함 (= 한정된 범위의 수만 표현이 가능)
# 2. 디지털 방식으로 소수를 표현해야 하므로 정밀도의 한계가 있다.

# e.g) 소수 43.2에서 소수 43.1을 빼면 얼마인지 파이썬으로 구현하시오.

print(43.2-43.1)

#0.1000000000000000142 가 나옴. 0.1이 아님.
# 부동 소수형은 위와 같이 정밀도의 한계를 가지고 있다. 이것은 비단 파이썬만의 문제는 아니다.
# 부동 소수형이 어떻게 구성되고 동작되는가를 정의하는 표준인 IEEE 754를 따르는 모든 컴퓨터 시스템의 문제이다.
# IEEE 754는 제한된 메모리를 이용해서 (파이썬의 경우 8바이트) 실수를 표현하기 때문에 제한된 정밀도를 갖게 된다.

###

#018 복소수형 자료 이해하기

#복소수는 크게 2가지로 나뉘어 진다.

        #실수                허수
        ↑                    ↑
#정수  유리수  무리수      i를제공하면 -1이 되는 수

#복소수는 실수부와 허수부로 되어있고, 허수부는 숫자뒤에 문자 i를 이용하는데 파이썬에서는 j를 사용한다.

#e.g)
c1 = 1 + 7j
    ↑  ↑
#실수형 ↑
    #허수형

print (c1.real) : #복소수형 자료에서 실수부만 취한다.
print (c1.imag) : #복소수형 자료에서 허수부만 취한다.
c2 = complex (2,3) # 실수부가 2이고 허수부가 3인 복소수를 만든다.

print(c2) # (2+3j)

###

#문제 52. 아래의 수학식을 파이썬으로 구현해서 답을 출력하시오.

#(1-2i ) 의 제곱 - 2*(1-2i) -12

```



```
c1 = complex(1,-2)
result = c1**2 - 2*c1 -12

print(result)
#%
```

# [ 019 대입 연산자 이해하기(=) ]

# 변수에 값을 대입하는데 사용하는 기호는 = 이다.  
 # 수학에서 =은 '=' 을 기준으로 왼쪽과 오른쪽이 같다는 의미.  
 # 하지만, 파이썬을 포함한 컴퓨터 프로그래밍 언어에서 '=' 은 [ '=' 왼쪽 변수에 오른쪽의 값을 대입하겠다는 뜻 ]  
 #e.g)

a = 7 #a는 7이다! 가 수학적 해석이라면, 프로그래밍에서는 'a라는 변수에 7이라는 값을 대입하겠다'는 뜻.

# 1. '=' : 대입연산자  
 # 2. '==' : 같다. (수학에서의 '=')  
 # 3. in : 여러개의 값을 비교

문제53. 주사위 1개를 10번 던지시오 !

```
import random

dice = [ 1, 2, 3, 4, 5, 6 ]
for i in range(1, 11): # 아래의 실행문을 10번 수행하겠다.
    result = random.choice(dice) # 실행문1
    print(result)
```

# 문제 54. 주사위 한개를 10번 던지시오.

```
import random #random이라는 파이썬 모듈을 지금 이 코드에서 생성하겠다.

#모듈 :
# ctrl 누른채로 random에 커서를 가져가면, 파란색으로 바뀜, 클릭해보면 모듈에 대한 상세 코드가 새창에 뜬.
# 파이썬의 좋은 점 중 하나. 누군가가 짜놓은 코드를 import 해와서 편하게 사용할 수 있다. (내가 복잡한 코드를 짤 필요가 없음)

dice=[1,2,3,4,5,6] #dice라는 변수에 1~6의 값을 대입하겠다.
for i in range(1,11): #for loop 문 아래의 실행문을 1이상 11미만 의 횟수만큼(즉 10번) 반복 수행하라.
    result = random.choice(dice) #결과라는 변수에 random 모듈을 활용해 dice 변수속 값을 선택한 결과를 대입하겠다.(실행문 1)
    print(result) #result라는 변수의 값을 출력하라. (실행문 2)
# print와 result은 같은 라인에 있다. for loop문의 명령에 따라 '순차적'으로 실행되어야 하기 때문.
# 만약 result와 print가 라인이 다르다면 값이 다르게 출력된다.
# 만약 print가 for i~와 같은 라인에 온다면, for loop문의 실행문이 아님. (for loop문에서 내린 명령에 영향을 받지 않음)
# 이 경우 for문이 10번 수행되고 난뒤에 나온 마지막 값을 출력함.
```

# 문제 55. 주사위를 10번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오.

```
import random

dice=[1,2,3,4,5,6] #주사위라는 변수에 1~6이라는 값을 각각 할당한다.
cnt=0 #횟수를 담기위한 변수를 cnt라는 이름으로 만들고 숫자 0을 할당한 것. (변수명은 cnt가 아니어도됨.)

for i in range(1,11):
    result=random.choice(dice) #for loop문의 실행문1
```

```

if result == 3: #for loop문의 실행문 2 #if문은 여기서 끝나기 때문에 끝을 맺어줘야 한다. (끝을 맺는 방법은 문장 끝에 콜론 ':' 입력)
    cnt=cnt+1 #if문의 영향권안에 들어가, if문의 실행문이 되어야 하기에 같은 라인이 아니고 1탭(or 4칸) 뒤에 cnt 실행문이 있는 것.
print(cnt)

###
실습!

import random

dice=[1,2,3,4,5,6]
cnt=0

for i in range(1,11):
    result=random.choice(dice)
    if result == 3:
        cnt=cnt+1
print(cnt)

###

#문제 55. 주사위를 10번 던져서 주사위의 눈이 짝수가 나오는 횟수를 출력하시오.

import random

dice=[1,2,3,4,5,6]
cnt=0

for i in range(1,11):
    result=random.choice(dice)
    if result%2 == 0:
        cnt=cnt+1
print(cnt)

#혹은

import random

dice=[1,2,3,4,5,6]
cnt=0

for i in range(1,11):
    result=random.choice(dice)
    if result in (2,4,6): #result의 결과가 2,4,6 중 하나가 있으면 // # 'in' 은 여러개의 값을 비교할때 사용한다!
        cnt=cnt+1 #cnt변수에 +1 해라
print(cnt)    # for loop 문이 모두 수행된 후에 cnt 변수가 가지고 있는 값을 출력하라.

###

# 문제 56. 주사위 2개를 동시에 던져서 두 눈의 합이 10이 되는 '횟수'를 출력하시오.(주사위 던지느 횟수 == 20번)

import random

dice_a=[1,2,3,4,5,6]
dice_b=[1,2,3,4,5,6]

cnt=0 #횟수를 담기 위한 cnt 라는 변수에 숫자 0을 할당한다.
    # 만약 cnt 에 아무것도 담고싶지 않으면(후에 뭐가 들어갈지 몰라서) cnt=None 를 쓰면 됨. (데이터 타입 상관없이 모두 받아들임)
for i in range(1,21): #주사위를 20번 던지기! 를 수행하기위한 loop(반복) 문을 만든 것.
    result_a=random.choice(dice_a) #실행문1
    result_b=random.choice(dice_b) #실행문2
    if result_a + result_b == 10:
        cnt=cnt+1 # if 문의 영향권 안에 있는 실행문
print(cnt) #if문의 영향권 안에 '없는' 실행문.
    #for 문과 같은 라인에 위치 = for문이 끝나면 print(cnt)문을 실행하라! 는 뜻.

```

#같은 라인 = 같은 레벨 // 우측으로 치우침 = 명령을 받는 실행문

###

# 문제 57. 주사위 2개를 동시에 던져서 두 눈의 합이 10이 되는 '확률'을 출력하시오. (주사위는 100번 던지시오)

import random

dice\_a=[1,2,3,4,5,6]

dice\_b=[1,2,3,4,5,6]

cnt=0

for i in range(1,101):

    result\_a=random.choice(dice\_a)

    result\_b=random.choice(dice\_b)

    if result\_a + result\_b == 10:

        cnt=cnt+1

print('주사위 2개를 동시에 100번 던져 눈의 합이 10이 될 확률은',cnt/100, '%입니다.')

###

# [ 020 사칙 연산자 이해하기(, -, \*, /, \*\*) ]

#오라클           vs           파이썬

# +                           +

# -                           -

# \*                           \*

# /                           /

# mod                        %

# power                      power or 숫자\*\*승수

# sprt (루트)                sqrt

# log2                       log2 -> 밑수가 2인 로그함수

# log10                      log10 -> 밑수가 10인 로그함수

# e.g) 밑수가 10이고 진수가 10인 log 값을 출력하시오. (log 함수를 사용하려면 math 모듈을 import 해야 한다.)

import math

print( math.log10(10) ) # math.log10(진수 값) 입력해준다.

# import math 라는 모듈을 쓰지 않으면 위의 결과는 출력되지 않는다.

###

# 문제 58. 아래의 수학적식을 파이썬으로 구현하시오.

#  $2 \times \log_2(10) + 1/3 \times \log_2(10)$

import math

print( 2\*math.log2(10) + 1/3\* math.log2(10) )

# 전체 식의 시작부분에만 math를 써주면 될 줄 알았는데, 모듈이 적용되는 것은 일반적인 숫자들이 아닌 log 이기때문에

# log 앞에 math. 을 붙여주는 게 맞다.

a= math.log2(10)

print (a)

# 변수에 여러개의 값을 담을 때는 대괄호를 쓴다. like dice = [1,2,3,4,5,6]

# 변수에 하나의 값을 담을 때는 괄호없이 '=' 할당 연산자만 사용한다.

###

# [ 021 연산자 축약 이해하기(=, -=, \*=, /=) ]

# 변수에 값을 사칙 연산하여 그 결과를 동일한 변수에 대입할 때, 연산자를 축약해서 대입할 수 있다.

#e.g)

# cnt = cnt + 1 을 아래와 같이 쓸 수 있다.

# cnt += 1

# e.g 2)

#original ver            #축약 ver

cnt = cnt + 1 -----> cnt += 1 # 과 같다. (실제로는 cnt+=1 처럼 붙여써야 인식된다. 공백 두면 인식못함)

cnt = cnt - 1 -----> cnt -= 1 # 과 같다.

cnt = cnt \* 2 -----> cnt \*= 2 # 과 같다.

cnt = cnt / 4 -----> cnt /= 4 # 과 같다.

# 파이썬은 간결함! 을 모토로 하고 있기에 나온 명령어들 중 하나. 쓰고 안쓰고는 본인 자유.

#%%

# 문제 59. 주사위를 10번 던져서 주사위눈이 3이 나오는 횟수를 출력하시오. (단 축약 연산자를 이용해서 출력하시오.)

import random

dice=[1,2,3,4,5,6]

cnt=0

for i in range(1,11):

    result=random.choice(dice)

    if result == 3:

        cnt +=1 # 요정도 공백은 괜찮음. 하지만 cnt += 1 처럼 다 떨어트려서 쓰면 인식 못함.

print(cnt)

#%%

#[ 022 True와 False 이해하기 ]

# 참을 나타내는 것이 True 이고, 거짓을 나타내는 것이 False이다.

# 조건을 판단해서 그 조건이 참이면 True, 거짓이면 False 를 리턴한다.

# True 는 1, False 는 0 이라는 값을 각각 가진다.

# True 와 False 로 표현하면 더 직관적이고 프로그램 코드의 가독성을 높일 수 있다. (권장!)

# e.g)

a=True

b=False

print(a)

print(1 == 1) # 라고 하면 결과값에 'True' 라고 나온다. '1은 1과 같다' 는 참이기 때문!

print(1 == 2) # 라고 하면 결과값에 'False' 라고 나온다. '1은 2와 같다' 는 거짓이기 때문!

#%%

# [ 023 관계 연산자 이해하기(==, !=, <, <=, >, >=) ]

#오라클            vs            파이썬

# >                            >

# >=                          >=

# <                            <

# <=                          <=

# =                            == -> 요거 하나 빼고는 다 오라클과 같음!

# !=                           !=

# in                           in

# 문제 60. 주사위를 20번 던져서, 주사위의 눈이 4이상 나오는 횟수를 출력하시오.

# '~이상' 이니까 >= 를 써야함

```
import random
```

```
dice=[1,2,3,4,5,6]
cnt=0
```

```
for i in range(1,21):
    result=random.choice(dice)
    if result >= 4:
        cnt+=1 # cnt=cnt+1
print(cnt)
```

```
###
```

# random.uniform(0,1) -> 0과 1사이의 난수를 출력하는 함수.

# 문제 61. (통계 정규분포 문제 연습) 주사위 1개를 288회 던져서 주사위의 눈이 5이상 나오는 '횟수' 를 구하시오.

```
import random
```

```
dice=[1,2,3,4,5,6]
cnt=0
```

```
for i in range (1,289):
    result=random.choice(dice)
    if result >=5:
        cnt = cnt+1 # or cnt+=1 (현업에선 이렇게 많이 쓰니 가급적 이렇게 쓰는게 좋음)
print(cnt)
```

```
###
```

# 문제 62. (통계 정규분포 문제 연습) 주사위 1개를 288회 던져서 주사위의 눈이 5이상 나오는 '확률' 을 구하시오.

```
import random
```

```
dice=[1,2,3,4,5,6]
cnt = 0
```

```
for i in range (1,289):
    result=random.choice(dice)
    if result >=5:
        cnt+= 1
print(cnt/288)
```

```
###
```

# 문제 63. (통계 정규분포 문제) 주사위 1개를 288회 던져서 주사위의 눈이 5이상 나오는 '횟수'를 출력하는데

# 이 작업을 100번해서 5이상 나오는 작업을 100번 수행하

```
import random
```

```
dice=[1,2,3,4,5,6]
```

```
for j in range(1,101):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    print(cnt)
```

```
###
```

# cnt 위 위치에 대하여...

```

#원래

# 만약 cnt 가 for j loop 문 위에 있다면?

import random

dice=[1,2,3,4,5,6]
cnt=0

for j in range(1,101):
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    print(cnt)

# cnt가 비워진 상태로 시작이 되어야 하는데 for i 이 288번 반복 수행되서 나온 결과값 + i 에서 dml 결과를 j loop 문에 의해서 100번 수행한 결과
# 가 누적되어서 입력된 것!

###

# 문제 64 (점심시간문제) 위에서 출력된 주사위를 굴려서 5이상의 수가 나오는 횟수 100개를 비어있는 리스트 a에 담고, a리스트의 개수를 출력하시오.

import random

dice=[1,2,3,4,5,6]
a= [ ] # 리스트를 담기위한 a 라는 변수(일종의 빈컵)을 만든다.

for j in range(1,101): #반복실행할 실행문을 100번 수행되게 한다. (cnt=0 부터~~~ a.append(cnt)까지)
    cnt=0 # 실행문1
    for i in range(1,289): #실행문2
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt) #실행문 3 // #for i loop 문이 288번 수행된 것 x100회 수행을 한 for j loop 문의 cnt 라는 값을 a 라는 변수에 append(입력) 한다.
print(len(a)) #a 변수에 담겨있는 값을 출력하는데 그 '개수' 만 출력한다. len(?) 는 괄호안의 변수에 들어있는 값의 '개수' 를 출력하라는 뜻!
    #오라클로 치면 count 같은 것.
    #print문이 for j loop와 같은 라인에 있는 이유는! for j ~ 에 소속된 실행문들이 '모두 수행된 이후 생성된 결과'를 출력해야 하기 때문.
    # for j loop과 동등한 level 이 아닌 for j 에 속하는 하위 level 이 되면 다른 값이 출력된다.

#위의 코드는 주사위를 288번 던져서 나오는 주사위의 눈중에 5이상의 눈이 나올 횟수 100개를 'a' 라는 리스트에 입력한 코드이다.

# 근데 5이상의 눈이 나올? 횟수 100개를 하면 주사위는 랜덤하게 288번 던지는 와중에 5이상 (5와 6 2개) 가 100번이 안채워지는 경우도 있을 텐데
#왜 매번 a 에는 100이라는 값이 채워지는가?

#실행순서
# 1.import random
# 2. dice=[ ~~~]
# 3. a = []
# 4. for j~~
# 5. print (len (a))

# for i ~ loop 내에서는 실행문 1~3이 '동시에' 수행된다. (단 순서대로! 즉, 굳이 따지자면 실행문 1 > 실행문 2 > 실행문 3으로!)

#들여쓰기를 잘해야 원하는 값을 얻을 수 있다! 동일한 level 인지 아닌지를 잘 구분해서 tab or 4spaces 를 잘 지켜줄 것!

###

```

# 문제 65. 동전을 100번 던져서 앞면이 나오는 '횟수'를 출력하시오.

```
import random

coin=['앞면','뒷면']
cnt=0

for i in range(1,101):
    result=random.choice(coin)
    if result == '앞면':
        cnt+=1
print(cnt)

###
```

#문제 66. 동전 한개와 주사위 한개를 동시에 100번 던져서 동전은 앞면이, 주사위의 눈은 5가 나오는 '횟수'를 출력하시오.

```
import random

coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]
cnt=0

for i in range(1,101):
    result_c=random.choice(coin)
    result_d=random.choice(dice)
    if result_c == '앞면' and result_d == 5:
        cnt=cnt+1
print(cnt)

###
```

# 문제 67. 동전 한개와 주사위 한개를 동시에 100번 던져서 동전은 앞면이, 주사위의 눈은 5가 나오는 횟수를 출력하는 행위를 50번해서 횟수가 50개가 '출력'되게 하시오.

```
import random

coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]

for j in range(1,51):
    cnt=0 #왜 cnt=0 이 여기에 있어야 하는가?!
    for i in range(1,101):
        result_c=random.choice(coin)
        result_d=random.choice(dice)
        if result_c == '앞면' and result_d ==5:
            cnt+=1
    print(cnt)

###
```

# 문제 68. 위의 횟수 50개를 비어있는 리스트 a list 에 '담으시오.'

# 출력 = print // # 담다 = append

```
import random

coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]
a=[ ]

for j in range(1,51):
```

```

cnt=0 #왜 cnt=0 이 여기에 있어야 하는가?!
for i in range(1,101):
    result_c=random.choice(coin)
    result_d=random.choice(dice)
    if result_c == '앞면' and result_d ==5:
        cnt+=1
a.append(cnt)
print(a) #print(a) 를 a.append(cnt) 와 같은 레벨에 두면 a list 에 cnt 의 결과가 담기는 과정을 볼 수 있다.
#지금은 a list 에 들어간 그 결과치를 보고 싶은 것이기에 for j와 같은 레벨에 print 를 두고 출력 한 것.

```

###

# 문제 69. 담겨진 a list 의 요소의 개수를 구하라.

```
import random
```

```

coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]
a=[ ]

```

```

for j in range(1,51):
    cnt=0
    for i in range(1,101):
        result_c=random.choice(coin)
        result_d=random.choice(dice)
        if result_c == '앞면' and result_d ==5:
            cnt+=1
    a.append(cnt)
print(len(a) )

```

# a에 담긴 리스트를 보면 다음과 같다.

```

#[14, 11, 6, 6, 5, 8, 7, 6, 10, 11, 8, 13, 5, 6, 7, 14, 9, 6, 6, 8, 6, 5, 4, 6,
# 9, 12, 7, 9, 6, 5, 12, 9, 10, 6, 7, 13, 9, 10, 6, 7, 6, 6, 6, 8, 9, 6, 8, 5, 3, 10]

```

###

# numpy 란?

# Deep learning 에 필요한 수학연산을 쉽게 구현하기 위해서 만들어 놓은 '모듈'

# 모듈 : 특정 목적을 위해 만들어둔 코드의 집합

# e.g) numpy를 사용하는 예제

# 아래의 리스트들의 요소들의 평균값을 출력하시오.

```
b=[7,4,5,3,2]
```

#평균 구하는 공식대로면  $7+4+5+3+2 / 5$  하면됨.

import numpy as np #numpy 모듈을 이 코드에서 사용하겠다.

#근데 numpy의 별칭을 np라고 하겠다.

```

b=[7,4,5,3,2]
print( np.mean(b)) #np. -> numpy 모듈을 사용해서
# mean(평균값 출력 모듈의 명령어) -> np.mean 으로 쓰이지 않으면 의미 없음!

```

###

#e.g 2 ) 아래의 리스트들의 요소들의 평균값을 구하시오 (numpy 사용하지 말고!)

```

b=[7,4,5,3,2]
cnt=0

```

```

for i in b:
    cnt= cnt+i
print(cnt/5)

```



```
# when i = 7 , cnt = 7
# when i = 4 , cnt = 7 + 4
# when i = 5 , cnt = 11 + 5
# when i = 3 , cnt = 16 + 3
# when i = 2 , cnt = 19 + 2
```

# 가 반복되어 최종 cnt = 21이 됨.

```
# or
```

```
b=[7,4,5,3,2]
sum(b)/5
```

```
###
```

# 문제 70. 위의 a list 요소들의 평균값을 출력하시오. (numpy를 활용해서 구현하시오)

```
import random
import numpy as np
```

```
coin=['앞면','뒷면']
dice=[1,2,3,4,5,6]
a=[ ]
```

```
for j in range(1,51):
    cnt=0
    for i in range(1,101):
        result_c=random.choice(coin)
        result_d=random.choice(dice)
        if result_c == '앞면' and result_d ==5:
            cnt+=1
    a.append(cnt)
print(np.mean(a) ) #평균구하기
print(np.var(a)) #분산구하기
print(np.std(a)) #표준편차 구하기
```

# 수학적인 계산은 존재하는 모듈로 수월하게 할 수 있다!

# 그러니 데이터 사이언티스트로서 내가 잘 해야 할 것은 데이터를 모으고 정제하는 작업!

```
###
```

# 문제 71. 문제 64번 코드를 가져와서 a list 속 요소들의 평균(np.mean) / 분산(np.var) /표준편차(np.std)를 구하시오.

# 문제 64번의 코드 = 주사위를 288번 던져 5의 눈이 나오는 횟수를 확인하는 작업을 100번 수행! 한 것을 a list 에 담고 a list 의 개수를 출력한 것!

```
import random
import numpy as np
```

```
dice=[1,2,3,4,5,6]
a=[ ]
```

```
for j in range(1,101):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt)
print(np.mean(a))
print(np.var(a))
print(np.std(a))
```

```
###
```

# 문제 72. 위의 a list 에 담긴 요소 100개를 출력하시오.

```
import random
```

```
dice=[1,2,3,4,5,6]
a= [ ]
```

```
for j in range(1,101):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt)
print(a)
```

```
#[102, 96, 98, 90, 100, 99, 102, 91, 103, 105, 91, 108, 112, 88, 99, 96, 91, 83, 94, 88, 100, 99, 105, 89, 98, 95, 102,
# 90, 102, 99, 99, 87, 106, 90, 91, 100, 88, 94, 109, 98, 93, 101, 97, 100, 92, 94, 97, 84, 92, 108, 95, 88, 91, 93, 89,
# 92, 104, 105, 96, 85, 92, 90, 91, 95, 92, 88, 83, 100, 90, 91, 99, 98, 86, 94, 89, 103, 105, 93, 96, 83, 93, 97, 93,
# 102, 91, 86, 79, 97, 98, 97, 98, 109, 95, 97, 103, 109, 111, 95, 107, 90]
```

# 맨 마지막에(for j loop와 같은 level에 둔것 의미) a list 를 print 하면 위와 같이 a list 에 있는 모든 요소들을 한번에 볼 수 있다.

# 이 요소들을 하나씩 고집어 내는 작업은 어떻게 하는가?!

```
#%%
```

#문제 73. 위의 a list 의 요소들을 하나씩 빼내서 출력하시오.

# Hint: for 문을 사용할 것!

```
import random
```

```
dice=[1,2,3,4,5,6]
a= [ ]
```

```
for j in range(1,101):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt)
```

```
for k in a: #for i loop 에서 a list 에 append 한 값들을 k로 불러온 것!
    print(k) #하나씩 출력되어서 결과가 나온다!
```

```
#%%
```

# 문제 74. a list 요소들의 숫자가 90이상이고 106 이하인 요소들의 갯를 구하시오.

```
import random
```

```
dice=[1,2,3,4,5,6]
a= [ ]
```

```
for j in range(1,101):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt)
```

```
cnt2=0
for k in a:
```

```

if k >= 90 and k <= 106: # OR if 90<= k <=106: 로 해도 됨.
    cnt2 = cnt2+1
print(cnt2)

```

# 글쓴다 생각하고 한단계 한단계씩 추가해서 구현하려고 해야지, 하나의 실행문에 모든걸 구현하려고 하면 너무 복잡해지고 어렵다.

```

###

```

# 문제 75. 그러면 문제 74의 경우가 발생하는 확률을 구출력하시오.

```

import random

```

```

dice=[1,2,3,4,5,6]
a= [ ]

```

```

for j in range(1,1001):
    cnt=0
    for i in range(1,289):
        result=random.choice(dice)
        if result >= 5:
            cnt+=1
    a.append(cnt)

```

```

cnt2=0
for k in a:
    if k >= 90 and k <= 106: # OR if 90<= k <=106: 로 해도 됨.
        cnt2 = cnt2+1
print(cnt2/1000) # OR print(cnt2/len(a)) 로 해도 됨.

```

```

###

```

[ 024 논리 연산자 이해하기(and, or, not) ]  
 [025 비트 연산자 이해하기(&, |, ~, ^, > >, < < )  
 실습과정 중에 충분히 숙지했고, 쉬운 개념이라 넘어감!

```

###

```

# [ 025 판다스(pandas) 사용법 ]

# Pandas

- # 1. 데이터 분석을 위한 파이썬 모듈
- # 2. 엑셀의 스프레드 시트와 같은 관계형 데이터베이스의 하나로 데이터 처리능력이 뛰어남.
- # 3. 판다스는 Dataframe 이라는 기본 자료구조를 사용한다.
- # (오라클의 테이블과 판다스의 Dataframe 이 서로 유사하다.)

# e.g) emp3.csv 를 파이썬을 이용해서 판다스의 데이터 프레임으로 만드시오.

카페에 올린 emp3.csv 와 dept3.csv 를 c드라이브 밑에 data라는 폴더를 만들어 옮겨둔다.

# Pandas

- # 1. 데이터 분석을 위한 파이썬 모듈
- # 2. 엑셀의 스프레드 시트와 같은 관계형 데이터베이스의 하나로 데이터 처리능력이 뛰어남.
- # 3. 판다스는 Dataframe 이라는 기본 자료구조를 사용한다.
- # (오라클의 테이블과 판다스의 Dataframe 이 서로 유사하다.)

# e.g) emp3.csv 를 파이썬을 이용해서 판다스의 데이터 프레임으로 만드시오.

import pandas as pd #판다스 모듈을 현 코드에서 사용하겠다!

```
emp=pd.read_csv("c:\\data\\emp3.csv") # c드라이브 밑에 emp3.csv 를 읽어서 emp 변수에 할당하겠다.
print(emp)
```

# 설명 : 판다스의 read\_csv 함수는 os 의 csv 파일을 읽어서 파이썬에서 판다스 데이터 프레임으로 만들겠다는 뜻.

```
###
```

```
   index empno  ename   job ...   hiredate  sal  comm deptno
0    1  7839  KING  PRESIDENT ... 1981-11-17 0:00 5000   NaN    10
1    2  7698  BLAKE  MANAGER ... 1981-05-01 0:00 2850   NaN    30
2    3  7782  CLARK  MANAGER ... 1981-05-09 0:00 2450   NaN    10
3    4  7566  JONES  MANAGER ... 1981-04-01 0:00 2975   NaN    20
4    5  7654  MARTIN SALESMAN ... 1981-09-10 0:00 1250 1400.0   30
5    6  7499  ALLEN  SALESMAN ... 1981-02-11 0:00 1600 300.0   30
6    7  7844  TURNER SALESMAN ... 1981-08-21 0:00 1500   0.0   30
7    8  7900  JAMES  CLERK ... 1981-12-11 0:00  950   NaN    30
8    9  7521  WARD  SALESMAN ... 1981-02-23 0:00 1250 500.0   30
9   10  7902  FORD  ANALYST ... 1981-12-11 0:00 3000   NaN    20
10  11  7369  SMITH  CLERK ... 1980-12-09 0:00  800   NaN    20
11  12  7788  SCOTT  ANALYST ... 1982-12-22 0:00 3000   NaN    20
12  13  7876  ADAMS  CLERK ... 1983-01-15 0:00 1100   NaN    20
13  14  7934  MILLER CLERK ... 1982-01-11 0:00 1300   NaN    10
```

```
[14 rows x 9 columns]
```

```
# 출력됨.
```

```
###
```

# SQL 을 배웠던 이유 : 데이터에서 정보를 얻어내기 위함 (데이터 검색력)

# 파이썬의 판다스도 데이터를 검색하는 모듈!

# 문제 76. (SQL -> 판다스 문법으로 바꾸기) 사원 테이블에서 이름, 월급을 출력하시오.

```
#SQL
```

```
select ename, sal
  from emp;
###
```

```
#Pandas
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp[ ['ename','sal'] ] )
```

#설명 : 데이터프레임명 [ ['컬럼명1', '컬럼명2'] ]

# 대괄호를 2개쓰는것에 주의할 것!

```
###
```

# 문제 77. 아래의 SQL을 판다스로 구현하시오.

```
SQL >
```

```
select ename, sal, job, hiredate
  from emp;
###
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp[ ['ename','sal','job','hiredate'] ] )
```

```
###
```

#문제 78. 아래의 SQL을 판다스로 구현하시오.

```
select ename, sal,
  from emp
 where job='SALESMAN'
```

```
###
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print(emp[['ename','sal']][emp['job']=='SALESMAN'])
```

```
###
```

# 문제 79. 월급이 3000이상인 직원들의 이름과 월급을 출력하시오.

```
# SQL
# select ename, sal
# from emp
# where sal=3000
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print(emp[['ename','sal']][emp['sal']>= 3000])
```

```
###
```

# 문제 80. 아래의 SQL을 판다스로 구현하시오.

```
# SQL
```

```
#select ename ,sal
#  from emp
#  where sal between 1000 and 3000;
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename','sal']][ emp['sal'].between(1000,3000)]) #검색조건 입력하는 부분에 ['sal'] 바로 뒤에 콤마 + between (범위시작점,범위끝점)
```

```
###
```

# 문제 81. 월급이 1000에서 3000사이가 아닌 직원들의 이름과 월급을 출력하시오.

```
#SQL
#select ename, sal
#  from emp
#  where sal not between 1000 and 3000;
```

# 판다스에서 not 은 '~(물결)'로 표시한다~!

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename','sal']][ ~emp['sal'].between(1000,3000)]) # emp 앞에 물결 (~) 을 하나 넣어주면 not between 과 같은 의미가 된다.
```

```
###
```

■ 오라클의 기타 비교 연산자와 파이썬 판다스와 비교

오라클	vs	판다스
1. between ~ and ~		emp['sal'].between (1000,3000)
2. in		emp['job'].isin ( ['SALESMAN','ANALYST'])
3. is null		emp['comm'].isnull()
4. like		apply 함수

```
###
```

# 문제 82. 직업이 SALESMAN, CLERK 인 직원들의 이름, 직업을 출력하시오.

```
#SQL
#select ename, job
#  from emp
```

```
# where job in('SALESMAN','CLERK');
```

#판다스

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp [['ename', 'sal']] [emp['job'].isin( ['SALESMAN','CLERK']) ] )
```

###

# 문제 83. 이번에는 직업이 CLERK, SALESMAN 이 아닌 직원들의 이름과 직업을 출력하시오.

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp [['ename', 'sal']] [~emp['job'].isin( ['SALESMAN','CLERK']) ] ) #문제 82와 동일한 코드에 emp 앞에 물결을 붙여주었다.
```

###

# 문제 84. 커미션이null 인 직원들의 이름, 커미션을 출력하시오.

```
select ename, comm
from emp
where comm is null;
###
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp [['ename', 'comm']] [emp['comm'].isnull () ] )
```

#오라클의 null 은 파이썬의 NaN 과 같다.

###

# 문제 85. 커미션이 null이 아닌 직원들의 이름과 커미션을 출력하시오.

```
select ename, comm
from emp
where comm is null;
###
```

```
import pandas as pd
emp=pd.read_csv("c:\\data\\emp3.csv")
print (emp [['ename', 'comm']] [~emp['comm'].isnull () ] )
```

```
# print (데이터프레임명 [['컬럼명1', '컬럼명2']] [~emp['where절에들어가는 조건명'].isnull () ] )
```

###

# (오늘의 마지막 문제) - 확률 문제

#문제 86. 6개의 제품이 들어있는 상자가 있는데 그 중에 2개가 불량품이라고 하자, 제품검사를 위해서 3개를 추출했을때 적어도 1개가 불량품일 확률을 구하시오.

# (복원사건이다. 6개중 1개뽑아서 확인 후 다시 넣고, 2트에서 6개중 1개 뽑아서 또 확인!)

# 수학적식이 아닌 파이썬으로 아래와 같이 상자안에 제품 6개를 넣고 랜덤으로 추출해서 수행하시오.

# 사건 A = 전체 6개의 제품중 3개를 뽑았는데 그중에 적어도 1개가 불량품인 것

# 힌트 사건 A의 여사건 = 전체 사건 중 사건 A를 제외한 나머지 사건

# 힌트 사건 A 여사건의 확률 = 1-P(A)

# 즉 1 - (불량품이 하나도 없을 확률)

```
import random
```

```
box=['정상','정상','정상','정상','불량','불량']
```

```

cnt=0
for j in range(1,10001):
    result_1=random.choice(box)
    result_2=random.choice(box)
    result_3=random.choice(box)
    if result_1 == '정상' and result_2 == '정상' and result_3 == '정상':
        cnt+=1
print(1-cnt/10000 )

```

20.11.30 파이썬강의 Day4 (문제86~문제121)

## ■ 복습

" 파이썬은 괄호 대신에 들여쓰기로 블록을 구분한다 "

↓  
실행영역

예: for i in range(1, 11):  
     print('a') # <-- for loop 문의 관찰하에 있는 실행영역이다.  
     print('b') # <-- for loop 문의 관찰하에 있는 실행영역이 아닙니다.  
                 # for loop 문이 다 끝나고 나면 작동되는 문장

SQL 과 파이썬의 차이점?

1. SQL은 비절차적 언어 : 3 select  ename, sal  
                           1 from emp  
                           2 where  ename='SCOTT';
2. 파이썬은 절차적 언어 : 코드의 흐름대로 절차적으로 실행한다.

```

for i in range(1, 10):
    print('a')
print('b')
for k in range(1, 100):
    print('c')

```

## 1. 파이썬 기본 자료형 5가지

1. 문자형 : a ='scott'
2. 숫자형 : cnt = 0
3. 리스트형 : a = [ 1, 2, 3 ] 대괄호
4. 튜플 : b = (1, 2, 3 ) 소괄호
5. 딕셔너리 : c = { 'apple' : '사과', 'eye' : '눈' } 중괄호  
                   ↑    ↑  
                   키   값

2. 파이썬 연산자 : >, <, >=, <=, ==, !=

3. if 문 :   if 조건 :  
           실행문

4. loop 문 : 프로그램을 실행하다 보면 반복적으로 특정 실행문을  
           실행해야할 때가 있다. 그때 loop 문을 사용해야한다.

```

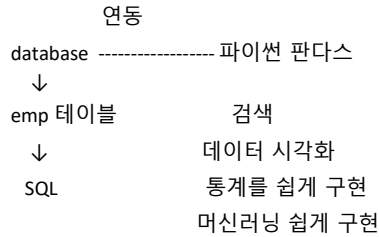
for loop 문
예: for i in range(1, 11):
    print('a')           i = 1 일때 print('a')
                       i = 2 일때 print('a')
                       :
                       i =10 일때 print('a')

while loop 문

```

while 조건 :  
실행문

5. 판다스 사용법 : 판다스는 엑셀과 같은 데이터를 검색할때 주로 사용



"현업에서 주로 이렇게해서 분석을 많이 한다."

아직까지는 DATABASE 와 연동을 하지는 않았고 emp.csv 를 판다스로  
읽어들여서 검색을 했다.

문제86. 이름이 SCOTT 인 사원의 이름과 월급을 출력하시오 !

```
SQL> select ename, sal
      from emp
      where ename='SCOTT';
```

```
pandas>
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename','sal'] ] [ emp['ename']=='SCOTT' ] )
```

## ■ 26. 시퀀스 자료형 이해하기

시퀀스 자료형은 어떤 객체가 순서를 가지고 나열되어 있는 것을  
말한다. 예를들어 문자열 'abcd' 는 문자 a, b, c, d 가 순서를  
가지고 차례대로 나열되어 있는 것이다.

예:

```
a = 'scott'           s c o t t
print (a) # scott     0 1 2 3 4
```

위의 scott 이라는 문자를 담은 a 변수에서 첫번째 요소만 출력하고 싶다면?

```
print ( a[0] ) # s
print ( a[1] ) # c
```

문제87. 위의 scott 을 담은 문자형 변수 a 에서 알파벳 o 를 출력하시오 ~

```
a = 'scott'
print (a[2])
```

설명: 문자형 변수에서 특정 철자만 가져오고 싶으면

문자형변수 옆에 대괄호를 쓰고 순서에 해당하는 숫자를  
적어주면 된다.

문제88. 아래의 문자형 변수에서 맨 끝의 철자인 h 를 출력하시오 !

```
b = 'smith'           s m i t h
print ( b[4] )        0 1 2 3 4
print ( b[-1] )       -5 -4 -3 -2 -1
```

문제89. 판다스를 이용해서 emp3.csv 에서 이름을 출력하는데



이름의 첫번째 철자만 출력하시오 !

```
SQL> select substr(ename, 1, 1)
      from emp;
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print( emp['ename'] )
print( type(emp['ename']) ) # <class 'pandas.core.series.Series'>
```

↓

오라클로 치면 컬럼 이다.

설명: type 명령어는 변수가 문자형, 숫자형, 리스트형, 딕셔너리형, 튜플형인지 확인하는 함수 이다.

```
for i in emp['ename']: # emp3.csv 에서 ename 만 가져와서
    print ( type(i) )      # ename 의 갯수만큼 loop 문을 수행해서
                          # ename 을 출력한다.
```

```
a = 'scott'
print( type(a) ) #<class 'str'>
print ( a[0] )
```

설명 : <class 'str'> 문자형으로 변경해줘야 문자형에서 특정 요소를 가져오는 문법을 사용할 수 있다.

예: a[0]

판다스의 <class 'pandas.core.series.Series'> 를 문자형으로 변경하려면 for loop 문을 이용해서 하나씩 가져오면 된다.

```
예: for i in emp['ename']:
    print (i)
```

문제90. 판다스를 이용해서 emp3.csv 를 가져와서 이름의 끝글자를 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print ( i[-1] )
```

문제91. 위의 결과에서 앞에 이름도 같이 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print ( i, ' ', i[-1] )
```

문제92. 이름의 첫번째 철자가 s 로 시작하는 직원들의 이름을 출력하시오 !

```
SQL> select ename
      from emp
      where ename like 'S%';
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    if i[0] =='S':
        print( i )
```

문제93. 이름의 끝글자가 T 로 끝나는 직원들의 이름을 출력하시오 !

```
SQL> select ename
      from emp
```

```
where ename like '%T';
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    if i[-1] == 'T':
        print(i)
```

#### ■ 27. 시퀀스 자료 인덱싱 이해하기

인덱싱(indexing)이란 시퀀스 자료형에서 인덱스를 통해 해당하는 값을 얻는 방법이다. 파이썬에서는 인덱스를 0 부터 시작하며 음수인 인덱스도 사용가능하다. 음수 인덱스는 '끝에서 몇 번째' 라는 의미를 같습니다.

예: a = 'SALEMAN'

```
print ( a[2] ) # L 이 출력됨
```

문제94. 이름의 두번째 철자가 M 인 사원의 이름을 출력하시오

```
SQL> select ename
      from emp
      where ename like '_M%';
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    if i[1] == 'M':
        print(i)
```

#### ■ 28. 시퀀스 자료 슬라이싱 이해하기

인덱싱은 인덱스에 해당하는 요소 하나를 취하는 방법이지만 슬라이싱은 시퀀스 자료에서 일정 범위에 해당하는 부분을 취하는 방법이다.

e.g): a = 'scott'                    s c o t t  
      print ( a[1:3] )            0 1 2 3 4  
      설명: 1번째 자리부터 3미만의 자리까지 요소들을 출력

문제95. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select substr(ename, 1, 3 )
      from emp;
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print(i[0:3] )
```

문제96. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select substr(ename, -2, 2)        K I N G
      from emp;                        -2 -1
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print(i[-2: ])        K I N G
    ↑                    -2 -1
```

콜론(:) 다음에 아무것도 안넣으면 그냥 끝까지 읽어라라는 뜻이다.

문제97.( 점심시간 문제) 아래의 SQL을 판다스로 구현하시오 !

답글 달고 식사하러 가시오. ~~

```
SQL> select substr( ename, 2, 3)
      from emp;
```

SQL>

#### ■ 29. 시퀀스 자료 연결 이해하기(+)

자료형이 동일한 두개의 시퀀스 자료는 + 연산자로 순서있게 연결하여 새로운 시퀀스 자료를 만들 수 있다.

문자열 + 문자열, 리스트 + 리스트, 튜플 + 튜플 과 같이 두개의 동일한 시퀀스 자료형에 대해 '+' 연산자로 연결이 가능하다.

```
e.g): a = 'i love '
      b = 'python'
      print ( a + b) # i love python
```

문제98. 아래의 두개의 리스트를 연결하여 출력하시오..

```
a = [ 2, 3, 4, 5 ]
b = [ 9, 2, 4, 8 ]

print ( a + b )

[2, 3, 4, 5, 9, 2, 4, 8]
```

문제99. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select ename || sal
      from emp;
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
```

```
for i , k in zip(emp['ename'], emp['sal'] ):
    print ( i + str(k) )
```

설명: zip 함수를 for loop 문에 사용하게 되면 두개의 범위 데이터를 한번에 받아서 loop 를 수행할 수 있다.

str 함수를 사용한 이유는 문자형 + 문자형을 연결이 가능한데 문자형 + 숫자형은 연결이 안된다. 그래서 숫자형을 문자형으로 변환하기 위해서 str 함수를 사용했다.

#### ■ 30. 시퀀스 자료 반복 이해하기(\*)

동일한 시퀀스 자료를 반복하여 새로운 시퀀스 자료로 만들고자 하면 별표(\*) 를 연산자로 사용한다.

e.g): print ( 'a' \* 7 ) # 소문자 알파벳 a 를 7번 출력한다.

결과: aaaaaaa

```
print( [1, 2, 3 ] * 5 ) # 리스트 [ 1, 2, 3] 안의 요소를 5개로 늘려서 출력
```

결과: [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]

문제100. 주사위의 눈 6개를 100개 담은 리스트 dice100 을 만들고

dict100 리스트를 출력하시오 !

```
dice100 = [ 1, 2, 3, 4, 5, 6 ] * 100
print (dice100)
```

문제101. 초등학생 키가 10개가 들어있는 아래의 tall 리스트의 요소 10개를  
10000개로 증가시켜서 tall10000 리스트에 담고 출력하시오 !

```
tall = [ 129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
```

```
tall10000 = tall*10000
print (tall10000)
```

문제102. 위의 모집단의 평균값, 분산, 표준편차를 출력하시오 !  
( numpy 를 사용하시오. ~~)

```
import numpy as np
```

```
tall = [ 129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
```

```
tall10000 = tall*10000
print(tall10000)
print ( type(tall10000) ) # <class 'list'>
```

```
print( np.mean(tall10000) ) # 136.66 모집단의 평균
print( np.var(tall10000) ) # 23.782399999999985 모집단의 분산
print( np.std(tall10000) ) # 4.876720209321013 모집단의 표준편차
```

설명: np.mean(리스트) 를 사용하면 평균값이 출력된다.

문제103. 위의 모집단 tall10000 에서 표본을 20개를 랜덤으로 추출하시오 !

```
import numpy as np
import random
```

```
tall = [ 129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
```

```
tall10000 = tall*10000 # 초등학생 키 데이터 10000 개 생성(모집단 생성)
```

```
for i in range(1, 21): # 20번 반복 하면서
    print( random.choice( tall10000) ) # 모집단에서 랜덤으로 키하나를 추출
```

문제104. 위의 랜덤으로 추출한 표본 20개를 비어있는 a 리스트에 담으시오 !

```
import numpy as np
import random
```

```
tall = [ 129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall*10000 # 초등학생 키 데이터 10000 개 생성(모집단 생성)
a = [] # 표본을 담기 위해 a 라는 비어있는 리스트를 만듭니다.
```

```
for i in range(1, 21): # 20번 반복 하면서
    a.append( random.choice( tall10000) ) # 모집단에서 랜덤으로 키하나를
    # 추출해서 a 리스트에 담는다.
```

```
print(a)
```

문제105. 위의 표본 20개의 평균값을 출력하시오 !

```
import numpy as np
import random
```

```
tall = [ 129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall*10000 # 초등학생 키 데이터 10000 개 생성(모집단 생성)
a = [] # 표본을 담기 위해 a 라는 비어있는 리스트를 만듭니다.
```

```

for i in range(1, 21): # 20번 반복 하면서
    a.append( random.choice( tall10000) ) # 모집단에서 랜덤으로 키하나를
        # 추출해서 a 리스트에 담는다.

print(np.mean(a)) # 136.365

```

### ■ 31. 시퀀스 자료 크기 이해하기(len)

모든 시퀀스 자료는 고정된 길이 또는 크기를 가지고 있다.  
 시퀀스 자료의 크기는 시퀀스 자료를 구성하는 요소의 갯수 이다.

```

e.g): a = 'scott'
      print ( len(a) ) # 5

      b = [ 2, 3, 4, 8, 1 ]
      print ( len(b) ) # 리스트의 요소의 갯수가 출력된다.

```

문제106. 아래의 SQL을 판다스로 구현하시오 !

```

SQL> select ename, length(ename)
      from emp;

```

```

pandas>
import pandas as pd

```

```

emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print ( i, len(i) )

```

### ■ 32. 멤버체크 이해하기(in)

in 은 자료에 어떤 값이 있는지 없는지 확인할 때 사용하는 키워드이다.  
 활용하는 방법은 다음과 같습니다.

```

e.g): 값 in 자료
설명: 값이 자료에 있으면 True 이고 없으면 False 가 된다.

```

```

e.g): listdata = [ 1, 2, 3, 4 ]
      a = 5 in listdata
      b = 4 in listdata
      print (a) # False
      print (b) # True

```

문제107. 모평균이 30이고 모표준편차가 7인 모집단을 구성하시오 !

```

import numpy as np

```

```

avg = 30
std = 7
N = 1000000

```

```

mogipdan = np.random.randn(N) * std + avg
print ( len( mogipdan ) )

```

설명: np.random.randn(숫자) 를 쓰면 이 숫자만큼 가우시안 정규분포에  
 따르는 난수들이 숫자만큼 생성된다.

```

[28.92911183 31.75287259 28.78657524 ... 31.03549309 30.59468915
 26.79542575]

```

문제108. 위의 모집단의 모평균을 출력하시오 !

```

import numpy as np

```

```

avg = 30
std = 7

```

```
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg  
print ( np.mean( mogipdan ) )
```

```
# 30.00016957084103
```

문제109. 아래의 SQL 을 판다스로 구현하시오 !

```
SQL> select  ename, job  
      from emp  
      where job in ('SALESMAN', 'ANALYST');
```

```
Pandas>  
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")  
print ( emp[ ['ename', 'job']] [ emp['job'].isin( ['SALESMAN', 'ANALYST']) ] )
```

문제110. 아래의 SQL을 판다스로 구현하시오!

```
SQL> select  ename, job  
      from emp  
      where job not in ('SALESMAN', 'ANALYST');
```

```
Pandas>  
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")  
print ( emp[ ['ename', 'job']] [ ~ emp['job'].isin( ['SALESMAN', 'ANALYST']) ] )
```

### ■ 33. 문자열 이해하기

문자열은 문자나 기호 순서로 나열되어 있는 시퀀스 자료 이다.

※ 문자열을 선언하는 방법 3가지

1. '문자열' <----- 싱글 쿼테이션 마크
2. "문자열" <----- 더블 쿼테이션 마크 한개
3. """ 문자열 """ <----- 더블 쿼테이션 마크를 3개

문자에 아래와 같이 싱글 쿼테이션 마크(') 가 포함되어져 있으면 위의 두번째인 더블 쿼테이션 마크를 사용하면 된다.

```
a = "My son's name is John"  
print (a)
```

문자에 아래와 같이 더블 쿼테이션 마크가 있으면 위의 세번째인 """ """ 더블 3개를 사용하면된다.

```
b = """ My son's name is "jone" """  
print (b)
```

문제111. 아래와 같은 글씨가 출력되게하시오 !

"모집단"의 모평균은  
"모집단"의 모분산은  
"모집단"의 모표준편차는

답:

```
a = """ "모집단"의 모평균은 """  
b = """ "모집단"의 모분산은 """  
c = """ "모집단"의 모표준편차는 """
```

```
print (a)  
print (b)
```

```
print (c)
```

문제112. 모평균이 30 이고 모표준편차가 7인 모집단(1000000개)의 모평균과 모분산과 모표준편차를 아래와 같이 출력하시오 !

```
"모집단"의 모평균은 30.000019293
"모집단"의 모분산은 49.00192922
"모집단"의 모표준편차는 7.000191929
```

답:

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
```

```
a = "" "모집단"의 모평균은 ""
b = "" "모집단"의 모분산은 ""
c = "" "모집단"의 모표준편차는 ""
print (a , np.mean(mogipdan) )
print (b, np.var(mogipdan) )
print (c, np.std(mogipdan) )
```

결과

```
"모집단"의 모평균은 30.0002577581802
"모집단"의 모분산은 49.033137737571856
"모집단"의 모표준편차는 7.002366581204662
```

문제113.모평균이 30 이고 모표준편차가 7인 모집단(1000000개)에서 표본을 49 개를 뽑으시오 ~

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
print ( np.random.choice( mogipdan , 49) )
```

문제114. 위에서 뽑은 49개의 평균값을 출력하시오 !

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
```

```
print ( np.random.choice( mogipdan , 49).mean() ) # 29.26777626361961
```

설명: numpy 의 random 안의 choice 함수를 이용해서 mogipdan에서 49개를 표본샘플링하고 그 표본의 평균값을 출력한다.

문제115. 위에서 출력한 표본의 평균값을 하나가 아니라 100개가 출력되게하시오 !

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
```

```
for i in range(1, 101):
    print ( np.random.choice( mogipdan , 49).mean() )
```

문제116. 이번에는 100개를 출력하지 말고 비어있는 리스트 a를 선언하고  
a에 100개를 담으시오 !

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
a = []
for i in range(1, 101):
    a.append( np.random.choice( mogipdan , 49).mean() )
```

```
print(a)
```

문제117. 위에서 구한 표본평균의 평균값과 표준편차를 아래와 같이  
출력하시오 !

"표본평균" 의 평균값은 ?

"표본평균" 의 표준편차는 ?

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
a = []
for i in range(1, 101):
    a.append( np.random.choice( mogipdan , 49).mean() )
```

```
print(""" "표본평균" 의 평균값은 """, np.mean(a))
print(""" "표본평균" 의 표준편차값은 """, np.std(a) )
```

```
"표본평균" 의 평균값은 29.967904550298996
"표본평균" 의 표준편차값은 1.0408616938518351
```

#### ■ 34. 문자열 포매팅 이해하기

문자열 포매팅이란 변하는 값을 포함하는 문자열을 표현하기 위해  
하나의 양식으로 문자열을 만드는 것이다.  
문자열 포매팅에서는 변하는 값을 나타내기 위해 사용하는 기호를  
'포맷 문자열' 이라고 하며, 자주 사용하는 포맷 문자열은 다음과 같습니다.

포맷 문자열	설명
%s	문자열에 대응됨
%c	문자나 기호 한개에 대응됨
%f	실수에 대응됨
%d	정수에 대응됨
%%	% 라는 기호 자체를 표시함

```
e.g): txt1='자바'
      txt2='파이썬'
      num1 = 5
      num2 = 10
      print( '나는 %s 보다 %s 에 더 익숙합니다' %(txt1,txt2) )
```



```
print('나는 ', txt1, ' 보다 ', txt2, '에 더 익숙합니다')
```

나는 자바 보다 파이썬 에 더 익숙합니다

나는 자바 보다 파이썬 에 더 익숙합니다

+ 추가

f-string 포매팅 (f 문자열 포매팅)

문법 : print( f'문자열{ 변수이름 or 출력하고 싶은 것 입력 }')

f'문자열{변수} 문자열 형태

이때, {} 안에 들어갈 값을 사전에 미리 지정해둘수도, 직접 문자열을 입력할수도 있다.

e.g)

```
month = 1
```

```
while month <= 12:
```

```
    print( f'2020년 {month} 월')
```

```
    month = month + 1
```

문제118. 아래의 변수를 이용해서 아래와 같이 결과가 출력되게하시오 !

```
num1 = 5
```

```
num2 = 10
```

결과: 5 는 10 보다 작습니다.

답:

```
num1 = 5
```

```
num2 = 10
```

```
print( '%d 는 %d 보다 작습니다' %(num1, num2) )
```

문제119. 문제117번의 결과가 아래와 같이 출력되게하시오 !

( 문자열 포맷을 이용하시오.)

표본평균의 평균값은 29.967904550298996 이고 분산값은 29.967904550298996

이고 표준편차는 1.0058238308290384 이다.

```
import numpy as np
```

```
avg = 30
```

```
std = 7
```

```
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
```

```
a = []
```

```
for i in range(1, 101):
```

```
    a.append( np.random.choice( mogipdan , 49).mean() )
```

```
print( '표준평균의 평균값은 %f 이고 분산값은 %f 이고 표준편차값은 %f 입니다'
```

```
      %(np.mean(a), np.var(a), np.std(a) ) )
```

표준평균의 평균값은 30.056687 이고 분산값은 0.776711 이고 표준편차값은 0.881312 입니다

문제120. 위의 결과가 아래와 같이 소수점 두번째 자리까지만 나오게 하시오 !

표준평균의 평균값은 30.06 이고 분산값은 0.78 이고 표준편차값은 0.89 입니다

```
print( '표준평균의 평균값은 %f 이고 분산값은 %f 이고 표준편차값은 %f 입니다'
```

```
      %( round(np.mean(a),2) , round(np.var(a),2) , round(np.std(a),2) ) )
```

표준평균의 평균값은 30.240000 이고 분산값은 0.840000 이고 표준편차값은 0.920000 입니다

```
a = 3939.292929
print( a[2:4] )
```

```
import numpy as np
```

```
avg = 30
std = 7
N = 1000000
```

```
mogipdan = np.random.randn(N) * std + avg
a = []
for i in range(1, 101):
    a.append( np.random.choice( mogipdan , 49).mean() )
```

```
print ( '표준평균의 평균값은 %.2f 이고 분산값은 %.2f 이고 표준편차값은 %.2f 입니다'
        %( round(np.mean(a),2) , round(np.var(a),2) , round(np.std(a),2) ) )
```

표준평균의 평균값은 30.04 이고 분산값은 1.25 이고 표준편차값은 1.12 입니다

문제121. ( 통계를 코드로 구현 문제4번의 링크2) 오늘의 마지막 문제

```
import numpy as np
```

```
mean=18
std=3
N=100000
passenger=np.random.randn(N)*std+mean
cnt=0
```

```
a=[]

for i in range(1,10001):
    result=np.random.choice(passenger,36).mean()
    if result >17:
        cnt=cnt+1
print(cnt/10000)
```

20.12.01 파이썬강의 Day5 (문제122~문제147)

#### ■ 파이썬 복습

1. 파이썬의 자료형 5가지 : 문자형, 숫자형, 리스트형, 튜플형, 사전형
2. 파이썬의 연산자: >, <, >=, <=, =, !=
3. 파이썬에서의 들여쓰기의 용도와 중요성

다른 자바나 c 언어는 괄호로 특정 블록을 지정하는데  
파이썬은 들여쓰기로 특정 블록(실행영역) 을 지정한다.

```
for i in range(1,11):
    print ('a') #for loop문의 관할하에 있는 실행영역
print ('b') #for loop 문과 상관없는 별도의 독립적인 실행영역
```

#### 4. if 문 사용법

```
if 조건:
    실행코드
elif 조건:
    실행코드
elif 조건:
    실행코드
else:
```

## 실행코드

5. 반복문 : for loop 문, while loop 문

6. 판다스 사용법 : 데이터 검색, 시각화, 통계와 머신러닝 구현을 쉽게  
해주는 파이썬의 모듈

7. 문자열 변수에서 특정 철자를 취하는 방법(인덱싱)

```
a='scott'  
print(a[0])
```

1교시 신호 보냈습니다.

8. 문자열 변수에서 특정 철자들을 취하는 방법(슬라이싱)

```
a='scott'  
print(a[0:2]) # sc
```

9. 문자열 변수의 길이 확인하는 방법(len)

```
a='scott'  
print(len(a)) # 5
```

10. 멤버 체크 하는 방법(in)

```
a=[1, 2, 3, 4, 5]  
b=3 in a  
print(b) # True
```

11. 문자열 포맷 이해하기

```
print('표준평균의 평균값은 %.2f 이고 분산값은 %.2f 이고 표준편차값은 %.2f 입니다'  
      %(round(np.mean(a),2), round(np.var(a),2), round(np.std(a),2) ) )
```

표준평균의 평균값은 30.04 이고 분산값은 1.25 이고 표준편차값은 1.12 입니다

### ■ 35. 이스케이프 문자 이해하기

이스케이프 문자는 키보드로 입력하기 어려운 기호를 나타내기 위해

역슬래쉬 '\'로 시작하는 문자이다.

파이썬에서 자주 사용되는 이스케이프 문자는 다음과 같습니다.

이스케이프 문자	설명
\n	줄바꾸기
\t	탭
\엔터	줄 계속
\\	\ 기호자체
' 또는 \"	' 기호 또는 " 기호 자체

```
print('나는 파이썬을 배운다.\n파이썬은 자바보다\  
훨씬 쉽습니다')
```

나는 파이썬을 배운다.

파이썬은 자바보다 훨씬 쉽습니다

### ■ 36. 리스트 이해하기 ([ ])

리스트는 파이썬에서 가장 많이 활용되는 시퀀스 자료형 중 하나이다.

리스트는 [ ]로 표시하며 [ ] 안에 요소를 콤마(,)로 구분하여 순서있게 나열한다.

e.g): k=[ 'a', 'b', 'c', 'd', 'e' ]

```

0 1 2 3 4
print ( k )
print ( type(k) ) # <class 'list'>
print ( k[1] ) # b

```

문제122. 동전의 앞면과 뒷면을 포함하는 리스트 coin 을 만드시오 !

```
coin = ['앞면', '뒷면']
```

문제123. 위에서 만든 coin 의 요소를 10000개로 늘려서 coin\_10000 변수에  
담으시오 !

```

coin = ['앞면', '뒷면']
coin_10000 = coin * 10000
print (coin_10000)

```

문제124. 위에서 만든 coin\_10000 리스트에서 표본을 10개를 랜덤 추출하시오 !

```

import numpy as np

coin = ['앞면', '뒷면']
coin_10000 = coin * 10000
print ( np.random.choice( coin_10000, 10) )

```

설명: np.random.choice 은 numpy 모듈안에 random 코드 안에 choice 함수를  
사용하겠다. np.random.choice( 모집단 리스트 , 샘플 갯수 )

```
['앞면','뒷면','뒷면','앞면','뒷면','앞면','앞면','앞면','뒷면','앞면']
```

문제125. 위에서 추출한 샘플 10개에서 앞면이 몇번 나오는지 출력하시오 !

```

import numpy as np

coin = ['앞면', '뒷면']
coin_10000 = coin * 10000
result = np.random.choice( coin_10000, 10)
print( type(result) )
cnt = 0
for i in result: # ['앞면','뒷면','뒷면','앞면','뒷면','앞면','앞면','앞면','뒷면','앞면']
    if i == '앞면':
        cnt = cnt + 1

print (cnt)

```

### ■ 37. 튜플 이해하기(( ))

튜플은 리스트와 비슷한 성질을 가지고 있는 자료형이지만  
요소의 값을 변경할 수 없다는 특징이 있다.  
리스트는 대괄호 [ ] 로 요소들을 감쌌는데 튜플은 소괄호 ( ) 로  
요소를 감쌌습니다.  
튜플은 데이터가 변경이 안되므로 튜플로 만든 데이터에 대한 신뢰도가  
높아집니다.

e.g): 리스트에서 특정 요소를 변경하는 방법

```

a = [ 1, 2, 3, 4, 5 ]
print ( type(a) ) # <class 'list'>
print( a[0] ) # 1

```

a[0] = 7 # a리스트의 0번째 요소를 7로 변경한다.  
print (a) # [ 7, 2, 3, 4, 5 ]  
          ↑  
          a 리스트의 0 번째 요소가 7로 변경이 되었다.

문제126. a 리스트에 인덱스 번호 3번째 요소를 17로 변경하시오 !

```
[ 7, 2, 3, 4, 5 ]
0 1 2 3 4 <-- 인덱스 번호
```

```
a[3] = 17
```

e.g)2: 튜플은 데이터를 변경할 수 없습니다.

```
b = ( 1, 2, 3, 4, 5 )
print ( b[0] ) # 1
b[0] = 7 # 에러가 나면서 실행되지 않습니다.
```

TypeError: 'tuple' object does not support item assignment

설명: 회사에서 변경이 되어서는 안되는 데이터는 프로그래밍할 때  
튜플로 만들어서 관리하면 된다.

신세계 백화점, 이마트에서 사용하는 포인트 적립 카드에서  
포인트 적립시 구매금액의 0.01 % 로 적립을 해준다고 하면  
이 0.01 은 절대로 프로그램에서 변경이 되어서는 안되는 데이터 이므로  
튜플로 관리해야한다.

문제127. 아래의 숫자 데이터들을 튜플로 생성하시오 !

튜플 변수 이름은 point 라고 하시오. !

```
0.01, 0.02, 0.03, 0.04, 0.05

point = ( 0.01, 0.02, 0.03, 0.04, 0.05 )
print ( type(point) ) # <class 'tuple'>
```

문제128. 위의 튜플 point 의 요소중 0.03 만 뽑아서 출력하시오 !

```
point = ( 0.01, 0.02, 0.03, 0.04, 0.05 )
print ( type(point) )
print ( point[2] )
```

문제129. 위의 튜플의 모든 요소를 다 출력하시오 !

결과:

```
0.01
0.02
0.03
0.04
0.05
```

```
point = ( 0.01, 0.02, 0.03, 0.04, 0.05 )
```

```
for i in point: # ( 0.01, 0.02, 0.03, 0.04, 0.05 )
    print ( i )
```

```
for j in 'scott':
    print ( j )
```

### ■ 38. 사전 이해하기({})

사전은 키와 값을 하나의 요소로 하는 순서가 없는 집합이다.  
그러므로 사전은 시퀀스 자료형이 아니며 인덱싱으로 값을 접근 할 수도  
없습니다. 사전은 '키:값' 쌍이 하나의 요소이다.

```
예: a = { 'apple': '사과', 'banana': '바나나', 'peach': '복숭아', 'pear': '배' }
      ↑   ↑
      키  값
print (a)
print ( type(a) ) # <class 'dict'>
```

```

print ( a.keys() ) # 키값들만 출력 dict_keys(['apple', 'banana', 'peach', 'pear'])
print ( a.values() ) # 값들만 출력 dict_values(['사과', '바나나', '복숭아', '배'])
a['grape']='포도' # 새로운 키와 값을 추가하는 방법
print(a)

```

e.g): b = { } # 비어있는 딕셔너리를 생성한다.

```

b['apple']='사과'
  ↑   ↑
  키   값
b['pear']='배'
b['grape']='포도'
print(b)

```

문제130. 아래의 두개의 리스트를 각각 만들고 아래와 같이 fruit 라는  
딕셔너리를 생성하십시오 ! ( for 문의 zip 사용하면 편하게 할 수 있음)

```

a = ['사과', '배', '포도', '복숭아', '바나나']
b = ['apple', 'pear', 'grape', 'peach', 'banana']

```

```

fruit = { }
for i, k in zip( a, b ):
    fruit[i] = k

print( fruit )

```

결과:

```

print ( fruit )
{'사과':'apple', '배':'pear', '포도':'grape', '복숭아':'peach', '바나나':'banana'}

```

### ■ 39. 함수 이해하기(def)

오라클 배울때 함수를 다음과 같이 사용했었습니다.

```

select max(sal)
from emp;

select substr('smith', 1, 3 )
from dual;

```

오라클 배울때는 오라클 개발자가 만들어준 함수를 사용했지만  
파이썬 배울때는 우리가 직접 함수를 생성한다.  
파이썬에도 파이썬 개발자가 미리 만들어 놓은 함수들이 있다.  
파이썬이 무료이고 전세계인 사용하는 프로그램이다 보니  
각국의 뛰어난 개발자들이 파이썬 함수를 직접 만들어서 공유를 해줍니다.  
우리는 그냥 import 명령어로 불러와서 그 개발자가 만든 함수를  
사용할 수 있다.

빅데이터 분석가 ( 분야가 다양한 곳에서 분석 )

```
import random
```

내 전공에 해당하는 코드들을 쉽게 구현할 수 있다.

오라클 함수	vs	파이썬 함수
upper		upper()
lower		lower()
initcap		capitalize()
substr		만들어써야함
replace		replace()
length		len()
rtrim		rstrip()

ltrim	lstrip()
rpad	만들어써야함
lpad	만들어써야함
instr	만들어써야함
round	round()
trunc	trunc()
mod	%
power	pow
to_char	str
to_number	int, float
to_date	datetime.strptime()
nvl	만들어써야함
decode	if 문
case	if 문

e.g). 아래의 SQL을 파이썬으로 구현하시오 !

```
SQL> select lower(ename)
      from emp;
```

```
Pandas>
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
for i in emp['ename']:
    print ( i.lower() )
```

문제131. (점심시간 문제) 아래의 SQL을 파이썬으로 구현하시오 !

```
SQL> select lower(ename), lower(job)
      from emp;
```

```
import pandas as pd
```

```
filename = '/content/gdrive/My Drive/emp3.csv'
```

```
emp = pd.read_csv(filename)
for i,k in zip(emp['ename'],emp['job']):
    print(i.lower(),k.lower())
```

\* 함수를 생성하는 방법

def 함수이름(매개변수1):

실행문

return 출력값이 있는 변수명

\* 콜론(:) 으로 끝을 맺는 경우 4가지

1. if 문 종료시

예: if 조건:

2. for 문 종료시

예: for i in range(1,11):

3. 함수 생성시

예: def 함수이름(입력 매개변수):

4. 클래스 생성시

예: class 클래스이름:

e.g):

\* 두개의 입력값을 받아서 더하는 함수를 생성

```
def add_number( n1, n2 ):
    result = n1 + n2 # n1 에 입력된 값과 n2 에 입력된 값을 더해서 result 에 입력한다.
    return result    # result 에 입력된 값을 출력한다.
```

\* 위에서 만든 함수를 실행하는 방법

```
print ( add_number( 1, 2 ) ) # 3
```

설명: 위와 같이 만들고 컴퓨터를 켜다가 켜다음 다시 아래와 같이 함수를 실행만 하게 되면 실행되지 않는다. 다시 add\_number 함수를 생성해줘야 한다.

```
print ( add_number( 1, 2 ) ) # 3
```

다시 아래와 같이 add\_number 함수를 생성하고 실행해야 한다.

```
def add_number( n1, n2 ):
    result = n1 + n2 # n1 에 입력된 값과 n2 에 입력된 값을 더해서 result 에 입력한다.
    return result    # result 에 입력된 값을 출력한다.
```

```
print ( add_number( 1, 2 ) ) # 3
```

문제132. for loop 문을 사용하여 1부터 10까지 출력하시오 !

```
for i in range(1, 11):
    print (i)
```

문제133. 1부터 10까지 다 더한 값을 출력하시오 !

```
cnt = 0
for i in range(1, 11):
    cnt = cnt + i
```

```
print (cnt)
```

문제134. 위의 코드를 이용해서 함수를 생성하는데 아래와 같이 숫자를 입력하고 함수를 실행하면 해당 숫자까지 1부터 다 더한 값이 출력되게하시오!

```
print ( all_add(10) )
```

```
55
```

```
def all_add(n1):
    cnt = 0
    for i in range(1, n1+1):
        cnt = cnt + i
    return cnt # 함수 생성시 return 절을 필수 이다.
```

```
print ( all_add(10) )
```

문제135. 다음의 문자열 변수를 생성하고 문자열 변수의 문자를 소문자로 출력하시오 !

```
a = 'SCOTT'
```

```
print ( a.lower() ) # scott
```

```
print ( 'SCOTT'.lower() ) # scott
```

문제136. 아래의 문자열을 대문자로 출력하시오!

```
a = 'scott'
print ( a.upper() )
print ( 'scott'.upper() )
```

문제137. 아래의 문자열에서 첫번째 철자만 출력하시오 !



```
a='scott'
print( a[0] )
```

문제138. 위에 출력한 첫번째 철자를 대문자로 출력하시오

```
a='scott'
print( a[0].upper() )
```

문제139. 아래의 문자열 변수에서 cott 만 출력하시오  
( 두번째 철자부터 끝까지 출력하시오 )

```
a='scott'
print (a[1:])
```

문제140. 아래의 함수를 생성하시오! (첫번째 철자 대문자 나머지 소문자로  
로 출력되게 하는 함수)

```
print ( initcap('scott') )
```

결과: Scott

```
def initcap(val):
    return val[0].upper() + val[1: ].lower()
```

```
print ( initcap('scott') )
```

#### ■ 40. 함수 인자 이해하기

인자란 위에서 사용한 입력 매개변수를 말한다.

```
e.g):
def add_text( t1, t2 ):
    return t1 + ' ' + t2
```

```
print ( add_text('파이썬', '자바') )
```

결과: 파이썬 자바

\* 매개변수에 아무것도 입력하지 않고 실행하면 기본값이 출력되겠음  
함수를 생성하는 방법

```
def add_text( t1, t2='최고'):
    return t1 + ' ' + t2
```

```
print ( add_text('파이썬', '자바') ) # 파이썬 자바
```

```
print ( add_text('파이썬') ) # 파이썬 최고
```

설명: t2에 값을 아무것도 안넣었더니 기본값으로 지정한 최고가 출력되었습니다

#### ■ 41. 지역변수와 전역변수 이해하기(global)

변수는 자신이 생성된 범위(코드블럭) 안에서만 유효한다.  
함수 안에서 만든 변수는 함수 안에서 살아 있다가  
함수 코드의 실행이 종료되면 그 생명을 다한다.  
이것을 '지역변수' 라고 한다.

예: 스타벅스내에서의 머그컵

이와는 반대로 함수 외부에서 만든 변수는 프로그램이 살아있는 동안에  
함께 살아있다가 프로그램이 종료되면 같이 소멸된다.  
이렇게 프로그램 전체를 유효범위로 가지는 변수를 '전역변수' 라고 한다.

예: 텀블러

e.g)1:

```
strdata = '전역변수' # func1 함수 외부에 있는 변수 (텀블러)
```

```
def func1():
```

```
    strdata2='지역변수' # func1 함수 내부에 있는 변수: 스타벅스 머그컵
    return strdata2
```

```
print ( func1() ) # 지역변수
```

\* 설명: 대부분 많은 코드들이 지역변수를 사용한다.

그러면 '전역변수'를 사용하는 경우는 언제 입니까?

프로그램 전체에서 공통적으로 사용되고 잘 변하지 않는 데이터는 전역변수로 사용한다.

e.g):

```
pi = 3.141592653589793
```

```
def cycle_func1( num1 ): # 원의 넓이를 구하는 함수
```

```
    global pi
    # global 이라는 키워드를 앞에 쓰면 된다.
    return pi * num1 * num1
```

```
def cycle_func2( num1 ): # 원의 1/4 인 부채꼴의 넓이를 구하는 함수
```

```
    global pi
    return 1/4 * pi * num1 * num1
```

```
print ( cycle_func1(5) ) # 78.53981633974483
```

```
print ( cycle_func2(5) ) # 19.634954084936208
```

#### ■ 42. 함수 리턴값 이해하기(return)

모든 함수는 이름을 갖고 있다. 이 이름을 불러주면 파이썬은

그 이름 아래에 정의되어있는 코드를 실행한다.

이때 함수를 부르는 코드를 호출자라고 한다.

함수가 호출자에게 결과를 돌려주는것을 반환(return) 이라고 한다.

```
pi = 3.141592653589793
```

```
def cycle_func1( num1 ): # 원의 넓이를 구하는 함수
```

```
    global pi
    return pi * num1 * num1
```

```
print ( cycle_func1(5) ) # 함수를 호출하는 호출자 코드 이다.
```

\* 절대값을 출력하는 함수 (무조건 양수로 출력하는 함수)

```
print ( abs(-9) ) # 9
```

```
print ( abs(9) ) # 9
```

문제141. abs 함수를 사용하지 말고 if 문을 이용해서 절대값을 출력하는

my\_abs 라는 함수를 생성하시오 !

```
print ( my_abs(-9) ) # 9
```

```
print ( my_abs(9) ) # 9
```

```
def my_abs(num1):
```

```
    if num1 > 0:      # 0보다 크면
```

```
        return num1  # 입력값 그대로 리턴
```

```
    else:            # 그게 아니라면 (0에 음수를 써도 0은 0)
```

```
        return -1 * num1  # -1을 곱해서 리턴
```

```
print( my_abs(-5) )
print( my_abs(5) )
```

#### ■ 43. 파이썬 모듈 이해하기

파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 한다.

이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게

하나의 파일로 묶어 놓은것을 모듈이라고 한다.

외부의 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로

가져와서 자유롭게 사용할 수 있도록 해야하는데 이런일을 파이썬에서는

모듈을 import 한다 라고 한다.

e.g):

```
import time # time 모듈을 임폴트 한다.
```

```
print('5초간 프로그램을 정지합니다')
```

```
time.sleep(5) # 5초동안 멈춰있는것이다.
```

```
print('5초가 지났습니다.')
```

e.g): 평균이 30이고 표준편차가 5인 가우시안 정규분포를 따르는 모집단을  
백만개 구성하는 코드

```
import numpy as np # numpy 모듈을 임폴트를 한다.
```

```
avg = 30
```

```
std = 5
```

```
N = 10000000
```

```
mogip = np.random.randn(N) * std + avg
```

```
print ( mogip )
```

문제142.서울시 초등학교 백만명의 키를 모집단을 구성하는데

평균키가 148.5 이고 표준편차가 30인 모집단을 만드시오 !

```
import numpy as np
```

```
avg=148.5
```

```
std = 30
```

```
N= 1000000
```

```
height = np.random.randn(N) * std + avg
```

```
print (height)
```

문제143. 위의 모집단에서 100명을 표본으로 추출하여 100명의 평균키를

비어있는 리스트 a 에 입력하는 작업을 10000 번 수행하여

a 리스트에 10000개의 표본의 평균키가 입력되게하시오 !

```
import numpy as np
```

```
avg=148.5
```

```
std = 30
```

```
N= 1000000
```

```
height = np.random.randn(N) * std + avg
```

```
a = []
```

```
for i in range (1, 10001):
```

```
    result = np.random.choice(height, 100).mean() # 모집단에서 100개를 추출해서 평균값 취함
```

```
    a.append(result) # 평균값을 10000 번 입력함
```

```
print (a)
```

```
print(len(a))
```

설명: 초등학교 100만명의 키 모집단에서 표본을 100개 추출해서 표본 평균을  
10000개 모았음

문제144. 통계학을 전문으로 구현하는 모듈인 scipy 모듈을 임폴트하여  
위의 표본의 평균키값에 대한 확률밀도값을 출력하시오 !

```
import numpy as np
from scipy.stats import norm # scipy의 stats 패키지로 부터
    # norm 이라는 모듈을 import 해라
# from 패키지 import 모듈이름

avg=148.5
std = 30
N= 1000000
height = np.random.randn(N) * std + avg
a = []

for i in range (1, 10001):
    result = np.random.choice(height, 100).mean() # 모집단에서 100개를 추출해서 평균값 취함
    a.append(result) # 평균값을 10000 번 입력함

x = np.arange(140, 160, 0.001) # 140~160 까지 0.001 간격으로 숫자를 생성
y = norm.pdf( x, np.mean(a) , np.std(a) )
# 초등학생 키의 표본평균값들에 대한 확률 밀도함수 값이 출력된다.
print(y)
```

문제145. 데이터 시각화 전문 모듈인 matplotlib 를 임폴트하여  
위의 표본 평균값 10000개의 데이터에 대한 확률 밀도함수 값으로  
정규분포 그래프를 그리시오 ~

```
import numpy as np
from scipy.stats import norm # scipy의 stats 패키지로 부터
    # norm 이라는 모듈을 import 해라
#matplotlib 모듈안에 pyplot 라는 함수를
    # 를 임폴트하는데 별칭을 plt 로 해라 ~

avg=148.5
std = 30
N= 1000000
height = np.random.randn(N) * std + avg
a = []

for i in range (1, 10001):
    result = np.random.choice(height, 100).mean() # 모집단에서 100개를 추출해서 평균값 취함
    a.append(result) # 평균값을 10000 번 입력함

x = np.arange(140, 160, 0.001) # 140~160 까지 0.001 간격으로 숫자를 생성
y = norm.pdf( x, np.mean(a) , np.std(a) )
# 초등학생 키의 표본평균값들에 대한 확률 밀도함수 값이 출력된다.
plt.plot( x, y, color='blue')
plt.show()
```

문제146. 위의 확률밀도함수 그래프의 아래쪽 영역도 색깔로 채우시오 ~

```
import numpy as np
from scipy.stats import norm # scipy의 stats 패키지로 부터
    # norm 이라는 모듈을 import 해라
import matplotlib.pyplot as plt #matplotlib 모듈안에 pyplot 라는 함수를
    # 를 임폴트하는데 별칭을 plt 로 해라 ~

avg=148.5
std = 30
N= 1000000
height = np.random.randn(N) * std + avg
a = []

for i in range (1, 10001):
    result = np.random.choice(height, 100).mean() # 모집단에서 100개를 추출해서 평균값 취함
```

```
a.append(result) # 평균값을 10000 번 입력함
```

```
x = np.arange(140, 160, 0.001) # 140~160 까지 0.001 간격으로 숫자를 생성
y = norm.pdf( x, np.mean(a) , np.std(a) )
# 초등학생 키의 표본평균값들에 대한 확률 밀도함수 값이 출력된다.
plt.plot( x, y, color='blue')
```

```
plt.fill_between( x, y, interpolate=True, color='skyblue', alpha=0.7)
# 설명: plt 모듈안의 fill_between 함수를 이용해서 확률밀도함수 그래프의 아래 영역을
      색깔로 채우는데 interpolate=True 를 이용해서 아래의 영역이 색깔로 채워지게
      되고 alpha 는 색깔 투명도인데 0.0 ~ 1.0 사이로 기술 할 수 있다.
plt.show()
```

문제147. 위에서 그린 확률밀도함수 그래프의 색깔을 변경하고 그래프를 사진으로 첨부해서 올리시오.~~

구글에서 matplotlib 색상표 라고 검색하면 색상이름을 확인할 수 있다.

```
import numpy as np
from scipy.stats import norm # scipy의 stats 패키지로 부터
      # norm 이라는 모듈을 import 해라
import matplotlib.pyplot as plt #matplotlib 모듈안에 pyplot 라는 함수를
      # 를 임포트하는데 별칭을 plt 로 해라 ~

avg=148.5
std = 30
N= 1000000
height = np.random.randn(N) * std + avg
a = []

for i in range(1, 10001):
    result = np.random.choice(height, 100).mean() # 모집단에서 100개를 추출해서 평균값 취함
    a.append(result) # 평균값을 10000 번 입력함
```

```
x = np.arange(140, 160, 0.001) # 140~160 까지 0.001 간격으로 숫자를 생성
y = norm.pdf( x, np.mean(a) , np.std(a) )
# 초등학생 키의 표본평균값들에 대한 확률 밀도함수 값이 출력된다.
plt.plot( x, y, color='blue')

plt.fill_between( x, y, interpolate=True, color='purple', alpha=0.7)
# 설명: plt 모듈안의 fill_between 함수를 이용해서 확률밀도함수 그래프의 아래 영역을
      색깔로 채우는데 interpolate=True 를 이용해서 아래의 영역이 색깔로 채워지게
      되고 alpha 는 색깔 투명도인데 0.0 ~ 1.0 사이로 기술 할 수 있다.
plt.show()
```

20.12.02 파이썬강의 Day6 (문제148~문제190)

모듈이란 ?

파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 한다.

이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게

하나의 파일로 묶어 놓은것을 모듈이라고 한다.

외부의 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로

가져와서 자유롭게 사용할 수 있도록 해야하는데 이런일을 파이썬에서는

모듈을 import 한다 라고 한다.

#### ■ 내가 직접 모듈 생성하기

지금 현재 작성한 코드가 어느 디렉토리에 어느 이름으로

저장되어있는지 확인 하는 방법?

```
e:\users\std\spyder-py3\temp.py
```

↑  
파일명

. 이 붙어있으면 숨김폴더

```
e:\Users\std\my_cal.py
```

```
def add_number(n1, n2):  
    result = n1 + n2  
    return result
```

위의 함수 스크립트를 e:\Users\std\ 밑에 my\_cal.py 라는  
이름으로 저장할 함

메뉴에 새로운창을 여시오. ~ 새로운 창에서 아래와 같이 임폴트를 한다.

```
import my_cal # my_cal 모듈을 임폴트한다.
```

```
print ( my_cal.add_number(1,2) ) # my_cal 모듈안에 add_number 함수를  
# 실행해라 ~
```

설명: 이번 창에서는 def 로 add\_number 함수를 만드는 코드는 없습니다.

my\_cal.py 모듈안에 있는데 그 모듈을 이 창에서 사용할 수 있도록  
import 를 했다.

문제148. my\_cal.py 모듈 스크립트 안에 곱하기를 하는 아래의 함수를  
추가하십시오 !

```
def gob_number(n1, n2):  
    result = n1 * n2  
    return result
```

문제149. 다른 새로운 창에서 my\_cal 모듈을 임폴트하고 gob\_number  
함수를 실행하십시오 ~

```
import my_cal
```

```
print ( my_cal.add_number(1,2) )  
print ( my_cal.gob_number(1,2) )
```

설명: 그동안 통계문제를 풀려고 import random 을 했는데  
random 은 여러개의 함수를 모아놓은 소스코드였고 아마도  
어딘가에 random.py 라는 이름으로 저장되어 있는것을  
우리가 import 한것 이다.

문제150. 이번에는 나누기를 하는 함수를 my\_cal.py 에 저장하고  
다른 새로운 창에서 아래와 같이 import 하고 실행 될 수 있도록  
하십시오 !

```
import my_cal
```

```
print ( my_cal.add_number(1,2) )  
print ( my_cal.gob_number(1,2) )  
print ( my_cal.devide( 10, 2 ) ) # 5.0
```

#### ■ 44. 파이썬 패키지 이해하기

우리가 음악파일을 저장할 때도 장르별로 폴더를 만들어서 별도로  
저장을 하듯이 파이썬 모듈도 음악처럼 갯수가 많아지면 폴더(모듈 꾸러미)  
별로 별도로 관리를 해야 관리자 편해지는데 이 폴더가 바로 '패키지'이다.

패키지 (폴더)      vs    모듈 (폴더 안의 my\_cal.py 같은 파이썬 스크립트)

## ■ 파이썬 패키지를 만드는 단계

1. 아래의 디렉토리에 my\_loc 라는 폴더를 생성한다.

```
e:\Users\stu
```

```
e:\Users\stu\my_loc
```

2. my\_loc 폴더 안에 my\_cal.py 를 옮겨 놓는다.

(my\_cal.py 를 my\_loc 폴더에 복사하고 기존에 있던 my\_cal.py는 지우시오)

3. 이 평범한 폴더가 패키지로 인정을 받으려면 반드시 갖고 있어야하는 파일이 있다. 그 파일이 \_\_init\_\_.py 라는 파일이다.

```
e:\Users\stu\my_loc
|
|
1. __init__.py
2. my_cal.py
```

4. 새로운 창에서 아래와 같이 스크립트를 수행한다.

```
from my_loc import my_cal # from 패키지 import 모듈
```

```
print ( my_cal.add_number(1,2) ) # my_cal 모듈 안에 있는 add_number 함수
# 를 실행해라 ~
```

어제 위와 비슷한 스크립트를 보았는데

```
from scipy.stats import norm # scipy 패키지안에 stats 라는 패키지에
# norm 이라는 모듈을 임포트해라 ~
```

```
print ( norm.pdf(x, 평균, 표준편차) ) # norm 모듈에 pdf (확률밀도함수)를
# 실행해라 ~
```

## ■ 45. 파이썬 모듈 임포트 이해하기 ① (import)

이미 만들어져 있는 어떤 함수를 우리가 작성하는 코드에서 자유롭게 활용할 수 있으려면 해당 함수가 포함된 모듈을 임포트해야한다. 임포트하는 방법은 다음과 같습니다.

```
import 모듈이름
from 패키지 import 모듈이름
import 패키지이름.모듈이름
```

e.g): 아래와 같이 우리가 만든 모듈이 아니라 다른 사람이 만든 모듈을 임폴트를 해서 썼는데 이 모듈은 어디에 있는것일까 ?

```
import pandas
import numpy
```

위와 같이 패키지 이름을 안주고 모듈만 import 했는데 잘 실행이 되었다. 위와같은 모듈은 어떤 모듈입니까 ?

1. 파이썬 내장 모듈
2. sys.path 에 정의되어 있는 모듈

\* import 를 만나면 파이썬 모듈을 찾는 순서

1. 파이썬 내장 모듈에 있는지 확인

2. sys.path 에 정의되어 있는 디렉토리를 뒤져봅니다.

\* 파이썬 내장 모듈이 무엇이 있는지 확인하는 방법

```
import sys
print ( sys.builtin_module_names)

('_abc', '_ast', '_bisect', '_blake2', '_codecs', '_codecs_cn', '_codecs_hk',
'_codecs_iso2022', '_codecs_jp', '_codecs_kr', '_codecs_tw',
'_collections', '_contextvars', '_csv', '_datetime', '_functools',
'_heapq', '_imp', '_io', '_json', '_locale', '_lsprof', '_md5',
'_multibytecodec', '_opcode', '_operator', '_pickle',
'_random', '_sha1', '_sha256', '_sha3', '_sha512',
'_signal', '_sre', '_stat', '_string', '_struct',
'_symtable', '_thread', '_tracemalloc',
'_warnings', '_weakref', '_winapi', 'array',
'atexit', 'audioop', 'binascii', 'builtins', '
cmath', 'errno', 'faulthandler', 'gc',
'itertools', 'marshal', 'math', 'mmap',
'msvcrt', 'nt', 'parser', 'sys', 'time', 'winreg', 'xxsubtype', 'zipimport', 'zlib')
```

\* sys.path 에 정의된 디렉토리가 무엇인지 확인하는 방법

```
import sys
for i in sys.path:
    print ( i )

e:\Users\stu
e:\Users\stu\anaconda3\python37.zip
e:\Users\stu\anaconda3\DLLs
e:\Users\stu\anaconda3\lib
e:\Users\stu\anaconda3

e:\Users\stu\AppData\Roaming\Python\Python37\site-packages
e:\Users\stu\anaconda3\lib\site-packages
e:\Users\stu\anaconda3\lib\site-packages\win32
e:\Users\stu\anaconda3\lib\site-packages\win32\lib
e:\Users\stu\anaconda3\lib\site-packages\Pythonwin
e:\Users\stu\anaconda3\lib\site-packages\IPython\extensions
e:\Users\stu\ipython
```

\* site-packages 란 무엇인가 ?

site-packages 란 파이썬의 기본 라이브러리 패키지 외에 추가적인 패키지를 설치하는 디렉토리 이다.

site-packages 디렉토리에 여러가지 소프트웨어가 사용할 공통 모듈을 넣어두면 물리적인 장소에 구애받지 않고 모듈에 접근하여 반입할 수 있습니다

아래의 명령어가 수행되려면 아래의 명령어를 수행하는 스크립트가

e:\Users\stu 밑에 있어야한다. 왜냐하면 my\_loc 폴더가  
바로 e:\Users\stu 밑에 있기 때문이다.

```
from my_loc import my_cal
```

그런데 e:\Users\stu 가 아니더라도 다른 디렉토리에서라도  
from my\_loc import my\_cal 명령어를 자유롭게 실행하려면  
my\_loc 폴더가 e:\Users\stu\anaconda3\lib\site-packages  
밑에 있으면 된다.

문제151. my\_loc 폴더를 site-packages 폴더 밑에 두시오. !!

```
e:\Users\stu\anaconda3\lib\site-packages
```

```
from my_loc3 import my_cal
```

```
print ( my_cal.add_number(1,2) )
print ( my_cal.gob_number(1,2) )
print ( my_cal.devide( 10, 2 ) ) # 5.0
```



## 46. numpy 모듈 이해하기

\* numpy 모듈이란 ? python 언어에서 기본적으로 지원하지 않는  
배열(array) 또는 행렬(matrix) 의 계산을 쉽게 해주는  
라이브러리 이다.  
딥러닝에서 많이 사용하는 선형대 수학에 관련된  
수식들을 python 에서 쉽게 프로그래밍 할 수 있게  
해줍니다.

e.g). 아래의 행렬을 만드시오 !

### 1. 리스트로만 했을때

```
a = [ [ 1, 2 ], [ 4, 7 ] ]  
print(a)
```

### 2. numpy array 로 했을때

```
import numpy as np  
a = [ [ 1, 2 ], [ 4, 7 ] ]  
a2 = np.array(a) # a 리스트를 numpy 배열로 구성함  
print(a2)
```

문제152. 아래의 행렬의 합을 출력하시오 !

```
import numpy as np  
a = [ [ 1, 2 ], [ 4, 5 ] ]  
b = [ [ 3, 1 ], [ 6, 2 ] ]  
a2 = np.array(a)  
b2 = np.array(b)  
print ( a2 + b2)
```

```
[[ 4 3]  
 [10 7]]
```

문제153. 아래의 행렬의 합을 출력하시오 !

```
import numpy as np  
a = [ [ 6, 3, 4 ], [ 5, 1, 7 ] ]  
b = [ [ 4, 5, 7 ], [ 9, 20, 4 ] ]  
a2 = np.array(a)  
b2 = np.array(b)  
print (a2 + b2)
```

문제154. 아래의 행렬의 곱을 먼저 손으로 구하시오 !

```
import numpy as np  
  
a = [ [ 1, 2 ], [ 4, 3 ] ]  
b = [ [ 2, 1 ], [ 3, 4 ] ]  
a2 = np.array(a)  
b2 = np.array(b)  
print ( np.dot(a2,b2) )
```

문제155. (점심시간 문제) 아래의 행렬의 곱을 출력하시오 !

```
import numpy as np  
  
a = [ [ 1, 2 ], [ 4, 3 ] ]  
b = [ [ 2, 1 ], [ 3, 4 ] ]  
a2 = np.array(a)  
b2 = np.array(b)  
print(a2*b2)
```

#### ■ 47. numpy 모듈 사용하기2

numpy 모듈로 최대값, 최소값, 평균값, 중앙값, 최빈값, 분산값, 표준편차, 공분산값, 상관계수의 통계값들을 쉽게 출력할 수 있다.

e.g)1. 아래의 리스트에서 최대값을 출력하시오 !

```
a = [ 28, 23, 21, 29, 30, 40, 23, 21 ]
```

```
import numpy as np
a2 = np.array(a)
print ( np.max(a2) )
```

e.g)2. 아래의 리스트에서 최대값, 최소값, 평균값, 최빈값, 중앙값을 출력하시오 !

```
a = [ 28, 23, 21, 29, 30, 40, 23, 21, 21]
```

```
import numpy as np
from scipy.stats import mode
```

```
a2 = np.array(a)
print ( np.max(a2) )
print ( np.min(a2) )
print ( np.mean(a2) ) # 평균값
print ( np.median(a2) ) # 중앙값
print ( mode(a) ) # ModeResult(mode=array([21]), count=array([3]))
```

#### ■ 48. 파일 열고 닫기(open, close)

파일은 텍스트 파일과 바이너리 파일 두가지 종류가 있다.

텍스트 파일은 사람이 읽을 수 있는 글자로 저장된 파일이고  
바이너리 파일은 컴퓨터가 읽고 이해할 수 있는 이진 데이터를 기록한 파일이다.

예를들어 윈도우에서 제공하는 메모장 프로그램을 이용하여 내용을 적고  
저장하면 텍스트 파일로 저장된다.

이미지 뷰어로 볼 수 있는 jpg 이미지 파일은 이진 데이터를 jpg 형식의  
파일로 저장한 바이너리 파일이다.

파이썬에 파일을 다루기 위해서 가장 먼저 해야할 일은 파일을 오픈 하는것  
이다. 파일을 오픈하기 위해서는 open() 함수를 이용한다.

문법: open(파일이름, 모드)

모드	설명
r	텍스트 모드로 읽기
w	텍스트 모드로 쓰기
rb	바이너리 모드로 읽기
wb	바이너리 모드로 쓰기

e.g)1. 이미지 파일을 파이썬에서 여는 방법

1. lena.png 파일을 내려받아 e:\\data\\ 밑에 저장한다.

2. 아래의 코드를 실행 한다.

```
import PIL.Image as pilimg # 이미지를 파이썬에서 시각화 하기위한 모듈
import numpy as np
import matplotlib.pyplot as plt # 데이터 시각화 전용 모듈을 임포트 한다.
```

```

im = pilimg.open('e:\\data\\lena.png') # lena.png 파일을 읽어서 im에 입력
pix = np.array(im) # 넘파이 배열로 변환 한다.
plt.imshow(pix) # 화면에 띄운다.

```

문제156. 페사진을 파이썬에서 시각화 하시오 !

```

import PIL.Image as pilimg # 이미지를 파이썬에서 시각화 하기위한 모듈
import numpy as np
import matplotlib.pyplot as plt # 데이터 시각화 전문 모듈을 임포트 한다.

```

```

im = pilimg.open('e:\\data\\1.png')
pix = np.array(im) # 넘파이 배열로 변환 한다.
print(pix)
plt.imshow(pix) # 화면에 띄운다.

```

#### ■ 49. 클래스 이해하기(class)

객체 지향 프로그램에서 중요한 단어가 바로 클래스 이다.  
클래스는 프로그래머가 지정한 이름을 만든 하나의 독립된 공간이며,  
이름공간(name space) 이라 부른다.  
클래스를 구성하는 주요요소는 클래스에서 변수 역할을 하는 클래스 멤버와  
함수와 동일한 역할을 하는 클래스 메소드 이다.

클래스는 설계도이고 객체는 설계를 바탕으로 만든 제품

예: 총 설계도                      vs                      총 (제품)  
(클래스 )    (객체)

구글 이미지                                      구글 이미지

e.g)1. 총 클래스(설계도) 를 생성하시오 !

총의 기능(함수): 1.총알을 장전하는 기능, 2.총쏘는 기능

설명: 파이썬 클래스를 만들때 self 키워드는 필수이다. 안쓰면 에러가 납니다.  
self 의 뜻은 자기자신이다. 이 총설계도로 총을 만들고 그 총을 사용할 때  
charge 와 shoot 기능을 사용할 것인데 이때 이 총이 내꺼라는 의미가 바로  
self 이다.

```

총 생산하는 명령어 --> gun1 = Gun() # 준혁이
gun2 = Gun() # 한결
gun3 = Gun() # 세원

```

준혁이가 gun1 총을 사서 다음과 같이 충전했다.

```

gun1.charge(10000)
gun2.shoot(1000)

```

실제로 총설계도에는 charge(self, num) 이라고 해서 입력 매개변수가 2개인데  
총을 사용할 때 충전할때는 gun1.charge(100) 이렇게 매개변수 하나에 값을 입력했다.  
그러면 self 입력매개변수에는 자동으로 gun1 이 들어간다.

```

class Gun(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.
    def charge( self, num ): # 총알을 충전하는 함수
        self.bullet = num

    def shoot( self, num): # 총을 쏘는 함수
        for i in range(num):
            if self.bullet > 0:

```

```

        print ('탕!')
        self.bullet -= 1
    elif self.bullet ==0:
        print('총알이 없습니다')
        break

```

e.g)2. 위에서 만든 총설계도를 가지고 총을 한정 만드시오 !

```

gun1 = Gun()
  ↑   ↑
제품(객체) 클래스(총 설계도)

```

e.g)3. gun1 이라는 제품에 총알 10발 충전 한다.

```

gun1.charge(10)
  ↑   ↑
제품이름 총설계도안의 함수이름

```

e.g)4. 총을 쏘보시오 ~

```

gun1.shoot(3)

```

탕!

탕!

탕!

문제157. 총설계도를 수정해서 총알을 아래와 같이 충전하면  
 몇발이 충전되었다. 라는 메시지가 출력되게하시오!

```

gun1.charge(10)

```

10발이 충전되었다.

class Gun(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.

```

    def charge( self, num ): # 총알을 충전하는 함수
        self.bullet = num
        print ( num, '발이 충전되었습니다')

```

```

    def shoot( self, num): # 총을 쏘는 함수
        for i in range(num):
            if self.bullet > 0:
                print ('탕!')
                self.bullet -= 1
            elif self.bullet ==0:
                print('총알이 없습니다')
                break

```

```

gun1 = Gun()
gun1.charge(10)

```

문제158. 이번에는 총을 쏘면 총알이 탕! 탕! 하면서 아래쪽에 몇발 남았습니까.  
 라는 메시지가 출력되게하시오 !

```

gun1 = Gun()
gun1.charge(10)

```

```

gun1.shoot(3)

```

탕!

탕!

탕!

7발 남았습니다.

class Gun(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.

```
def charge( self, num ): # 총알을 충전하는 함수
    self.bullet = num
    print ( num, '발이 충전되었습니다')
```

```
def shoot( self, num): # 총을 쏘는 함수
    for i in range(num):
        if self.bullet > 0:
            print ('탕!')
            self.bullet -= 1

        elif self.bullet ==0:
            print('총알이 없습니다')
            break
    print ( '총알이 ', self.bullet, ' 발 남았습니다')
```

```
gun1 = Gun()
gun1.charge(10)
gun1.shoot(3)
```

문제159. 총을 처음 생산했을때 총알이 반드시 0발 장전되겠금  
총설계도를 수정하시오!

class Gun(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.

```
def __init__(self): # 설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
    self.bullet = 0 # 만들때만 작동하고 그 다음엔 작동안함
    print ( '총이 만들어졌습니다' , self.bullet, '발 장전 되었습니다')
```

```
def charge( self, num ): # 총알을 충전하는 함수
    self.bullet = num
    print ( num, '발이 충전되었습니다')
```

```
def shoot( self, num): # 총을 쏘는 함수
    for i in range(num):
        if self.bullet > 0:
            print ('탕!')
            self.bullet -= 1

        elif self.bullet ==0:
            print('총알이 없습니다')
            break
    print ( '총알이 ', self.bullet, ' 발 남았습니다')
```

```
gun1 = Gun()
```

총이 만들어졌습니다 0 발 장전 되었습니다

문제160. 총클래스를 이용해서 카드 클래스를 만들고 아래와 같이  
카드를 충전하고 사용하시오 !

```
card1 = Card()
카드가 발급되었다.
```

```
card1.charge(10000)
10000 원 충전되었다.
```

```
card1.consume(1000)
1000원이 사용되었다.
```

답:

class Card(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.

```
def __init__(self): # 설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
```

```

self.cash = 0 # 만들때만 작동하고 그 다음엔 작동안함
print ( '카드가 만들어졌습니다' , self.cash, '원이 충전 되었습니다')

def charge( self, num ): # 충알을 충전하는 함수
    self.cash = num
    print ( num, '원이 충전되었습니다')

def consume( self, num): # 충을 쓰는 함수
    if self.cash > 0:
        self.cash -= num
        print (num , '원 사용되었습니다')
        print ( '잔액이 ' , self.cash, ' 원 남았습니다')

    elif self.cash ==0:
        print('잔액이 없습니다')

card1 = Card()
card1.charge(10000)
card1.consume(1000)

```

## ■ 51. 클래스 메소드(함수) 이해하기

card 클래스의 charge 와 consume 는 메소드(기능) 이다.

```

예: a = [ 1, 2, 3, 4 ]
print ( type(a) ) # <class 'list'>

```

설명: 리스트가 클래스였다고 출력이 된다. 리스트가 class 라는 말은  
리스트 클래스 안에도 분명히 메소드(기능) 가 존재할 것 이다.

```

card1.charge(1000)
  ↑   ↑
객체  메소드(기능)

```

```

a = [1, 2, 3, 4 ]
a.append(5)
print(a)
  ↑   ↑
객체  메소드(기능)

```

## \* 리스트 객체의 유용한 메소드

메소드          설명

1. append() : 리스트 맨 끝에 새로운 요소를 추가할 때 사용
2. count() : 리스트에서 특정 요소의 갯수를 카운트할 때 사용
3. insert() : 리스트의 특정위치에 요소를 입력할 때 사용
4. remove() : 리스트의 특정 요소를 제거할 때 사용
5. sort() : 리스트의 요소를 순차적으로 정렬할 때 사용
6. reverse() : 리스트의 요소를 역순으로 정렬할 때 사용
7. index() : 리스트의 특정 요소의 위치를 출력할 때 사용

a = [ 1, 2, 3, 1, 2, 2, 2, 3, 4 ] # a라는 리스트 객체(제품)가 생성됨

```
print( a.count(2) ) # a 리스트에 숫자 2가 몇개가 있는지 조회
```

문제161. 초등학생 키에 대한 모집단을 생성하시오. ! (천만개로 구성)  
키의 평균값은 140, 표준편차 5로 해서 생성하시오 !

```
import numpy as np
```

```
height = np.random.randn(10000000) * 5 + 140
print( height)
```

문제162. 위의 모집단에 표본을 100개를 추출해서 표본의 평균값을 출력하시오 !

```
import numpy as np

height = np.random.randn(10000000) * 5 + 140
print( np.random.choice( height, 100).mean() )
```

문제163. 위에서 표본 100개를 뽑아서 평균값을 구해서 a 라는 비어있는 리스트에 담은 작업을 10000번 수행하시오 !

```
import numpy as np

height = np.random.randn(10000000) * 5 + 140
a = []
for i in range(1, 10001):
    a.append( np.random.choice( height, 100).mean() )

print( len(a) )
```

문제164. 위에서 구한 표본 평균값들 10000개의 평균값과 표준편차를 s\_avg 와 s\_std 변수에 각각 담으시오 !

```
import numpy as np

height = np.random.randn(10000000) * 5 + 140
a = []
for i in range(1, 10001):
    a.append( np.random.choice( height, 100).mean() )

s_avg = np.mean(a)
s_std = np.std(a)
print ( s_avg )
print ( s_std )
```

문제165. 초등학교 키 데이터를 120부터 160까지 0.001 간격으로 생성해서 x 라는 변수에 넣으시오 !

```
x = np.arange( 120, 160, 0.001)
print(x)
```

문제166. 위에서 만든 x 에 있는 키값들을 x 축으로 두고 확률 밀도함수 그래프를 생성하는데 y축의 확률밀도함수값을 구할 때 문제 164번에서 구한 평균값, 표준편차를 이용하시오 ~

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

height = np.random.randn(10000000) * 5 + 140
a = []
for i in range(1, 10001):
    a.append( np.random.choice( height, 100).mean() )

s_avg = np.mean(a)
s_std = np.std(a)
x = np.arange( 138, 142, 0.001)
y = norm.pdf( x, s_avg, s_std)
plt.plot( x, y , color='blue')
plt.show()
```

## ■ 52. 클래스 생성자 이해하기

```
gun1 = Gun()
  ↑   ↑
제품 설계도
(객체) (클래스)
```

다른 제품(객체)를 생성하고 싶다면 ?

```
gun2 = Gun()
gun2.charge(100)
gun2.shoot(3)
↑   ↑
```

객체 메소드(기능)

\* 클래스를 사용해서 파이썬 코딩을 해야하는 이유 ?

객체 지향 언어의 장점인 상속을 활용할 수 있기 때문이다.

객체 지향 언어의 장점이 너무 프로그램 할때 너무 유용하기 때문에

객체를 만드는 것을 지향(선호)하는 프로그램 언어 이다.

신용카드 발급 및 사용 프로그램을 구축하고 싶다면 ?

팀장님 ----> 카드의 중요 기본 기능을 담은 설계도(클래스)를 생성

클래스 이름: Card()

↓

팀원1(나)                      팀원2(동료)

(영화 할인 카드 클래스 생성) (주유 할인 카드 클래스 생성)

상속 코드: class Movie\_Card( Card ):

면접 문제: 객체 지향언어의 장점이 무엇입니까? 상속 이다.

만약에 내가 팀장님이 만든 카드의 중요기능(메소드)을 담은 클래스(Card)

를 상속받기만 하면 나는 팀장님이 코딩한 카드의 주요기능을 코딩하지

않아도 되고 나는 단순히 영화할인에만 집중해서 코드를 구현하면 된다.

※ 파이썬 클래스에서 사용하는 키워드인 self 가 무엇입니까 ?

class Card(): # 클래스 이름은 첫번째 철자는 대문자 나머지는 소문자로 구성한다.

def \_\_init\_\_(self): # 설계를 가지고 제품을 처음 만들때 자동으로 작동되는 함수

self.cash = 0 # 만들때만 작동하고 그 다음엔 작동안함

print ( '카드가 만들어졌습니다' , self.cash, '원이 충전 되었습니다')

def charge( self, num ): # 총알을 충전하는 함수

self.cash = num

print ( num, '원이 충전되었습니다')

이준혁\_card = Card()

양건준\_card = Card()

이준혁\_card.charge(100000): # 준혁이가 준혁이 카드를 10만원 충전했다.

양건준\_card.charge(5000000): # 양건준이가 건준이 카드에 500만원 충전함

설명: 여기서 5000000은 num 변수에 들어가고 양건준\_card 객체 이름은

self 매개변수에 들어가서 양건준\_card 에 5000000 충전했다는것을

나타내는 것이다.

## ■ 52. 클래스 생성자 이해하기

클래스의 인스턴스 객체가 생성될 때 자동적으로 호출되는 메소드가

클래스 생성자 이다. 클래스 생성자는 \_\_init\_\_(self): 이다.

카드의 기능 ? 0. 카드가 처음 만들어질때 0 원으로 충전되겠끔 하는 기능

1. 충전

2. 소비

e.g): class Card():



```
def __init__(self): # 생성자 메소드
    self.cash = 0
    print( '카드가 발급되었습니다', self.cash, '원이 충전되어있습니다')
```

yys\_card = Card() # 객체(제품)를 생성하는 명령어

문제167. 총 클래스를 생성하는데 총 클래스로 아래와 같이 총(제품)을  
생성하면 "총이 만들어졌습니다. 총알은 0발 장전되었다."  
라는 메시지가 출력되게 총 클래스를 만드시오 !

```
yys_gun1 = Gun()
```

총이 만들어졌습니다. 총알은 0발 장전되었다.

```
class Gun():
    def __init__(self):
        self.bullet = 0
        print ( '총이 만들어졌습니다. 총알은 ', self.bullet, ' 발 장전되었습니다')
```

```
yys_gun1 = Gun()
```

총이 만들어졌습니다. 총알은 0 발 장전되었습니다

문제168. 문자열 포맷을 이용해서 위의 메시지를 더 자연스럽게  
출력되게하시오 !

```
class Gun():
    def __init__(self):
        self.bullet = 0
        print ( '총이 만들어졌습니다. 총알은 %d 발 장전되었습니다' %self.bullet )
```

```
yys_gun1 = Gun()
```

총이 만들어졌습니다. 총알은 0 발 장전되었습니다

#### ■ 53. 클래스 소멸자 이해하기

객체가 사라질 때 자동으로 호출되는 함수를 소멸자(\_\_del\_\_) 이라고 한다.

예: 총 클래스로 yys\_gun1 이라는 총을 만들고 총을 사용하다가  
이제 총을 폐기하고 싶으면 del 명령어로 총을 폐기하면 되는데  
이때 자동으로 작동되는 메소드(함수) 가 소멸자 함수 이다.

```
e.g): class Gun():
    def __init__(self):
        self.bullet = 0
        print ( ' 총 한정이 생성되었습니다')
```

```
    def __del__(self):
        print( '총이 폐기 되었습니다')
```

```
yys_gun1 = Gun()
```

```
del yys_gun1
총이 폐기 되었습니다
```

#### ■ 54. 클래스 상속 이해하기

클래스는 상속이라는 특성을 가지고 있는 이름 공간이다.  
클래스에서 상속이란 어떤 클래스가 가지고 있는 멤버(변수)나 메소드(함수)를  
상속받는 클래스가 모두 사용할 수 있도록 해주는것 이다.  
상속을 해주는 클래스를 부모 클래스 또는 슈퍼 클래스라고 하고  
상속을 받는 클래스를 자식 클래스 또는 서브 클래스라고 한다.

부모 클래스로 부터 상속을 받아 자식 클래스를 정의하는 방법은 다음과 같습니다.

e.g): class 자식클래스이름(부모클래스이름):

설명: 자식 클래스는 부모 클래스에서 정의된 모든 멤버와 메소드를 그대로 상속 받습니다.

1. 팀장님이 카드의 주요기능을 담고 있는 Card() 클래스를 생성한다.
2. 팀원인 나는 영화할인 카드 클래스를 만들것인데 팀장님이 구현한 카드의 주요기능을 Card() 클래스로 부터 다 상속받고 나는 팀장님이 구현한 코드는 구현하지 않습니다. 나는 그냥 영화할인에 집중해서 메소드를 만들면 된다.

e.g): 팀장님이 만든 카드 class (카드 설계도) 부모클래스

1. 카드가 발급되었을때 0원이 충전되게 하는 기능 : \_\_init\_\_(self)
2. 카드를 충전하는 기능 : charge(self, num)
3. 카드를 쓰는 기능 : consume(self, num)

```
class Card(): # 부모 클래스
    def __init__(self):
        self.cash = 0
        print ('카드가 발급되었다.')

    def charge(self, num):
        self.cash += num
        print ( num, '원이 충전되었다.')

    def consume(self, num):
        if self.cash >= num: # 잔액이 소비하는 돈보다 커야지만 쓸 수있게
            self.cash -= num
            print ( num, '원이 사용되었다.')
        else:
            print ('잔액이 부족합니다')
```

```
yy_card1 = Card()
yy_card1.charge(10000)
yy_card1.consume(8000)
```

문제169. 충전한 금액보다 더 많은 금액을 소비하면 어떻게 되는지 테스트하시오 !

```
class Card(): # 부모 클래스
    def __init__(self):
        self.cash = 0
        print ('카드가 발급되었다.')

    def charge(self, num):
        self.cash += num
        print ( num, '원이 충전되었다.')

    def consume(self, num):
        if self.cash >= num: # 잔액이 소비하는 돈보다 커야지만 쓸 수있게
            self.cash -= num
            print ( num, '원이 사용되었다.')
        else:
            print ('잔액이 부족합니다')
```

```
yy_card1 = Card()
yy_card1.charge(10000)
yy_card1.consume(80000)
```

카드가 발급되었다.

10000 원이 충전되었다.

잔액이 부족합니다

문제170. 팀장님이 만든 Card() 클래스를 상속받아 영화할인 카드를  
생성하시오 ! ( 영화할인 카드 클래스: Movie\_Card() )

```
class Movie_Card( Card ): # 부모클래스인 Card 클래스를 상속받아
    pass                 # Movie_Card 클래스를 만든다.
```

```
m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(8000)
```

카드가 발급되었다.

100000 원이 충전되었다.

8000 원이 사용되었다.

문제171. 이번에는 제대로 영화관에서 사용하면 할인이 될 수  
있도록 영화 클래스를 생성하시오 !

```
class Movie_Card( Card ): # 부모에게 __init__, charge , consume 를 상속받았다.
    def consume(self, num, place): # 부모에게 받은 consume 는 내가 만든 consume로
        if place=='영화관':      # 덮어쓰기(overriding ) 가 된다.
            num = 0.8 * num
            if self.cash >= num: # 잔액이 num 보다 크다면 카드를 사용하고
                self.cash -= num
                print( place, '에서 ', num, '원이 사용되었다.')
            else:                # 아니면 잔액이 부족하다고 출력해라 ~
                print('잔액이 부족합니다')
        else: # 영화관에서 사용한게 아니라면 아래의 코드가 수행되게 해라 ~
            if self.cash >= num:
                self.cash -= num
                print ( place, '에서', num, '원이 사용되었다.')
            else:
                print('잔액이 부족합니다')
```

```
m_card1 = Movie_card()
m_card1.charge(20000)
m_card1.consume(12000,'영화관')
m_card1.consume(2000,'편의점')
```

카드가 발급되었다.

20000 원이 충전되었다.

영화관 에서 9600.0 원이 사용되었다.

편의점 에서 2000 원이 사용되었다.

문제172. 위의 영화할인 카드에 할인 장소를 추가해서  
주유소에서도 20% 할인 될 수 있도록 코드를 수정하시오 !

```
m_card1 = Movie_card()
m_card1.charge(100000)
m_card1.consume(12000,'영화관')
m_card1.consume(30000,'주유소')
m_card1.consume(2000,'편의점')
```

답:

```

class Movie_Card( Card ): # 부모에게 __init__, charge , consume 를 상속받았다.
    def consume(self, num, place): # 부모에게 받은 consume 는 내가 만든 consume로
        if place in ('영화관', '주유소'):
            num = 0.8 * num
            if self.cash >= num: # 잔액이 num 보다 크다면 카드를 사용하고
                self.cash -= num
                print( place, '에서 ', num, '원이 사용되었다.')
            else:
                # 아니면 잔액이 부족하다고 출력해라 ~
                print('잔액이 부족합니다')
        else: # 영화관에서 사용한게 아니라면 아래의 코드가 수행되게 해라 ~
            if self.cash >= num:
                self.cash -= num
                print ( place, '에서', num, '원이 사용되었다.')
            else:
                print('잔액이 부족합니다')

m_card1 = Movie_card()
m_card1.charge(20000)
m_card1.consume(12000,'영화관')
m_card1.consume(2000,'편의점')

```

문제173. (점심시간 문제) 영화관과 주유소에서는 20% 할인되게하고  
스타벅스에서는 10% 할인되게 코드를 수정하시오

```

m_card1 = Movie_card()
m_card1.charge(100000)
m_card1.consume(12000,'영화관')
m_card1.consume(30000,'주유소')
m_card1.consume(6000,'스타벅스')
m_card1.consume(2000,'편의점')

```

#### ■ 50. 클래스 멤버와 인스턴스 멤버 이해하기

클래스에서 선언된 변수는 클래스 멤버(변수)와 인스턴스 멤버(변수)가  
있다. 클래스 멤버는 클래스 메소드 바깥에서 선언되고  
인스턴스 멤버는 클래스 메소드 안에서 self 와 함께 선언되는 변수이다.

e.g): 사원이 입사하면 입사한 사원에 대한 이메일을 자동으로 생성하고  
이름을 출력하는 함수와 월급을 인상하는 함수를 담은 클래스를  
생성

```

class Employees: # 옆에 괄호() 를 따로 안쓴 이후는 밑에 __init__ 함수에 입력 매개변수가
    # 여러개 여서 이다.
    raise_amount = 1.1 # 클래스 변수(클래스 멤버) , 클래스 변수에는 self 가 없습니다.

    def __init__(self, first, last, pay): # emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
        self.first = first # 위와 같이 객체(제품)를 생성할 때 입력값이 3개가
        self.last = last # 입력되면서 객체(제품) 이 만들어 집니다.
        self.pay = pay
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'
        self.raise_amount = 1.1 # 인스턴스 변수

    def full_name(self): #사원의 전체이름을 출력하는 함수
        print ( '{} {}'.format(self.first, self.last) ) # 앞에 중괄호 2개에 각각 self.first 와 self.last
        # 에 있는 값이 입력이 된다.

    def apply_raise(self): # 월급을 인상하는 함수
        self.pay = int( self.pay * self.raise_amount)

```

```
emp_chulsu = Employees('chulsu', 'kim', 5000000)
print( emp_chulsu.full_name() )
print( emp_chulsu.pay ) # 5000000
    ↑      ↑
    객체이름 인스턴스 변수이름
```

문제174. 위에서 생성한 객체 emp\_chulsu 의 email 변수에 있는 내용을 출력해보시오 !

```
print( emp_chulsu.email ) # chulsu.kim@gmail.com
```

문제175. 철수의 월급을 회사규정에 따라 인상시키시오 !

```
emp_chulsu.apply_raise() # emp_chulsu 객체의 apply_raise() 메소드를 실행한다.
print(emp_chulsu.pay) # 5500000
```

문제176. 이번에는 새로운 사원 철수2 로 emp\_chulsu2 객체를 생성하시오 !

```
emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
```

문제177. 철수2 사원이 월급을 인상시키기 위한 인스턴스 변수를 알아냈습니다.

자기는 월급을 10% 인상이 아니라 20% 인상 시키고 싶어서 아래와 같이  
emp\_chulsu2 객체의 멤버인 raise\_amount 라는 변수에 1.2 를 할당하고  
월급을 인상시키는 apply\_raise() 메소드를 실행을 했다.

```
emp_chulsu2.raise_amount = 1.2 # 클래스의 인스턴스 변수에 접근해서
    # 인스턴스 변수를 갱신을 했다. 1.2 로
```

```
print( emp_chulsu2.pay ) # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay) # 6000000
```

문제178. 그래서 위와 같이 악용이 될수 없도록 막는 방법이 무엇입니까 ?

class Employees:

```
    raise_amount = 1.1 # 클래스 변수 (클래스 내부에는 있지만
        # 메소드 바깥쪽에 있는 변수
```

```
    def __init__(self, first, last, pay): # 객체가 만들어질때
```

```
        self.first = first        # 바로 작동되는 함수
        self.last = last
        self.pay = pay
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'
        self.raise_amount = 1.1 # 인스턴스 변수
```

```
    def full_name(self): #사원의 전체이름을 출력하는 함수
        return '{} {}'.format(self.first, self.last)
```

```
    def apply_raise(self): # 월급을 인상하는 함수
```

```
        self.pay = int( self.pay * Employees.raise_amount) # 클래스 변수를 사용
#            ↑
#        아까는 여기가 self 였는데 지금은 클래스 이름인 Employees 가 있다.
#        이자리에 인스턴스 변수가 아니라 클래스 변수를 썼습니다.
```

```
emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
```

```
emp_chulsu2.raise_amount = 1.2 # 인스턴스 변수는 1.2 로 변경이 되었다.
    # 클래스 변수를 변경한것은 아니다.
```

```
print(emp_chulsu2.pay) # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay) # 5500000
```

설명: 위와 같이 민감한 변수(멤버)들은 인스턴스 변수가 아니라

클래스 변수(멤버)를 사용해서 계산되게 코딩해야한다.

클래스내의 변수는 2개가 있는데

1. 클래스 변수 : 메소드 바깥의 변수  
객체 생성이후에 안의 값을 변경할 수 없다.
2. 인스턴스 변수 : 메소드 안의 변수  
객체 생성이후에 안의 값을 변경할 수 있다.

앞으로는 클래스를 활용해서 코딩을 하는 습관을 들일 필요가 있다.

#### ■ 55. 예외처리 이해하기 ① (try~except)

프로그램을 작성하다 보면 뜻하지 않은 오류가 발생하는 코드가 있을 수 있다. 프로그램이 실행되는 동안 오류가 발생하면 프로그램이 더 이상 진행 될 수 없는 상태가 되는데 이를 예외상황이라고 한다. 프로그램에 예외가 발생하더라도 프로그램을 중단 시키지 않고 예외에 대한 적절한 처리를 하여 프로그램을 계속 진행 시킬 수 있도록 하는 구문이 try ~ except 이다.

e.g): try:  
문제가 없을 경우 실행할 코드  
except:  
문제가 생겼을 때 실행할 코드

e.g)1: try ~ except 를 사용해서 예외처리를 하지 않았을 때의 코드

```
def my_divide():  
    x = input('분자의 숫자를 입력하시오. ~')  
    y = input('분모의 숫자를 입력하시오. ~')  
    return int(x) / int(y)  
  
print( my_divide() )
```

분자의 숫자를 입력하시오. ~10

분모의 숫자를 입력하시오. ~0

ZeroDivisionError: division by zero

e.g)2. try ~except 를 사용했을 때의 e.g)

```
def my_divide():  
    try:  
        x = input('분자의 숫자를 입력하시오. ~')  
        y = input('분모의 숫자를 입력하시오. ~')  
        return int(x) / int(y)  
    except:  
        return '당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.'
```

설명: try 와 except 사이에 코드가 잘 실행이 된다면 except 이후의 문장은 실행하지 않고 try 와 except 사이에 코드가 실행이 안된다면 except 이후의 문장을 실행한다.

```
print( my_divide() )
```

분자의 숫자를 입력하시오. ~10

분모의 숫자를 입력하시오. ~0

당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.

문제179. 판다스를 이용해서 emp3.csv 에서 이름과 월급을 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename', 'sal']] )
```

문제180. 이름이 SCOTT 인 사원의 이름과 월급을 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename', 'sal']] [ emp['ename']=='SCOTT' ] )
```

문제181. input 함수를 이용해서 이름을 물어보게 하고 이름을 입력하면  
해당 사원의 이름과 월급일 출력되게 하시오 !

이름을 입력하시오.~ SCOTT

SCOTT 3000

```
import pandas as pd
```

```
name = input('이름을 입력하시오.~')
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename', 'sal']] [ emp['ename']== name ] )
```

문제182. 이번에는 소문자로 이름을 입력해도 출력되게하시오 !

```
힌트: print ( 'scott'.upper() )
```

이름을 입력하시오.~ scott

SCOTT 3000

답:

```
import pandas as pd
```

```
name = input('이름을 입력하시오.~')
```

```
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename', 'sal']] [ emp['ename']== name.upper() ] )
```

문제183. 그런데 이번에는 위의 코드를 실행하는데 없는 사원이름을  
입력하시오 !

이름을 입력하시오.~ jack

Empty DataFrame

Columns: [ename, sal]

Index: []

문제184. 위의 코드에 예외처리를 해서 없는 사원이름을 입력하면  
입력하신 이름의 사원은 존재하지 않습니다. 라는 메시지가  
출력되게하시오 !

답: 이것은 에러가 아니라서 예외처리가 되지 않습니다.

문제185. 숫자를 입력하면 해당 숫자의 제곱값이 출력되는 코드를 구현하시오

숫자를 입력하시오.~ 2

답:

```
num = int( input( '숫자를 입력하시오.~' ) )  
print ( num**2 )
```

문제186. try~except 예외처리를 이용해서 아래와 같이  
문자를 입력하면 '잘못된 값을 입력하셨습니다' 라는 메시지가  
출력되게 하시오 !

숫자를 입력하시오.~ a

잘못된 값을 입력하셨습니다.

```
try:  
    num = int( input('숫자를 입력하시오.') )  
    print ( num**2 )  
except:  
    print('잘못된 값을 입력하셨습니다')
```

설명: try 와 except 사이의 있는 코드에서 에러가 나지만  
except 이후의 문장을 실행한다.

#### ■ 56. 예외처리 이해하기 ② (try~except~else)

어떤 로직을 수행할 때 오류 상황이 아닐 경우에만 어떤 작업을  
수행하는 코드를 작성해야할 때가 있다.  
이런 경우에 try~except~else 구문을 활용한다.

e.g): try:  
 실행할 코드 블록  
except:  
 예외처리할 코드 블록  
else:  
 except 절을 만나지 않았을 경우 실행하는 코드 블록

e.g)1. 186번 문제 코드

```
try:  
    num = int( input('숫자를 입력하시오.') )  
    print ( num**2 )  
except:  
    print('잘못된 값을 입력하셨습니다')  
else:  
    print('결과 출력에 성공했습니다')
```

설명: try ~ except 사이의 코드에 에러가 안났다면 else: 이후의 문장을  
실행한다.

숫자를 입력하시오. 4  
16  
결과 출력에 성공했습니다

숫자를 입력하시오. a  
잘못된 값을 입력하셨습니다

문제187. 아까 했던 나누기 프로그램을 수정해서 나누기가 성공하면  
성공적으로 나누기를 하였습니다. 라는 메시지가 출력되게하시오

분자를 입력하시오.~ 10  
분모를 입력하시오.~ 2

5  
성공적으로 나누기를 하였습니다.



답:

```
try:
    x = input('분자의 숫자를 입력하시오. ~')
    y = input('분모의 숫자를 입력하시오. ~')
    print (int(x) / int(y))
except:
    print('당황하셨겠지만 잘못된 값을 입력하셔서 나누기를 할 수 없습니다.')
else:
    print('성공적으로 나누기를 하였습니다')
```

#### ■ 57. 예외처리 이해하기 ③ (try~except~finally)

오류 발생 유무와 상관없이 어떤 코드를 무조건 실행 시키려면

try ~ except ~ finally 구문을 활용한다.

무조건 실행시키는 코드는 finally 부분에 작성하면 된다.

```
예: try:
    print('안녕하시오. ~')
except:
    print('예외가 발생했습니다')
finally:
    print('저는 무조건 실행됩니다')
```

문제188. 나누기하는 프로그램을 실행할 때 오류가 나던 오류가 나지 않던  
무조건 아래의 메시지가 출력되게하시오 !

```
분자를 입력하시오. ~ 10          10
분모를 입력하시오. ~ 2           0

5                                나누기를 할 수 없습니다.
```

이준혁이 만든 프로그램이다. 이준혁이 만든 프로그램이다.

```
답:
try:
    x = int( input('분자의 숫자를 입력하시오.~'))
    y = int( input('분모의 숫자를 입력하시오.~'))
    print( x / y )
except:
    print ('나누기를 할 수 없습니다.')
finally:
    print ('이준혁이 만든 프로그램입니다')
```

문제189. 동전을 10번던져서 앞면이 2번 나올 확률을 출력하시오. !

( 동전을 10번 던지는 작업을 10000 번 수행되게 하시오)

```
import random

coin=['앞면', '뒷면']
cnt=0
a = []
for i in range(1, 11):
    result = random.choice(coin)
    a.append(result)

print (a)

['앞면', '뒷면', '뒷면', '뒷면', '뒷면', '앞면', '뒷면', '뒷면', '앞면', '앞면']

import random

coin=['앞면', '뒷면']
cnt=0
```

```

a = []
for i in range(1, 11):
    result = random.choice(coin)
    a.append(result)
print(a.count('앞면'))

```

['앞면', '뒷면', '뒷면', '뒷면', '뒷면', '앞면', '뒷면', '뒷면', '앞면', '앞면']

```
import random
```

```
coin=['앞면', '뒷면']
cnt = 0
```

```

for k in range(1, 10001):      # k = 1 일때 실행문2, 실행문3
    a=[]                      # k = 2 일때 실행문2, 실행문3
    for i in range(1, 11):     # 실행문2
        result = random.choice(coin)
        a.append(result)

```

```
#['앞면', '뒷면', '뒷면', '뒷면', '뒷면', '앞면', '뒷면', '뒷면', '앞면', '앞면']
```

```

if a.count('앞면')== 2:      # 실행문3
    cnt = cnt + 1

```

```
print(cnt/10000) # 확률
```

문제190. (오늘의 마지막 문제) 위의 코드를 함수로 생성하여 확률이 출력되게하시오 !

```

coin_prob(2) # 0.04314
coin_prob(4) # 0.2065
    ↑
    앞면의 갯수

```

```
import random
```

```

def coin_prob(num):
    coin=['앞면', '뒷면']
    cnt=0
    for k in range(1,10001):
        a=[]
        for i in range(1,11):
            result=random.choice(coin)
            a.append(result)
        if a.count('앞면')==num:
            cnt+=1
    print(cnt/10000)

```

```

coin_prob(2)
coin_prob(4)

```

20.12.03 파이썬강의 Day7 (문제191~문제220)

■ 어제 배웠던 내용중에 질문이 많았던 내용만 전체적으로 리뷰

1. 클래스 변수와 인스턴스 변수
2. a=[] 리스트의 위치

■ 58. 예외처리 이해하기 ④ (try~except Exception as e)

코드에서 예외가 발생하면 이에 대한 자세한 내용을 파악하는것이

중요한다. 파이썬은 발생 가능한 예외에 대해 exception 객체로 미리 정의해두고 있는데 이에 대한 내용은 다음 링크에서 확인할 수 있습니다

<http://docs.python.org/3/library/exceptions.html>

설명: 위에 사이트에 보면 여러가지 다양한 예외에 대해서 미리 예러 처리를 할 수 있도록 정의해놓은 예외들을 확인할 수 있다.

```
e.g):
try:
    x = int( input('분자의 숫자를 입력하시오 ') )
    y = int( input('분모의 숫자를 입력하시오 ') )
    print( x / y )
except:
    print ('잘못된 값을 입력하셔서 나누기를 시도하셨습니다')
```

분자의 숫자를 입력하시오 10

분모의 숫자를 입력하시오 0

잘못된 값을 입력하셔서 나누기를 시도하셨습니다

분자의 숫자를 입력하시오 10

분모의 숫자를 입력하시오 a

잘못된 값을 입력하셔서 나누기를 시도하셨습니다

설명: 위의 경우에는 분모값을 입력할 때 숫자 0 을 입력했을때와 문자 a 를 입력했을때 똑같이 '잘못된 값을 입력하셔서 나누기를 시도하셨습니다' 가 출력되게 했는데 좀더 구체화 해서 분모값으로 0 을 입력하면 0으로 나눌 수 없습니다가 나오고 분모값으로 a 를 입력하면 잘못된 값을 입력하셨습니다 가 나오게 하고 싶다면?

```
try:
    x = int( input('분자의 숫자를 입력하시오 ') )
    y = int( input('분모의 숫자를 입력하시오 ') )
    print( x / y )
except ZeroDivisionError:
    print ('0으로 나눌 수 없습니다.')
except:
    print ('잘못된 값을 입력하셨습니다')
```

분자의 숫자를 입력하시오 10

분모의 숫자를 입력하시오 0

0으로 나눌 수 없습니다.

분자의 숫자를 입력하시오 10

분모의 숫자를 입력하시오 a

Traceback (most recent call last):

```
File "e:\Users\stu\spyder-py3\temp.py", line 4, in <module>
    y = int( input('분모의 숫자를 입력하시오 ') )
```

문제191. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 1번부터 출력되게하는 코드를 작성하시오 !

숫자를 입력하시오. ~ 5

1  
2  
3  
4  
5

```
a = int( input('숫자를 입력하시오. '))  
for i in range(1, a+1):  
    print (i)
```

문제192. 위의 코드에 예외처리를 해서 숫자를 물어볼때 문자를 입력하면  
잘못된 값을 입력하셨습니다.라고 메시지가 출력되게하시오

```
try:  
    a = int( input('숫자를 입력하시오. '))  
    for i in range(1, a+1):  
        print (i)  
except:  
    print ('잘못된 값을 입력하셨습니다.')
```

설명: 위의 코드의 경우에는 숫자를 입력할 때 알파벳 a를 넣으면  
예외처리가 되어서 잘못된 값을 입력하셨습니다. 라는 말만 나오고  
에러대한 정확한 원인파악은 하기가 어렵습니다.

잘못된 값을 입력하셨습니다 말고도 정확한 에러에 대한 원인을 파악을  
하고 싶다면 아래와 같이 작성하면 된다.

```
try:  
    a = int( input('숫자를 입력하시오. '))  
    for i in range(1, a+1):  
        print (i)  
except Exception as e:  
    print ('잘못된 값을 입력하셨습니다.')
```

```
    print (e) # 에러가 난 이유를 출력해줍니다.
```

숫자를 입력하시오.~ a

잘못된 값을 입력하셨습니다.  
invalid literal for int() with base 10: 'a'

#### ■ 59. 예외처리 이해하기 ⑤ (try~except 특정 예외)

파이썬 입장에서 봤을때는 오류가 아닌데 프로그래머가 이건 오류이다  
라고 일부러 프로그램이 안돌게 오류메세지를 출력하는 경우에 사용한다.

계산하는 프로그램

```
프로그램 코드 .....  
프로그램 코드 ..... <--- 오류가 발생했으면  
프로그램 코드 .....  
프로그램 코드 .....
```

설명: 금융권 프로그램에서는 금액이 안맞는 프로그램이 있다면  
사고로 이어지게 되므로 금액이 안맞으면 프로그램을 종료해라 라고  
예외처리를 할 수 있다.

문법: 실행코드  
if 어떤조건:  
 raise Exception('예외가 발생했다.') # 여기서 종료  
else:  
 실행코드

문제193. 판다스를 이용해서 emp3.csv 의 데이터를 로드하는데  
이름을 물어보게하고 이름을 입력하면 해당 사원의 이름과 월급이  
출력되게하시오 !

이름을 입력하시오.~ scott

SCOTT 3000

```
import pandas as pd
```

```
name = input('이름을 입력하시오.~')
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp[ ['ename','sal']] [ emp['ename'] == name.upper() ] )
```

문제194. 위의 결과에서 월급만 출력하시오 !

```
import pandas as pd
```

```
name = input('이름을 입력하시오.~')
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp[ 'sal' ] [ emp['ename'] == name.upper() ].values[0]
print( result )
```

이름을 입력하시오.~ scott

3000

설명: emp 데이터 프레임에서 어떤 특정값을 딱 하나만 출력하려면  
위와 같이 작성해줘야 한다.

문제195. 위의 코드에 사용자 정의 예외처리를 해서 월급이 고소득자는  
해당 사원의 월급을 볼 수 없습니다. 라는 메시지가 출력되게하시오  
( 월급이 3000 이상인 사원들을 고소득자로 보고 작성하시오)

```
import pandas as pd
```

```
name = input('이름을 입력하시오.~')
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp[ 'sal' ] [ emp['ename'] == name.upper() ].values[0]
if result >= 3000:
    raise Exception('해당 사원의 월급은 볼 수 없습니다')
print ( emp[['ename','sal']] [ emp['ename'] ==name.upper() ] )
```

이름을 입력하시오.~ smith

이름을 입력하시오.~ scott

이름을 입력하시오.~ scott

Traceback (most recent call last):

```
File "e:\\Users\\stu\\spyder-py3\\temp.py", line 7, in <module>
    raise Exception('해당 사원의 월급은 볼 수 없습니다')
```

Exception: 해당 사원의 월급은 볼 수 없습니다

문제196. 위의 코드를 수정해서 이름을 물어보게하고 이름과 직업을  
출력하는 코드로 작성하는데 직업이 SALESMAN 이면  
해당 사원의 정보는 볼 수 없습니다. 라는 메시지가 출력되면서  
프로그램이 종료되게 하시오 !

이름을 입력하시오.~ king

이름을 입력하시오.~ allen

```
import pandas as pd

name = input('이름을 입력하시오. ~')
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp[ 'job' ] [ emp['ename'] == name.upper() ].values[0]

if result == 'SALESMAN':
    raise Exception('해당 사원의 정보는 볼 수 없습니다')

print ( emp[['ename','job']] [ emp['ename'] == name.upper() ] )
```

#### ■ 60. 값 입력받기(input)

파이썬 내장 함수 input 은 사용자가 키보드로 입력한 값을 문자열로 리턴한다. input() 의 인자는 사용자 입력을 돕기위한 안내 문구나 힌트 등을 표시하는 문자열이 된다.

```
예: name = input(' 사원이름을 입력하시오. ~' )
    num  = int( input(' 숫자값을 입력하시오. ~' ) )
```

문제197. 아래처럼 사원 이름을 물어보게하고 사원이름을 입력하면 해당 사원의 이름과 월급이 출력되게하시오 !

```
import pandas as pd

name = input('사원 이름을 입력하시오. ~')

emp = pd.read_csv('e:\\data\\emp3.csv')
print ( emp[['ename', 'sal']] [ emp['ename']== name.upper() ] )
```

문제198. 없는 사원 이름을 입력하면 해당 사원은 없습니다. 라는 메시지가 출력되게하시오 !

```
import pandas as pd

try:
    name = input('사원 이름을 입력하시오. ~')
    emp = pd.read_csv('e:\\data\\emp3.csv')
    result = emp[ 'ename' ] [ emp['ename'] == name.upper() ].values[0]
    print ( emp[['ename', 'sal']] [ emp['ename']== result ] )
except LookupError:
    print ('해당 사원은 없습니다')
```

설명: result = emp[ 'ename' ] [ emp['ename'] == name.upper() ].values[0] 이 코드에서 values[0] 을 사용하면 Series(컬럼) 가 아니라 값으로 출력이 되어서 result 에 담기게 된다. 없는 사원이름을 입력하면 result 에 값이 입력되지 않게 되므로 LookupError 예외처리가 되어서 해당사원은 없습니다. 라는 메시지가 출력되는 겁니다.

사원 이름을 입력하시오. ~ aaa

문제199. (점심시간 문제) 직업을 물어보게하고 직업을 입력하면 해당 사원의 이름과 직업과 월급이 출력되게하는 코드를 작성하는데 없는 직업을 입력하면 해당 직업은 사원 테이블에 없습니다. 라는 메시지가 출력되게하시오 !

직업을 입력하시오. ~ salesman

직업을 입력하시오. ~ aaaaaa

해당 직업은 사원 테이블에 없습니다.

## ■ 61. 자료형 확인하기(type)

파이썬의 자료형은 하나의 클래스이다.

파이썬은 숫자나 문자, 문자열, 리스트, 튜플, 사전, 함수 등을 각각의 하나의 클래스로 취급한다.

코드를 작성하다가 변수 이름만 보고 이 자료가 어떤 자료형인지 확인해야 하는 경우가 있다. 이때 파이썬 내장함수인 `type()` 을 활용하면 자료형을 쉽게 확인 할 수 있다.

```
e.g):
numdata = 57
print ( type(numdata) ) # <class 'int'>
```

```
numdata2 = 57.2
print ( type(numdata2) ) # <class 'float'>
```

```
strdata = '파이썬'
print ( type(strdata) ) # <class 'str'>
```

```
a = [ 1, 2, 3, 4 ]
print ( type(a) ) # <class 'list'>
```

문제200. 딕셔너리 자료형을 만들고 위와 같이 `type` 을 확인하시오 !

```
b = { '사과' : 'apple', '배' : 'pear' }
print ( type(b) ) # <class 'dict'>
```

설명: 리스트 --> 대괄호 []  
튜플 --> 소괄호 ()  
딕셔너리 --> 중괄호 { }

## ■ 62. 나눗셈에서 나머지만 구하기(%)

나누기 연산에서 나머지만 구하는 연산을 % 로 구한다.

```
예: print ( 12 % 3 ) # 0
     print ( 12 % 5 ) # 2
```

문제201. 아래와 같이 두개의 숫자를 각각 물어보게하고 아래의 메시지가 출력되게하시오 !

첫번째 숫자를 입력하시오. ~ 1113  
두번째 숫자를 입력하시오. ~ 23

1113 을 23으로 나누면 9가 나머지로 남습니다.

답:

```
a = int( input('첫번째 숫자를 입력하시오. ~') )
b = int( input('두번째 숫자를 입력하시오. ~') )
c = a % b
print ( '%d 를 %d 로 나누면 %d 가 나머지로 남습니다.' %(a,b,c) )
```

## ■ 63. 몫과 나머지 구하기(divmod)

e.g):

```
result1, result2 = divmod(1113, 23) # divmod 함수가 출력하는 값이 2개여서
↑   ↑           # 앞에 result1, result2 변수를 두개를
몫   나머지      # 썼음

print ( result1, result2)
```

```
print( divmod(1113, 23) )
```

(48, 9) <--- 값이 2개가 출력되고 있다.

↑ ↑  
몫 나머지

```
result1 = divmod(1113, 23)  
print (result1) # (48, 9)
```

```
result1, result2 = divmod(1113, 23)  
print ( result1, result2)
```

문제202. 아래와 같이 실행되게 코드를 수행하시오 !

첫번째 숫자를 입력하시오 ~ 1113

두번째 숫자를 입력하시오 ~ 23

1113을 23으로 나눈 몫은 48 이고 나머지는 9 이다.

첫번째 숫자를 입력하시오 ~ 1113

두번째 숫자를 입력하시오 ~ 0

0 으로는 나눌 수 없습니다.

```
try:  
    a = int( input('첫번째 숫자를 입력하시오 ~') )  
    b = int( input('두번째 숫자를 입력하시오 ~') )  
    result1, result2 = divmod(a,b)  
    print ( '%d를 %d 으로 나눈 몫은 %d 이고 나머지는 %d 이다.' %(a,b,result1, result2) )  
except ZeroDivisionError:  
    print ('0 으로 나눌 수 없습니다.')
```

#### ■ 64. Pandas 를 이용한 데이터 검색

SQL, Pandas 를 자유롭게 사용할 수 있어야 한다.

SQL ---> Pandas, Pandas ---> SQL

```
e.g): emp[ ['ename','sal'] ] [ emp['ename']=='SCOTT' ]  
        ↑           ↑  
        컬럼        조건
```

문제203. dept3.csv 를 판다스로 로드해서 dept 데이터 프레임  
전체를 출력하시오 !

```
import pandas as pd  
  
dept = pd.read_csv('e:\\data\\dept3.csv')  
print ( dept )
```

문제204. 부서위치가 DALLAS 의 부서번호와 부서명(dname) 을  
출력하시오 !

```
import pandas as pd  
  
dept = pd.read_csv("e:\\data\\dept3.csv")  
print ( dept[['deptno', 'dname']] [ dept['loc']=='DALLAS'] )
```

#### ■ 65. Pandas 를 이용한 조인

e.g): 이름과 부서위치를 출력하시오 !

```
SQL> select e.ename, d.loc  
      from emp e, dept d  
      where e.deptno = d.deptno;
```

Pandas>



```
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")
result = pd.merge( emp, dept, on='deptno')
# 설명: emp 데이터 프레임과 dept 데이터 프레임을 merge 를 써서
#       조인시키는데 on='deptno' 를 이용해서 연결고리가 되는 컬럼을
#       지정해주면 된다.

print (result)
print ( result[ ['ename', 'loc'] ] )
```

문제205. DALLAS 에서 근무하는 직원들의 이름과 부서위치를 출력하시오

SQL>

```
Pandas>
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")
result = pd.merge( emp, dept, on='deptno')

print ( result[ ['ename', 'loc'] ] [ result['loc']=='DALLAS' ] )
```

문제206. 월급이 3000 이상인 직원들의 이름과 월급과 부서위치를 출력하시오

SQL>

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

result = pd.merge( emp, dept, on = 'deptno' )
print( result[['ename', 'sal', 'loc']][result['sal'] >= 3000] )
```

문제207. 부서번호가 10번, 20번인 직원들의 이름과 부서위치와 부서번호를  
출력하시오 !

```
SQL> select e.ename, d.loc, e.deptno
      from emp e, dept d
      where e.deptno = d.deptno and e.deptno in (10, 20) ;
```

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

result = pd.merge( emp, dept, on = 'deptno' )
print( result[['ename', 'loc', 'deptno']][result['deptno'].isin( [10,20] )] )
```

문제208. 월급이 1000 에서 3000 사이인 직원들의 이름과 월급과 부서위치를  
출력하시오 !

```
SQL> select e.ename, e.sal, d.loc
      from emp e, dept d
      where e.deptno = d.deptno
            and e.sal between 1000 and 3000;
```

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

result = pd.merge( emp, dept, on = 'deptno' )
print( result[['ename', 'sal', 'loc']][result['sal'].between(1000,3000)] )
```

문제209. 아래의 SQL을 Pandas 로 구현하시오

```
SQL> select e.ename, d.loc
      from emp e, dept d
      where e.deptno (+) = d.deptno ;
```

Pandas>

```
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

result = pd.merge( emp, dept, on = 'deptno' , how='right')
print( result[['ename', 'loc']] )
```

문제210. 아래의 SQL을 Pandas 로 구현하시오 !

```
SQL> select e.ename, d.loc
      from emp e full outer join dept d
      on ( e.deptno = d.deptno );
```

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

result = pd.merge( emp, dept, on = 'deptno' , how='outer')
print( result[['ename', 'loc']] )
```

#### ■ 66. Pandas 를 이용한 서브쿼리

e.g): JONES 보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오

```
SQL> select ename, sal
      from emp
      where sal > ( select sal
                    from emp
                    where ename='JONES');
```

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

jsal = emp['sal'][emp['ename']=='JONES'].values[0]
# JONES 월급을 jsal 변수에 담는다
print( emp[['ename','sal']] [ emp['sal'] > jsal ] )
```

문제211. 아래의 서브쿼리를 Pandas 로 구현하시오

```
SQL> select ename, sal
      from emp
      where job = ( select job
                    from emp
                    where ename='SCOTT' );
```

```
Pandas>
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

scott_job = emp['job'][emp['ename']=='SCOTT'].values[0]
print ( emp[['ename','sal']] [ emp['job']==scott_job ] )
```

문제212. 아래의 서브쿼리의 결과를 Pandas 로 수행하시오 !

```
SQL> select ename, sal
      from emp
      where job = ( select job
                    from emp
                    where ename='SCOTT' )
      and ename != 'SCOTT';

import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
dept = pd.read_csv("e:\\data\\dept3.csv")

scott_job = emp['job'][emp['ename']=='SCOTT'].values[0]
print ( emp[['ename','sal']] [ (emp['job']==scott_job) & (emp['ename'] != 'SCOTT') ] )
```

설명: 판다스에서 and 는 & 이고 or 는 | 이다.

& 와 | 를 쓸때는 괄호로 묶어줘야 한다.

~ 은 not 이고 != 같지 않다는 연산자

#### ■ 67. Pandas 와 오라클 그룹함수의 비교

```
SQL> select max(sal)
      from emp;
```

```
Pandas>
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp['sal'].max() )
```

설명: emp['sal'].max() --> 최대월급  
emp['sal'].min() --> 최소월급  
emp['sal'].sum() --> 토달월급  
emp['sal'].var() --> 분산값  
emp['sal'].std() --> 표준편차값

문제213. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select max(sal)
      from emp
      where deptno = 20;
```

```
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp['sal'][emp['deptno']==20].max() )
```

문제214. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select min(sal)
      from emp
      where job='SALESMAN';
```

```
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
print ( emp['sal'][emp['job']=='SALESMAN'].min() )
```

카페 12기 게시판에 올린 emp12.csv 를 c 드라이브 밑에 data 밑에  
다운 받으시오.

문제215. emp12.csv 를 판다스 데이터 프레임으로 만들어서 출력하시오 !

```
import pandas as pd

emp12 = pd.read_csv("e:\\data\\emp12.csv")
print(emp12)
```

문제216. 우리반에서 최소 나이를 출력하시오 !

```
import pandas as pd

emp12 = pd.read_csv("e:\\data\\emp122.csv")
print(emp12['AGE'].min() )
```

문제217. 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select job, max(sal)
      from emp
      group by job;

import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp.groupby('job')['sal'].max().reset_index()
print(type(result))
print(result)
```

※ 설명: reset\_index() 키워드는 Series(컬럼) 로 출력하는게 아니라  
DataFrame(테이블) 으로 출력하는 키워드 이다.

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

```

0 ANALYST 3000
1 CLERK 1300
2 MANAGER 2975
3 PRESIDENT 5000
4 SALESMAN 1600

```

문제218. 아래의 SQL을 판다스로 구현하시오 !

```

SQL> select deptno, sum(sal)
      from emp
      group by deptno;

```

```

import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()
print(result)

```

문제219. 아래의 SQL을 판다스로 구현하시오 !

```

SQL> select deptno, sum(sal)
      from emp
      where deptno != 20
      group by deptno;

```

```

import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()
print (result[['deptno','sal']] [ result['deptno'] != 20 ] )

```

```

print(type(result))
<class 'pandas.core.frame.DataFrame'>

```

```

emp[컬럼][조건]
result[컬럼][조건]

```

문제220. (오늘의 마지막 문제) 어제 마지막 문제로 만든 함수를 이용해서 아래와 같이 출력되게하시오

```

def coin_prob1(num):
    import numpy as np
    N = 10000
    coin = ['앞면','뒷면']
    cnt = 0

    for k in range(1,N+1):
        a=[]
        for i in range(1,11):
            result = np.random.choice(coin)
            a.append(result)

        if a.count('앞면')== num:
            cnt += 1
    return cnt/N

for i in range(0,11):
    print(f'동전을 10번 던져서 {i}개 앞면이 나올 확률 ',coin_prob1(i))

```

20.12.04 파이썬 강의 Day 8(문제221~문제252)

#### ■ 파이썬 수업 복습

1. 파이썬의 자료형 5가지
2. 파이썬 연산자
3. if 문

4. loop 문
5. 문자열에서 특정 위치의 철자 검색, 슬라이싱 검색
6. 파이썬으로 함수 생성
7. 파이썬으로 클래스 생성
8. 파이썬의 예외처리
9. 판다스 사용법

면접문제:

클래스를 왜 사용해야하는지? (= 객체지향 언어의 장점이 무엇인지?)

답: 상속을 사용할 수 있기 때문이다.

상속을 사용했을때 어떤 장점이 있는가?

답: 부모 클래스가 작성한 기능(메소드)을 코딩하지 않아도 된다.

중요한 기능은 부모 클래스를 작성하는 팀장님이 구현하고  
 팀원인 나는 팀장님이 구현한 중요기능을 그대로 상속받고  
 팀장님이 시킨 상세코드에 집중하면 되기 때문!

\* 판다스로 조인하기 복습

판다스로 조인(join)을 하기 위해서는 merge 를 사용하면 된다.

문법: pandas.merge( emp\_left, dept\_right, how='inner', on='deptno')

옵션: how='inner' : emp 와 dept 데이터 프레임에 공통적으로 존재하는  
 교집합일 경우에만 추출하겠다.

how='outer' : 열의 데이터가 양쪽 데이터 프레임에 공통적으로  
 존재하는 교집합이 아니어도 추출하겠다.

how='left' : 왼쪽 데이터 프레임의 키열에 속하는 데이터값을  
 기준으로 병합하겠다.

how='right' : 오른쪽 데이터 프레임의 키열에 속하는 데이터 값을  
 기준으로 병합하겠다.

```
e.g)
#%%%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")
result = pd.merge( emp, dept, on='deptno', how='left')
print( result[['ename', 'loc']])
```

```
select e.ename, d.loc
from emp e, dept d
where e.deptno = d.deptno (+);
```

#%%%  
 문제221. 부서번호, 부서번호별 평균월급을 출력하시오 !

```
select deptno, avg(sal)
from emp
group by deptno;
```

```
pandas>
#%%%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].mean().reset_index()
print( result )
#%%%
```

설명: groupby 부터 시작해서 reset\_index() 까지가 한 세트이므로  
 항상 같이 사용하면 됩니다.

```
#%%%
deptno    sal
```

```

0   10  2916.666667
1   20  2175.000000
2   30  1566.666667

```

문제222. 위의 결과에서 평균월급을 출력할 때 정수부분만 출력하고 싶다면?  
(소수점 이전만 출력)

```

#%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].mean().reset_index().astype(int)
print( result )
#%

```

설명: astype(int) 는 출력되는 데이터에서 정수부분만 취해라라는 뜻입니다.  
정수형으로 변환해라 ~ 라는 뜻입니다.

```

deptno  sal
0   10  2916
1   20  2175
2   30  1566

```

문제223. 직업과 직업별 토털월급을 출력하시오 !

```

SQL> select job, sum(sal)
      from emp
      group by job;

#%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('job')['sal'].sum().reset_index()
print( result )

```

#%

문제224. 부서위치, 부서위치별 토털월급을 출력하시오 !

```

SQL> select d.loc, sum(e.sal)
      from emp e, dept d
      where e.deptno = d.deptno
      group by d.loc;

#%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")
result = pd.merge( emp, dept, on ='deptno')
result2 = result.groupby('loc')['sal'].sum().reset_index()
print ( result2 )

loc      sal
0  CHICAGO   9400
1  DALLAS  10875
2  NEW YORK   8750

```

문제225. 아래의 SQL을 판다스로 구현하시오 !

```

SQL> select d.loc, nvl(sum(e.sal),0)
      from emp e, dept d
      where e.deptno (+) = d.deptno
      group by d.loc;

#%
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")
result = pd.merge( emp, dept, how='right', on ='deptno')
result2 = result.groupby('loc')['sal'].sum().reset_index()
print ( result2 )

```

문제226. 아래의 SQL을 판다스로 구현하시오 !

```

SQL> select deptno, count(*)
      from emp
      group by deptno;

Pandas>
#%
import pandas as pd

```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['empno'].count().reset_index()
print(result)
#%
```

문제227. emp122.csv 를 내려받아 판다스 데이터 프레임으로 만들고  
출력하시오 !

```
#%
import pandas as pd
```

```
emp122 = pd.read_csv("c:\\data\\emp122.csv")
print (emp122)
#%
```

문제228. 통신사, 통신사별 인원수를 출력하시오 !

```
#%
import pandas as pd
```

```
emp122 = pd.read_csv("c:\\data\\emp122.csv")
result = emp122.groupby('TELECOM')['INDEX'].count().reset_index()
print(result)
```

```
#%
0   kt    15
1   lg     4
2   sk    11
```

문제229. 우리반 테이블에서 통신사가 kt 이고 나이가 30살 이상인  
학생들의 이름과 나이와 통신사를 출력하시오 !

```
#%
import pandas as pd
emp122=pd.read_csv('c:\\data\\emp122.csv')
result=emp122[['ENAME','AGE','TELECOM']][ (emp122['TELECOM']=='kt') & (emp122['AGE']>=30)]
print(result)
```

설명: and 는 판다스에서 & 이고 or 는 판다스에서 | 입니다.  
그리고 &, | 를 사용할 때는 양쪽 조건에 소괄호를 둘러줘야 합니다.

```
#%
```

#### ■ 68. 반올림수 구하기(round)

파이썬 내장함수 round() 는 인자로 입력된 숫자형 자리수 에서  
반올림한 결과를 리턴합니다.

예제:

```
print( round(16.554) ) # 소수점 첫번째 자리에서 반올림  17
print( round(16.554, 0) ) # 소수점 첫번째 자리에서 반올림  17
print( round(16.554, 1) ) # 소수점 두번째 자리에서 반올림  16.6
print( round(16.554, 2) ) # 소수점 세번째 자리에서 반올림  16.55
```

\* 파이썬에서 반올림할때 중요하게 알아야 내용

```
print ( round(142.5) ) # 143으로 예상했는데 142가 나온다
```

" R 과 파이썬을 짝수를 좋아합니다."

```
142.5 -----> 142      2      2.5    3
187.5 -----> 188      7      7.5    8
```

파이썬을 기본적으로 이것을 해결하는 함수가 없습니다.  
0.5 일때는 짝수를 좋아하겠금 반올림되고 0.51 일때는 그냥 반올림됩니다.

예제. 판다스를 이용하지 않고 파이썬으로만 emp3.csv 에서 이름과 월급을  
출력하시오 !

```
import csv
file = open("c:\\data\\emp2.csv") # os 에 있는 emp2.csv 를 읽어서
    # file 이라는 변수에 넣는다.
emp_csv = csv.reader(file) # file 변수에 있는 csv 파일을 읽어서 emp_csv
    # 변수에 넣는다.
print(emp_csv) # 이 상태에서 그냥 프린트 하면 메모리가 주소만 나옵니다.
```

```
for emp_list in emp_csv: # csv 파일의 내용을 한행씩 리스트에 담아서
    print( emp_list[1], emp_list[5] )      # 출력합니다.
###
문제230. 판다스를 이용하지 말고 이름과 월급*12.3 를 출력하시오
```

```
['7839', 'KING', 'PRESIDENT', '0', '1981-11-17', '5000', '0', '10']
['7698', 'BLAKE', 'MANAGER', '7839', '1981-05-01', '2850', '0', '30']
['7782', 'CLARK', 'MANAGER', '7839', '1981-05-09', '2450', '0', '10']
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']
['7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-10', '1250', '1400', '30']
['7499', 'ALLEN', 'SALESMAN', '7698', '1981-02-11', '1600', '300', '30']
['7844', 'TURNER', 'SALESMAN', '7698', '1981-08-21', '1500', '0', '30']
['7900', 'JAMES', 'CLERK', '7698', '1981-12-11', '950', '0', '30']
['7521', 'WARD', 'SALESMAN', '7698', '1981-02-23', '1250', '500', '']
###
import csv
```

```
file = open("c:\\data\\emp2.csv") # os 에 있는 emp2.csv 를 읽어서
    # file 이라는 변수에 넣는다.
emp_csv = csv.reader(file) # file 변수에 있는 csv 파일을 읽어서 emp_csv
    # 변수에 넣는다.
print(emp_csv) # 이 상태에서 그냥 프린트 하면 메모리가 주소만 나옵니다.

for emp_list in emp_csv: # csv 파일의 내용을 한행씩 리스트에 담아서
    print( emp_list[1], int(emp_list[5])*12.3 )      # 출력합니다.
```

설명: csv 파일의 내용을 읽어서 출력할 때 기본적으로 월급도 문자형으로  
출력되므로 산술연산을 하려면 int() 함수를 이용해서 숫자형으로  
변환해주어야 합니다.

```
KING 61500.0
BLAKE 35055.0
CLARK 30135.0
JONES 36592.5
MARTIN 15375.0
ALLEN 19680.0
TURNER 18450.0
JAMES 11685.0
WARD 15375.0
FORD 36900.0
SMITH 9840.0
SCOTT 36900.0
ADAMS 13530.0
MILLER 15990.000000000002
```

문제231. 위의 결과를 다시 출력하는데 소수점 첫번째 자리에서 반올림되게  
하시오 !

```
###
import csv
file = open("c:\\data\\emp2.csv") # os 에 있는 emp2.csv 를 읽어서
    # file 이라는 변수에 넣는다.
emp_csv = csv.reader(file) # file 변수에 있는 csv 파일을 읽어서 emp_csv
    # 변수에 넣는다.
print(emp_csv) # 이 상태에서 그냥 프린트 하면 메모리가 주소만 나옵니다.
```

```
for emp_list in emp_csv: # csv 파일의 내용을 한행씩 리스트에 담아서
```



```
print( emp_list[1], round(int(emp_list[5])*12.3) )      # 출력합니다.
```

```
###
```

문제232. 직업이 SALESMAN 인 직원들의 이름과 직업을 출력하는데

판다스 이용하지 말고 emp2.csv 을 읽어서 출력하시오 !

```
###
```

```
import csv
```

```
file = open("c:\\data\\emp2.csv")
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if emp_list[2]=='SALESMAN':
```

```
        print( emp_list[1], emp_list[2] )
```

문제233.(점심시간 문제) 부서번호가 20번인 직원들의 이름과 월급과

부서번호를 출력하시오 ~

판다스를 이용한 방법:

판다스를 이용하지 않은 방법:

```
import csv
```

file=open("c:\\data\\emp2.csv") #os에 있는 emp2.csv를 읽어서 file 변수에 넣는다

emp\_csv=csv.reader(file) # file 변수에 있는 csv 파일을 읽어서 emp\_csv 변수에 넣는다

```
for emp_list in emp_csv:
```

```
    if int(emp_list[7])==20:
```

```
        print(emp_list[1],int(emp_list[6]), emp_list[7])
```

```
deptno 0  ename 1  job  2 mgr3   hiredate 4   sal 5   comm 6  deptno 7
```

```
['7839', 'KING', 'PRESIDENT', '0', '1981-11-17', '5000', '0', '10']
```

```
['7698', 'BLAKE', 'MANAGER', '7839', '1981-05-01', '2850', '0', '30']
```

```
['7782', 'CLARK', 'MANAGER', '7839', '1981-05-09', '2450', '0', '10']
```

```
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']
```

```
['7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-10', '1250', '1400', '30']
```

```
['7499', 'ALLEN', 'SALESMAN', '7698', '1981-02-11', '1600', '300', '30']
```

```
['7844', 'TURNER', 'SALESMAN', '7698', '1981-08-21', '1500', '0', '30']
```

```
['7900', 'JAMES', 'CLERK', '7698', '1981-12-11', '950', '0', '30']
```

```
['7521', 'WARD', 'SALESMAN', '7698', '1981-02-23', '1250', '500', '30']
```

```
['7902', 'FORD', 'ANALYST', '7566', '1981-12-11', '3000', '0', '20']
```

```
['7369', 'SMITH', 'CLERK', '7902', '1980-12-09', '800', '0', '20']
```

```
['7788', 'SCOTT', 'ANALYST', '7566', '1982-12-22', '3000', '0', '20']
```

```
['7876', 'ADAMS', 'CLERK', '7788', '1983-01-15', '1100', '0', '20']
```

```
['7934', 'MILLER', 'CLERK', '7782', '1982-01-11', '1300', '0', '10']
```

## ■69. 실수형 자료를 정수형 자료로 변환하기(int)

\* 파이썬의 변환함수 정리

	데이터 유형	변환함수
정수형	int	int()
문자형	str	str()
실수형	float	float()
리스트형	list	list()
튜플형	tuple	tuple()

코드 작성시 수학연산을 하다보면 정수끼리만 계산해야하는 경우가 있습니다.

이때 우리가 가진 데이터가 실수형이라면 실수형 자료를 정수형으로

변환한 후에 계산을 해주어야 합니다.

파이썬 내장함수 int() 는 인자로 입력된 실수형 자료를 정수형 자료로

변환해줍니다. int() 는 입력된 실수형 자료의 소수부분은 버리고

정수부분만 취하여 정수값으로 리턴합니다.

예제: print ( int(-5.4) ) # -5

문제234. 판다스를 이용해서 emp3.csv 를 읽어다가 이름과 월급을 출력하는데 월급을 출력할때 소수점 이하는 버리고 정수부분만 출력되게하시오 !

```
###
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename', 'sal']])
```

설명: int() 함수를 따로 안써도 정수형으로 출력되고 있습니다.  
판다스를 이용하지 않았을때와는 다르게 숫자는 바로 숫자형으로 출력해주고 있습니다.

#### ■ 70. 정수형 자료를 실수형 자료로 변환하기(float)

이미지 처리나 공학용 프로그램을 작성할 때 실수형 끼리만 계산해야하는 경우가 많습니다. 이때 우리가 가진 데이터가 정수형이라고 하면 정수형 자료를 실수형 자료로 변환한 후에 계산해 주어야 합니다. 파이썬 내장함수 float() 은 인자로 입력된 정수형 자료를 실수형으로 변환해 줍니다.

예: print ( float(10) ) # 10.0 이 출력됨

문제235. 판다스를 이용해서 이름과 월급을 출력하는데 월급을 출력할때 실수형으로 출력하시오 !

```
###
import pandas as pd
emp = pd.read_csv("c:\\data\\emp3.csv")

emp['sal'] = emp['sal'].apply(float)
    # emp 데이터프레임의 sal 컬럼의
    # 데이터를 float(실수형)로 변환해서 emp 데이터 프레임
    # 에 sal 컬럼의 데이터로 변경해라 ~
print ( emp[['ename', 'sal']])
```

설명: emp['sal'].apply 는 emp 데이터 프레임에 sal 시리즈에 apply 함수를 적용해서 데이터 유형을 변경할 수 있습니다.  
emp['sal'].apply(float) 이렇게하면 데이터 유형을 실수형으로 변경하는 것입니다.

#### ■ 71.정수 리스트에서 소수만 걸러내기(filter)

파이썬 내장함수인 filter() 는 리스트와 같은 자료형에서 특정 조건을 만족하는 값만 편리하게 추출할 수 있는 방법을 제공합니다. filter() 의 첫번째 인자는 특정 조건의 값을 추출하는 함수가 입력되며 두번째 인자(입력매개변수) 리스트와 같은 자료형이 입력됩니다.

예제: 숫자가 나열되어있는 리스트에서 짝수만 추출해내는 코드를 작성

```
a = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

1. 숫자를 입력하면 짝수이면 결과를 출력하고 홀수면 출력하지 않는 함수를 생성

```
def get_even(num):
    if num%2 == 0:
        return num
    else:
        return      # 리턴 다음에 아무것도 쓰지 않아서 아무것도
                    # 리턴되지 않습니다.
```

```
print (get_even(2) ) # 2
```

```
print (get_even(7) ) # None
```

2. filter 함수와 get\_even 함수를 이용해서 위의 a 리스트에서 짝수를 추출하기

```
a = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
result = filter( get_even, a )
#           ↑   ↑
#           함수이름 리스트이름
```

```
print (result) # <filter object at 0x0000024E99DFF748>
print ( list(result) ) # list 함수로 리스트로 변환해야 결과가 출력됩니다.
```

```
[2, 4, 6, 8, 10]
```

문제236. 아래의 리스트에서 숫자가 300 이상이면 출력하고  
300 미만이면 출력되지 않게 하시오 !

```
b = [ 100, 352, 254, 456, 123, 234, 567, 903 ]
```

```
def get_over(num):
    if num >= 300:
        return num
    else:
        return
```

```
result = filter( get_over, b )
print (list(result)) # [352, 456, 567, 903]
#%%
```

문제237. 우리반 데이터에서 나이가 30살 이상인 나이만 따로 결과로  
리스트로 출력하시오

45분까지 쉬세요 ~~

```
[ 31, 30, ..... ]
```

```
import pandas as pd
```

```
emp122= pd.read_csv("c:\\data\\emp122.csv")
a = []
for i in emp122['AGE']:
    if i >= 30:
        a.append(i)
```

```
print (max(a))
```

## ■ 72. 최대값, 최소값 구하기(max, min)

max() 와 min() 은 인자로 입력된 자료에서 최대, 최소값을 구해주는  
함수 입니다. max() 와 min() 의 인자로 리스트와 같은 시퀀스 자료가  
입력되면 시퀀스 자료의 요소 가운데 최대값 또는 최소값을 구하여  
리턴합니다.

오라클	vs	파이썬
max		max
min		min
count		len
sum		sum
avg		mean

예제:

```
a = [ 8, 7, 12, 55, 21, 34, 15, 9, 22 ]
print ( max(a) )
```

문제238. 사원 테이블에서 최대월급을 출력하시오 !( emp3.csv)

1. 판다스를 이용했을때

```
#%%
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print( emp['sal'].max() )
      .min()
      .mean()
      .sum()
```

## 2. 판다스를 이용하지 않았을때

```
import csv
file = open("c:\\data\\emp2.csv")
emp = csv.reader(file)
a = []
for i in emp:
    a.append( int(i[5]) )

print( max(a) )
```

문제239. 우리반 데이터(emp122.csv) 에서 최소나이를 출력하시오 !

### 1. 판다스 이용했을때

```
import pandas as pd
emp122 = pd.read_csv("c:\\data\\emp122.csv")
print ( emp122['AGE'].min() )
```

### 2. 판다스 이용하지 않았을때

emp122.csv 를 복사해서 emp1222.csv 로 붙여넣는다.

```
import csv
file = open("c:\\data\\emp1222.csv")
emp = csv.reader(file)
a = [ ]
for i in emp:
    a.append( int(i[2]) )
print(min(a))
```

UnicodeDecodeError: 'cp949' codec can't decode byte 0xec in position 3: illegal multibyte sequence

emp1222.csv 를 메모장으로 열어서 다른 이름으로 저장하기를 누르고

인코딩을 확인하면 UTF-8 로 되어있을텐데 이것을 ANSI 로 변경하고

저장하시오 !

```
###
import csv
file = open("c:\\data\\emp1222.csv")
emp = csv.reader(file)
a = [ ]
for i in emp:
    a.append( int(i[2]) )
print(min(a))
```

## 73. 판다스에서 결측치(NaN) 확인하기

emp3.csv 에 comm 처럼 결측치(NaN)를 처리하는 판다스 함수를 배웁니다.

↓  
Not a Number

예: 판다스를 이용해서 이름과 커미션을 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename', 'comm']] )
```

```
   ename  comm
0   KING   NaN
1  BLAKE   NaN
2  CLARK   NaN
3  JONES   NaN
```

```

4 MARTIN 1400.0
5 ALLEN 300.0
6 TURNER 0.0
7 JAMES NaN
8 WARD 500.0
9 FORD NaN
10 SMITH NaN
11 SCOTT NaN
12 ADAMS NaN
13 MILLER NaN

```

문제240. 커미션이 결측치(NaN) 인 사원들의 이름과 커미션을 출력하시오

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename', 'comm']] [ emp['comm'].isnull() ] )
```

문제241. 커미션이 결측치(NaN) 이 아닌 사원들의 이름과 커미션을  
출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print ( emp[['ename', 'comm']] [ ~ emp['comm'].isnull() ] )
```

\* 데이터 받았으면 데이터 분석을 하기 전에 데이터 전처리를 해야하는데  
전처리중에서 결측치 확인하는 단계가 있습니다.

문제242. emp3.csv 에 결측치가 있는지 확인하는 방법 ?

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print( emp.isnull() )
```

문제243. emp3.csv 에 결측치가 몇개인지 확인하는 방법 ?

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
print( emp.isnull().sum() )
```

```

index      0
empno      0
ename      0
job        0
mgr        1
hiredate   0
sal        0
comm       10
deptno     0

```

설명: 결측치가 있으면 데이터 분석하기가 어렵고 머신러닝을 이용한  
데이터 분석인 경우 좋은(정확도가 높은) 머신러닝 모델이 나오기  
어렵습니다. 결측치를 처리를 해줘야 합니다.

문제244. 타이타닉 데이터에 결측치가 어느 컬럼에 많은지 확인하시오 !

```
import pandas as pd
```

```
tat = pd.read_csv("c:\\data\\train.csv")
print( tat.isnull().sum() )
```

```

PassengerId  0
Survived     0
Pclass       0
Name         0
Sex          0
Age         177

```

```
SibSp      0
Parch      0
Ticket      0
Fare        0
Cabin      687
Embarked    2
dtype: int64
```

#### ■ 74. 판다스에서 파생변수 추가하는 방법

파생변수란 기존의 데이터를 가지고 새롭게 가공해서 만든 새로운 컬럼입니다.

예제: 판다스에서 컬럼 추가하는 방법

emp 데이터 프레임에 sal 의 데이터와 똑같은 데이터로 sal2 라는 컬럼을 추가하려면 ?

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
emp['sal2'] = emp['sal'] # emp 데이터 프레임에 sal2 컬럼을 추가하는데
                        # 데이터는 emp 데이터 프레임에 sal 로 하시오
print (emp)
```

문제245. 판다스를 이용해서 이름과 부서위치를 출력하시오!  
(emp3.csv 와 dept3.csv 를 이용해서 조인하세요 )

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")

result = pd.merge( emp, dept, how='inner', on='deptno')
print ( result[['ename', 'loc']] )
```

문제246. emp 데이터 프레임에 loc 컬럼을 추가하고 해당 사원의 부서위치로 값을 갱신하시오 !

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")

result = pd.merge( emp, dept, how='inner', on='deptno')
emp['loc'] = result['loc'] # emp 데이터프레임에 loc 컬럼 추가하면서
print (emp)              # result 의 loc 로 값을 갱신합니다.
```

설명: 파생변수를 왜 추가를 하나면?

emp 테이블에서 퇴사할것 같은 사원이 누군인지 예측하시오 !

머신러닝을 이용해서 예측을 하면 됩니다.

머신러닝이 예측을 잘하려면 좋은 데이터를 주고 학습시켜야 합니다.

자기의 월급이 자기가 속한 직업의 평균월급보다 더 작은 월급을 받는 사원이면 퇴사할 가능성 높다.

직업별 평균월급이 emp 데이터 프레임에 추가 되어 있으면 머신러닝이 예측하기 좋은 데이터가 추가가 된것 입니다.

문제247. 직업, 직업별 평균월급을 판다스로 출력하시오 !

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
```

```
result = emp.groupby('job')['sal'].mean().reset_index()
```

```
result['sal'] = result['sal'].astype(int) # result 데이터프레임에 sal 을 정수형으로
print(result) # 변환해서 result 데이터프레임에 sal 에
# 반영하겠다.
```

```
   job    sal
0  ANALYST 3000
1   CLERK 1037
2  MANAGER 2758
3  PRESIDENT 5000
4  SALESMAN 1400
```

문제248. emp 와 result 를 서로 조인해서 조인된 전체 데이터 프레임을 출력하시오 !

```
result2 = pd.merge( emp, result, how='inner', on='job')
print ( result2 )
```

설명: emp 에도 sal 이 있고 result 에도 sal 이 있어서 emp 의 sal 은 컬럼명이 sal\_x 로 변경되었고 result 의 sal 은 sal\_y 로 변경되었습니다. sal\_y 는 해당 직업의 평균월급입니다.

문제249. emp 데이터 프레임에 컬럼을 하나 추가하는데 job\_avgsal 로 추가하고 문제248번에서 구한 직업별 평균월급인 result2['sal\_y'] 의 값으로 값을 갱신하시오 !

```
emp['job_avgsal'] = result2['sal_y']
print (emp)
```

설명: 현업에서 머신러닝 데이터 분석가들이 하는 일중 상당수가 바로 이런 파생변수를 추가하는 작업입니다. 좋은 파생변수를 추가해야 머신러닝이 예측을 잘 할 수 있습니다. 게임회사에서 어떻게 응용하냐면 그 게임을 이탈될것 같은 유저를 머신러닝으로 찾아서 형평성에 어긋나지 않도록 하면서 그 유저가 인식하지 못하도록 조용히 혜택을 줍니다.

문제250. emp 데이터 프레임에 해당 사원이 근무하는 부서번호의 평균월급을 sal\_avg 라는 이름으로 파생변수를 생성하시오 !

```
#%%
```

```
import pandas as pd
emp = pd.read_csv("e:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].mean().reset_index() # 부서별 평균월급을 result에 할당한다.
result['sal'] = result['sal'].astype(int) # result 에서 sal 의값을 int 로 변경한다.
# 기존에 emp에 sal이라는 값이 존재하기에
# 구분을 위해 자동으로 기존의 sal은 sal_x, 새로 생긴 부서평균 sal 은 sal_y로 이름이 변경되었다.
result2 = pd.merge(emp,result,how='inner',on='deptno') # 부서별 평균 월급이 sal_y로 emp에 merge되었다.
#
```

```
emp['sal_avg'] = result2['sal_y'] # emp 테이블에서 sal_avg 라는 컬럼을 새로만들고(파생변수) 그 안에 result2 의 sal_y 값을 할당해라
```

```
#print(emp)
# print(emp['sal_avg'])
```

```
#%%
```

문제251. emp 데이터 프레임을 출력하는데 월급이 높은 사원부터 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.sort_values('sal', ascending=False)
print ( result )
```

설명: 판다스를 사용할 때 데이터를 정렬하려면 위와 같이 sort\_values 함수를 이용하면 됩니다.

ascending=True : 낮은값에 높은값 순으로 정렬하겠다.  
ascending=False : 높은값에 낮은값 순으로 정렬하겠다.

문제252. (오늘의 마지막 문제) 아래의 SQL을 판다스로 구현하시오 !

```
SQL> select job, sum(sal)
      from emp
      group by job
      having sum(sal) >= 6000
      order by sum(sal) desc;
```

Pandas>

```
###

import pandas as pd
emp=pd.read_csv("e:\\data\\emp3.csv")

result = emp.groupby('job')['sal'].sum().reset_index()
result2 = result.sort_values('sal', ascending = False)

print ( result2[['job', 'sal']] [result2['sal'] >= 6000 ] )

###
```

20.12.07 파이썬 강의 Day 9(문제253~문제283)

#### ■ 75. 문자열에서 특정 위치의 문자 얻기

문자열에서 특정 위치의 문자를 얻는 방법은 인덱싱을 이용하는 것입니다.  
인덱스는 0 부터 시작합니다. 파이썬 인덱스는 음수도 가능합니다.

예제:

```
print( 'scott'[0]) # s
print( 'scott'[2]) # o
```

a = 'scott' # a 라는 문자형 변수를 생성하면서 scott 을 a 라는 빈껍데에 담았다.  
print (a[0]) # s  
print (a[2]) # o

문제253. 아래의 txt 변수에서 w 를 출력하시오 !

```
txt = 'A tale that was not right'

print (txt[12])
```

문제254. 위의 txt 변수에서 맨끝의 철자인 t 를 출력하시오 !

```
txt = 'A tale that was not right'
print (txt[-1])
```

문제255. emp2.csv 에서 이름만 출력하시오 !

판다스를 이용하지 말고 emp2.csv 파일에 직접 읽어들여서  
출력하시오 !

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
```



```
print (emp_list[1] )
```

```
BLAKE
CLARK
JONES
MARTIN
ALLEN
TURNER
JAMES
WARD
FORD
SMITH
SCOTT
ADAMS
MILLER
```

문제256. 위의 출력된 이름의 데이터 유형이 무엇인지 확인하시오 !

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print (type(emp_list[1] ))
```

문제257. 문제255번을 다시 수행해서 emp2.csv 에서 이름을 출력하는데  
이름의 첫번째 철자만 출력하시오 !

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print (emp_list[1][0]) # a[0]
```

문제258. 위의 결과를 다시 출력하는 첫번째 철자가 소문자로 출력되게하시오

( 힌트: 'SCOTT'.lower() )  
문자열.lower()

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print (emp_list[1][0].lower()) # a[0].lower()
```

#### ■ 76. 문자열에서 지정한 구간의 문자열 얻기(슬라이싱)

문자열에서 특정 구간에 있는 문자열을 얻으려면 슬라이싱을  
이용하면 됩니다.

예제:

```
print( 'scott'[0:2] ) # sc 0부터 2미만까지
a = 'scott'
print ( a[0:2] ) # sc
print ( a[3:] ) # tt 3부터 문자끝까지 출력됨
print ( a[:3] ) # sco 문자 처음부터 읽어서 3미만까지 출력
```

문제259. 아래의 SQL을 파이썬으로 구현하시오 !

( 판다스 이용하지말고 그냥 emp2.csv 를 읽어서 구현)

```
SQL> select ename, substr( ename, 1, 3)
      from emp;
```

파이썬>

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[1], emp_list[1][0:3] )
```

## ■ 77. 문자열에서 홀수 번째 문자만 추출하기

주어진 문자열에서 홀수 번째 문자만 추출하는 방법은 슬라이싱의 스텝을 이용하면 됩니다.

예제:

```
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
result = txt[0:] # 0번째 부터 문자끝까지 출력
print (result )
result2 = txt[0 : 2 ] # 문자열에서 2칸씩 건너뛰면서 출력
print(result2)
```

문제260. 위의 txt 문자열에서 짝수번째 철자들만 출력하시오 !

```
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
result3 = txt[1: : 2]
print(result3)
```

## ■ 78. 문자열을 거꾸로 만들기

슬라이싱을 이용하면 매우 간단하게 거꾸로된 문자열을 얻을 수 있습니다.  
문자열 txt 를 처음부터 끝까지 스텝 -1 로 슬라이싱 하면 됩니다.

예제:

```
txt = 'aAbBcCdDeEfFgGhHiIjJkK'
print ( txt[ : ] ) # 문자열 전체 다 출력
print ( txt[ : ] ) # 문자열 전체가 다 출력
print ( txt[ : : -1 ] ) # 문자열 전체가 다 출력되는데 처음부터 읽어서
# 끝까지 1스텝으로 읽는다.
result = txt[ : : -1] # 문자열 전체를 뒤에서 부터 읽어서 앞으로
# 1스텝씩 읽는다.
print (result)
```

문제261. 이름을 출력하는데 이름의 철자를 거꾸로 출력하시오 !  
( emp2.csv 를 읽어서 하세요 ~ 파이썬으로만 하세요 )

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[1][ : : -1] )
```

## ■ 79. 두 개의 문자열 합치기(+)

두개의 문자열을 합치는 방법은 매우 간단합니다.  
문자열1 + 문자열2 + 문자열3 으로 + 연산자를 이용하면 됩니다.

예제:

```
a = 'scott'
b = 'king'
print (a + b ) # scottking
```

문제262. 아래의 SQL 을 파이썬으로 구현하시오 ! (emp2.csv 를 이용하세요)

```
SQL> select ename || job
from emp;
```

파이썬>

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[1] + emp_list[2] )
```

문제263. 아래와 같이 결과가 출력되게하시오 !

```
SQL> select  ename || '의 직업은' || job || '입니다.'
        from emp;
```

KING 의 직업은 PRESIDENT 입니다.

BLAKE 의 직업은 MANAGER 입니다.

:

파이썬>

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[1] +' 의 직업은 ' + emp_list[2] + ' 입니다' )
```

■ 80.문자열을 반복해서 새로운 문자열로 만들기(\*)

예제:

```
print ( '★' * 10 ) # ★★★★★★★★★★
print ( '여러분 ~' * 10 ) # 여러분 ~여러분 ~여러분 ~여러분 ~여러분 ~여러분 ~여러분 ~여러분 ~여러분 ~
```

문제264. for loop문을 이용해서 숫자 1부터 5까지 출력하시오 !

```
1
2
3
4
5
답:
for i in range(1, 6):
    print (i)
```

문제265. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자까지 숫자가  
출력되게하시오 !

숫자를 입력하세요 ~ 5

```
1
2
3
4
5
답:
a = int( input('숫자를 입력하세요 ~ '))

for i in range(1, a+1):
    print (i)
```

문제266. 위의 코드를 수정해서 숫자를 물어보게하고 아래와 같이  
★ 이 출력되게하시오 !

숫자를 입력하시오 ~ 5

```
★
★★
★★★
★★★★
★★★★★

a = int( input('숫자를 입력하세요 ~ '))

for i in range(1, a+1):
    print (i * '★')
```

문제267. (알고리즘 22번 문제) 아래와 같이 사각형이 출력되게하시오 !

답글로 올리세요 ~

가로의 숫자를 입력하세요 ~ 3

세로의 숫자를 입력하세요 ~ 5

```
★★★★
★★★★
★★★★
★★★★
★★★★
```

답:

```
a= int(input('가로의 숫자를 입력하세요~'))
```

```
b= int(input('세로의 숫자를 입력하세요~'))
```

```
for i in range(1, b+1):
    print(a * '★')
```

■ 81. 문자열에서 특정 문자가 있는지 확인하기(in)

문자열에서 특정 문자가 있는지 없는지 확인하려면 in 키워드를  
이용합니다.

예제:

```
txt = 'abcdefghijklmnopqr'
```

```
if 'b' in txt:
```

```
    print ('존재합니다.')
```

```
else:
```

```
    print ('존재하지 않습니다')
```

문제268. emp2.csv 에서 이름을 출력하는데 이름에 s 를 포함하고 있는  
사원들의 이름을 출력하시오 !

```
SQL> select  ename
      from emp
      where ename like '%S%';
```

파이썬>

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    if 'S' in emp_list[1]: # if 'b' in txt:
        print( emp_list[1] )
```

```
JONES
JAMES
SMITH
SCOTT
ADAMS
```

설명: 위의 코드는 금융감독원에서 금융사에 요청하는것중에 하나가  
보험회사의 경우 보험 상담원들이 상담할때 적절한 보험용어를 써서  
고객을 응대를 해야해서 고객과의 통화를 다 녹음을 하고  
그 녹음된 내용을 text 로 변환해서 그 text 안에 단어들을 일일이  
확인하는 작업을 한다. ( 예: 30일 이라고 해야하는데 한달이라고 했다  
던가 ) 이럴때 응용될 수 있는 코드입니다.

문제269. 문제268번 코드를 수정해서 출력하는데 이름에 s 자가 포함된  
사원의 이름이 몇명인지 출력하시오 !

```

결과: 5
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
cnt = 0
for emp_list in emp_csv:
    if 'S' in emp_list[1]: # 이름에 s 자가 포함되어져 있으면
        cnt = cnt + 1

print (cnt)

```

## ■ 82. 문자열에서 특정 문자열이 있는지 확인하기(in)

보험회사에서 보험 상담원분들이 상담하실때의 상담내용중에 부적절한 용어가 있는지 없는지를 확인하여 금융감독원에 보고를 해야 하는데 이 확인작업을 사람이 하려면 많은 인력과 시간이 드니까 파이썬으로 쉽게 할 수 있도록 프로그래밍할 때 문자열에서 문자열이 있는지 확인하는 코드를 작성하면 됩니다.

예제: 겨울왕국 스크립트에서 elsa 라는 단어(문자열)가 몇번 나오는지 카운트 해봅니다. ( 파이썬 게시판에 winter.txt 로 올려입니다)

예제1. 겨울왕국(winter.txt) 스크립트를 파이썬으로 불러와서 출력하시오

```

winter = open('c:\\data\\winter.txt')
for i in winter:
    print (i)

```

예제2. 출력되는 겨울왕국 스크립트를 전부 소문자로 출력하시오 !

```

winter = open('c:\\data\\winter.txt')
for i in winter: # winter.txt 의 스크립트를 한행씩 가져오면서
    print ( i.lower() ) # 소문자로 변환해서 출력합니다.

```

예제3. 위에서 소문자로 출력된 스크립트 한행을 하나 가져와서 아래와 같이 실행하면 뭐가 나올까 ?

```

if 'elsa' in 'olaf skates and helps elsa coach anna.':
    print ('존재합니다')
else:
    print('존재하지 않습니다.')

```

예제4. 겨울왕국 스크립트에 elsa 가 몇번 나오는지 카운트 하시오 !

```

winter = open('c:\\data\\winter.txt')
cnt = 0
for i in winter: # winter.txt 의 스크립트를 한행씩 가져오면서
    if 'elsa' in i.lower() :
        cnt = cnt + 1

print (cnt) # 318

```

```

if 'elsa' in 'olaf skates and helps elsa coach anna. elsa is princess':
    cnt = cnt + 1

```

설명: 위의 코드는 한 행에 elsa 가 두번 나와도 cnt 는 1개만 카운트 됩니다.  
정확하게 하려면 위의 문자열에 elsa 가 2개가 나오니까 cnt 가 2개가 되어야합니다.

예제5. winter.txt 를 한행씩 읽어서 어절 단위로 출력되게하시오 !

```

winter = open("c:\\data\\winter.txt")
winter2 = winter.read().split(' ') # 스크립트를 공백단위로 분리해라~
for i in winter2:

```

```
print (i)
```

예제6. 위의 어절들이 소문자로 출력되게하시오 !

```
winter = open("c:\\data\\winter.txt")
winter2 = winter.read().split(' ') # 스크립트를 공백단위로 분리해라 ~
for i in winter2:
    print ( i.lower() )
```

문제270. (점심시간 문제) 겨울왕국 스크립트에서 elsa 라는 문자열(단어)이  
몇번 나오는가? 세원이 코드를 실행하세요 ~

```
winter = open("c:\\data\\winter.txt") # winter.txt 를 winter 로 불러온다.
winter2 = winter.read().split("\n") # 엔터로 먼저 줄 단위로 분리를 해야함
```

```
cnt = 0
for i in winter2:
    for j in i.split(' '):
        if 'elsa' in j.lower():
            cnt +=1
            print(j,cnt)
```

Elsa 329

오전에 배웠던 내용으로만 elsa 가 몇건 들어있는지 카운트

```
winter = open("c:\\data\\winter.txt") # winter.txt 를 winter 로 불러온다.
winter2 = winter.read().split(" ")
cnt = 0
for i in winter2:
    if 'elsa' in i.lower():
        cnt = cnt + 1
        print(i.lower(), cnt)
```

elsa 304

confidence. anna elsa? elsa love. elsa 305

설명: 위의 결과를 보면 304 다음에 elsa 가 3개가 있으므로 305가 되면 안되는

데 305가 된 이후는 confidence. anna elsa? elsa love. elsa 를  
하나의 문장으로 보았기 때문입니다. 어절별로 분리한걸로 코드를  
구현했지만 실제로는 어절별로 분리가 되지 않았습니다.  
그래서 위와같은 실수를 하지 않으려면 해결하는 방법은 ?

먼저 스크립트를 행단위로 먼저 분리를 하고나서 어절별로 분리를  
해야 합니다.

```
winter = open("c:\\data\\winter.txt")
winter2 = winter.read().split("\n") # 행단위로 분리하는 코드
```

```
cnt = 0
for i in winter2:
    for k in i.split(' '):
        print(k)
```

위에 방법은 세원이가 한 방법이고 아래는 건준이가 한 방법

```
winter=open('c:\\data\\winter.txt')
winter2=winter.read().split() # 아무것도 안쓰면 어절별로 나뉜다.
cnt=0 # 아무것도 안쓰면 기본값이 어절별입니다.
for i in winter2: # 어절별로 분리한 것들을 하나씩 가져오면서
    if 'elsa' in i.lower():
        cnt+=1
print(cnt)
```

4교시 신호 보냈습니다.

### ■ 83. 문자열 길이 구하기(len)

예제:

```
a='scott'
print ( len(a) ) # 5
```

문제271. 파이썬으로 emp2.csv 를 읽어 이름, 이름의 길이를 출력하시오 !

```
SQL> select  ename, length(ename)
        from emp;
```

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print( emp_list[1], len(emp_list[1]) )
```

### ■ 84. 문자열이 알파벳인지 검사하기(isalpha)

문자열은 문자나 숫자, 기호들로 구성이 됩니다. 코드를 작성하다 보면 특정 문자열이 한글이나 알파벳과 같이 사람의 언어를 표현하기 위해 사용되는 문자로만 구성되어 있는지 확인해야하는 경우가 있습니다. 파이썬 문자열 객체가 제공하는 메소드인 `isalpha()` 는 주어진 문자열이 사람의 언어 문자로만 구성되어있는지 확인해줍니다.

예제:

```
txt1 = 'Warcraft three'
txt2 = '안녕'
txt3 = '3PO'
```

```
print ( txt1.isalpha() ) # False 공백이 하나 있어서 False 가 나옴
print ( txt2.isalpha() ) # True 한글이나 다른 나라언어도 True 로 나옴
print ( txt3.isalpha() ) # False 숫자3 때문에 False
```

문제272. 스티븐 잡스 연설문인 jobs.txt 를 한행 한행씩 출력하시오 !

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
stev2 = stev.read().split("\n") # 한행씩 분리하는 작업
for i in stev2: # 한행씩 분리된 스크립트를 읽어서
    print(i)    # 한행씩 출력한다.
```

설명: 위의 코드를 실행했는지 cp949 에러가 나면 encoding 에 UTF8 도 써보고 ANSI 도 써보고 CP949 도 써보세요 ~

문제273. 위에서 한행씩 출력하고 있는 스크립트를 철자 하나씩 출력하시오

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
stev2 = stev.read().split("\n")
for i in stev2:
    for k in i: # 스크립트 한행을 읽어서 철자를 하나씩 불러오는 코드
        print(k) # 그 철자를 출력한다.
```

문제274. 위에서 출력된 철자가 알파벳이면 cnt 를 증가시켜서 스티븐 잡스 연설문에 알파벳이 몇개가 있는지 출력하시오 !

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
stev2 = stev.read().split("\n")
cnt = 0
for i in stev2:
    for k in i: # 스크립트 한행을 읽어서 철자를 하나씩 불러오는 코드
        if k.isalpha() == True:
            cnt = cnt + 1
```

```
print(cnt) # 9255
```

설명: 스티븐 잡스 연설문에서 숫자나 공백문자나 마침표와 같은 구두점을 뺀 알파벳 철자만 카운트 했습니다.

데이터 분석 사례중에 유명한 사례중 하나가 아프리카 케냐의 은행에서 고객들에게 대출을 해줄때 그 사람의 신용을 확인해서 대출을 해주는데 신용을 확인할 때 sns 의 그 사람에 대한 글을 분석해서 긍정적인 평가가 많은지 부정적인 평가가 많은지를 파악해서 대출심사에 반영을 하니까 훨씬 은행의 대출금 환수가 원활해졌던 사례에서 사용되고 있습니다.

#### ■ 85. 문자열이 숫자인지 검사하기(isdigit)

문자열 객체의 isdigit() 메소드는 문자열을 구성하는 요소가 모두 숫자인지 체크하고 True 또는 False 로 리턴합니다.

번호	내장함수	설명
1	isalpha()	알파벳(한글)이 맞는지 확인
2	isdigit()	숫자가 맞는지 확인
3	isspace()	공백이 맞는지 확인

문제275. 스티븐 잡스 연설문에는 숫자가 몇개가 있는지 출력하시오 !

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
stev2 = stev.read().split("\n")
cnt = 0
for i in stev2:
    for k in i: # 스크립트 한행을 읽어서 철자를 하나씩 불러오는 코드
        if k.isdigit() == True:
            cnt = cnt + 1
            print ( k, cnt )

print(cnt)
```

#### ■ 86. 문자열이 알파벳 또는 숫자인지 검사하기(isalnum)

알파벳(한글) 과 숫자를 동시에 확인하는 문자열 함수는 isalnum() 입니다.

예제:

```
a = 'A story is 2003'
for i in a:
    if i.isalnum() == True:
        print ( i )
```

\* 케냐 은행의 데이터 분석 사례를 파이썬 코드로 구현해 봅니다.  
( 긍정단어집, 부정단어집을 다운로드 받으세요 )

예제1. 긍정단어(positive-word.txt) 를 파이썬으로 읽어서 한행씩 출력하시오

```
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n")

for i in pos:
    print(i)

#%%%

e.g 2)
#%%%
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n") # positive 스크립트행을 엔터로 구분한 단어들을 pos에 담아라!
```



print(pos) # 결과물이 너무 수가 많지만, (무튼) 내용들이 list 안에 담겨 있는 것을 볼 수 있다.

print(type(pos)) # <class 'list'> 라고 나온다.

# 긍정단어들이 리스트에 담겨 있고, 그 리스트의 이름은 pos 이다.

###

e.g 3) 아래의 단어가 긍정 단어 리스트 중에 있는지 확인하시오

###

```
if 'wonderful' in ['wonder', 'wonderful', 'wonderfully', 'wonderous', 'wonderously', 'wonders', 'wondrous', 'woo']:
    print('존재합니다.')
```

```
else:
```

```
    print('존재하지 않습니다.')
```

# e.g2 & e.g3 을 활용해서 긍정 단어 부정단어 개수나 유무 여부를 체크할 수 있다.

###

```
positive = open("c:\\data\\positive-words.txt")
```

```
pos = positive.read().split("\n") # positive 스크립트행을 엔터로 구분한 단어들을 pos에 담아라!
```

```
if 'wonderful' in pos:
```

```
    print('존재합니다.')
```

```
else:
```

```
    print('존재하지 않습니다.')
```

###

e.g 4) (케냐 사례 구현연) 스티브 잡스 연설문 (jobs.txt) 를 읽어서 한 단어씩 출력하시오.

문제 272코드와 같음.

###

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
```

```
stev2 = stev.read().split("\n")
```

```
for i in stev2:
```

```
    for k in i.split(' '):
```

```
        print(k)
```

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
```

```
stev2 = stev.read().split() # 아무것도 쓰지 않으면, 어절별로 분리해서 stev2에 저장하라는 뜻.
```

```
print(stev2) #list 에 담겨있다.
```

```
for i in stev2: # stev2 리스트에서 하나씩 빼낸다. (어절별로 분리되어서 1단위가 1개가 됨)
```

```
    print(i)
```

###

문제 276번. 스티브 잡스 연설문에는 긍정단어가 몇개가 있는지 확인하시오.

###

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
```

```
stev2 = stev.read().split() # 스티브 잡스 연설문을 어절별로 분리해 stev2에 입력하라. stev2는 리스트이고, 잡스의 연설문이 어절별로
```

```
positive = open("c:\\data\\positive-words.txt") #저장되어 있음.
```

```
pos = positive.read().split("\n") #positive 에서 '행' 별로 분리해서 pos 에 담아라.
```

```
    # pos는 리스트이고, 긍정단어들이 들어있는 곳
```

```
cnt=0
```

```
for i in stev2: #stev2 에서 단어를 하나씩 빼온다.
```

```
if i.lower() in pos: #만약 stev2 에서 빼온 단어중에 pos 와 일치하는 단어가 있다면 (stev2 자료에 대문자도 있어서 lower 돌려줌)
    cnt+=1 #cnt 를 1개 증가시키시오.
```

```
print('긍정단어의 수는', cnt, '개 입니다.')
```

```
###
```

087 문자열에서 대소문자 변환하기 (upper, lower)

문자열 객체의 upper() method(기능) 은 문자열에 있는 모든 알파벳을 대문자로 변환한 결과를 리턴한다.

문자열 객체의 lower() method(기능) 은 문자열에 있는 모든 알파벳을 소문자로 변환한 결과를 리턴한다.

```
e.g)
###
txt = 'A lot of things occur each day'
```

```
result1 = txt.upper()
result2 = txt.lower()
```

```
print(result1) # A LOT OF THINGS OCCUR EACH DAY
print(result2) # a lot of things occur each day
```

```
###
```

문제 277. emp2.csv 에서 이름과 월급을 출력하시오.

```
###
```

```
import csv
```

```
file = open("c:\\data\\emp2.csv")
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    print (emp_list[1], emp_list[5])
```

```
###
```

문제 278. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 이름과 월급이 출력되게 하는데 입력값이 소문자이든 대문자이든 상관없이 같은 값이 잘 출력되게 하시오.

```
###
```

```
import csv
```

```
file = open("c:\\data\\emp2.csv")
```

```
emp_csv = csv.reader(file)
```

```
name = input('이름을 입력하시오~')
```

```
for emp_list in emp_csv:
    if emp_list[1].lower() == name.lower():
        print (emp_list[1], emp_list[5])
```

```
###
```

088 문자열에서 좌우 공백 제거하기 (lstrip, rstrip, strip)

함수 설명

lstrip 문자열에서 존재하는 왼쪽 공백을 제거

rstrip 문자열에서 존재하는 오른쪽 공백을 제거

strip 문자열에서 존재하는 양쪽 공백을 제거

```
e.g)
#%%
txt7 = '          양쪽에 공백이 있는 문자열 입니다.          '
print(txt7.lstrip())
print(txt7.rstrip())
print(txt7.strip())
```

```
#%%
```

양쪽에 공백이 있는 문자열 입니다.

양쪽에 공백이 있는 문자열 입니다.

양쪽에 공백이 있는 문자열 입니다.

보통은 strip 을 많이 쓴다.

```
#%%
```

문제 279. 스티브 잡스 연설문에서는 정관사 the 가 몇번 나오는가?

```
#%%
```

```
stev = open("c:\\data\\jobs.txt", encoding='UTF8')
```

```
stev2 = stev.read().split() # 스티브 잡스 연설문을 어절별로 분리해 stev2에 입력하라. stev2는 리스트이고, 잡스의 연설문이 어절별로
```

```
cnt=0
```

```
for i in stev2: #stev2 에서 단어를 하나씩 빼온다.
```

```
    if i.lower().strip() in 'the': #만약 stev2 에서 빼온 단어중에 pos 와 일치하는 단어가 있다면 (stev2 자료에 대문자도 있어서 lower 돌려줌)
```

```
        cnt+=1 #cnt 를 1개 증가시키시오.
```

```
        print(i.lower(),cnt)
```

```
print('잡스의 연설문에서 정관사 the 의 개수는' , cnt , '개 입니다.')
```

```
#%%
```

089 문자열에서 문자 개수 구하기 (count)

문자열 객체의 count() method 는 문자열에서 특정 문자의 개수를 리턴한다.

```
#%%
```

```
txt = ' A lot of things occur each day. Today is beautiful day'
```

```
print( txt.count('day'))
```

```
#%%
```

문제 280. 안철수 연설문에는 국민이라는 단어가 몇번 나오는지 카운트 하시오. (ahn.txt)

```
#%%
```

```
ahn = open("c:\\data\\ahn.txt", encoding='UTF8')
```

```
ahn2 = ahn.read() # 위의 예시와 다르게 split() 없이 그냥 안철수 연설문을 읽어서 통채로 ahn2에 입력한다.
```

```
print( ahn2) #안철수 연설문 전체가 출력됨.
```

```
print( ahn2.count('국민')) # 연설문 전체에서 국민이라는 단어가 몇개 있는지 출력한다.
```

```
#%%
```

■ 파이썬에서 막대 그리프 그리는 방법

```
#%%
```

```
import matplotlib.pyplot as plt
```

```
y_value = [21.6, 23.6, 45.8, 77.0]
x_index = [0,1,2,3]
```

```
plt.bar( x_index, y_value,color='purple')
```

```
plt.show()
```

```
###
```

문제 281. 동전을 10번 던져서 앞면이 2개가 나오는 지난번 마지막 문제의 확률 10개를 가지고 막대 그래프를 그리시오.

```
###
```

```
import matplotlib.pyplot as plt
```

```
y_value = [0.00191,0.01,0.07,0.16]
x_index = [0, 1, 2,3]
```

```
plt.bar( x_index, y_value,color='purple')
```

```
plt.show()
```

```
###
```

문제 282. 위의 그래프 결과에 제목, x축 라벨과 y축 라벨을 같이 출력하시오.

```
###
```

```
import matplotlib.pyplot as plt
```

```
y_value = [0.00191,0.01,0.07,0.16]
x_index = [0, 1, 2,3]
```

```
plt.bar( x_index, y_value,color='purple')
```

```
plt.title('coin Probability') #그래프 이름
```

```
plt.xlabel('cnt') #그래프 x축 라벨
```

```
plt.ylabel('probability') #그래프 y축 라벨
```

```
show()
```

```
###
```

(오늘의 마지막 문제)

문제 283. 지난번 마지막 문제인 동전을 10번 던져서 앞면이 2개 나오는

아래의 확률 결과를 가지고 막대그래프로 시각화 하시오!

```
###
```

```
import matplotlib.pyplot as plt
```

```
y_value = []
```

```
x_index = []
```

```
def coin_prob(num):
```

```
    import random
```

```
    coin = [ '앞면', '뒷면' ]
```

```
    cnt = 0
```

```
    for k in range(1, 10001):
```

```
        a = []
```

```
        for i in range(1, 11):
```

```
            result = random.choice(coin)
```

```
            a.append(result)
```

```
        if a.count('앞면')== num:
```

```
            cnt += 1
```

```
    return cnt/10000
```

```
for j in range(0, 11):
```

```
    y_value.append(coin_prob(j))
```

```
    x_index.append(j)
```

```
plt.bar( x_index, y_value,color='purple')
plt.title('coin Probability') #그래프 이름
plt.xlabel('tries') #그래프 x축 라벨
plt.ylabel('probability') #그래프 y축 라벨
show()
```

20.12.08 파이썬 강의 Day 10(문제284~문제318)

문제284. 우리반 데이터(emp122.csv)를 파이썬으로 로드해서  
이메일만 출력하시오 !

```
import csv
file = open("c:\\data\\emp122.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[6] )
```

문제285. 위에서 출력한 이메일에서 @의 위치 인덱스 번호를  
출력하시오 !

```
import csv
file = open("c:\\data\\emp122.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[6].find('@') )
```

문제286. 그러면 이번에는 이메일에서 .의 위치 인덱스를 출력하시오

```
import csv
file = open("c:\\data\\emp122.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[6].find('.') )
```

문제287. 이번에는 이메일을 출력하는데 이메일의 첫번째 철자부터  
세번째 철자까지만 출력하시오 !

```
import csv
file = open("c:\\data\\emp122.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print ( emp_list[6][0:4] )
```

문제288. 우리반 데이터 이메일을 출력하는데 이메일에서 도메인만 출력하시오

```
gmail
gmail
naver
naver
:
:
```

답:

```
import csv
file = open("c:\\data\\emp123.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    a = emp_list[6].find('@') # @의 위치 인덱스 번호
    b = emp_list[6].find('.') # .의 위치 인덱스 번호
    print ( emp_list[6][ a +1 : b ] )
```

문제289. 미승이의 이메일인 miseung.hailey@gmail.com도 같이  
출력하려면 어떻게 해야하는가 ?

```
import csv
```

```

file = open("c:\\data\\emp123.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    a = emp_list[6].find('@') # @ 의 위치 인덱스 번호
    b = emp_list[6].rfind('.') # . 의 위치 인덱스 번호를 뒤에서 부터 찾음
    print ( emp_list[6][ a +1 : b ] )

```

문제290. rfind도 domain.co.kr 같은곳은 domain.co까지 끊겨서  
문제생기지 않을까요?

## ■ 91. 문자열을 특정 문자(열)로 분리하기(split)

코드를 작성할 때 가장 많이 접하게 되는 경우가 구분자(separator)  
로 구분되어있는 문자열 파싱(parsing) 을 하는 일입니다.  
이때 문자열 객체의 split() 를 활용하면 구분자를 기준으로  
문자열을 쉽게 분리하여 파싱할 수 있습니다.

```

a = "1,"이준혁",29,"남","경영학과"
print( type(a) ) # <class 'str'>
result = a.split(',') # 콤마(,) 로 구분해서 리스트에 담는다.

print (result) # ['1', "'이준혁'", '29', "'남'", "'경영학과'']

```

설명: 1,"이준혁",29,"남","경영학과" <--- 이 데이터를 문자열 변수에 할당  
하고 싶는데 문자열들 중에 더블 쿼테이션 마크가 있다면 양쪽에  
싱글 쿼테이션 마크를 3개를 써서 둘러줘야 합니다.

\* 정리:

1. a = 문자열.split(',') : 문자열을 콤마(,) 로 구분해서 구분하는 문자들을  
요소로 해서 a 리스트에 담는다.
2. a = 문자열.split(' ') : 문자열을 공백으로 구분해서 구분하는 문자들을  
요소로 해서 a 리스트에 담는다.
3. a = 문자열.split() : 문자열을 어절별로 구분해서 구분하는 문자들을  
요소로 해서 a 리스트에 담는다.

\* 웹로그 데이터 분석하는 선배기수가 모사의 웹 사이트에 접속하는  
고객들이 해당 사이트에서 어느 상품을 많이 클릭을 하는지등을  
조회하거나 아니면 웹사이트에서 어느 페이지를 많이 보는지등을  
분석하는 업무를 한다.

문제291. 아래의 url 변수에 있는 문자열은 슬래쉬(/) 로 구분되어있는데  
이 문자열의 요소를 아래의 리스트 처럼 구성하시오 !

```
url = 'http://www.naver.com/news/today=20191204'
```

결과: ['http:', '', 'www.naver.com', 'news', 'today=20191204' ]

답:

```

url = 'http://www.naver.com/news/today=20191204'
result = url.split('/')
print (result)

```

```
['http:', '', 'www.naver.com', 'news', 'today=20191204']
```

## ■ 92. 문자열을 특정 문자(열)로 결합하기(join)

문자열 객체의 join() 메소드는 split() 와는 반대로 문자열이  
요소인 리스트를 인자로 받아 리스트 모든 용소를 특정 문자열로  
연결하여 새로운 문자열로 만들어 리턴합니다.

예제:

```
a = ['http:', ' ', 'www.naver.com', 'news', 'today=20191204']
bond = '/'
url = bond.join(a)
print(url)
```

<http://www.naver.com/news/today=20191204>

설명: 문자열 함수인 join 을 이용해서 리스트의 요소를 불러와서  
다시 문자열로 만들수 있습니다.

문제292. 우리반 데이터에서 이름만 출력하시오 !

```
import csv
file = open("c:\\data\\emp123.csv", encoding="UTF8")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1])
```

문제293. 위에서 출력된 이름을 a 라는 비어있는 리스트에 담고  
a 리스트를 출력하시오 !

```
import csv
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append(emp_list[1])
```

```
print(a)
```

['이준혁', '한결', '현지연', '성기창', '유혜영', '김정민', '이성원', '김정원', '김미승', '김예린', '신현종', '구윤모', '김승순', '허정민', '이신성', '황나현', '장수진', '권준환', '송  
종미', '정희원', '김홍비', '김소라', '권세원', '정다희', '허혁', '양건준', '김주원', '박혜진', '하상준', '홍재연']

문제294. 위의 a 리스트에 담겨진 이름을 하나씩 뽑아서 콤마(,) 로 연결해서  
아래와 같이 문자열로 출력되게하시오

이준혁,한결,현지연,성기창,유혜영,.....

```
import csv
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append(emp_list[1])
bond=','
result = bond.join(a)
print(result)
```

문제295. 위의 결과가 김씨부터 출력되게하시오~

(한글을 정렬을 해서 출력하시오 ㄱ ㄴ ㄷ ㄹ 순서로)

```
import csv
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append(emp_list[1])
a.sort() # a 리스트 안의 요소를 정렬을 합니다.
bond=','
result = bond.join(a)
print(result)
```

구윤모,권세원,권준환,김미승,김소라,김승순,김예린,김정민,김정원,김주원,김홍비,박혜진,성기창,송종미,신현종,양건준,유혜영,이성원,이신성,이준혁,장수진,정다희,  
정희원,하상준,한결,허정민,허혁,현지연,홍재연,황나현

### ■ 93. 문자열에서 특정 문자(열)를 다른 문자(열)로 바꾸기(replace)

문자열 객체의 replace() 메소드는 문자열에서 특정 문자나 문자열을 다른 문자나 문자열로 변경합니다.

예제:

```
txt = 'My password is 1234'
```

```
result = txt.replace('1', '0') # txt 문자열의 숫자 1을 숫자 0 으로 변경해라  
print ( result )
```

문제296. emp2.csv 에서 이름과 월급을 출력하는데 월급을 출력할 때  
0 대신에 \* 로 출력하시오 !

```
SCOTT 3***  
SMITH 125*  
:      :  
import csv  
file = open("c:\\data\\emp2.csv")  
emp_csv=csv.reader(file)  
  
for emp_list in emp_csv:  
    print( emp_list[1], emp_list[5].replace('0','*') )
```

문제297. 아래의 리스트를 가지고 아래와 같이 출력하시오 !

```
a = [ 'name:홍길동', 'age:17', 'major:경영학', 'nation:한국' ]
```

결과:

```
name ----> 홍길동  
age ----> 17  
major ----> 경영학  
nation ----> 한국
```

```
a = [ 'name:홍길동', 'age:17', 'major:경영학', 'nation:한국' ]
```

```
for i in a:  
    print ( i.replace(':', '---->') )
```

### ■ 94. 문자열을 바이트 객체로 바꾸기(encode)

```
file = open("c:\\data\\emp1222.csv", encoding="UTF8")
```

인코딩이란 ?

인코딩이란 사람이 인지할 수 있는 형태의 데이터를 약속된 규칙에 의해 컴퓨터가 사용하는 0 과 1로 변환하는 과정을 말합니다.  
컴퓨터의 발명, 발전이 미국의 학계와 기업을 중심으로 이루어지다 보니 문자 집합(컴퓨터에서 문자를 담는 저장소)도 미국을 기준으로 제정되었습니다.  
미국에서 제정된 ASCII(미국 정보 교환 표준부호)는 1960년대에 제정된 문자집합으로 이후 개발된 문자 집합의 토대를 이루고 있습니다.  
ASCII 는 7비트 만을 이용해서 음이 아닌수(0~127)의 문자집합내의 문자 할당 약속입니다.

예제: A 는 숫자 65 입니다. a 는 숫자 97 입니다.

구글에서 아스키 코드표를 검색해보세요 ~~

예제: chr 함수를 이용해서 숫자 65부터 127까지가 어떤 문자인지  
출력하시오 !

```
for i in range(65, 128):  
    print ( i, '---->', chr(i) ) # 컴퓨터가 알고있는 숫자가 문자로 무엇인지
```



이 아스키 코드표로 미국 엔지니어들은 행복하게 살아가고 있었지만  
다른 나라 사람들은 그러지 못했습니다.

## 1. 서유럽 국가

독일어의 움라우트, 스페인어의 물결표, 터키어의 시딜러, 프랑스어의 악센트  
는 7비트로 표현 못합니다.

## 2. 아시아

아시아에서는 한자, 일본어, 한글이 7비트에 들어맞지가 않았습니다.  
더 큰 문자셋이 필요했는데 16비트로된 문자집합을 제공해서 해결했습니다.  
그 문자 집합셋이 바로 CP949, EUC-KR 입니다.

```
file = open("c:\\data\\emp1222.csv", encoding="CP949")
```

포스코 더 샵 아파트에서 사는 사람들의 주소를 컴퓨터 입력할 때  
어떻게 보이냐면 포스코 더 ? 아파트로 보였다.  
그래서 해결한게 바로 유니코드 문자 집합셋입니다.  
바로 그게 UTF8 , UTF16 입니다.

```
file = open("c:\\data\\emp1222.csv", encoding="UTF8")
```

파이썬에서 한글이 ?? 로 출력이 되면 UTF8 로 인코딩하면 됩니다.

문제298. (점심시간 문제) 우리반 데이터에서 이름을 출력하는데  
아래와 같이 나이도 같이 출력되게하시오 !

구윤모(26),권세원(34),권준환,김미승,김소라,김승순,김예린,김정민,김정원,김주원,김홍비,박혜진,성기창,송종미,신현종,양건준,유혜영,이성원,이신성,이준혁,장수진,  
정다희,정희원,하상준,한결,허정민,허혁,현지연,홍재연,황나현

올리고 식사하러가세요 ~~~~

못했으면 점심 먹고 와서 하세요 ~

## ■ 95. 바이트 객체를 문자열로 바꾸기(decode)

```
file = open("c:\\data\\emp1222.csv", encoding="UTF8")
```

인코딩(encoding) ?

사람이 알아 볼 수 있는 언어 -----> 컴퓨터가 알아볼 수 있는 언어(숫자)

디코딩(decoding) ?

컴퓨터가 알아 볼 수 있는 언어 -----> 사람이 알아 볼 수 있는 언어

예제:

```
txt = 'A'
```

b\_txt = txt.encode() # txt 변수에 들어있는 대문자 A 를 인코딩한다.

```
print(b_txt) # b'A' --> binary 'A' 라는 뜻
```

(이진) --> 0과 1로 표현하는 데이터

컴퓨터가 알아볼 수 있는 바이너리 데이터

```
print(txt) # 사람이 알아볼 수 있는 언어인 알파벳 A
```

```
c_txt = b_txt.decode() # 컴퓨터가 알아볼 수 있는 언어--> 사람이 알아볼수
```

# 있는 언어로 변경

```
print(c_txt) # A
```

```
file = open("c:\\data\\emp1222.csv", encoding="UTF8")
```

설명: emp1222.csv 를 파이썬으로 로드하는데 파이썬으로 load 하려면

컴퓨터가 알아볼 수 있는 언어로 변경해야 합니다.  
이때 문자집합셋을 유니코드 문자집합셋인 UTF8 을 이용해서  
인코딩 즉 컴퓨터가 알아볼 수있는 언어로 변경하겠다는 것입니다.

#### ■ 96. 순차적인 정수 리스트 만들기(range)

[ 0, 1, 2, 3 ] 이나 [ 100, 101, 102, 103 ] 과 같이 순차적인 정수 리스트를  
만드는 가장 간단한 방법은 파이썬 내장함수인 range() 를 이용하는 것입니다.

예제:

```
print ( range(1, 11) ) # range(1, 11)
print ( list( range(1, 11) ) ) # [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

문제299. 아래와 같이 주사위의 눈을 담은 리스트를 만드시오 !

```
print (dice)
```

결과: [ 1, 2, 3, 4, 5, 6 ]

답 :

```
dice = list( range(1, 7) )
print (dice)
```

문제300. 주사위 2개를 만들고 주사위 2개를 동시에 던져서  
두 주사위의 눈의 합이 10 이 나오는 확률을 구하시오 !  
( 주사위 2개를 동시에 10000 번 던지세요 ~)

```
import random
```

```
dice1 = list( range(1, 7) )
dice2 = list( range(1, 7) )
cnt = 0
for i in range(1, 10001):
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt = cnt + 1
```

```
print(cnt/10000) # 0.084
```

문제301. 아래의 결과를 출력하시오!

```
[ 2, 4, 6, 7, 10, 12, 14, 16, 18 ]
```

```
a = list( range(2, 20, 2) ) # range(시작숫자, 끝숫자, 스텝)
print (a)                  # 시작숫자부터 끝숫자 미만까지 스텝만큼 증가해서
                           # 출력한다.
```

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

#### ■ 97. 리스트에서 특정 위치의 요소 얻기

리스트의 특정 위치에 있는 요소값을 구하려면 인덱싱을 이용합니다.  
인덱스는 0 부터 시작합니다.

```
a = [ 1, 2, 'a', 'b', 'c', [ 4, 5, 6 ] ]
print (a)
```

예제1. 위의 리스트 a 에서 숫자 2를 출력하시오 !

```
a = [ 1, 2, 'a', 'b', 'c', [ 4, 5, 6 ] ]
```

```
print ( a[1] )
```

예제2. 위의 리스트 a 에서 숫자 4를 출력하시오 !

```
a = [ 1, 2, 'a', 'b', 'c', [ 4, 5, 6 ] ]
```

```
print(a[5]) # [4, 5, 6]
print(a[5][0]) # 4
```

문제302. 아래의 리스트에서 숫자 7을 출력하시오 !

```
b = [ 2, 3, 4, [ 5, 6 ], [ 7, 8 ], 9 ]
```

```
print(b[4][0])
```

#### ■ 98. 리스트에서 특정 요소의 위치 구하기(index)

리스트 객체의 index() 메소드는 리스트에서 요소의 값을 알고 있을때  
그 요소가 최초로 나타나는 위치의 인덱스를 리턴합니다.

예제:

```
a = [ 1, 2, 'a', 'b', 'c', [ 1, 2, 3 ] ]
    0 1 2 3 4 5
print ( a.index(2) ) # 1
print ( a.index('a') ) # 2
```

문제303. 아래의 리스트에서 숫자 4의 인덱스 번호를 출력하시오!

```
a = [ 1, 2, 'a', 'b', 'c', [ 4, 5, 6 ] ]
```

```
print(a.index(4))
```

ValueError: 4 is not in list

답: 리스트 안에 있는 리스트의 인덱스 번호는 출력못함

#### ■ 99. 리스트에서 특정 위치의 요소를 변경하기

```
list_a = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g' ]
print ( list_a[0] ) # a
```

```
list_a[0] = 'z'
print (list_a) # [ 'z', 'b', 'c', 'd', 'e', 'f', 'g' ]
```

문제304. list\_a 의 알파벳 d 를 k 로 변경하시오 !

```
list_a = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g' ]
```

```
list_a[ list_a.index('d') ] = 'k'
print( list_a ) # ['a', 'b', 'c', 'k', 'e', 'f', 'g']
```

#### ■ 100. 리스트에서 특정 구간에 있는 요소 추출하기

리스트에서 특정 구간에 있는 요소를 추출하여 새로운 리스트를  
만들려면 슬라이싱을 이용하면 됩니다.

예제:

```
list_a = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g' ]
print ( list_a [0 : 4] ) # ['a', 'b', 'c', 'd']
print ( list_a [ 2 : ] ) # 리스트의 인덱스 2번째 요소부터 읽어서 끝까지 읽으라
print ( list_a [ : 3 ] ) # 리스트의 처음부터 읽어서 인덱스 번호 3 미만까지
                        # 읽으라
```

```
['a', 'b', 'c', 'd']
['c', 'd', 'e', 'f', 'g']
['a', 'b', 'c']
```

문제305. 위에는 알파벳 a 부터 g 까지 담은 리스트를 만들었는데

이번에는 알파벳 a 부터 z 까지를 담은 리스트를 list\_a 로  
생성하시오 ~

```
['a', 'b', 'c', 'd', .....,'z']
```

import string # string 모듈안에 알파벳이 들어있습니다.

```
list_a = []
for i in string.ascii_lowercase:
    list_a.append(i)
print ( list_a )
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```

문제306. 위에서 만든 list\_a 에서 요소를 검색하는데 맨 끝에 알파벳 z  
빼고 모든 요소를 다 출력하시오 !

```
['a', 'b', 'c', 'd', .....,'y']
```

import string # string 모듈안에 알파벳이 들어있습니다.

```
list_a = []
for i in string.ascii_lowercase:
    list_a.append(i)
print ( list_a[0 : list_a.index('z')] )
```

#### ■ 101. 리스트에서 짝수 번째 요소만 추출하기

리스트에서 짝수번째 요소를 추출하는 방법은 두번째 요소부터  
끝까지 스텝 2 로 슬라이싱 하면 됩니다.

```
list_a = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h' ]
print ( list_a[1: : 2] ) # 인덱스 번호 1번째 요소부터 끝까지 2 스텝으로 읽어라

['b', 'd', 'f', 'h']
```

문제307. 1부터 100까지의 숫자중에 짝수만 아래의 list\_a 담아서 출력하시오

```
print (list_a)

[ 2, 4, 6, 8 , 10, 12, ..... 100 ]
```

답:  
list\_a = list( range(2, 101, 2 ) )  
print (list\_a)

#### ■ 102. 리스트 요소 순서를 역순으로 만들기 ① (reverse)

리스트 객체의 reverse() 메소드는 리스트의 모든 요소 순서를 거꾸로  
만듭니다.

예제:  
list\_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']  
list\_a.reverse()  
print (list\_a)

문제308. 우리반 테이블에서 이름을 출력하는데 ㅎ 부터 ㄱ 까지 출력될수  
있도록 하시오

```
import csv
file=open('c:\\data\\emp122.csv',encoding='UTF8')
emp_122 =csv.reader(file)
a=[]
for emp_list in emp_122:
    a.append(emp_list[1])
a.sort()
a.reverse()
```

```
result = ','.join(a)
print(result)
홍재연,하상준,박혜진,김주원,양건준,허혁,정다희,권세원,김소라,김홍비,정희원,송종미,권준환,장수진,황나현,이신성,허정민,김승순,구윤모,신현종,김예린,김미승,김
```

문제309. emp2.csv 에서 이름과 월급을 출력하시오 !

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv=csv.reader(file)

for emp_list in emp_csv:
    print( emp_list[1], emp_list[5] )
```

문제310. 위의 데이터중 월급을 비어있는 리스트인 a 리스트에 담으시오

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv=csv.reader(file)
a = [ ]
for emp_list in emp_csv:
    a.append( emp_list[5] )
print(a)
```

문제311. 위의 a 리스트에 있는 월급을 월급이 높은 순서대로 정렬해서  
a 리스트에 들어있게 하고 a 리스트를 출력하시오

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv=csv.reader(file)
a = [ ]
for emp_list in emp_csv:
    a.append( int( emp_list[5] ) )
a.sort() # 먼저 낮은값에 높은값 순으로 정렬하고
a.reverse() # 낮은값에서 높은값 순으로 정렬된 a 리스트의 데이터를
# 역순으로 만듭니다.
print(a)
```

#### ■ 103. 리스트 요소 순서를 역순으로 만들기 ② (reversed)

리스트 요소 순서를 역순으로 만드는 또 다른 방법은 파이썬 내장 함수인  
reversed() 이용하는 방법입니다.  
reverse() 는 인자로 입력된 시퀀스 자료형의 요소 순서를 역순으로  
새로운 시퀀스 자료형을 만들어 리턴합니다.

reverse() 함수 : 원본 데이터를 역순으로 만듭니다.  
reversed() 함수 : 역순으로 만들지만 원본 데이터를 역순으로 만들지 않습니다.

예제:  
list\_a = ['a', 'b', 'c', 'd', 'e']  
list\_a.reverse() # list\_a 의 요소를 역순으로 변경한다.  
print (list\_a) # ['e', 'd', 'c', 'b', 'a']

```
list_b = ['a', 'b', 'c', 'd', 'e']
result = reversed( list_b ) # list_b 의 요소를 역순으로 만들어서 result 에 저장
print( list(result) ) # ['e', 'd', 'c', 'b', 'a']
print (list_b) # ['a', 'b', 'c', 'd', 'e']
```

45분까지 쉬세요 ~~

#### ■ 104. 리스트 합치기( )

두개의 리스트를 연결하여 새로운 리스트를 만드는 방법은 + 연산자를  
이용해 두개의 리스트를 더하면 됩니다.

예제:

```
listdata1 = [ 'a', 'b', 'c', 'd' ]
listdata2 = [ 'f', 'g', 'h', 'i' ]
listdata3 = listdata1 + listdata2
print (listdata3)
```

```
['a', 'b', 'c', 'd', 'f', 'g', 'h', 'i']
```

문제312. 아래와 같이 엄마와 아기가 함께하는 수영교실 나이 리스트를  
생성하시오 ! 일일이 숫자를 쓰지말고 파이썬 코드로 간단하게  
만들어보세요

```
[ 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

```
답:
listdata1 = [34]
listdata2 = [2]
print ( listdata1 * 10 + listdata2 * 2 )
```

#### ■ 105. 리스트 반복하기(\*)

list\*n 은 리스트를 n 번 반복해서 새로운 리스트를 만듭니다.  
예를들어, list\*2 는 list + list 와 동일한 결과가 출력됩니다.

```
예제:
listdata = list( range(0,3) )
result = listdata * 3
print (result) # [0, 1, 2, 0, 1, 2, 0, 1, 2]
```

문제313. 아래의 엄마와 아기가 함께하는 수영교실의 나이의 평균값을  
출력하시오 !

```
swim_age = [ 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2 ]

import numpy as np
a = np.array(swim_age) # numpy array(배열) 로 만들어준다.
b = np.mean(a)
print(b) # 18.0
```

문제314. 아래의 엄마와 아기의 수영교실의 중앙값을 출력하시오 !  
( 중앙값은 가운데 있는 값)

```
swim_age = [ 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2 ]

import numpy as np
a = np.array(swim_age) # numpy array(배열) 로 만들어준다.
b = np.median(a)
print(b) # 18.0
```

문제315. 엄마와 아기가 함께하는 수영교실 나이 데이터의 최빈값을 출력하시오

```
swim_age = [ 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2 ]

from scipy import stats
import numpy as np
a = np.array(swim_age)
result = stats.mode(a)
print (result) # ModeResult(mode=array([2]), count=array([10]))
```

문제316. 우리반 나이 데이터를 비어있는 리스트 a 에 담으시오 !  
( emp1222.csv 를 읽으세요 ~)

```
import csv
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append( int(emp_list[2] ) )
print(a)
```

[29, 31, 35, 29, 28, 28, 25, 28, 27, 27, 27, 26, 33, 30, 27, 27, 25, 28, 29, 24, 24, 29, 36, 26, 26, 26, 44, 28, 29, 28]

문제317. 우리반 나이 데이터의 평균값과 중앙값과 최빈값을 출력하시오 !

```
import csv
import numpy as np
from scipy import stats
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append( int(emp_list[2] ) )
a2 = np.array(a)
print('평균값:', np.mean(a2))
print('중앙값:', np.median(a2))
print('최빈값:', stats.mode(a2) )
```

파이썬 자료형 5가지

#### ■ 106. 리스트에 요소 추가하기(append)

리스트 객체의 append() 메소드는 인자로 입력된 값을 리스트의 맨 마지막 요소로 추가합니다.

예제:

```
listdata = []
listdata.append('a')
listdata.append('b')
print(listdata) # ['a', 'b']
```

#### ■ 파이썬으로 히스토그램 그래프 그리기

히스토그램 그래프는 계급을 가로축에 도수를 세로축에 나타낸뒤  
각 계급의 크기를 가로의 길이로 하고 도수를 세로의 길이로 하는  
직사각형을 차례대로 그려서 나타낸 그래프

예제:

1. 평균이 150이고 표준편차가 5인 초등학생 10만명의 키를 생성한다.

```
import numpy as np
height = np.random.randn(100000) * 5 + 150
```

2. 계급의 크기를 나타내는 가로의 길이를 설정

```
bins = [ 142, 144, 146, 148, 150, 152, 154, 156, 158, 160 ]
```

3. 히스토그램 그래프의 세로축에 해당하는 도수(건수)를 구하는 코드

```
hist, bins = np.histogram( height, bins )
print(hist, bins )
```

```
[ 6144 9650 13318 15366 15601 13326 9715 6079 3221]
[142 144 146 148 150 152 154 156 158 160]
```

142 ~ 144 미만 : 6144 명

144 ~ 146 미만 : 9650 명

: :

158 ~ 160 미만 : 3221 명

4. 위의 데이터를 가지고 히스토그램 그래프를 그린다.

```
import matplotlib.pyplot as plt
import numpy as np
height = np.random.randn(100000) * 5 + 150
bins = [ 142, 144, 146, 148, 150, 152, 154, 156, 158, 160 ]
hist, bins = np.histogram( height, bins ) # 도수 확인하려고 한번 본 데이터
plt.grid() # 격자모양이 생김
plt.hist( height , bins , rwidth=0.9, alpha=0.7, color='red')
```

↑

## 초등학생키 10만명 데이터

rwidth --> 히스토그램 그래프의 넓이

alpha --> 색깔 투명도

문제318. (오늘의 마지막 문제) 우리반 나이 데이터를 가지고 히스토그램 그래프를 그리시오 !

x축의 계급(간격)은 24 ~ 44 (2살 간격으로)

24 ~ 26

26 ~ 28

:

42 ~ 44

```
import csv
import numpy as np
import matplotlib.pyplot as plt

emp12 = open("c:\\data\\emp123.csv",encoding = 'UTF-8')
emp12 = csv.reader(emp12)
next(emp12)      # 컬럼명 날리기
a=[]
for i in emp12:
    a.append(int(i[2]))
x_val = list(range(int(min(a)),int(max(a))+1,2))
a = np.array(a)

plt.xticks(x_val)      # x축 눈금 조절. 검색해서 찾았습니다.
plt.yticks(range(0,12)) # y축 눈금 조절.
plt.hist(a,x_val,rwidth = 0.8,)
```

온라인 Q/A 게시판에 올린 답 naver.co.kr 도 검색되게 하는 방법?

종미 코드

```
import csv
file = open("c:\\data\\emp123.csv",encoding='UTF8')
emp_csv=csv.reader(file)
for emp_list in emp_csv:
    a = emp_list[6].find('@') # 이메일에서 @ 의 위치 인덱스 번호
    result = emp_list[6][a:] # @naver.co.kr <-- @ 를 포함해서 끝까지 스캔
    b = result.find('.')      # dot(.) 의 위치 인덱스 번호
    print(result[1:b])        # @naver.co.kr 에서 인덱스 번호 1부터 b 미만까지
                                0 1
                                @naver.com
```

20.12.09 파이썬 강의 Day 11 (문제319~문제350)

### ■ 107. 리스트의 특정 위치에 요소 삽입하기(insert)

리스트 객체의 insert() 메소드는 리스트의 특정 위치의 새로운 요소를 입력합니다.

예제:

```
list_a = [ 'a', 'b', 'c', 'd', 'e' ]
```

```
list_a.append('f') # append 메소드는 무조건 맨뒤쪽에 요소를 추가한다.
```

```
print(list_a) # [ 'a', 'b', 'c', 'd', 'e', 'f' ]
```



```
list_a.insert( 3, 'k')
print (list_a) # ['a', 'b', 'c', 'k', 'd', 'e', 'f']
```

설명: list\_a.insert(3,'k') 는 list\_a 리스트의 인덱스번호 3번으로 k 를 구성해라

문제319. 아래의 a 리스트의 요소 화성 다음에 소행성을 요소로 입력하시오!

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
a.insert( a.index('화성') + 1, '소행성')
```

```
print(a)
```

#### ■ 108. 리스트의 특정 위치의 요소 제거하기(del)

\* 파이썬의 자료형(자료구조) 5가지와 해당 메소드와 if 문과 loop 문

1. 문자형 count, index,.....
2. 숫자형 int(), round() + if 문과 loop문
3. 리스트형 count, append, insert, del + 함수 생성, 클래스, 예외처리
4. 튜플형
5. 사전형

※ del 메소드와 remove 메소드의 차이 ?

둘다 리스트안의 요소를 삭제하는 것인데  
remove 는 요소명으로 삭제하는 것이고  
del 은 리스트의 인덱스 번호로 삭제하는 것입니다.

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del(a[4]) # a 리스트에 인덱스 번호 4번의 해당하는 요소를 삭제한다.
print(a)
```

문제320. 아래의 a 리스트에서 목성을 지우시오 !

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
del(a[ a.index('목성') ])
print(a)
```

#### ■ 109. 리스트에서 특정 요소 제거하기(remove)

리스트에서 특정 요소의 값을 알고 있을때 그 요소를 제거하는 방법은  
리스트 객체의 remove() 메소드를 이용하면 됩니다.

예제: 아래의 a 리스트에서 목성을 지우시오 !

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
a.remove('목성')
print(a)
```

문제321. 아래의 빨간공 2개와 파란공 3개가 들어있는 주머니에서  
공을 하나 랜덤으로 빼면 그 공이 주머니에서 지워지게 하시오 !

```
box = [ 'red', 'blue', 'red', 'blue', 'blue' ] 45분까지 쉬세요 ~~
```

```
import random

result = random.choice(box)
if result == 'red':
    box.remove('red')
else:
```

예린이와 현종이 답 45분까지 쉬세요~~

```
box = ['red', 'blue', 'red', 'blue', 'blue']
```

```
import random
a = random.choice(box)
box.remove(a)
print(box)
```

문제322. 아래의 주머니에서 임의로 하나의 공을 뽑았을 때  
 그 공이 파란색일 확률은 어떻게 되는가? (복원 추출)  
 공을 뽑은 작업을 10000 번 수행하시오!

```
box = ['red', 'blue', 'red', 'blue', 'blue']

import random
cnt = 0
for i in range(1, 10001):
    result = random.choice(box) # 실행문1
    if result == 'blue': # 실행문2
        cnt = cnt + 1

print ( cnt / 10000) # 0.5936
```

문제323. 파란공 70개와 빨간공 30개가 들어있는 주머니를 만드시오

```
a = ['blue'] * 70
b = ['red'] * 30
box = a + b
print (box)
```

[illegible]

문제324. 위의 box 에 들어있는 요소가 몇개인지 확인하시오!

```
a = ['blue'] * 70
b = ['red'] * 30
box = a + b
print(box)
print( len(box) ) # 100
```

문제325. 위의 box 에 있는 요소들을 무작위로 섞으시오!

```
from random import shuffle # from 패키지 import 모듈

a = ['blue'] * 70
b = ['red'] * 30
box = a + b
print(box)

shuffle(box) # box 안의 요소들을 무작위로 섞는다.
print(box)
```

문제 326. 위의 box 에서 공 3개를 추출했을때 그 공 3개가 다 파란색일 확률은 어떻게 되는가? (공을 뽑은 작업을 10000 번 수행하세요)  
복원 추출하세요 ~

```
from random import shuffle # from 패키지 import 모듈
import random

a = ['blue'] * 70
b = ['red'] * 30
box = a + b
```

```

print (box)
shuffle(box) # box 안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1, 10001):
    result1 = random.choice(box)
    result2 = random.choice(box)
    result3 = random.choice(box)
    if result1=='blue' and result2=='blue' and result3=='blue':
        cnt = cnt + 1
print (cnt/10000) # 0.3481

```

result 여러번 쓰지않고 하는 방법은 없을까용 ?

```

from random import shuffle # from 패키지 import 모듈
import random

```

```

a = ['blue'] * 70
b = ['red'] * 30
box = a + b
shuffle(box) # box 안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1, 10001):
    result = random.sample( box, 3 ) # box 에서 공을 3개씩 추출한다.
    if result.count('blue') == 3: # result 리스트의 요소가 blue 인 갯수 3개이면
        cnt = cnt + 1 # cnt 의 값을 누적시켜라 ~
print( cnt /10000 ) # 0.327

```

문제327. 이번에는 위의 box 에서 공을 3개를 뽑았을때 그중 2개가  
blue 일 확률을 출력하시오 !( 비복원 추출, 공을 뽑는 작업을 10000번  
수행하세요 ~)

```

from random import shuffle # from 패키지 import 모듈
import random

```

```

a = ['blue'] * 70
b = ['red'] * 30
box = a + b
shuffle(box) # box 안의 요소들을 무작위로 섞는다.
cnt = 0
for i in range(1, 10001):
    result = random.sample( box, 3 ) # box 에서 공을 3개씩 추출한다.
    if result.count('blue') == 2: # result 리스트의 요소가 blue 인 갯수 2개이면
        cnt = cnt + 1 # cnt 의 값을 누적시켜라 ~
print( cnt /10000 ) # 0.4577

```

설명: ranom 패키지의 sample 함수는 비복원 추출 입니다.

문제328. 위의 문제를 복원추출로 수행하시오

```

import random
from random import shuffle      45분까지 쉬세요 ~
a = []
for i in range(10000):
    box = ['blue']*70 + ['red']*30
    shuffle(box)
    cnt = 0
    for i in range(3):
        res = random.choice(box)
        if res == 'blue':
            cnt += 1
    if cnt == 2:
        a.append(cnt)
print(len(a)/10000)

```

random.sample(population, k, \*, counts=None)  
Return a k length list of unique elements chosen  
from the population sequence or set.

Used for random sampling without replacement.

```
import random
blue = ['blue']*70
red = ['red']*30
box = blue + red
cnt = 0          # 결이의 복원 추출 코드

for i in range(1,10001):
    a = []
    for k in range(3):          # 3회 뽑음
        a.append(random.choice(box)) # a 안에 집어넣음
    if a.count('red') == 1:      # a에서 red 숫자 계산해서 1이면(blue = 2개라는 말)
        cnt += 1                # cnt에 1 추가

print(cnt/10000)
```

# 넘파이의 랜덤 초이스는 복원 추출합니다.

```
import numpy as np
box = []
a = ['blue']*70
b = ['red']*30
box = a+b          # 문제328 번 세원이 코드

cnt = 0
cnts = 0
for i in range(1,10001):
    result = list(np.random.choice(box,3,replace=True)) # replace = True 면 복원
    cnts += 1
    if result.count('blue')==2:
        cnt += 1
print(cnt/cnts)
```

문제329. 세원이가 올린 코드를 참조해서 아래의 문제를 해결하시오!

6개의 제품이 들어있는 상자가 있는데 그 중에 2개가 불량품이라고  
했을때 제품 검사를 위해 3개를 추출했을때 1개가 불량품일 확률을  
출력하시오 ! ( 복원 추출로 하세요 ~)

```
box = ['정상', '불량품', '정상', '정상', '불량품', '정상']
```

```
import numpy as np
box = ['정상', '불량품', '정상', '정상', '불량품', '정상']

cnt = 0
cnts = 0
for i in range(1,10001):
    result = list(np.random.choice(box,3,replace=True)) # replace = True 면 복원
    cnts += 1
    if result.count('불량품')==1:
        cnt += 1
print(cnt/cnts) # 0.446
```

#### ■ 110. 리스트에서 특정 구간에 있는 모든 요소 제거하기

리스트 객체에서 특정 구간에 있는 요소들을 제거하려면  
del 과 슬라이싱을 이용하면 됩니다.

예제:

```
a=['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
del(a[2:6]) # 인덱스번호 2번부터 6번 미만까지 요소를 지움
print(a) # ['태양', '수성', '토성', '천왕성', '해왕성']
```

문제330. 위의 a 리스트의 요소를 다 지우시오 !

```
a=['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del( a[:] )
print(a) # []
```

문제331. (점심 시간 문제) 초등학교 10만명의 체중 데이터를 가지고  
히스토그램 그래프를 그리시오 ~~

( 초등학교 10만명의 평균 체중은 45kg 이고 표준편차는 5 입니다.)

```
30kg ~ 60kg 까지 간격 2kg 으로 x 축의 간격(계급) 을 정하세요 ~
30 ~ 32
32 ~ 34      코드와 그래프를 같이 올리세요 ~~
:            (한결이가 사용했던 xtick 을 이용해서 그리세요 ~)
58 ~ 60
```

#### ■ 111. 리스트에 있는 요소 개수 구하기(len)

파이썬 내장함수인 len() 은 시퀀스 자료형의 크기를 구하는 함수입니다.  
len() 을 리스트에 적용하면 리스트의 모든 요소의 개수를 리턴합니다.

예:

```
a=['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
print( len(a) )
```

문제332. 점심시간에 만들었던 초등학교 10만명의 체중 데이터를 모수로  
보고 10만명의 체중 데이터에서 100명을 샘플링하여 평균을  
구하시오 !

```
import numpy as np
N = 1000000
std = 5
avg = 45
population_weight = np.random.randn(N) * std + avg
result = np.random.choice(population_weight, 100, replace=True).mean()
print (result)
```

문제333. 위의 모집단에서 표본 100명을 추출하여 표본평균을 구하는  
작업을 1000번 수행해서 1000개의 표본평균을 a 라는 비어있는  
리스트에 넣으시오 !

```
import numpy as np
N = 1000000
std = 5
avg = 45
population_weight = np.random.randn(N) * std + avg
a = []
for i in range(1, 1001):
    result = np.random.choice(population_weight, 100, replace=True).mean()
    a.append(result)
print(a)
```

문제334. 위의 a 리스트의 요소의 갯수를 출력하시오

```
import numpy as np
N = 1000000
std = 5
avg = 45
population_weight = np.random.randn(N) * std + avg
a = []
for i in range(1, 1001):
    result = np.random.choice(population_weight, 100, replace=True).mean()
```

```

a.append(result)
print(len(a))

```

문제335. 위의 표본평균 1000개를 가지고 히스토그램 그래프를 그리시오

```

x 축 계급의 범위 : 30 ~ 60 (간격 2)
import matplotlib.pyplot as plt
import numpy as np
N = 1000000
std = 5
avg = 45
population_weight = np.random.randn(N) * std + avg
a = []
for i in range(1, 1001):
    result = np.random.choice(population_weight, 100, replace=True).mean()
    a.append(result)
bins = list(range(40,51,1))
plt.xticks(bins) #x축 tick
plt.grid()
plt.hist( a, bins, rwidth = 0.9, color = 'black')
plt.xlabel('kg') #x축 label
plt.ylabel('cnt') #y축 label

```

#### ■ 112. 리스트에서 특정 요소 개수 구하기(count)

리스트.count('요소') 은 객체 리스트에서 '요소' 의 갯수를 구하는 메소드입니다.

```

box=['yellow', 'yellow', 'red', 'red', 'red' ]
print( box.count('red')) # 3

```

문제336. 우리반 데이터의 나이를 비어있는 리스트 a 에 입력하고 나서

우리반 나이 데이터중에서 27 살이 몇명있는지 출력하시오 !

( 판다스 사용하지 말고 emp1222.csv 를 로드해서 하세요)

```

import csv
file = open("c:\\data\\emp1222.csv", encoding="CP949")
emp_csv = csv.reader(file)
a = []
for emp_list in emp_csv:
    a.append( int(emp_list[2] ) )

print (a.count(27) ) # 5

```

#### ■ 113. 리스트 제거하기(del)

리스트 자체를 메모리에서 제거하려면 다음과 같이 del 명령어를 수행합니다.

```

예제:
a = [ 1, 2, 3, 3, 3, 4 ]
del a
print (a)

```

NameError: name 'a' is not defined <--- 메모리에서 완전히 지우는 명령어

설명: 스파이더 오른쪽 아래에 있는 지우개 기능은 현재까지 스파이더를 사용하면서 정의된 모든 변수를 다 지우는 기능입니다.

문제337. 파이썬에서 정의된 모든 변수들을 확인하시오 !

```

a = [ 1, 2, 3, 4, 5 ]
b = 'scott'
c = 30

print( dir() )

```

#### ■ 114. 리스트 요소 정렬하기 ① (sort)

list.sort() 는 리스트 객체 list 의 모든 요소를 정렬합니다.

예:

```
a = [ 2, 4, 1, 3, 5 ]
a.sort()
print(a) # [1, 2, 3, 4, 5]
```

만약에 역순으로 정렬하고 싶다면 ?

```
a = [ 2, 4, 1, 3, 5 ]
a.sort(reverse=True)
print(a) # [5, 4, 3, 2, 1]
```

문제338. emp2.csv 에서 월급을 비어있는 리스트인 a 에 담고  
월급을 역순으로 정렬해서 리스트의 요소로 구성하시오 !

```
import csv
file = open("c:\\data\\emp2.csv")
emp_csv=csv.reader(file)          45분까지 쉬세요 ~~
a = [ ]
for emp_list in emp_csv:
    a.append( int(emp_list[5] ) )
a.sort(reverse=True)
print(a)
```

#### ■115. 리스트 요소 정렬하기 ② (sorted)

파이썬 내장함수 sorted()를 이용하 리스트를 정렬 할 수 도 있습니다.

리스트 객체의 sort() 메소드와 다른점은 리스트 객체의 sort() 는  
원본 리스트를 정렬한 형태로 변경하지만 sorted() 는 원본 리스트는  
그대로 두고 정렬한 결과 리스트만 리턴합니다.

예제:

```
a = [ 2, 1, 3, 5, 4 ]
a.sort()
print(a) # [ 1, 2, 3, 4, 5]
```

```
a = [ 2, 1, 3, 5, 4 ]
b = sorted(a) # a 리스트의 요소를 정렬해서 b 에 입력한다.
print(b) # [ 1, 2, 3, 4, 5]
print(a) # [ 2, 1, 3, 5, 4 ] # 원본은 그대로 둡니다.
```

#### ■ 116. 리스트 요소 무작위로 섞기(shuffle)

파이썬 기본 모듈인 random 은 난수를 발생하는 모듈입니다.  
이 모듈이 제공하는 shuffle 은 리스트의 요소를 무작위로 섞는데 활용됩니다.

예제:

```
from random import shuffle
a = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
shuffle(a)
print(a) # [10, 6, 1, 4, 9, 8, 3, 5, 2, 7]
```

문제339. 아래의 artist 라는 리스트에 가수이름들을 추가하고  
music 리스트에 음악 리스트를 추가하시오 !

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
```

문제340. 아래에 artist 를 키 로 하고 music 을 값으로 해서 아래와 같은 딕셔너리를 구성하시오!

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
music = [ 'yesterday', '너랑나', 'beat it' ]
```

```
gini = { '비틀즈': 'yesterday', '아이유': '너랑나', '마이클 잭슨': 'beat it' }
```

답:

```
gini = {} # 비어있는 딕셔너리를 gini 라는 이름으로 생성한다.
```

```
gini['비틀즈'] = 'yesterday'
```

```
gini['아이유'] = '너랑나'
```

```
gini['마이클 잭슨'] = 'beat it'
```

```
print(gini) # {'비틀즈': 'yesterday', '아이유': '너랑나', '마이클 잭슨': 'beat it'}
```

문제341. 위와 같이 일일이 손으로 노가다 하지 말고 아래의 2개의 리스트를

이용해서 for 문과 zip 을 이용해서 한번에 gini 딕셔너리를 구성하시오

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
```

```
music = [ 'yesterday', '너랑나', 'beat it' ]
```

```
print(gini)
```

```
{'비틀즈': 'yesterday', '아이유': '너랑나', '마이클 잭슨': 'beat it'}
```

답:

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
```

```
music = [ 'yesterday', '너랑나', 'beat it' ]
```

```
gini = {} # 비어있는 딕셔너리 gini 를 만든다. }
```

```
for i, k in zip( artist, music):
```

```
    gini[ i ] = k
```

```
print (gini)
```

■ 117. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate)

파이썬 내장함수 enumerate() 는 시퀀스 자료형을 인자로 받아

각 요소를 인덱스와 함께 쌍으로 추출할 수 있는 반복가능한 자료인

enumerate 객체를 리턴합니다. enumerate 객체는 주로 for 문과

함께 자주 사용되고 list() 를 이용해서 리스트 객체를 변환할 수 있습니다.

예제:

```
emp12=['김주원', '김미승', '김홍비', '권세원']
```

```
result = list( enumerate( emp12) )
```

```
print( result )
```

```
[ (0, '김주원'), (1, '김미승'), (2, '김홍비'), (3, '권세원') ]
```

예제2:

```
emp12=['김주원', '김미승', '김홍비', '권세원']
```

```
for i, k in enumerate(emp12):
```

```
    print ( i, k )
```

0 김주원

1 김미승

2 김홍비

3 권세원

문제342. 아래의 music 리스트에서 요소를 하나씩 빼내는데 앞에 인덱스 번호도 같이

출력되게하시오 !

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

```
music =['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
```

결과:

0 yesterday



```

1 imagine          45분까지 쉬세요 ~~
2 너랑나
3 마슈멜로우
4 beat it
5 smooth criminal

```

```

for i, k in enumerate(music):
    print(i, k)

```

문제343. artist 리스트를 가지고 아래와 같이 결과를 출력하시오 !

```

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

```

결과:

```

비틀즈      0
비틀즈      1
아이유      2
아이유      3
마이클 잭슨  4
마이클 잭슨  5

```

```

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
for i, k in enumerate(artist):
    print (k, i)

```

문제344. 아래의 두개의 리스트를 가지고 아래와 같이 결과를 출력하시오 !

```

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']

```

결과:

```

비틀즈      yesterday --> music[0]
비틀즈      imagine   --> music[1]
아이유      너랑나    --> music[2]
아이유      마슈멜로우 --> music[3]
마이클 잭슨  beat it   --> music[4]
마이클 잭슨  smooth criminal --> music[5]

```

```

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
for i, k in enumerate(artist):
    print (k, music[i])

```

최종

```

{ '비틀즈': ['yesterday','imagine'], '아이유': ['너랑나','마슈멜로우'], '마이클잭슨': ['beat it', 'smooth criminal'] }
  ↑      ↑
  키      값

```

설명: 위와 같이 딕셔너리를 구성하려면 그냥 딕셔너리인 중괄호 하나로 { } 비어있는 딕셔너리를 만들면 안되고 defaultdict 를 이용해야합니다. defaultdict 는 collections 패키지에 있어서 아래와 같이 임폴트를 해야합니다.

```

from collections import defaultdict
gini = defaultdict(list)
gini['비틀즈'].append('yesterday')
print(gini)
gini['비틀즈'].append('imagine')

```

```
print(gini)
```

```
defaultdict(<class 'list'>, {'비틀즈': ['yesterday', 'imagine']})
```

문제345. 그럼 위에서 만든 default 딕셔너리에 아이유를 키로 하고 너랑나, 마슈멜로우를 값으로 해서 추가하시오 !

```
{'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우']}
```

```
from collections import defaultdict
gini = defaultdict(list)
gini['비틀즈'].append('yesterday')
gini['비틀즈'].append('imagine')
gini['아이유'].append('너랑나')
gini['아이유'].append('마슈멜로우')
gini['마이클 잭슨'].append('beat it')
gini['마이클 잭슨'].append('smooth criminal')
print(gini)
```

문제346. 위와 같이 일일이 손으로 노가다 하지 말고 enumerate 를 이용해서 gini 딕셔너리를 구성하시오 !

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
```

결과

```
{'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클잭슨': ['beat it', 'smooth criminal']}
```

```
from collections import defaultdict
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
gini=defaultdict(list)
for i, k in enumerate( artist ):
    gini[ k ].append( music[i] )

print(gini)
```

```
{'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클잭슨': ['beat it', 'smooth criminal']}
```

문제347. 위의 gini 딕셔너리에서 음악만 추출하시오 ( 딕셔너리의 값만 추출 )

딕셔너리에서 키값을 추출할 때는 : 딕셔너리이름.keys()

딕셔너리에서 값만 추출할 때는 : 딕셔너리이름.values()

```
for i in gini.values():
    print(i)

['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']
```

```
for i in gini.values():
    for k in i: # ['yesterday', 'imagine'], ['너랑나', '마슈멜로우']
        print(k)
```

문제348. 위의 코드 음악들을 비어있는 리스트인 a 에 담으시오 !

```
['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
```

답: 40분까지 쉬세요 ~~~

```
a = []
for i in gini.values():
    for k in i: # ['yesterday', 'imagine'], ['너랑나', '마슈멜로우']
```

```

a.append(k)
print(a)

```

문제349. 아래의 코드를 수행해서 gini.values()의 요소들을 프린트 해보시오 !

```

from collections import defaultdict
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마슈멜로우', 'beat it', 'smooth criminal']
gini=defaultdict(list)
for i, k in enumerate( artist ):
    gini[ k ].append( music[i] )

for j in gini.values():
    print(j[0])

['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']

```

문제350. (오늘의 마지막 문제)

아래와 같이 결과를 출력하시오 ! 바로 옆에 다른 아티스트의 노래가 나와줘야 합니다. 아래 처럼 문자열로 출력하세요 !

yesterday, 너랑나, beat it, imagine, 마슈멜로우, smooth criminal

```

{'비틀즈': ['yesterday', 'imagine'],
'아이유': ['너랑나', '마슈멜로우'],
'마이클 잭슨': ['beat it', 'smooth criminal']}]

```

```

for j in gini.values():
    print(j[0])
['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']

```

```

artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it','smooth criminal']

```

```

from collections import defaultdict

gini = defaultdict(list)
for i,k in enumerate(artist):
    gini[k].append(music[i]) #여기까지는 song & artist 리스트를 사전형으로 만드는 작업

a = []
bond= ' '
for h in range(2):
    for j in gini.values():
        a.append(j[h])
    random_song_list=bond.join(a)
    print(random_song_list)

```

20.12.10 파이썬 강의 Day 12 (문제351~문제390)

118 리스트의 모든 요소의 합 구하기 (sum)

파이썬 내장 함수 sum()을 이용해 list의 모든 요소의 합을 출력한다.

e.g)

```

# %%

```

```
listdata = [2,2,1,3,8,4,3,9,2,20]
```

```
result = sum(listdata) #listdata 리스트의 안에 있는 모든 요소들을 더해라
```

```
print(result)
```

```
###
```

문제 351. 아래의 리스트에 짝수번째 요소의 합을 구하시오.

```
###
```

```
listdata = [2,2,1,3,8,4,3,9,2,20]
```

```
result = listdata[1 :: 2] #076 슬라이싱의 응용! 1: <- 1번째 인덱스부터 // :2 인덱스 번호 2개마다 하나씩 출력하라
```

```
print(result)
```

```
###
```

문제 352. (알고리즘문제) 아래의 리스트에서 1부터 10까지의 숫자중에 없는 숫자 하나가 있다. 그 없는 숫자를 찾아라

```
###
```

```
a = [2,1,4,5,6,7,9,10,3]
```

```
result = sum( range(1,11)) - sum(a)
```

```
print('a list에서 1부터 10사이의 숫자중 없는 수는', result, '입니다.')
```

# loop 문 없이도 문제를 풀 수 있다.

# 1부터 10사이의 수 이니까, 1부터 10까지의 합을 모두 구한후, a list 의 모든 합을 구한 값을 빼준다.(a 에는 1부터 10사이의 숫자중 1개가 없으므로!)

```
###
```

119 리스트 요소가 모두 참인지 확인하기(all, any)

리스트의 모든 요소가 참인지 또는 모든 요소가 거짓인지 판단해야 하는 경우, 파이썬 내장함수 all()이나 any() 를 사용하면 된다.

all() 은 인자로 입력되는 리스트의 모든 요소가 '참' 인 경우에만 True 를 리턴하고, 거짓이 1개라도 포함되어 있으면 False 를 리턴한다.

any() 는 인자로 입력되는 리스트의 모든 요소가 '거짓' 인 경우에만 False 를 리턴하고 '참' 이 하나라도 존재하면 True 를 리턴한다.

e.g)

```
###
```

```
listdata1 = [True, True, True]
```

```
listdata2 = [True, False, True]
```

```
print( all(listdata1 )) # True 리턴
```

```
print( all(listdata2 )) # False 리턴
```

```
print( any(listdata1 )) # True 리턴 - 3다 True 이니까
```

```
print( any(listdata1 )) # True 리턴 - 3개중 2개가 True
```

```
###
```

e.g2)

```
###
```

```
listdata1 = [1, 1, 1]
```

```
listdata2 = [1, 0, 1]
```

```
print( all(listdata1 )) # True 리턴
```

```
print( all(listdata2 )) # False 리턴
```

```
print( any(listdata1 )) # True 리턴 - 3다 1 이니까
```

```
print( any(listdata1 )) # True 리턴 - 3개중 2개가 1
```

```
#####
```

True 는 숫자로는 1이고, False는 숫자로 0이다.

문제 353. while loop 문을 사용해서 무한 loop 문을 수행하시오!

```
#####
```

```
while True :  
    print( 'aaaaaaaaaaaaaaaaaaaaa')
```

# 결과값에 aaaaaaaaaaaaaaaaaaaaaa 가 끝없이 계속 출력된다.

# 멈추기 위해서는 결과창 우측에 빨간색 네모 (정지버튼) 을 누르는 수 밖에 없다.

```
#####
```

문제 354. 이번에는 위의 True 대신에 숫자 1을 넣고 무한 루프문을 수행하시오.

```
#####
```

```
while 1 :  
    print( 'aaaaaaaaaaaaaaaaaaaaa')
```

```
#####
```

120 사전에 요소 추가하기

파이썬 책의 큰 목차 :

1. 문자형과 문자열 함수들
2. 숫자형과 숫자형 함수들 + if , loop
3. 리스트형과 리스트형 함수들 + 모듈 & 클래스
4. 사전형과 사전형 함수들 + 예외처리

+

판다스 사용법 & 웹스트롤링 -> 데이터 분석가로서 알아야할 스킬들

```
#####
```

사전형은 키 : 값 으로 되어진 요소로 구성되어 있다.

사전형은 리스트형 처럼 인덱스번호로 요소를 접근하는 것이 아니라 '키 값' 으로 요소의 값에 접근한다.

(리스트의 순서 = 인덱스 번호 였다면, 사전형의 순서 = 키의 순서 )

e.g)

```
#####
```

sol = {} # 중 괄호를 써서 비어있는 딕셔너리(사전)을 생성한다.

sol['태양'] = 'sum' #키는 태양, 값은 sum을 sol 딕셔너리에 입력한다.

```
print(sol) # {'태양': 'sum'}
```

```
sol['수성'] = 'mercury'
```

```
sol['금성'] = 'venus'
```

```
print(sol) # {'태양': 'sum', '수성': 'mercury', '금성': 'venus'}
```

```
#####
```

문제 355. 아래의 2개의 리스트를 가지고 sol 딕셔너리를 만드시오(지난주 내용 복습)

```
#####
```

```
sol_eng = ['sun', 'mercury', 'venus', 'earth', 'mars']
```

```
sol_kor = ['태양','수성','금성','지구','화성']
```

```
sol = {}
```

```
for i,k in zip(sol_kor,sol_eng): # zip 을 이용해 2개의 리스트를 동시에 반복 수행한다. i = sol_kor, k = sol_eng 가 되는 것
    sol[i] = k # i의 첫번째를 1번째 키 이면 = k 의 첫번째 값은 값으로 한다.
```

```
print(sol)
# {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```
###
```

121 사전의 특정 요소값 변경하기

사전 자료형의 특정 요소값을 변경하는 방법은 다음과 같다.

e.g) 사전형 이름['변경할 키값'] = '변경할 값'

```
###
```

```
sol = {'태양': 'sum', '수성': 'mercury', '금성': 'venus'}
```

```
sol['태양'] = 'aaaaa'
```

```
print(sol) # {'태양': 'aaaaa', '수성': 'mercury', '금성': 'venus'}
```

```
###
```

문제 356. 아래의 딕셔너리 값중에 Fire 를 피땀눈물로 변경하시오.

```
###
```

```
dict = {'방탄소년단': 'Fire', '소녀시대': 'Gee'}
```

```
dict['방탄소년단'] = '피땀눈물'
```

```
print(dict) # {'방탄소년단': '피땀눈물', '소녀시대': 'Gee'}
```

```
###
```

문제 357. 아래의 딕셔너리 값중에서 Fire를 피땀눈물로 변경하시오.

```
###
```

```
dict = {'소녀시대': ['다시만난 세계','Gee'], '방탄소년단': ['DNA', 'Fire']}
```

```
dict['방탄소년단'][1] = '피땀눈물' # 방탄소년단 키값에 대응하는 '값'의 1번째 인자를 (0,1 이므로 2번째) 피땀눈물로 바꿔라
```

```
print(dict) # {'소녀시대': ['다시만난 세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
###
```

122 사전의 특정 요소 제거하기(del)

사전 dict 에서 특정요소 ( key : value ) 를 제거하는 방법은 다음과 같다.

e.g)

```
###
```

```
sol = {'태양': 'sum', '수성': 'mercury', '금성': 'venus'}
```

```
print(sol) # {'태양': 'sum', '수성': 'mercury', '금성': 'venus'}
```

```
del sol['태양']
```

```
print(sol) # {'수성': 'mercury', '금성': 'venus'}
```

```
###
```

문제 358. 아래의 딕셔너리에서 '다시만난 세계'의 값만 지우시오.

```
###
```

```
dict = {'소녀시대': ['다시만난 세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
```

```
del dict['소녀시대'][0] # 다시만난세계 는 소녀시대 키값의 0번째 인자값이므로
```

```
print(dict) # {'소녀시대': ['Gee'], '방탄소년단': ['DNA', 'Fire']}
```

```
###
```

123 사전의 모든 요소 제거하기(clear)

사전의 모든 요소를 제거하여 빈 사전으로 만드는 방법은 사전 객체의 clear() 메소드를 이용하면 된다.

e.g)

```
###
```

```
dict = {'소녀시대': ['다시만난 세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
```

```
dict.clear()
```

```
print(dict) # {} <- 빈 사전이 되었다.
```

```
###
```

문제 359. 위와 같이 딕셔너리의 요소를 지우는게 아니라 딕셔너리 자체를 메모리에서 지우려면 어떻게 해야 할까?

```
###
```

```
dict = {'소녀시대': ['다시만난 세계', 'Gee'], '방탄소년단': ['DNA', 'Fire']}
```

```
del dict
```

```
print(dict) # <class 'dict'> <- 메모리에서 지워지면 이렇게 나옴. 실제로 Variable explorer 에서 dict 가 사라졌다.
```

# 요는 clear 와 del이 어떤 차이인지를 이해하는 것

```
###
```

124 사전에서 키만 추출하기 (keys)

아래의 사전에서 키값만 출력하고 싶다면, 사전이름.keys() 라고 하면 된다.

```
###
```

```
sol = {'태양': 'sum', '수성': 'mercury', '금성': 'venus'}
```

```
print(sol.keys()) # dict_keys(['태양', '수성', '금성'])
```

```
###
```

문제 360. 위의 sol 사전에 있는 키를 추출해서 아래와 같이 결과를 출력하시오.

결과: ['태양', '수성', '금성', '지구', '화성']

```
###
```

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```
print(sol.keys()) # 그냥 dict_keys(['태양', '수성', '금성', '지구', '화성']) 라고 출력하는 것
```

```
###
```

# 앞의 dict\_keys~~를 없애고 리스트값으로만 출력하기 위해서는 loop 문을 사용하면 된다.

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```

a= []

for i in sol.keys():
    a.append(i)
print(a)

###

sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

for i in sol.keys():
    print(i,end=',') # 태양,수성,금성,지구,화성,

# 꼭 빈 리스트에 담지 않아도 똑같이 출력할 수 있다.
# print의 end 옵션에 준 구분자(여기서는 ',') 는 값들을 가로로 출력하면서 구분자로 구분해서 가로로 출력한다.
# end 뒤에 아무것도 쓰지 않으면 enter 로 인식되어 그냥 print(i) 로 출력한것 처럼 세로로 나온다.

###

문제 361. 위의 문제에서 나온 결과에서 맨 끝에 있는 콤마(,) 는 출력되지 않게 하시오. (게시판에 올리기!) 풀어볼 문제!
기존 출력 결과 : 태양,수성,금성,지구,화성,
결과 : 태양,수성,금성,지구,화성

###

sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

풀이 1
###
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

for i,k in enumerate(sol.keys()): #sol 사전형에서 enumerate 사용해서 인덱스 키값 으로 1줄씩 출력하라
    if i+1 != len(sol): #만약 i(인덱스) +1 이 sol 사전의 전체 개수와 같지 않으면 len(sol) = 5
        print ( k , end = ',') # sol의 키값들을 ',' 로 구분해서 출력하라.
    else: #그게 아니라면 = i (인덱스) + 1 이 sol 사전의 전체 개수와 같으면
        print(k) # sol 의 값들만 출력해라.
###
# 해석
for i,k in enumerate(sol.keys()):
    if i+1 != len(sol):
        print(k, end = ',')

i가 1일때 ->i+1 은 1 < len(sol) # 1 < 5 ->so, 태양, 출력
i가 1일때 ->i+1 은 2 < len(sol) # 2 < 5 ->so, 수성, 출력
i가 1일때 ->i+1 은 3 < len(sol) # 3 < 5 ->so, 금성, 출력
i가 1일때 ->i+1 은 4 < len(sol) # 4 < 5 ->so, 지구, 출력
i가 1일때 ->i+1 은 5 = len(sol) # 5 = 5 -> if 조건에 부합 x -> else 조건 출력 ->so, 화성 출력 (심표 없음)

풀이 2 슈퍼 심플
###

sol={'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
a=[]
for i in sol.keys():
    a.append(i) # sol 의 키값을 i 번호에 맞게 하나씩 a 에 할당
print(.join(a))

###

```



125 사전에서 값만 추출하기 (values)

아래의 사전형에서 값에 해당하는 부분만 추출하려면 사전형이름.values() 를 사용하면 된다.  
#%%

```
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}  
  
print(sol.values()) # dict_values(['sun', 'mercury', 'venus', 'earth', 'mars'])  
  
#%%
```

문제 362. 아래의 sol 딕셔너리에서 값들만 아래와 같이 출력되게 하시오.

```
#%%  
  
sol = {'태양': 'sun', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}  
  
for i in sol.values():  
    print(i,end=',')  
  
#%%
```

문제 363. 아래의 gini 딕셔너리에서 값만 추출하시오.

```
['yesterday', 'imagine']  
['너랑나', '마슈멜로우']  
['beat it', 'smooth criminal']  
#%%  
  
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}  
  
for i in gini.values():  
    print(i)  
  
#%%
```

문제 364. 위의 결과를 다시 출력하는데 결과 리스트 3개중에 0번째 요소만 출력하시오.

```
결과 :  
yesterday  
너랑나  
beat it  
  
#%%  
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}  
  
for i in gini.values():  
    print(i[0]) #  
  
#%%
```

문제 365. 지난주 목요일 마지막 문제의 자신의 답을 가져와서 gini 딕셔너리에서 music 만 출력하는데 artist 별로 노래가 겹치지 않게 출력하시오.  
#%%

```
artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']  
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it','smooth criminal']  
  
from collections import defaultdict  
  
gini = defaultdict(list)  
for i,k in enumerate(artist):  
    gini[k].append(music[i]) #여기까지는 song & artist 리스트를 사전형으로 만드는 작업  
  
a = []  
bond= ' , '
```

```

for h in range(2):
    for j in gini.values():
        a.append(j[h])
        random_song_list=bond.join(a)
print(random_song_list)

```

###

문제 366. 아래의 gini dictionary 에서 값들을 리스트로 만드시오.

###

```

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

```

print(gini.values()) # 는 dict\_values([('yesterday', 'imagine'), ('너랑나', '마슈멜로우'), ('beat it', 'smooth criminal')]) 로 나옴.

```

gini2 = ( list(gini.values()))

```

print( gini2) # [['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth criminal']] 로 앞에 dict\_values 가 사라짐.

###

문제 367. 위에서 출력된 gini2 리스트의 요소들을 shuffle 로 섞어 보시오.

###

```

from random import shuffle

```

```

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}

```

```

gini2 = ( list(gini.values()))

```

```

shuffle(gini2)

```

```

print(gini2)

```

###

(점심시간문제)

문제 368. 지난시간에 만들었던 마지막문제의 자신의 답과 리스트의 요소를 무작위로 섞는 shuffle 함수를 이용해서 코드를 수행할때 마다 곡이 무작위로 뿔혀서 출력되게 하시오. 단, 아티스트 별로 노래가 겹치면 안된다. 즉, 같은 가수의 곡이 연달아 나오면 안된다)

겹치지 않는다 = 무작위 추출? /

###

```

from random import shuffle

```

```

artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']

```

```

music = ['yesterday','imagine','너랑나','마슈멜로우','beat it','smooth criminal']

```

```

from collections import defaultdict

```

```

gini = defaultdict(list)

```

```

for i,k in enumerate(artist):

```

```

    gini[k].append(music[i]) #여기까지는 song & artist 리스트를 gini 라는 사전형으로 만드는 작업

```

```

# gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']} 이렇게 된것

```

```

gini2 = ( list(gini.values())) # gini에서 값들을 출력한 것을 리스트로 만드는 것.

```

```

#shuffle(gini2) #만들어둔 리스트를 랜덤으로 무작위하게 섞는 작업

```

```

# [['yesterday', 'imagine'], ['beat it', 'smooth criminal'], ['너랑나', '마슈멜로우']] 의 list in list 가 섞이는 것이기에

```

```

# 가수별 곡의 묶음(예시. ['yesterday', 'imagine']) 자체는 섞이지 않고 계속 같은 순서를 유지.

```

```

a = []

```

```

b = []

```

```

for i in gini2:

```

```

    a.append(i[0]) # 각 가수의 1번째 곡만 모은 리스트

```

```

    b.append(i[1]) # 각 가수의 2번째 곡만 모은 리스트

```

```

    result = a+b # a list 의 값 모두 입된 후 b list 의 값이 입력되기에 같은 가수의 곡이 섞일수 없음.

```

print(result) # 다만 gini2 의 리스트는 2곡이 1묶음으로 인지되어 섞이기에 a가 섞이는 경우 (3c1) \* b가 섞이는 경우(3c1) 가지만큼 밖에 섞이지 않음

### 더 완벽하게 섞기 도전! + shuffle쓰면서!

```
from random import shuffle
```

```
artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']
```

```
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it','smooth criminal']
```

```
from collections import defaultdict
```

```
gini = defaultdict(list)
```

```
for i,k in enumerate(artist):
```

```
    gini[k].append(music[i]) #여기까지는 song & artist 리스트를 gini 라는 사전형으로 만드는 작업
```

```
# gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']} 이렇게 된것
```

```
gini2 = []
```

```
for h in range(2):
```

```
    for j in gini.values():
```

```
        gini2.append(j[h]) # ['yesterday', '너랑나', 'beat it', 'imagine', '마슈멜로우', 'smooth criminal']
```

```
shuffle(gini2)
```

```
print(gini2) #같은 가수의 곡이 서로 이어서 나오지 않게 하는 것을 강제하는 명령어가 없어서 붙어서 나올 가능성 있음
```

###

```
from random import shuffle
```

```
artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']
```

```
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it','smooth criminal']
```

```
from collections import defaultdict
```

```
gini = defaultdict(list)
```

```
for i,k in enumerate(artist):
```

```
    gini[k].append(music[i])
```

```
gini2 = list( gini.values()) #gini 에서 값들만 따로 추출하여 리스트로 생성
```

```
shuffle(gini2) # gini2 의 값들을 랜덤하게 섞는 작업
```

```
d_gini = dict(gini2) # gini2 의 값들을 다시 사전형으로 만들어줌
```

```
# {'너랑나': '마슈멜로우', 'yesterday': 'imagine', 'beat it': 'smooth criminal'}
```

```
print(d_gini)
```

```
#복습하면서 다시 해보기!
```

###

```
126 사전 요소를 모두 추출하기(items)
```

딕셔너리 자료형에서 데이터를 추출하는 방법은 다음과 같다.

1. key 만 추출 :dict.keys()

2. 값만 추출 : dict.values()

3. 둘다 추출 :dict.items()

e.g)

###

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```
print(gini.keys())
```

```
print(gini.values())
print(gini.items())
print(gini)
###
dict_keys(['비틀즈', '아이유', '마이클 잭슨'])
dict_values([['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth criminal']])

dict_items([('비틀즈', ['yesterday', 'imagine']), ('아이유', ['너랑나', '마슈멜로우']), ('마이클 잭슨', ['beat it', 'smooth criminal'])])
소괄호 안에 키와 값을 함께 출력하고 있다.
```

```
print(gini) 한 경우
{'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

###

문제 369. 위의 gini 딕셔너리의 items 를 뽑아서 리스트에 담으시오.

###

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```
result = list(gini.items())
```

```
print(result)
```

```
[('비틀즈', ['yesterday', 'imagine']), ('아이유', ['너랑나', '마슈멜로우']), ('마이클 잭슨', ['beat it', 'smooth criminal'])]
```

양 끝에 [] 가 둘러져 리스트가 되었다.

###

문제 370. 위의 result에서 음악 첫번째 곡들만 아래와 같이 출력하시오.

```
yesterday
```

```
너랑나
```

```
beat it
```

###

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```
result = list(gini.items())
```

```
for i in result:
```

```
# print(i[1]) 는 [] 안에 있는 곡의 리스트들만 1줄씩 출력된다. 리스트의 1번째 요소의 1번째 값이 1줄씩 출력.
print(i[1][0])
```

###

127 사전 정렬하기(sorted)

파이썬 내장함수 sorted() 는 사전 자료를 인자로 입력받아 정렬할 수 있다.

사전의 각 요소는 키 & 값으로 되어져 있다.

sorted() 함수에 사전을 인자로 입력하면 기본적으로 사전의 키를 오름차순으로 정렬한 결과를 리스트로 리턴한다.

e.g)

###

```
dict2 = {'소녀시대': '소원을 말해봐', '방탄소년단': 'DNA', '오마이걸': '살짝 설렘'}
```

1. 위의 dict2 에서 키만 추출

```
print(dict2.keys())
```

```
dict_keys(['소녀시대', '방탄소년단', '오마이걸'])
```

2. 위의 dict2 의 키값을 아래와 같이 정렬해서 출력하시오.

```
dict3 = sorted(dict2)
```

```
print(dict3)
```

or

```
result = dict2.values()
print(sorted(result))
```

```
['방탄소년단', '소녀시대', '오마이걸']
```

```
###
```

문제 371. 위의 결과를 reverse 되게 해서 출력하시오.

```
['오마이걸', '소녀시대', '방탄소년단']
```

```
###
```

```
dict2 = {'소녀시대':'소원을 말해봐','방탄소년단':'DNA','오마이걸':'살짝 설렘'}
```

```
result = dict2.keys()
print(sorted(result,reverse = True))
```

```
['오마이걸', '소녀시대', '방탄소년단']
```

```
###
```

문제 372. 우리반 데이터 (emp1222.csv) 에서 이름과 통신사를 출력하시오.

```
###
```

```
import csv
```

```
file = open("c:\\data\\emp1222.csv")
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    print( emp_list[1],emp_list[5] )
```

```
# * defaultdict 를 이용해서 딕셔너리의 값을 리스트로 구성하는 코드 (복습)
```

```
###
```

```
from collections import defaultdict
```

```
gini = defaultdict(list)
```

```
gini['kt'].append('허혁')
```

```
gini['kt'].append('정다희')
```

```
gini['sk'].append('김승순')
```

```
gini['sk'].append('성기창')
```

```
print (gini)
```

```
###
```

문제 373. 우리만에서 통신사의 데이터를 key 로 하고 학생이름을 값으로 해서 defaultdict 을 생성하시오.

딕셔너리 이름은 telecom 으로 하시오

```
###
```

```
from collections import defaultdict
import csv
```

```
file = open("c:\\data\\emp1222.csv")
```

```
emp_csv = csv.reader(file)
```

telecom = defaultdict(list) #딕셔너리를 생성할 건데 defaultdict 기능을 활용해서 만들 것이다.

```
for emp_list in emp_csv:
```

```
    telecom[emp_list[5]].append(emp_list[1]) #리스트에서 5번째 인자값을 키로 하고 1번째 인자값을 값으로 해서 사전을 만들어라
print (telecom)
```

```
###
```

문제 374. 위에서 구성한 telecom 딕셔너리에서 통신사가 kt인 학생들을 출력하는데, 이름을 ㄱ, ㄴ, ㄷ 순으로 정렬되어 출력되게 하시오.

```
###
```

```
from collections import defaultdict
import csv
```

```
file = open("c:\\data\\emp1222.csv")
```

```
emp_csv = csv.reader(file)
```

```
telecom = defaultdict(list)
```

```
for emp_list in emp_csv:
```

```
    telecom[emp_list[5]].append(emp_list[1])
```

```
result = telecom['kt']
```

```
print( sorted(result) )
```

```
###
```

128 문자 코드값 구하기(ord)

파이썬 내장함수 ord() 는 문자를 컴퓨터가 인식하는 코드값으로 변환한다.

e.g)

encoding : 사람이 알아볼 수 있는 언어를 컴퓨터가 알아볼 수 있는 언어로 변환하는 것

```
A -----> 65
```

```
###
```

```
print( ord('A')) #65
```

```
###
```

문제 375. 알파벳 대문자를 출력하시오.

```
###
```

```
import string
```

```
print( string.ascii_uppercase) # ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
print( string.ascii_lowercase) ->소문자 출력
```

문제 376. 위의 코드와 ord() 함수를 이용해서 아래의 결과를 출력하시오.

```
###
```

```
import string
```

```
for i in string.ascii_uppercase: # A ~ Z 까지 대문자가 나오도록 하는 것
```

```
    print(i, '----->', ord(i))
```

```
A -----> 65
```

```
B -----> 66
```

```
C -----> 67
```

```
D -----> 68
```

```

E -----> 69
F -----> 70
G -----> 71
H -----> 72
I -----> 73
J -----> 74
K -----> 75
L -----> 76
M -----> 77
N -----> 78
O -----> 79
P -----> 80
Q -----> 81
R -----> 82
S -----> 83
T -----> 84
U -----> 85
V -----> 86
W -----> 87
X -----> 88
Y -----> 89
Z -----> 90

```

```
###
```

129 코드값에 대응하는 문자 얻기(chr)

파이썬 내장함수 chr() 은 ord() 함수의 반대기능을 수행한다.

chr() 의 인자로 정수값을 입력하면 이 정수값에 해당하는 문자를 리턴한다.

e.g)

```
print( chr(65)) # A가 출력
```

```
###
```

문제 377. 아래와 같이 결과를 출력하시오!

```
###
```

```
import string
```

```
for i in range(65,91):
    print( i, '----->', chr(i) )
```

```

65 -----> A
66 -----> B
67 -----> C
68 -----> D
69 -----> E
70 -----> F
71 -----> G
72 -----> H
73 -----> I
74 -----> J
75 -----> K
76 -----> L
77 -----> M
78 -----> N
79 -----> O
80 -----> P
81 -----> Q
82 -----> R
83 -----> S
84 -----> T
85 -----> U
86 -----> V
87 -----> W
88 -----> X
89 -----> Y
90 -----> Z

```

```
###
```

130 문자열로 된 식을 실행하기(eval)

코드를 작성하다보면 파일에서 읽은 수식이나 문자열을 그대로 실행해야 하는 경우가 있다.

예를들어 '2+3'과 같이 두개의 숫자를 더하는 문자열을 텍스트 파일에서 읽어 이를 실행하여 이 수식의 결과인 '5'를 구하는 경우 등.

```
e.g)
####

a='2+3'
print(a) # 2+3 이라는 문자열이 출력됨.

print( eval( a)) # 5가 출력됨
```

```
####

문제 378. 아래의 결과를 출력하시오.
34 + 256 - 71 * 34 =? 다.
```

```
####

a = '34 + 256 - 71 * 34'

print( a , '=', eval(a), '이다.')

34 + 256 - 71 * 34 = -2124 이다.
```

그냥 식을 계산하려 한다면 각 숫자를 개별의 문자로 할당한뒤 ( a = 34, b=256 .. e등) 문자별 사칙연산을 해야 했을 것.

```
####

문제 379. 아래의 리스트의 숫자를 뽑아내서 아래의 문자열을 생성하시오.
```

```
a = [34,22,45,27,31,33,55]

결과 : 34 + 22 +45 + 27 +31 +33 +55
```

```
####

a = [34,22,45,27,31,33,55]
```

```
bond = '+'
result = bond.join(a)
```

# sequence item 0: expected str instance, int found 라고 에러가 남 -> 문자형일 것으로 기대했으나 int형밖에 없다는 것. -> 문자형으로 바꿔주면됨

```
####

a = [34,22,45,27,31,33,55]
b=[] # 그냥 b= str(a) 로하면 문자형이 들어있는 리스트로 변경되지 않는다. 그래서 별도의 빈 리스트를 만들어서 루프문으로 할당해준 것.
```

```
for i in a:
    b.append(str(i))
print('+'.join(b))
```

# join 을 하려면 문자형이어야 한다는 것을 배움! 숫자형은 불가!

```
####

문제 380. 아래의 리스트를 가지고 아래의 결과를 출력하시오.
```

34+22+45+27+31+33+55 = ?

```
####

a = [34,22,45,27,31,33,55]
```



```

b=[]

for i in a:
    b.append(str(i)) # a 의 각 요소를 문자형으로 변경해서 b 리스트에 담는다.
    result = '+'.join(b) # 34+22+45+27+31+33+55 이 result 에 입력되는 것
print( result, '=', eval(result),'이다.')

```

###

문제 381. 우리반 데이터 (emp1222.csv) 에서 나이를 읽어들여 아래와 같이 결과가 출력되게 하시오.

###

```

import csv

file = open("c:\\data\\emp1222.csv")
emp_csv = csv.reader(file)

b = []
for emp_list in emp_csv:
    b.append(str (emp_list[2]))
    result = '+'.join(b)
print( result, '=', eval(result), '이다.' )

29+31+35+29+28+28+25+28+27+27+27+26+33+30+27+27+25+28+29+24+24+29+36+26+26+26+44+28+29+28 = 859 이다.

```

###

131 이름없는 한줄짜리 함수 만들기(lambda)

파이썬에서 함수를 정의하는 방법은 일반적으로 다음과 같다.

e.g)

def 함수이름(입력매개변수):

실행코드

###

```
a = [1,2,3,4,5]
```

```

def odd_func(a):
    if a%2 ==0:
        return a

```

```

print( odd_func(2)) # 2가 나옴
print( odd_func(3)) # None가 나옴
print( odd_func(a)) # 에러가 남. unsupported operand type(s) for %: 'list' and 'int' %(나누기)는 리스트나 int 유형만 가능

```

###

```

def add_func(a,b):
    return a+b

```

```
print( add_func(2,3)) #5 가 나옴
```

설명 : 함수를 만들때 위와 같이 함수 이름을 부여하고 생성을 하는 것만 배웠다. 이처럼 함수를 생성하는 이유는 함수를 한번 만들면 다음부터는 함수이름을 가지고 함수를 호출만 하면 되기 때문에 생성을 했던 것.

```

print( add_func(5,6))
print( add_func(7,2))

```

지금부터 만들 함수는 함수이름 없이 함수를 생성하는 코드이다.

함수 이름을 부여하지 않고 함수를 생성한다는 것은 이 함수는 지금 이 코 드에서 한번만 실행하고 이후에는 쓸일이 없다는 뜻이다.

이러한 경우에 lamda 를 사용해서 아래와 같이 1줄짜리 일회성 함수를 생성한다.

```
e.g)
#%%
```

```
k = lambda a,b : a+b
print( k(2,3)) # 2+3인 5가 출력된다.
```

문법 : 변수이름 = lambda 입력매개변수: 실행문 < - 1줄에 끝난다.

```
#%%
```

문제 382. 아래의 함수를 이름 없는 한줄짜리 함수로 만들어 보시오.

```
def gob_func(a,b) :
    return a*b
```

```
#%%
```

```
k = lambda a,b : a*b
```

```
print( k(2,3))
```

```
#%%
```

```
def odd_func(a):
    if a&2==0:
        return '짝수입니다.'
    else:
        return '홀수입니다.'
```

```
#%%
```

```
k = lambda a,b : '짝수입니다.' if a%2 == 0 else '홀수입니다.'
```

```
print( k(2,3))
```

```
#%%
```

문제 384. 아래의 함수를 lambda 로 구현하시오.

```
def high_income(a):
    if a >= 3000:
        return '고소득자입니다.'
    else:
        return '일반소득자 입니다.'
```

```
#%%
```

```
k = lambda a : '고소득자입니다.' if a >=3000 else '일반소득자입니다.'
```

```
print( k (4000))
print( k (2000))
```

```
#%%
```

문제 385. 아래의 함수를 lambda 로 구현하시오.

```
#%%
```

```
def child_tall(num):
    if 140 - 1.96*5 <= num <= 140+1.96*5:
        return '신뢰구간 95%안에 있습니다.'
    else:
        return '신뢰구간 95%안에 없습니다.'
```

```
print ( child_tall(147)) # 신뢰구간 95%안에 있습니다.
```

```
print ( child_tall(157)) # 신뢰구간 95%안에 없습니다.
```

```
###
```

```
k = lambda num : '신뢰구간 95%안에 있습니다.' if 140-1.96*5 <= num <= 140+1.96*5 else '신뢰구간 95%안에 없습니다.'
```

```
print(k (144))
```

답 : 한줄 짜리 함수로는 너무 길어서 이럴 경우는 lambda 로 하지 않는것이 더 현명하다.

```
###
```

문제 386. 초등학생 키 데이터를 120 ~ 160 사이로 해서 0.001 간격으로 생성하시오.

```
###
```

```
import numpy as np
```

```
x = np.arange(120,160,0.001)
```

```
print(x)
```

```
print(len(x)) # 40000건의 데이터 생성됨
```

```
###
```

문제 387. 위에서 만든 40000건의 데이터에 대한 확률밀도함수값을 출력하시오  
( 평균값 140, 표준편차 5 )

```
###
```

```
from scipy.stats import norm
```

```
import numpy as np
```

```
x = np.arange(120, 160, 0.001)
```

```
y = norm.pdf( x, 140, 5) # 확률밀도함수
```

```
print(y)
```

```
###
```

문제 388. 위의 데이터중 x 값과 y 값을 가지고 시각화 하시오.

```
###
```

```
from scipy.stats import norm
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
x = np.arange(120,160,0.001)
```

```
y = norm.pdf(x,140,5)
```

```
plt.plot(x,y,color='purple')
```

```
plt.show()
```

```
###
```

문제 389. 위의 그래프에서 신뢰구간 95%에 해당하는 부분만 색깔로 칠하시오.

```
###
```

```
from scipy.stats import norm
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
x = np.arange(120,160,0.001)
```

```
y = norm.pdf(x,140,5)
```

```
plt.plot(x,y,color='purple')
plt.fill_between(x,y,where = (x>= 140-1.96*5) & (x<= 140+1.96*5), color='red', alpha=0.5)
plt.show()
```

# where 에 색칠하는 구간을 정해서 입력해준다.

```
###
(오늘의 마지막 문제)
문제 390. 위의 코드를 이용해서 confidence_interval 함수를 생성하고, 아래와 같이 신뢰구간을 넣으면 해당 신뢰구간에 해당하는 그래프가 출력되게 하시오.
```

신뢰구간 68% -->  $\mu \pm 1 * \text{표준편차}$   
신뢰구간 95% -->  $\mu \pm 1.96 * \text{표준편차}$   
신뢰구간 99% -->  $\mu \pm 2.58 * \text{표준편차}$

```
confidence_interval(68)
confidence_interval(95)
confidence_interval(99)
```

```
###
```

# 셋다 동시에 출력하는 경우

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt
```

def confidence\_interval(num): # 3개 동시에 출력시 구분 될 수 있도록 색 / 투명도 구분 신뢰도 낮을수록 진하게 / 신뢰도 높을수록 진한 색

```
if num == 68:
    plt.fill_between(x,y,where = (x>= 140-1*5) & (x<= 140+1*5), color='yellow', alpha=0.9)
elif num == 95:
    plt.fill_between(x,y,where = (x>= 140-1.96*5) & (x<= 140+1.96*5), color='orange', alpha=0.5)
elif num == 99:
    plt.fill_between(x,y,where = (x>= 140-2.58*5) & (x<= 140+2.58*5), color='blue', alpha=0.2)
```

```
x = np.arange(120,160,0.001)
y = norm.pdf(x,140,5)
plt.plot(x,y,color='purple')
```

```
confidence_interval(68)
confidence_interval(95)
confidence_interval(99)
```

```
###
```

#개별로 입력하여 출력하는 경우

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt
```

def confidence\_interval(num): # 3개 동시에 출력시 구분 될 수 있도록 색 / 투명도 구분 신뢰도 낮을수록 진하게 / 신뢰도 높을수록 진한 색

```
x = np.arange(120,160,0.001)
y = norm.pdf(x,140,5)

plt.plot(x,y,color='purple')

if num == 68:
    z = 1
elif num == 95:
    z = 1.96
elif num == 99:
    z = 2.58
plt.fill_between(x,y,where = (x>= 140-z*5) & (x<= 140+z*5), color='red', alpha=0.5)
```

```
plt.show()
```

```
confidence_interval(68)
confidence_interval(95)
confidence_interval(99)
```

20.12.14 파이썬 강의 Day 13(문제391~문제414)

12월 15일

132 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map)

파이썬 내장함수 map() 은 리스트 A와 함수 f가 주어지면 리스트 A의 요소를 함수 f에 입력해서 출력하는 결과를 간단하게 표현하여 출력해준다.

문법 : map(함수명, 함수에 제공할 매개변수 값(들) )

매개변수의 값들을 바꿔가면서 함수를 반복실행할때 유용하게 사용할 수 있다.

```
e.g)
#%%
```

```
def gob(x):
    return x*x
```

```
a = [1,2,3,4,5]
result = map(gob, a)
```

```
print(result) # <map object at 0x0000020E70E955B0>
```

# 결과를 보려면 list로 한번 감싸줘야 한다.

```
print( list( result)) # [1, 4, 9, 16, 25] 이 출력됨.
```

```
#%%
```

map() 함수가 리턴해주는 결과를 보려면 '반드시' list 함수로 리턴된 값을 변환해서 리스트 형태로 봐야 한다.

리스트의 요소 하나 하나를 함수에 입력값으로 입력한다(즉, mapping 시키는 것)

```
#%%
```

문제 391. 숫자를 입력해서 해당 숫자가 3000 보다 크면 숫자 1을 리턴하게 하고 입력하는 숫자가 3000보다 작다면 0을 리턴하는 함수를 생성하시오.

```
#%%
```

```
def high_income(num):
    if num >= 3000:
        return(1)
    else:
        return(0)
```

```
print ( high_income(3500))
print ( high_income(2000))
```

```
#%%
```

문제 392. 위에서 만든 함수를 emp 함수에 적용해서 아래의 리스트 요소를 high\_income 함수의 입력 매개변수로 입력해서 아래의 결과가 출력되게 하시오.

```
a = [4000,5000,2000,3500,1000]
[1, 1, 0, 1, 0]
```

```
#%%
```

```
def high_income(num):
```

```

if num >= 3000:
    return(1)
else:
    return(0)

```

```
a = [4000,5000,2000,3500,1000]
```

```
result = map(high_income,a)
```

```
print( list(result))
```

```
###
```

통계 실험시에 map() 함수를 활용하는 예시

문제 393. 동전을 지정된 횟수만큼 던져서 앞면이 나오는지 뒷면이 나오는지 출력하는 함수를 생성하시오.

```
print( coin_cnt(5)) ->앞면 앞면 뒷면 앞면 뒷면
```

```
###
```

```
import random
```

```

def coin_toss (num):
    coin = ['앞면','뒷면']
    for i in range (1,num+1):
        result = random.choice(coin)
        print(result) # return(result) 를 쓰면 1번밖에 출력이 되지 않는다.

```

```
coin_toss(5) # 결과 확인
```

```
###
```

문제 394. 위의 함수를 수정해서 숫자를 입력하면 해당 숫자만큼 동전을 던져서 동전이 앞면이 나오는 확률을 출력하시오.

```
###
```

```
import random
```

```

def coin_toss(num):
    coin=['앞면','뒷면']
    cnt = 0
    for i in range(1, num+1):
        result = random.choice(coin)
        if result == '앞면':
            cnt+=1
    return cnt/num

```

```
coin_toss(100)
```

```
#왜 아무값도 안나오지?
```

```
###
```

문제 395. 위에서 만든 coin\_toss 함수에 아래의 a 리스트의 요소들을 적용해서 결과로 확률이 출력되게 하시오.

```
a = [10,100,1000,10000,100000]
```

```
###
```

```
import random
```

```

def coin_toss(num):
    coin=['앞면','뒷면']
    cnt = 0
    for i in range(1, num+1):
        result = random.choice(coin)

```

```

    if result == '앞면':
        cnt+=1
    return cnt/num

```

```

a = [10,100,1000,10000,100000]

```

```

result = map(coin_toss,a)

```

```

print( list(result)) # [0.5, 0.53, 0.498, 0.4988, 0.49831]

```

```

#%

```

동전을 던지는 횟수가 많아질 수록 0.5에 근사한 값이 출력된다.

```

#%

```

문제 396. 주사위를 던져서 주사위의 눈이 5가 나올 확률을 출력하는 함수를 만들고, 아래의 a list 의 주사위를 던진 횟수를 map 함수를 적용해서 확률이 점점 1/6로 근사해지는지 실험하시오.

```

#%

```

```

import random

```

```

def dice_toss(num): #주사위를 num번 던질때 5의 눈이 나오는 횟수 / 던지횟수를 리턴하는 함수 생성
    dice=[1,2,3,4,5,6]
    cnt = 0
    for i in range(1,num+1):
        result=random.choice(dice)
        if result == 5:
            cnt+=1
    return cnt/num

```

```

a = [10,100,1000,10000,100000] # 주사위 던지는 실험을 시행할 매개변수들의 리스트

```

```

result = map(dice_toss,a) # map 함수적용

```

```

print( list( result))

```

```

#%

```

문제 397. 아래의 불량품이 들어있는 박스에서 제품을 3개를 뽑았을때 3개중에 2개가 불량품일 확률을 출력하는 함수를 만드시오.(복원추출)

```

box = ['정상','정상','불량품','정상','불량품','정상','정상','불량품']

```

```

print ( box_cnt(10) )

```

```

#%

```

import numpy as np #함수 안에 넣지 않고 밖에 두는 것이 더 좋다.

```

def box_cnt(num):
    box = ['정상','정상','불량품','정상','불량품','정상','정상','불량품']
    cnt=0
    for i in range(1, num+1):
        result = list( np.random.choice ( box, 3, replace = True))
        if result.count('불량품') == 2:
            cnt+=1
    return cnt / num

```

```

print ( box_cnt(100) )

```

```

#%

```

문제 398. 위에서 만든 box\_cnt 함수에 아래의 a 리스트의 횟수를 map 함수로 적용해서 확률이 출력되게 하시오.

```

a = [10,100,1000,10000,100000]

```

```
###
```

import numpy as np #함수 안에 넣지 않고 밖에 두는 것이 더 좋다.

```
def box_cnt(num):
    box = ['정상','정상','불량품','정상','불량품','정상','정상','불량품']
    cnt=0
    for i in range(1, num+1):
        result = list( np.random.choice ( box, 3, replace = True))
        if result.count('불량품') == 2:
            cnt+=1
    return cnt / num
```

```
a = [10,100,1000,10000,100000]
result = map( box_cnt,a)
```

```
print( list(result)) # [0.0, 0.26, 0.241, 0.2583, 0.26442]
```

```
###
```

### 133. 텍스트 파일을 읽고 출력하기 (read)

텍스트 파일을 읽고 그 내용을 화면에 출력하고자 하면 제일먼저 텍스트 읽기모드로 파일을 연다.

텍스트 파일을 오픈하면 텍스트 파일을 읽어 내용을 화면에 출력하면 된다.

e.g)

```
###
```

```
f = open('c:\\data\\jobs.txt',encoding='UTF8')
```

data = f.read() #파일을 한번에 전부 읽어오는 함수 ( read() )

```
print(data)
```

f.close() # 파일을 닫는다. 닫는 코드를 쓰지않으면 스파이더를 실행하는동안 계속 파일이 열려있게 된다 -> 메모리를 계속 차지한다 -> 낭비!

# 텍스트 파일이 크면 메모리를 많이 사용하게 된다. 그러므로 사용이 끝나면 반드시 close() 를 입력해 닫아준다.

```
###
```

스티브 잡스 연설문을 별로 긴 편이 아니어서 한번에 읽어와도 관계가 없는데 텍스트 파일의 용량이 매우 클 경우 read() 로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기시킬 수 있다.

웹에서 스크롤링한 데이터를 분석하고자 할때 위와 같이 open 함수를 이용해서 파이썬으로 데이터를 불러온다.

```
###
```

문제 399. 중앙일보에서 인공지능으로 검색한 기사인 mydata3.txt를 파이썬으로 불러서 출력하시오.

```
###
```

```
f = open('c:\\data\\mydata3.txt',encoding='UTF8')
```

```
data = f.read()
```

```
print(data)
```

```
f.close()
```

```
###
```

문제 400. 위의 중앙일보기사에서 빅데이터 라는 단어가 몇번 나오는지 카운트 하시오.

```
###
```

```
f = open('c:\\data\\mydata3.txt',encoding='UTF8')
```

```
data = f.read()
```



```
print(data.count('빅데이터')) # 11 번
print(data.count('인공지능')) # 24 번

f.close()
```

###

#### 134 텍스트 파일을 한줄씩 읽고 출력하기 ① (readline)

텍스트 파일의 용량이 매우 클 경우 read() 로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기할 수 있다.  
이 경우 텍스트 파일 내용을 한줄 단위로 읽고 작업을 수행하면 된다.  
readline() 은 텍스트 파일에서 한 줄을 읽는다. 한줄을 읽고 나면 파일을 읽기 시작하는 위치는 그 다음줄을 맨 처음이 된다.

```
e.g)

###

f = open('c:\\data\\jobs.txt', encoding='UTF8')

data = f.readline()

print(data)

f.close()
```

###

read() 함수는 텍스트 파일 전체를 읽어오는 반면 readline() 은 텍스트 파일에서 한줄만 읽어 온다.

문제 401. 위의 스티브 잡스 연설문 데이터를 모두 읽어 오시오!(readline() 을 사용해서!)

###

```
f = open('c:\\data\\jobs.txt', encoding='UTF8')
data = f.readline()
```

while data: # data 변수안에 data가 있으면 True, 없으면 False 이다. True -> while 문 아래의 실행문 수행

```
    print(data) #처음에 인지되는 data를 출력한다.
```

```
    data = f.readline() # 그 다음줄을 읽어서 data변수에 입력한다. 를 계속 반복 til data가 없어서 False 가 될때 까지
f.close()
```

###

맨위에 data = f.readline() 코드는 처음에 딱 1번만 실행되고 while문 안에서는 실행되지 않는다.

data = f.readline() 가 실행되면서 'steve jobs'2005 stanford commencement address'가 입력되어서 1번 출력이 된 것.

그 다음부터는 while 문 안에 data = f.readline() 가 작동 o 되어서 반복적으로 스티브 잡스의 연설문 파일속 내용을 1줄씩 읽어서 data 변수에 넣는것.

즉 맨위에 data = f.readline() 가 없다면 밑의 while 문은 실행되지 않는다.

문제 402. 위에서 한줄씩 읽은 데이터에서 인공지능이라는 단어가 나오면 count하게 하시오.

###

```
f = open('c:\\data\\mydata3.txt', encoding='UTF8')
data = f.readline()
```

while data: # data 변수안에 data가 있으면 True, 없으면 False 이다. True -> while 문 아래의 실행문 수행

```
    print(data.count('인공지능')) #처음에 인지되는 data를 출력한다.
```

```
    data = f.readline() # 그 다음줄을 읽어서 data변수에 입력한다. 를 계속 반복 til data가 없어서 False 가 될때 까지
```

```
f.close()
```

```
###
```

문제 403. 위의 count 한 건수를 다 누적시켜서 출력하시오.

```
###
```

```
f = open('c:\\data\\mydata3.txt', encoding='UTF8')
data = f.readline()
```

```
cnt=0
```

```
while data: # data 변수안에 data가 있으면 True, 없으면 False 이다. True -> while 문 아래의 실행문 수행
```

```
    #print(data.count('인공지능')) #처음에 인지되는 data를 출력한다.
```

```
    cnt= cnt + data.count('인공지능')
```

```
    data = f.readline() # 그 다음줄을 읽어서 data변수에 입력한다. 를 계속 반복 til data가 없어서 False 가 될때 까지
    print(cnt)
```

```
f.close()
```

```
###
```

문제 404. 위의 코드를 수정해서 단어를 물어보게 하고 단어를 입력하면 해당 단어가 몇번 나오는지 출력되게 하시오.

```
###
```

```
f = open('c:\\data\\mydata3.txt', encoding='UTF8')
data = f.readline()
```

```
word = input('단어를 입력하세요 ~')
```

```
cnt=0
```

```
while data: # data 변수안에 data가 있으면 True, 없으면 False 이다. True -> while 문 아래의 실행문 수행
```

```
    #print(data.count('인공지능')) #처음에 인지되는 data를 출력한다.
```

```
    cnt= cnt + data.count(word)
```

```
    data = f.readline() # 그 다음줄을 읽어서 data변수에 입력한다. 를 계속 반복 til data가 없어서 False 가 될때 까지
    print('이 기사에서 ',word,'단어는', cnt, ' 번 나왔습니다.')
```

```
f.close()
```

```
###
```

135 화면에서 사용자 입력을 받고 파일로 쓰기 (write)

사용자로부터 입력받은 테스트 파일로 저장하려면 파일을 텍스트 쓰기 모드로 오픈하고 파일 객체의 write() 를 이용해 데이터를 파일에 기록한다.

e.g)

```
###
```

```
text = input('파일에 저장할 내용을 입력하세요~')
```

```
f = open('c:\\data\\mydata.txt','w')
```

```
f.write(text)
```

```
f.close()
```

# 위의 w 는 write의 약자이다. c 드라이브 밑에 data 밑에 mydata.txt 파일로 쓰겠다(write)는 뜻.

# 위의 코드는 웹에서 스크롤링한 데이터를 파일로 저장할 때 응용하게 될 것.

```
###
```

문제 405. 아래의 파란색 공과 빨간색 공이 들어있는 박스에서 5개의 공을 뽑았을 때 그중 2개가 파란색 공일 확률을 출력하는 함수를 만들고 공을 뽑는 횟수를 10,100,1000,10000,100000 로 했을때의 확률을 map 함수를 이용해서 출력하시오.

파란색공 24개, 빨간색공 26개

```
###
```

import numpy as np #함수 안에 넣지 않고 밖에 두는 것이 더 좋다.

```
a = ['빨강']*26
```

```
b = ['파랑']*24
```

```
box = a+b
```

```
def box_cnt(num):
```

```
    cnt=0
```

```
    for i in range(1, num+1):
```

```
        result = list( np.random.choice ( box, 5, replace = True))
```

```
        if result.count('파랑') == 2:
```

```
            cnt+=1
```

```
    return cnt / num
```

```
pick_cnt = [10,100,1000,10000,100000]
```

```
result = map( box_cnt,pick_cnt)
```

```
print( list(result))
```

```
###
```

136 텍스트 파일에 한줄씩 쓰기(writelines)

파일 객체의 writelines() 는 텍스트 문자열이나 텍스트 문자열이 요소로 되어있는 리스트를 인자로 받아 파일에 한 줄씩 기록한다.

리스트가 인자인 경우 writelines() 는 리스트의 요소를 하나의 문자열로 결합하여 파일을 한꺼번에 기록한다.

\* writelines 는 리스트 자료형도 파일에 저장할 수 있다.

e.g)

```
###
```

```
data = []
```

```
while True:
```

```
    text = input( ' 저장할 내용을 입력하세요~')
```

```
    if text == '': # 텍스트에 아무것도 입력하지 않으면
```

```
        break # break 문이 실행되서 무한루프가 종료 되는 것.
```

```
    data.append( text +'\\n') # break가 되어 종료되면 text가 엔터와 함께 datalist 에 appending되는 것.
```

```
f = open('c:\\data\\mydata99.txt','w') # mydata99.txt 를 생성하겠다. 'w' 대신에 'a'를 쓰면 add 가 가능해진다.
```

```
f.writelines(data) # data 리스트의 내용을 mydata99.txt 에 저장하겠다.
```

```
f.close()
```

# 만약 내용을 쭉 입력하고 아무것도 입력하지 않고 엔터를 치면 mydata99.txt 가 생성되며 그동안 입력된 내용이 입력되어 저장된다.

# open 의 옵션중 w 는 그냥 쓰기이고, a 는 추가(add) 를 의미한다.

```
###
```

```
listdata = [2,2,1,3,8,5,7]
```

```
result = sorted( listdata) #리스트를 정렬하는 sorted()
```

```
print (result ) # [1, 2, 2, 3, 5, 7, 8]
```

```
f = open("c:\\data\\mydata2.txt","w") # mydata2.txt를 생성하겠다.
```

```
f.write( str(result) ) # result 에 있는 내용을 mydata2.txt 에 문자형으로 변환하고 write하겠다. (문자형 변환은 필수!)
```

```
f.close()
```

# str 로 변환하지 않을 경우

# TypeError: write() argument must be str, not list

```
###
```

문제 406. 아래의 리스트를 mydata9.txt 로 저장하시오.

```
###
```

```
listdata2 = ['a','b','c','d','e','f','g']
```

```
result = sorted(listdata2)
```

```
print (result)
```

```
f = open("c:\\data\\mydata9.txt","w")
```

```
f.write( str(result) )
```

```
f.close()
```

```
###
```

137 텍스트 파일 복사하기(read, write)

텍스트 파일을 복사하는 방법을 파이썬으로 구현하려면 read 와 write를 이용하면 된다. read() 로 텍스트 파일을 읽고, write() 로 읽은 내용을 다른 파일 이름으로 저장하면 된다.

e.g)

```
###
```

```
f = open('c:\\data\\mydata.txt','r') # r = read
```

```
h = open('c:\\data\\mydata_copy.txt','w') # w = write -> mydata_copy 에 write 하겠다.
```

```
data = f.read()
```

```
h.write( data )
```

```
f.close()
```

```
h.close()
```

```
###
```

138 바이너리 파일 복사하기(read, write)

이미지나 동영상도 파이썬으로 복사 붙여넣기를 할 수 있다.

이미지나 동영상을 복사 붙여넣기 할때는 파일의 크기가 크기때문에 한번에 읽어 들일수는 없고 일부분만 일정한 크기로 읽으면서 복사 붙여넣기를 한다. (옛날 프린터기 쓸때 프린터기가 1줄씩 출력해서 1장안에 그림을 출력하는 느낌??)

e.g)

```
###
```

bufsize = 1024 # 1kb 크기의 버퍼 사이즈를 지정한다.

```
f = open('c:\\data\\lena.png','rb') # rb = read binary
```

```
h = open('c:\\data\\lena_copy.png','wb') # wb = write binary
```

```
data = f.read(bufsize) # f에서 이미지 1kb 를 읽어서 data 저장한다.
```

```
while data: # data에 data가 발견된느 동안은 아래 실행문을 무한루프 돌린다.
```

```
    h.write(data) # lena_cpy.png 에 1kb씩의 데이터를 write 한다.
```

```
    data = f.read(bufsize) # lena.png에서 1kb의 데이터를 read 한다.
```

```
f.close()
```

```
h.close()
```

```
###
```

lena.png 의 총 크기는 334kb 이므로 1kb씩 읽어들여서 쓰는 작업을 총 334번 하게 되는 것.

```
###
```

139 파일을 열고 자동으로 닫기 (with ~ as )

with ~ as 절을 사용하게 되면 f.close() 를 명시하지 않아도 되므로 프로그래머가 실수로 f.close() 를 작성하지 않아서 발생하는: 메모리 부족오류를 예방할 수 있다.

1. with ~ as 절을 사용 안했을 때

###

```
f = open("c:\\data\\mydata99.txt",'r')
data = f.readlines()
print(data)
f.close()
```

###

2. with ~ as 절을 사용했을 때 #1과의 차이는 f.close() 가 없다는 것 + with ~ as 절이 사용되었다는 것.

###

```
with open("c:\\data\\mydata99.txt","r") as f:
    data = f.readlines()
    print(data)
```

###

with ~ as 절을 사용하면 자동으로 파일 close 를 해준다. :

둘다 같은 결과가 출력되기는 함!

파이썬 기본 문법 끝~!!

이후는 책에도 없고, 웹스크롤링을 위한 추가적인 학습 & 예제들

###

140 HTML 기본문법

데이터 분석 순서

데이터 수집 -> 데이터 유형 및 속성 파악 -> 데이터 변환 -> 데이터 저장 -> 데이터 정제 -> 데이터 분석

\* 데이터 수집 기술

웹스크롤링 기술

\* HTML 이란 Hyper Text Markup Language 의 약자

여러개의 태그(tag)를 연결해서 모아놓은 문서

e.g ) 아주 간단한 html 문서를 만들어 보기

1. 메모장을 열고 아래와 같이 코딩을 한다.

```
<html><head><title> 양건준님의 오늘 일정 </title></head> # title -> 인터넷상의 웹 부분(head)에 표시되는 사이트 이름 (head 에서의 타이틀)
<body>
<p class="title"> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </p> # 한글 부분이 본문(body)에서의 타이틀.
</body>
</html>
```

2. 메모장의 파일 이름을 a.html 로 바탕화면에 저장한다.

\* 저장시 파일 형식을 '모든파일'로 해야한다!

<html> 로 시작하면 </html> 로 종료 해줘야 한다.

<body> <head> <title> 등도 마찬가지로!

e.g2) 위의 글씨를 진하게 하시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head> # title -> 인터넷창의 탭 부분(head)에 표시되는 사이트 이름 (head 에서의 타이틀)
<body>
<p class="title"><b> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </b></p> # 한글 부분이 본문(body)에서의 타이틀. # b =bold
</body>
</html>
```

바탕화면의 a.html 을 연결프로그램 -> 메모장으로 하여 연다.

작성한 코드를 입력하고 저장한다.

기존에 열려있던 a.html 창을 새로고침한다!

e.g 3) 위의 글씨에 밑줄을 그어보시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"><b><u> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </u></b></p>
</body>
</html>
```

밑줄을 사용하려면 태그 u 를 사용한다.

e.g 4) 위의 글씨를 이탤릭체로 변경하시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u></b></p>
</body>
</html>
```

글씨를 이탤릭체로 변경하려면 태그 i 를 이용한다.

e.g 5) p태그를 추가해서 제목과 내용을 나누시오.

```
<html><head><title> 양건준님의 오늘 점심 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u></b></p>
<p class="content"> 양건준님은 오늘 점심시간에 지하식당으로 내려가서 반 친구들과 같이 우육탕을 먹었습니다.
코로나에 대한 걱정도 있었지만 용기내서 먹었습니다.</p>
</body>
</html>
```

제목은 class title 에 적고 내용은 class content 에 적는다.

e.g 6) 위에서 만든 html 문서의 본문에 링크를 거시오. 링크는 우리반 카페주소로!

```
<html><head><title> 양건준님의 오늘 점심 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u></b></p>
<p class="content"> 양건준님은 오늘 점심시간에 지하식당으로 내려가서 반 친구들과 같이 우육탕을 먹었습니다.
코로나에 대한 걱정도 있었지만 용기내서 먹었습니다.
<a href="http://cafe.daum.net/oracleoracle" class="cafe1" id="link1">
</p>
</body>
</html>
```

워드 문서를 만들때도 그 문서내에 단원도 있고 세부목차도 있듯이 class 에 그 html 문서의 특정 단원이라고 보면 되고 id 는 링크를 줄때 부여하는 제목인데 id 는 값이 unique 하다.

문제 407. 다음과 같이 내용과 링크를 수정하시오

양건준님은 오늘 점심시간에 지하식당으로 내려가서 반 친구들과 같이 우육탕을 먹었습니다. 코로나에 대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이지는 바로가기를 누르세요. 바로가기

```
<html><head><title> 양건준님의 오늘 점심 </title></head>
<body>
<p class="title"><b><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u></b></p>
<p class="content"> 양건준님은 오늘 점심시간에 지하식당으로 내려가서 반 친구들과 같이 우육탕을 먹었습니다.
코로나에 대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이지는 바로가기를 누르세요.
바로가기 <a href="http://cafe.daum.net/oracleoracle" class="cafe1" id="link1">
</p> 1
</body>
</html>
```

###

## 141 beautiful soup 모듈 배우기

beautiful soup 모듈이란 파이썬 코드를 복잡하게 작성하지 않아도 편하게 웹스크롤링 할 수 있도록 여러 함수들을 제공하는 웹스크롤링 전문 모듈이다.

카페에서가서 ecologicalpyramid.html 문서를 내려받아 c 드라이브의 data 폴더에 저장하시오.

e.g 1) ecologicalpyramid.html 코드를 beautiful soup 모듈에서 사용할 수 있도록 파싱(parsing) 을 하고 그 내용을 출력하시오.

###

from bs4 import BeautifulSoup # 대소문자 맞춰줘야 한다. # bs4 패키지에서 BeautifulSoup 모듈을 불러오겠다.

```
f = open("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser") # BeautifulSoup 모듈을 사용할 수 있도록 파싱을 하는 것
print(soup)
```

###

e.g 2 ) ecologicalpyramid.html 코드에서 class 이름 name 에 접근해서 데이터를 긁어오시오.

웹상에 출력된 내용들 중 plants, algae, deer .... 등만 출력하고 싶음!

코드를 보고 해당 자료들이 어떤 클래스에 있는지 코우 클래스를 파악 후 그 값들을 웹크롤링 하기!

###

from bs4 import BeautifulSoup # 대소문자 맞춰줘야 한다. # bs4 패키지에서 BeautifulSoup 모듈을 불러오겠다.

```
f = open("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = (soup.find_all(class_="name"))
print(result)
```

###

출력물!

```
[<div class="name">plants</div>,
<div class="name">algae</div>,
<div class="name">deer</div>,
<div class="name">rabbit</div>,
<div class="name">fox</div>,
<div class="name">bear</div>,
<div class="name">lion</div>,
<div class="name">tiger</div>]
```

find\_all 로 했기 때문에 전체 코드에서 class="name" 에 해당하는 부분을 전부 긁어와서 리스트에 담아 준 것

웹스크롤링에서 얻고자 하는 부분은 class="name" 이 아닌 그 뒤의 본문 plants, algae, deer ... 등!

e.g 3) 위에서 긁어온 자료중에서 html 코드 말고 text 만 출력하시오.

```
###
```

```
from bs4 import BeautifulSoup # 대소문자 맞춰줘야 한다. # bs4 패키지에서 BeautifulSoup 모듈을 불러오겠다.
```

```
f = open("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_="name") # result 는 list type
```

```
#for i in result: # result list 안의요소들을 하나씩 가져오는 것.
# print(i) #출력해보면 html code + 텍스트를 한줄씩 출력하난.
```

```
for i in result:
    print( i.get_text() ) # text 부분만 정제해서 출력한다.
```

```
###
```

문제 408. ecologicalpyramid.html 코드에서 numbers 클래스에 있는 숫자들만 긁어와서 출력하시오.

```
###
```

```
from bs4 import BeautifulSoup # 대소문자 맞춰줘야 한다. # bs4 패키지에서 BeautifulSoup 모듈을 불러오겠다.
```

```
f = open("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_="number") # result 는 list type
for i in result:
    print( i.get_text() )
```

```
###
```

문제 409. 위에서 긁어온 숫자들을 a 라는 비어있는 리스트에 저장한 후 a 안에 요소들을 정렬하고, a list 를 출력하시오.

```
###
```

```
from bs4 import BeautifulSoup # 대소문자 맞춰줘야 한다. # bs4 패키지에서 BeautifulSoup 모듈을 불러오겠다.
```

```
f = open("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_="number") # result 는 list type
```

```
a=[]
for i in result:
    a.append( int( i.get_text() ) )
a.sort()
print(a)
```

```
###
```

문제 410. 중앙일보사 홈페이지로가서 신문기사 하나를 보시오.

기사를 하나 찾은 후 ctrl+s 를 눌러 시가의 html 을 aaa77.html 로 저장하난.

문제 411. aa77.htmlid을 beautifulsoup 모듈의 함수를 쓸 수 있도록 파싱 하고 출력하시오.

```
###
```

```
from bs4 import BeautifulSoup
```

```
f = open("c:\\data\\aaa77.html",encoding='UTF8')
soup = BeautifulSoup(f, "html.parser") # BeautifulSoup 모듈을 사용할 수 있도록 파싱을 하는 것
print(soup)
```

```
###
```



실전 코드는 정말 복잡하구나..

```
###
```

문제 412. 위의 기사의 본문을 가져오기위해서 기사 본문의 class 이름이 무엇인지 확인하시오. (class 확인 꿀팁!)

1) 크롬의 개발자 모드로 들어간다 (F12 버튼) -> 우측에 html 코드창이 나온다.

2) Ctl + Shift + C를 누르고(혹은 좌상단 좌상향 방향가리키는 화살표 버튼) 찾고자하는 class 가 있는 위치(기사본문)를 클릭하면 html문서에 class를 확인할 수 있는 html 문서 쪽으로 바로 이동한다.

현재 html에서 본문의 코드는 <div id="article\_body" itemprop="articleBody" class="article\_body mg fs4"> 이다.

```
###
```

문제 413. 클래스 이름 article\_body mg fs4 로 접근하여 text를 읽어오시오.

```
###
```

```
from bs4 import BeautifulSoup
```

```
f = open("c:\\data\\aaa77.html",encoding='UTF8')
soup = BeautifulSoup(f, "html.parser") # BeautifulSoup 모듈을 사용할 수 있도록 파싱을 하는 것
result = soup.find_all(class_='article_body mg fs4')
for i in result:
    print(i.get_text() )
```

```
###
```

(오늘의 마지막 문제)

문제 414. 위에서 스크롤링한 중앙일보 기사를 c drive 밑에 mytext23.txt 로 저장하시오.

```
###
```

```
from bs4 import BeautifulSoup
```

```
f = open("c:\\data\\aaa77.html",'r',encoding='UTF8')
```

```
soup = BeautifulSoup(f, "html.parser") # BeautifulSoup 모듈을 사용할 수 있도록 파싱을 하는 것
result = soup.find_all(class_='article_body mg fs4')
for i in result:
    result2 = i.get_text()
# print(result2)
h = open("c:\\data\\mytext23.txt","w",encoding='UTF8')
h.write(result2)
f.close()
h.close()
```

20.12.15 파이썬 강의 Day 14 (문제415~문제421)

빅데이터 수집을 위해 반드시 알아야 할 기술! 웹스크롤링~!

웹스크롤링을 위해 사용한 파이썬 모듈 -> Beautiful soup 모듈

웹스크롤링을 편하게 하기 위한 방법

-> 크롬의 개발자모드! (F12) 로 들어가서 화살표를 이용(ctl+shift+c) 이용해서 웹상에서 스크롤링하기 원하는 지점의 html을

142 웹스크롤링 실전 1단계 (ebs 레이디 버그 게시판)

우리 회사의 신제품이 출시되었을때 그 제품에 대한 사람들의 반응을 데이터 분석하고자 할때 웹스크롤링과 데이터 시각화를 이용하면 된다.

e.g 1) ebs 레이디버그 시청자게시판의 url을 가지고 직접 html 문서를 내려받을 수 있도록 코드를 구현

(어제는 ctrl+s를 눌러서 웹페이지를 pc에 저장하고 구현을 했는데, 실전에서는 직접 웹에서 바로 html문서를 내려받을 수 있도록 구현한다.)

###

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list\_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url

url= urllib.request.Request(list\_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)

#위의 url의 html 문서들을 result 변수에 담았다.

print(result) # 위의 url의 html 전체 문서가 다 출력되고 있음

# 여기까지는 ctrl+s 해서 웹상의 html문서를 내컴퓨터에 다운 받는 작업을 거치지 않고 파이선으로 바로 불러오는 작업을 한 것

###

e.g 2) 위에서 긁어온 html문서를 BeautifulSoup의 함수를 이용해서 웹스크롤링을 할 수 있도록 BeautifulSoup을 쓸 수있게 파싱하시오.

###

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

list\_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url

url= urllib.request.Request(list\_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)

#위의 url의 html 문서들을 result 변수에 담았다.

print(result) # 위의 url의 html 전체 문서가 다 출력되고 있음

soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것

print(soup)

###

e.g 3) 지금 페이지의 시청자 게시판의 글내용에 해당하는 부분의 태그와 클래스 이름을 알아내시오

1. 웹페이지에서 F12를 눌러 개발자 모드로 진입

2. 좌측 상단 화살표 or ctrl+shift+c 누른뒤 찾고자하는 위치에 가서 클릭

3. 개발자 모드 창에서 찾고자한 내용(위치)와 제일 가까운 태그와 클래스 이름 확인

결과 : <p class="con"> 재미있다 </p>

-> 태그 : p // 클래스 = con

e.g 4) p 태그(가 매우 여러개인데) 중에 클래스가 con에 해당하는 부분을 스크롤링 하시오.

###

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

```

list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.

soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
# e.g 4의 답

# result2 = soup.find_all( class_="con") 은 어제 배운 것
result2 = soup.find_all('p', class_="con") # 문제가 p 태그 중에서 클래스가 con에 해당하는 부분! 이므로!

print(result2) #결과물을 보면 </p> ] 로 대괄호로 닫혀있음 = list형태로 저장되어있다는 뜻.

#%%%

find 함수는 맨 처음 하나만 가져오는데 find_all 은 p 태그에 class 이름 con에 해당하는 부분을 모두 가져온다.

e.g 5) 위의 결과에서 html 문서 말고 텍스트만 가져오시오

#%%%

```

```

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

```

```

list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.

soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것

# result2 = soup.find_all( class_="con") 은 어제 배운 것
result2 = soup.find_all('p', class_="con")
# e.g 5) 의 답 (이하)

for i in result2:
    print(i.get_text() )

#%%%

```

```

e.g 6) 위에서 출력되고 있는 텍스트들이 좀 더 깔끔하게 나오도록 하시오.
#%%%

```

```

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

```

```

list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

```

```
result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.
```

```
soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

```
# result2 = soup.find_all( class_="con") 은 어제 배운 것
result2 = soup.find_all('p', class_="con")
# e.g 5) 의 답 (이하)
```

```
for i in result2:
    print(i.get_text(" ",strip=True) ) # text( ) 안의 "" 사이 공백은 재량껏! 근데 1칸정도 주는게 가독성이 더 좋다.
```

```
# strip = True 는 출력된 결과물(get_text) 의 공백들을 모두 자르겠다. 벗겨내겠다는 뜻.
```

```
###
```

```
e.g 7) 위에서 출력되고 있는 텍스트들이 params 라는 비어있는 리스트에 담기게 하시오.
```

```
###
```

```
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

```
list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)
```

```
# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄
```

```
result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.
```

```
soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

```
# result2 = soup.find_all( class_="con") 은 어제 배운 것
result2 = soup.find_all('p', class_="con")
# e.g 5) 의 답 (이하)
```

```
params = []
for i in result2:
    params.append(i.get_text(" ",strip=True) )
print(params)
```

```
###
```

```
e.g 8) 게시글을 올린 날짜를 스크롤링하기 위해서 게시글 날짜가 있는 html 문서의 태그이름과 클래스 이름을 확인하시오.
```

```
태그 이름은 span // class 이름은 data 이다.
```

```
e.g 9) 위의 날짜를 모두 스크롤링해서 params2 라는 리스트에 담으시오.
###
```

```
# 1. 웹에서 html 문서를 가져와 beautifulsouop 으로 파싱
```

```
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

```
list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuid=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)
```

```
# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

result = urllib.request.urlopen(url).read().decode("utf-8") # decode -> 컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.

soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

```
# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.
result1 = soup.find_all('span', class_="date")
result2 = soup.find_all('p', class_="con")
```

```
# 3. 시청자게시판의 날짜와 본문을 params / params2 리스트에 담는다.
```

```
params1 = []
params2 = []
for i,k in zip(result1,result2):
    params.append(i.get_text(" ",strip=True) )
    params2.append(k.get_text(" ",strip=True) )
print(params)
print(params2)
```

```
####
```

```
e.g 10) 위의 날짜와 본문 내용이 아래와 같이 출력되게 하시오.
```

```
'2020.12.11 19:52', '재미있 다'
```

```
####
```

```
# 1. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱
```

```
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

```
list_url="https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuld=106" # <-레이디버그 시청자게시판 url
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)
```

```
# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄
```

```
result = urllib.request.urlopen(url).read().decode("utf-8") # decode -> 컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
#위의 url의 html 문서들을 result 변수에 담았다.
```

```
soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

```
# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.
result1 = soup.find_all('span', class_="date")
result2 = soup.find_all('p', class_="con")
```

```
# 3. 시청자게시판의 날짜와 본문을 params / params2 리스트에 담는다.
```

```
params1 = []
params2 = []
for i,k in zip(result1,result2):
    params1.append(i.get_text(" ",strip=True) )
    params2.append(k.get_text(" ",strip=True) )
```

```
# 4. 날짜와 본문을 한줄로 묶어서 출력하기
for j,h in zip(params1,params2):
    print(j+' '+h)
```

```
####
```

제일 중요!

e.g 11) 여태까지는 딱 1페이지(게시판 1번째 페이지)의 내용을 가져와서 웹스크롤링을 했는데, 전체 게시판의 페이지 내용을 웹스크롤링하려면??  
(여기서는 22번 페이지까지 있음.)

즉 레이디버그 전체 게시판의 글을 다 스크롤링해서 예제 10번 결과처럼 출력되게 하시오.

list\_url 을 between 을 쓴다던가해서 범위로 바꿔야 하나??

<https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuld=106>

<https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuld=106> 2페이지 url

<https://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=3&hmpMnuld=106> 3페이지 url

위의 url 들을 잘 살펴보면 페이지 번호만 다르고 나머지는 모두 동일하다는 것을 확인할 수 있다.

그렇다고 이걸 페이지마다 노가다로 일일이 url 입력해서 긁어오는것은 너무 비효율적!

e.g 12) 아래의 결과만 출력하는데,

```
####
```

```
for i in range(1,23):
```

```
    print('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' +str(i) + '&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&')
```

```
####
```

위에서 발견한것 처럼, 페이지번호만 다르기 때문에 그 숫자부분만을 for 문과 '+' 을 사용해 새로 입력+ 연결 해주면서 출력하면 되는 것  
단 전체 url은 문자형이기때문에 for loop을 이용하여 입력하는 숫자도 문자로 변형되어야 하기때문에 str(m) 를 돌려준다.

e.g 13) e.g 12 번 코드를 e.g 10) 코드에 적용해서 레이디 버그 전체 게시판의 글들이

```
####
```

# 1. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# e.g 12) 의 코드를 list\_url부분에 적용한다.

```
for m in range(1,23):
```

```
    list_url='http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' +str(m) + '&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
```

# 이후의 코드를 새로넣은 for 문의 영역안에 둘수 있도록, 전부 드레그 한 후 tap!

```
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)
```

```
# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄
```

```
result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)
```

```
    #위의 url의 html 문서들을 result 변수에 담았다.
```

```
soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = "date")
```

```
result2 = soup.find_all('p', class_ = "con")
```

# 3. 시청자게시판의 날짜와 본문을 params / params2 리스트에 담는다.

```

params1 = []
params2 = []
for i,k in zip(result1,result2):
    params1.append(i.get_text(" ",strip=True) )
    params2.append(k.get_text(" ",strip=True) )

```

# 4. 날짜와 본문을 한줄로 묶어서 출력하기

```

for j,h in zip(params1,params2):
    print(j+' '+h)

```

# 1번일때의 날짜와 리뷰 게시물 쪽 가져오고, 2번일때의 ~~ 반복 ... 22번일때의 날짜와 리뷰 함께 출력!

# params 리스트들이 'for 문안쪽'에 있는 것은 m에 새로운 숫자가 들어갈때마다 새로운 내용이들어간다는 것

# 페이지 1번돌때의 명령문에 따라 결과를 1줄 출력하고, 페이지 2번이 되면 다시 빈리스트로 인식하여 2번의 내용을 append하고 1줄 출력.. 을 반복!

###

e.g 14) params1 과 params2 에 레이디버그 시청자 게시판의 모든 게시 날짜와 본문내용이 들어가게 코드를 수정하시오.

###

# 0. 웹스크롤링에 필요한 모듈을 import

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 1. 레이디버그 게시판 게시날짜와 게시글 전체를 다 담을 리스트 2개를 생성한다.

```

params1 = []

```

```

params2 = []

```

# e.g 12) 의 코드를 list\_url부분에 적용한다.

for m in range(1,23): #페이지번호를 1~22번까지 변경하기 위해 for 문 사용

```

    list_url='http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+str(m) + '&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819
    &searchKeyword=&searchCondition=&searchConditionValue=0&'

```

# 이후의 코드를 새로넣은 for 문의 영역안에 둘수 있도록, 전부 드레그 한 후 tap!

```

url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)

```

```

# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄

```

```

result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)

```

#위의 url의 html 문서들을 result 변수에 담았다.

```

soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것

```

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```

result1 = soup.find_all('span', class_ ="date")

```

```

result2 = soup.find_all('p', class_ ="con")

```

```

for i,k in zip(result1,result2):

```

```

    params1.append(i.get_text(" ",strip=True) )

```

```

    params2.append(k.get_text(" ",strip=True) )

```

# 3. params 1 & 2에 append된 22개 페이지의 날짜와 본문 내용을 한줄로 묶어서 출력하기

```

for j,h in zip(params1,params2):

```

```

    print(j+' '+h)

```

# e.g 13) 은 레이디 버그 게시판 전체글을 화면에 출력하는 코드

# e.g 14) 는 레이디 버그 게시판 전체글을 params1 과 params2에 담고 화면에 출력하는 코드

###

(점심시간문제)

문제 415. 레이디 버그 게시판의 전체 게시글은 총 몇건인가?

###

# 0. 웹스크롤링에 필요한 모듈을 import

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 1. 레이디버그 게시판 게시날짜와 게시글 전체를 다 담을 리스트 2개를 생성한다.

params1 = []

# e.g 12) 의 코드를 list\_url부분에 적용한다.

for m in range(1,23): #페이지번호를 1~22번까지 변경하기 위해 for 문 사용

list\_url='http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+str(m) + '&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'

url= urllib.request.Request(list\_url)

result = urllib.request.urlopen(url).read().decode("utf-8")

soup = BeautifulSoup(result, "html.parser")

# 2. 시청자 게시판의 본문 내용을 가져온다.

result1 = soup.find\_all('p', class\_="con")

for i in result1: # 가져온 내용을 params1에 append 한다.

params1.append(i.get\_text(" ",strip=True) )

# 3. 시청자 게시판의 전체 글의 수가 몇인지 카운트

print( len(params1))

###

문제 416. 게시글 429개 전체를 c드라이브 밑에 mytext34.txt 라는 이름으로 저장하시오.

###

# 0. 웹스크롤링에 필요한 모듈을 import

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 1. 레이디버그 게시판 게시날짜와 게시글 전체를 다 담을 리스트 2개를 생성한다.

params1 = []

params2 = []

# e.g 12) 의 코드를 list\_url부분에 적용한다.

combine=[]

f = open("c:\\data\\mytext34.txt","w",encoding='UTF8') # write하기위한 파일을 만드는 명령어 'w' 는 write



for m in range(1,23): #페이지번호를 1~22번까지 변경하기 위해 for 문 사용

```
list_url='http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page='+str(m) + '&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&'
```

# 이후의 코드를 새로넣은 for 문의 영역안에 둘수 있도록, 전부 드레그 한 후 tap!

```
url= urllib.request.Request(list_url) # 사람이 알아볼 수 있는 위의 url을 파이썬이 알아볼 수 있도록 변환(parshing)하는것(1번째 작업)
```

```
# print(url) # <urllib.request.Request object at 0x00000187F00DCE80> 확인차 한번 출력해봄
```

```
result = urllib.request.urlopen(url).read().decode("utf-8") # decode ->컴퓨터의 언어를 사람이 알아볼수있도록 변환하는 것 (2번째 작업)  
#위의 url의 html 문서들을 result 변수에 담았다.
```

```
soup = BeautifulSoup(result, "html.parser") # e.g 2) 의 답 부분 // BeautifulSoup을 쓰도록 파싱 한 것
```

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ ="date")
```

```
result2 = soup.find_all('p', class_ ="con")
```

```
for i,k in zip(result1,result2):
```

```
    params1.append(i.get_text(" ",strip=True) )
```

```
    params2.append(k.get_text(" ",strip=True) )
```

# 3. params 1 & 2에 append된 22개 페이지의 날짜와 본문 내용을 한줄로 묶어서 write 하기.

```
for j,h in zip(params1,params2):
```

```
    f.write(j+' '+h+'\n') # 출력을 할때 결과물마다 한 단씩 엔터처리를 할 수있도록 '+'\n' 을 붙여준다.
```

```
f.close()
```

###

143 웹스크롤링 실전2 (중앙일보사)

중앙일보사 홈페이지에서 인공지능으로 검색했을때 나오는 기사들을 조회하고, 그 기사들을 전부 웹스크롤링한다. 가 목표!

e.g 1) 중앙일보사 홈페이지에서 인공지능으로 검색했을때 나오는 url 을 가져오시오. (레이디 버그때처럼 페이지 번호들이 하단에 떠야함)

<https://news.joins.com/Search/TotalNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews>

page 가 나와야함 2페이지로 넘어갔다가 다시 1페이지로 넘어오면 위와 같은 url이 나옴.

e.g 2) 위의 사이트에서 보이는 상세기사를 클릭하고 그 상세 기사의 url 을 가져오시오.

url1 = <https://news.joins.com/article/23947044>

url2 = <https://news.joins.com/article/23946979>

url3 = <https://news.joins.com/article/23946876>

3개의 기사 url을 볼때 http~~~ 2394 까지는 같은데 그 뒤에서 규칙성을 발견하기 힘들. 이런 경우는 어떻게 스크롤링 해야 하나?

e.g 3) 인공지능으로 검색했을때 나오는 상세기사들의 url을 스크롤링 하시오.

###

# 1. 웹스크롤링에 필요한 모듈을 import 한다.

```
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

# 2. 중앙일보에서 인공지능으로 검색했을때 나오는 첫 페이지의 html코드를  
# beautiful soup 에서 이용할 수 있도록 파싱.

```
list_url = 'https://news.joins.com/Search/TotalNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews'
url= urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, "html.parser")
```

# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.  
# 개발자 모드로 찾아보니 태그는 h2, class 는 headline mg 이다.

```
result1 = soup.find_all('h2', class_="headline mg") # 상세기사 페이지의 1페이지에 있는 세부기사의 제목들이 리스트로 스크롤링 되었다.
print(result1)
```

# 4. result1 의 유형은 list이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼낸다.  
# 하나씩 빼면서 href의 값을 얻어 낸다.

```
for i in result1: #
    for k in i: # result1 리스트의 요소를 하나씩 빼는 코드
        # print(k) # h2 태그에서 a 태그의 html 코드(<a href~~ 부분)의 값을 출력하기 위한 코드 #
        print(k.get("href"))
```

###  
왜 for i 밑에 for k를 쓰면 a href 가 나오는가?

그냥 for i in result1:  
 print(i)  
를 하면

```
[<h2 class="headline mg"><a href="https://news.joins.com/article/23947044" target="_blank">'AI 발전에 써라' 동원 김재철 명예회장, KAIST에 500억 기부</a></h2>
```

이런식으로 리스트에 담기면서 나옴.

```
for i in result1:
    print(i.get("href"))
    # h2 태그에서 href 값을 찾으려는 시도 -> 찾아지는게 없어서 none 가 나옴.
```

그래서

```
for i in result1:
    for k in i:
        # 를 하면
        # h2 태그가 감싸고 있는 안쪽의 a 태그를 가지고 오게 되는 것.
```

위의 실행문에서 result 는  
html 문서가 리스트에 들어있고 그것들을 불러오는 것!  
그래서 i 의 기준은 일반적으로 for i 할때처럼 숫자가 아니라 태그를 기준으로 뽑혀지는 것

일반적인 파이썬에서 리스트를 불러오는 것과는 별개임.

이해한바대로 적어보자면

지금 위의 result1 은 html 문서의 태그를 읽어온 결과물 이니까 일반적으로 파이선에서 쓰는 for i 를 썼을때  
처럼 i가 순번으로 숫자 1,2,3 입력되는 것이아니라 첫번째 태그, 두번째 태그, 세번째 태그 ... 이런식으로 i에 대입되어서  
출력이 되는것 같음!

[구윤모(데싸12기)] [오후 2:37]

```
a=[1,2,3,4,5]
a=[[1],[2],[3],[4],[5]]
```

```
for i in a:
    를 하면 a=[1,2,3,4,5] 가 출력되는 것이고
```

```
for i in a:
    for k in i:
        를 하면 a=[[1],[2],[3],[4],[5]] 가 되어서 list in list 를 출력
```

```
for i in result1:
    print(i)
    를 하면
i가 1일때 <h2 class="headline mg"> <a href="https://news.join.com/article/23947044" target="_blank">'AI 발전에 써라' ~ ~ </a></h2>
i가 2일때 <h2 class="headline mg"> <a href="https://news.join.com/article/23946979" target="_blank">美 무인기, ~ ~ ~ </a></h2>
i가 3일때 <h2 class="headline mg"> <a href="https://news.join.com/article/23946876" target="_blank">라인, 日 증 ~ ~ ~ </a></h2>
```

이런식으로 출력되는 것이고

```
<h2 class ~ ~ ~ > ~ </h2> 가 1개의 단위
<a href ~ ~ ~ > ~ ~ </a> 가 h2 로 둘러준 리스트 안의 유일한 1개의 내용물인데
리스트 인 리스트가 된것
```

그래서 a = [[1],[2],[3],[4],[5],[6]]

정확한 개념은 아니지만 위의 a list 에서 [ 가 <h2 class ~ ~ ~ > ] 는 <~h2> 이고  
1 -> <a href= ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ > ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ </a> 인 것

```
for i in result1:
    for k in i:
        print(k)
```

#####

e.g 4) 위의 상세기사 url 을 params 라는 비어있는 리스트에 다 append 시키시오.  
#####

```
# 1. 웹스크롤링에 필요한 모듈을 import 한다.
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈
import urllib.request # <- 웹상의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

```
# 2. 중앙일보에서 인공지능으로 검색했을때 나오는 첫 페이지의 html코드를
# beautiful soup 에서 이용할 수 있도록 파싱.
```

```
list_url = 'https://news.join.com/Search/TotalNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews'
url= urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, "html.parser")
```

```
# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.  
# 개발자 모드로 찾아보니 태그는 h2, class 는 headline mg 이다.
```

```
result1 = soup.find_all('h2', class_="headline mg") # 상세기사 페이지의 1페이지에 있는 세부기사의 제목들이 리스트로 스크롤링 되었다.  
print(result1)
```

```
# 4. result1의 유형은 list이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼낸다.  
# 하나씩 빼면서 href의 값을 얻어 낸다.
```

```
params = [] # a 태그의 href에 해당하는 내용들을 담을 리스트  
for i in result1:  
    for k in i: # result1 리스트의 요소를 하나씩 빼는 코드  
        # print(k) # h2 태그에서 a 태그의 html 코드(<a href~~ 부분)의 값을 출력하기 위한 코드 #  
        params.append(k.get("href")) # 추출해낸 a 태그 부분을 비어있는 params에 append.
```

```
print(params)
```

```
###
```

e.g 5) 상세기사 url 중에 하나를 복사해 오고 그 상세기사 url의 웹페이지로 접속해서 본문기사의 태그이름과 클래스 이름이 무엇인지 확인하시오.

```
url = https://news. Joins.com/article/23947044
```

태그는 div

클래스는 article\_body mg fs4

이다.

e.g 6) 위의 상세기사 본문을 스크롤링 해서 출력하시오.

```
###
```

```
# 1. 웹스크롤링에 필요한 모듈을 import 한다.  
from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈  
import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈
```

```
# 2. 중앙일보에서 상세기사 url로 검색했을때 나오는 기사1개 페이지의 html코드를  
# beautiful soup에서 이용할 수 있도록 파싱.
```

```
list_url = 'https://news. Joins.com/article/23947044'
```

```
url= urllib.request.Request(list_url)  
result = urllib.request.urlopen(url).read().decode("utf-8")  
soup = BeautifulSoup(result, "html.parser")
```

```
# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.  
# 개발자 모드로 찾아보니 태그는 div, class 는 article_body mg fs4 이다.
```

```
result1 = soup.find_all('div', class_="article_body mg fs4") # 상세기사의 본문 내용이 담겨있다.  
params=[]
```

```
for i in result1:  
    params.append(i.get_text(" ",strip=True) )
```

```
print(params)
```

```
###
```

e.g 7) 상세기사 url 을가져와서

```
###
```

#1. 웹스크롤링에 필요한 모듈을 임포트 합니다.

```
from bs4 import BeautifulSoup
import urllib.request
```

#2. 중앙일보에서 인공지능으로 검색했을 나오는 첫페이지의 html 코드를

# beautiful soup 에서 이용할 수 있도록 파싱합니다.

#3. 편리한 작업을 위해 함수를 만든다.

```
def j_scroll():
    list_url = 'https://news.joins.com/Search/JoongAngNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongAngNews'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup(result, "html.parser")
```

```
result1 = soup.find_all('h2', class_='headline mg')
```

```
params=[]
for i in result1:
    for k in i:
        params.append(k.get("href"))
return(params)
```

```
print(j_scroll())
```

###

e.g 8) 상세기사 url (e.g 7) 로 본문 기사를 스크롤링해서 리스트에 담았던 e.g 6) 번을 j\_detail\_scroll() 함수로 만드시오.

###

# 1. 웹스크롤링에 필요한 모듈을 import 한다.

from bs4 import BeautifulSoup # <- 그냥 웹스크롤링 할때 필요한 기본 모듈

import urllib.request # <- 웹사이트의 url을 파이썬이 인식할 수 있도록 해주는 모듈

# 2. 중앙일보에서 상세기사 url로 검색했을때 나오는 기사1개 페이지의 html코드를

# beautiful soup 에서 이용할 수 있도록 파싱.

```
def j_detail_scroll():
    list_url = 'https://news.joins.com/article/23947044'
```

```
url= urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, "html.parser")
```

# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.

# 개발자 모드로 찾아보니 태그는 div, class 는 article\_body mg fs4 이다.

```
result1 = soup.find_all('div', class_="article_body mg fs4") # 상세기사의 본문 내용이 담겨있다.
```

```
params=[]
```

```
for i in result1:
    params.append(i.get_text(" ",strip=True))
```

```
return(params)
```

```
print(j_detail_scroll())
```

###

e.g 9) 지금 만든 j\_detail\_scroll() 함수는 상세기사 딱 1개의 본문을 출력하는 함수인데, 이 j\_detail\_scroll() 함수에 j\_scroll() 함수를 실행했을때 나오는 상세 url 여러개를 제공할 수 있도록 코드를 수정하시오.

###

#1. 웹스크롤링에 필요한 모듈을 임포트 합니다.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def j_scroll():
```

```
    list_url = 'https://news.joins.com/Search/JoongAngNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongAngNews'
```

```
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")
```

```
    result1 = soup.find_all( 'h2', class_ = 'headline mg')
```

```
    params=[]
```

```
    for i in result1:
```

```
        for k in i:
```

```
            params.append(k.get("href"))
```

```
    return params # 상세기사 url들이 담긴다.
```

```
def j_detail_scroll():
```

```
    #2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
```

```
    # beautiful soup 에서 이용할 수 있도록 파싱합니다.
```

```
    list_url = j_scroll() #<- 위에서 짰 j_scroll() 함수의 리턴되는 내용들 = params 가 여기에 리스트로 할당되는 것
```

```
    params2 =[]
```

```
    for i in list_url:
```

```
        url = urllib.request.Request(i)
```

```
        result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
        soup = BeautifulSoup( result, "html.parser")
```

```
    #3. 상세기사의 본문을 가져올 수 있도록 태그와 클래스를 찾습니다.
```

```
    # 찾아보니 태그는 div 이고 클래스 이름은 article_body mg fs4 입니다.
```

```
    result1 = soup.find_all( 'div', class_ = 'article_body mg fs4')
```

```
    #4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
```

```
    # 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.
```

```
    for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
```

```
        params2.append( i.get_text(" ", strip=True) )
```

```
    return params2
```

```
print( j_detail_scroll())
```

```
###
```

코드가 복잡하니 이래르 돕기위한 설명 구간

#1. 웹스크롤링에 필요한 모듈을 임포트 합니다.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def j_scroll():
```

```
    list_url = 'https://news.joins.com/Search/JoongAngNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongAngNews'
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup( result, "html.parser")
```

```
    result1 = soup.find_all( 'h2', class_ = 'headline mg')
```

```

params=[]
for i in result1:
    for k in i:
        params.append(k.get("href"))
return params # 상세기사 url들이 담긴다.

```

```

['https://news.ajins.com/article/23947227',
'https://news.ajins.com/article/23947225',
'https://news.ajins.com/article/23947223',
'https://news.ajins.com/article/23947215',
'https://news.ajins.com/article/23947044',
'https://news.ajins.com/article/23946979',
'https://news.ajins.com/article/23946876',
'https://news.ajins.com/article/23946841',
'https://news.ajins.com/article/23946757',
'https://news.ajins.com/article/23946642']

```

params 에 들어가는 url 들!

```

def j_detail_scroll():
    #2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
    # beautiful soup 에서 이용할 수 있도록 파싱합니다.
    list_url = j_scroll() # <- 위에서 짰 j_scroll() 함수의 리턴되는 내용들 = params 가 여기에 리스트로 할당되는 것
    params2=[]
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")

        #3. 상세기사의 본문을 가져올 수 있도록 태그와 클래스를 찾습니다.
        # 찾아보니 태그는 div 이고 클래스 이름은 article_body mg fs4 입니다.

        result1 = soup.find_all( 'div', class_ = 'article_body mg fs4')

        #4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
        # 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.

        for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )

    return params2

```

```

print( j_detail_scroll())

```

###

중앙일보의 기사를 스크롤링하는 내용 총정리

1. 검색키워드(예:인공지능)를 가지고 검색한 후 그 url 을 얻어낸다.
2. 상세기사 url 을 리스트에 담는 j\_scroll() 함수를 생성한다.
3. 상세기사 url 로 기사 본문을 스크롤링하는 j\_detail\_scroll() 함수를 생성한다.
- 4.

문제 417. 동아일보에서 검색키워드(인공지능) 을 가지고 검색을 한 후 그 url 을 얻어낸다.

```

https://www.donga.com/news/search?p=1'&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=2&more=1&sorting=1&search_date=1&v1=&v2=&range=1
https://www.donga.com/news/search?p=16'&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=2&more=1&sorting=1&search_date=1&v1=&v2=&range=1
https://www.donga.com/news/search?p=31'&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=2&more=1&sorting=1&search_date=1&v1=&v2=&range=1

```

1페이지당 15개씩 숫자가 늘어남 (패턴 발견)

문제 418. 상세기사 url 을 리스트에 담은 d\_scroll() 함수를 생성하시오.

테그는 p // 클래스는 tit

```
###

from bs4 import BeautifulSoup
import urllib.request

def d_scroll():
    list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=2&more=1&sorting=1&search_date=1&v1=&v2=&range=1'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")

    result1 = soup.find_all( 'p', class_ = 'tit')

    params=[]
    for i in result1:
        for k in i:
            params.append(k.get("href"))
    return params  # 상세기사 url들이 담긴다.

print(d_scroll())

###
```

문제 419. 상세기사 url에 h 기사본문을 스크롤링하는 d\_detail\_scroll() 함수를 생성한다.

```
###

def d_detail_scroll():
    #2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
    # beautiful soup 에서 이용할 수 있도록 파싱합니다.
    list_url = d_scroll() # <- 위에서 짰 j_scroll() 함수의 리턴되는 내용들 = params 가 여기에 리스트로 할당되는 것
    params2 =[]
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")

        #3. 상세기사의 본문을 가져올 수 있도록 테그와 클래스를 찾습니다.
        # 찾아보니 테그는 div 이고 클래스 이름은 article_body mg fs4 입니다.

        result1 = soup.find_all( 'div', class_ = 'article_txt')

        #4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
        # 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.

        for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )

    return params2

print(d_detail_scroll())

###

완성본
from bs4 import BeautifulSoup
import urllib.request

def d_scroll():
    list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=2&more=1&sorting=1&search_date=1&v1=
```



[=&v2=&range=1'](#)

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
```

```
result1 = soup.find_all( 'p', class_ = 'txt')
```

```
params=[]
for i in result1:
    for k in i:
        params.append(k.get("href"))
return params # 상세기사 url들이 담긴다.
```

```
def d_detail_scroll():
```

```
#2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
```

```
# beautiful soup 에서 이용할 수 있도록 파싱합니다.
```

```
list_url = d_scroll() # <- 위에서 짰 j_scroll() 함수의 리턴되는 내용들 = params 가 여기에 리스트로 할당되는 것
```

```
params2 =[]
```

```
for i in list_url:
```

```
    url = urllib.request.Request(i)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup( result, "html.parser")
```

```
#3. 상세기사의 본문을 가져올 수 있도록 태그와 클래스를 찾습니다.
```

```
# 찾아보니 태그는 div 이고 클래스 이름은 article_body mg fs4 입니다.
```

```
result1 = soup.find_all( 'div', class_ = 'article_txt')
```

```
#4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
```

```
# 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.
```

```
for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
```

```
    params2.append( i.get_text(" ", strip=True) )
```

```
return params2
```

```
print( d_detail_scroll())
```

```
###
```

문제 420. 한겨레 신문사에서 인공지능으로 검색했을때 나오는 기사 본문을 스크롤링하는 함수 두개를 생성하시오.

h\_scroll() : 상세기사 url 을 가져오는 함수

h\_detail\_scroll() : 상세기사 url 로 기사본문을 가져오는 함수

1페이지 url :<http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=0>

2페이지 url :<http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=1>

3페이지 url :<http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=2>

```
###
```

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

```
def h_scroll():
```

```
    list_url = 'http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=0'
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup( result, "html.parser")
```

```

result1 = soup.find_all('dt')

params=[]
for i in result1:
    for k in i:
        params.append('http:'+ k.get("href")) ) #앞에 http:를 명시해서 url을 가져올 수 있도록 한다.
return params # 상세기사 url들이 담긴다.

def h_detail_scroll():
    #2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
    # beautiful soup 에서 이용할 수 있도록 파싱합니다.
    list_url = h_scroll() # <- 위에서 짰 j_scroll() 함수의 리턴되는 내용들 = params 가 여기에 리스트로 할당되는 것
    params2 =[]
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")

        #3. 상세기사의 본문을 가져올 수 있도록 태그와 클래스를 찾습니다.
        # 찾아보니 태그는 div 이고 클래스 이름은 article_body mg fs4 입니다.

        result1 = soup.find_all('div', class_ = 'text')

        #4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
        # 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.

        for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
            params2.append( i.get_text(" ", strip=True) )

    return params2

print( h_detail_scroll())

###

```

(오늘의 마지막 문제)

문제 421. 본인이 원하는 신문에 원하는 키워드로 검색했을때 나오는 본문기사들을 수집하는 함수 2개를 생성하시오.

밑에 페이지번호 1,2,3만 스크롤링 하시오)

hd\_scroll() : 상세기사 url 가져오는 함수

hd\_detail\_scroll() : 상세기사 url 로 본문 기사를 가져오는 함수.

헤럴드 경제 '백신' 검색

1 페이지 url : <http://biz.heraldcorp.com/search?q=%EB%B0%B1%EC%8B%A0&dt=1&nt=1&np='1'&hq=>

2 페이지 url : <http://biz.heraldcorp.com/search?q=%EB%B0%B1%EC%8B%A0&dt=1&nt=1&np='2'&hq=>

3 페이지 url : <http://biz.heraldcorp.com/search?q=%EB%B0%B1%EC%8B%A0&dt=1&nt=1&np='3'&hq=>

###

```

from bs4 import BeautifulSoup
import urllib.request
import time

```

```

def hd_scroll():
    params=[]
    for i in range(1,4):
        list_url = 'http://biz.heraldcorp.com/search?q=%EB%B0%B1%EC%8B%A0&dt=1&nt=1&np='+str(i)+'&hq='

        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode('UTF-8', errors='ignore')
        soup = BeautifulSoup( result, "html.parser")

        result1 = soup.find_all('p', class_ ='ntitle')

```

```

for i in result1:
    for k in i:
        params.append(k.get("href"))
return params

```

```

def hd_detail_scroll():
    #2. 상세 기사 url 로 검색했을 나오는 페이지의 html 코드를
    # beautiful soup 에서 이용할 수 있도록 파싱합니다.
    list_url = hd_scroll()
    params2 = [ ]
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode('UTF-8', errors='ignore')
        soup = BeautifulSoup( result, "html.parser")

        #3. 상세기사의 본문을 가져올 수 있도록 태그와 클래스를 찾습니다.
        # 찾아보니 태그는 div 이고 클래스 이름은 con_left 입니다.

        result1 = soup.find_all( 'div', class_ = 'con_left')

        #4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를
        # 하나씩 빼내면서 본문의 텍스트를 얻어냅니다.

        for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
            # time.sleep(10)
            params2.append( i.get_text(" ", strip=True) )

    return params2

print( hd_detail_scroll())

```

###

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xc7 in position 94318: invalid continuation byte 라는 에러가 나와서

CP949 로도 바꿔 보았으나 다른 에러 발생

UnicodeDecodeError: 'cp949' codec can't decode byte 0xad in position 576: illegal multibyte sequence

이 두가지 에러를

result = urllib.request.urlopen(url).read().decode('UTF-8', errors='ignore') 를 사용함으로써 해결!

20.12.16 파이썬 강의 Day 15(~문제422)

SQL	vs	Python
select		파이썬의 자료형 5가지 기반으로 관련 함수들
from		소개를 하면서 수업
where		매일 다른 함수를 배움
를 무한 반복 하며 학습		SQL대비 익숙해지고 숙달될 시간이 없음 -> 그래서 어렵게 느끼는 것
		↓↓↓
		앞에서 배웠던 함수들을 처음부터 다 암기할 수 없음
		매일 쓰다보면 자연스럽게 자주쓰는 함수들은 기억이남 (나의 경우 : for i in range~)
		↓↓↓
		파이썬 코드의 큰 틀을 암기하면 된다.
		1. 필요한 모듈 import
		2. 데이터 발생(주사위던지기, 사람 키 데이터 10만명 만들기 등) 또는 수집 (웹스크롤링 등)
		3. 수집된 데이터를 리스트에 append 시켜서 저장

4. 저장한 리스트의 데이터를 가지고 통계분석을 한다.
5. 결과를 출력한다.

데이터 분석가가 갖춰야 할 필수 기술 ?

1. 웹스크롤링 기술
2. 통계로 데이터 분석
3. 머신 러닝으로 데이터 분석
4. 딥러닝으로 데이터 분석

를 위해서는 위의 파이썬 코드 큰 틀 1~5를 잘 숙지하고 있으면 가능하다!! 겁먹지 말자!

###

어제 웹스크롤링한 사이트

1. 레이디버그 게시판
2. 중앙일보 신문사
3. 동아일보 신문사
4. 한겨레 신문사
5. (마지막문제) 각자 스크롤링하기 원하는 사이트에서 하기!

대부분의 학생들이 5번을 실패함. 공통적으로 웹스크롤링이 안되었던 이유를 발견!

beautiful soup 문법중에서 좀 더 알아야 할 내용

e.g 1) 웹스크롤링할 사이트에 접속 한다.

<https://futurechosun.com/>

e.g 2) 위의 사이트에서 '봉사' 로 기사를 검사

page1 url : <https://futurechosun.com/page/1?s=%EB%B4%89%EC%82%AC>

page2 url : <https://futurechosun.com/page/2?s=%EB%B4%89%EC%82%AC>

page3 url : <https://futurechosun.com/page/3?s=%EB%B4%89%EC%82%AC>

e.g 3) 위의 첫페이지의 html 코드를 파이썬으로 불러오는 작업

###

1. 웹스크롤링에 필요한 모듈을 import 한다.

```
from bs4 import BeautifulSoup
```

```
import urllib.request # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈
```

###

2. 첫 페이지의 url 을 파싱한다. -> url 을 파이썬이 알아볼 수있는 언어로 디코드 한다.

```
list_url = 'https://futurechosun.com/page/1?s=%EB%B4%89%EC%82%AC '
```

```
url = urllib.request.Request(list_url)
```

```
result = urllib.request.urlopen(url).read().decode("utf-8")
```

- 2-1. '2.' 까지의 작업이 잘 입력되었는지 확인하기 위해 실행

```
print(result) html코드가 출력되었다. // 중간 중간 print를 해서 내 코드가 맞는지 확인하는 작업을 꼭 해줘야 함.
```

###

3. 웹스크롤링 전문 모듈인 beautiful soup 모듈( 중 find\_all )을 이용할 수 있게끔

내려 받은 html 코드를 파싱한다.

'이용할 수 있게 한다' means -> beautiful soup 모듈의 함수중 하나인 find\_all을 사용해서

태그 이름과 클래스 이름을 가지고 검색하기 원하는 지점을 빠르게 찾아갈 수 있도록 명령어를 입력해주는 것.

실행!

```
list_url = 'https://futurechosun.com/page/1?s=%EB%B4%89%EC%82%AC '
```

```
url = urllib.request.Request(list_url)
```

```
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser") < - 이걸 입력해준다.
```

print(soup) html 코드가 출력되었다. // 2번과의 차이점은 beautifulsoup 을 이용해 파이썬이 내가 원하는 위치의 데이터를 추출해올 수 있도록 된 것.

####

4. 조선일보 더 나은 미래 첫페이지의 보이는 기사들(12개)의 상세 기사 url 을 알아내기 위해 태그 이름 & 클래스 이름이 무엇인지 알아낸다.

href~ 로 되어있는 부분을 찾으려한다.

1) 크롬에서 F12 눌러 개발자 모드로 진입

2) ctrl+shift+C 누르고 상세기사의 제목 부분 클릭 -> 트랙킹된 html 코드에서 가장 가까운(위쪽방향) 태그와 클래스 이름을 찾는다.

```
<div class="elementor-post__text"><div class="elementor-post__title"><a href="https://futurechosun.com/archives/52491"> “편견에 주눅 들었던 결혼 이주 여성들... ‘봉사’로 자존감 되찾았다” </a></div><div class="elementor-post__meta-data"><span class="elementor-post-author"> 박선하 기자 </span><span class="elementor-post-date"> 2020년 12월 1일 </span></div><div class="elementor-post__excerpt"><p>[우리사회 쉼주민] 박시은 ‘다빛나’ 대표 사람은 타인과 사회로부터 상처를 받으면 주눅 들게 된다. 상처를 성장의 발판으로 삼은 사람들의 이야기가 영웅담처럼 읽히는 것도 그만큼 그런 일이 드물고 어렵다는</p></div></div>
```

가 그 대상인데, 보면 위쪽에서 가장 가까운 태그와 클래스를 보면

태그는 div 이고, 클래스 이름 은 elementor-post\_\_title 임을 알 수 있다.

####

5. 태그 이름 div 에 클래스 이름은 elementor-post\_\_title 에 해당하는 beautiful soup 으로 파싱된 html 코드를 모두 긁어온다.

1~ 4 앞의 코드들 +

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
# result1=soup.find('div',class_='elementor-post__title').find('a')
result1=soup.find_all('div',class_='elementor-post__title')
[1] 위의 list_url 페이지에서 result1에 div 태그 & 클래스명 elementor-post__title 에 해당하는 모든 html 을 모두 긁어온다.
```

```
print(resul1)
을 해보면!
```

```
[ ] 안에 [ <div class="elementor-post__title"><a href="https://futurechosun.com/archives/52491"> “편견에 주눅 들었던 결혼 이주 여성들... ‘봉사’로 자존감 되찾았다” </a></div>,
<div class="elementor-post__title"><a href="https://futurechosun.com/archives/47330"> [글로벌 이슈] 전 세계 코로나19 확산으로... 해외 봉사 ‘올스톱’ 위기 </a></div> ]
이런 식으로 쭉 12개의 항목들이 [ ] 안에 들어있다.
```

```
<div class="elementor-post__title">~~</a></div> 까지가 1개의 항목
```

```
<div> 태그가 <div> <a href=~~></a> </div> 이런식으로 <a> 태그를 두르고 있는 형식으로 파싱된 html 코드들이 요소들로 구성되어 있다.
```

여기서 우리가 최종적으로 가지고 오고 싶은 것은 <a href="https://futurechosun.com/archives/47330"> 이 부분! (a 태그부분)

이를 위해 a태그로 접근하려면 그 요소들을 서 for 문을 사용해 하나씩 빼오는 작업이 필요!

☆ 리스트 형태의 자료에서 요소를 하나씩 빼올려면 for loop 문을 사용해야 한다!

####

6. result1 리스트의 요소들( beautiful soup 으로 파싱된 html 코드 '들') 을 for loop 문으로 하나씩 빼낸다.

```
list_url = 'https://futurechosun.com/page/1?s=%EB%B4%89%EC%82%AC'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result, "html.parser")
```

```
# print(soup)
# result1=soup.find('div',class_='elementor-post__title').find('a')
result1=soup.find_all('div',class_='elementor-post__title')

for i in result1:
    print(i)
를 하면
<div class="elementor-post__title"> <a href="https://futurechosun.com/archives/52491"> “편견에 주눅 들었던 결혼 이주 여성들... ‘봉사’로 자존감 되찾았다죠” </a>
</div>
이런식으로 한줄씩 출력된다.
```

하지만 우리가 원하는 부분은 ,,,,

```
####
```

7. for i in result1 을 이용해서 빼낸 요소인 html 코드 3줄에서 a 태그에 해당하는 부분만 가져오게 하시오!

```
for i in result1:
    print(i.find_all('a')) <- 어제 배우지 않았던 내용. 그냥 find_all 이 아니라 find_all() 괄호안에 내가 원하는 부분을 딱 지정해서 입력할 수 있음.
```

```
[<a href="https://futurechosun.com/archives/52491"> “편견에 주눅 들었던 결혼 이주 여성들... ‘봉사’로 자존감 되찾았다죠” </a>]
```

이런식으로 [] 리스트로 둘러져서 하나의 html 코드가 1줄씩 출력된다.

```
####
```

8. 위의 a 태그의 html 코드들을 담은 리스트는 요소를 딱 1개만 담고 있다.

그러므로 리스트 안의 그 요소만 딱 뽑아내서 출력을 하시오!

e.g)

a = [2] 인 리스트에서 2만 뽑아낼려면?

print(a[0]) 을 하면 된다.

적용!

```
for i in result1:
    print(i.find_all('a')[0])
```

```
<a href="https://futurechosun.com/archives/52491"> “편견에 주눅 들었던 결혼 이주 여성들... ‘봉사’로 자존감 되찾았다죠” </a>
```

```
<a href="https://futurechosun.com/archives/47330"> [글로벌 이슈] 전 세계 코로나19 확산으로... 해외 봉사 ‘울스톱’ 위기 </a>
```

```
.
.
.
```

이런식으로 딱 a 태그에 해당되는 부분만 뽑아짐! (div를 빼고)

```
####
```

9. 위에서 뽑아낸 a 태그의 html 문서에서 href의 값만 얻어 내시오!

```
for i in result1:
    print (i.find_all('a')[0].get("href"))
```

결과!

```
https://futurechosun.com/archives/52491
https://futurechosun.com/archives/47330
https://futurechosun.com/archives/47004
https://futurechosun.com/archives/46834
https://futurechosun.com/archives/43905
https://futurechosun.com/archives/43785
https://futurechosun.com/archives/39923
```

<https://futurechosun.com/archives/38545>

딱 원하는 부분만 뽑아졌다!!

###

10. 위의 상세기사 url 을 params 라는 리스트에 append 시킨다.

```
params=[]

for i in result1:
    params.append(i.find_all('a')[0].get("href"))

print(params)
```

9번의 결과물들이 (params 리스트 인) [] 안에 담겨서 출력된다.

###

11. 첫페이지 뿐만이 아니라 1페이지부터 3페이지까지 (1페이지당 24개, 24\*3 개) 의 기사 url 을 params 리스트에 append 시키시오.

params=[] <- params가 10번처럼 for 문과 for 문 사이에 그대로 있으면 첫번째 for 문이 반복될때마다 params 가 리셋되어서 72개가 아닌 맨 마지막 24개만 담긴다.

```
for i in range(1,4):
    list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")
    # result1=soup.find('div',class_='elementor-post__title').find('a')
    result1=soup.find_all('div',class_='elementor-post__title')
    for i in result1:
        params.append(i.find_all('a')[0].get("href"))
```

###

12. 조선일보 더 나은 미래 사이트는 제목뿐만 아니라 사진을 클릭했을때도 상세기사로 들어가는 구조로 되어있어서 기사는 12개인데 1페이지당 24개의 url이 출력 된다.

```
params=[]
for i in range(1,4):
    list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result, "html.parser")
    # result1=soup.find('div',class_='elementor-post__title').find('a')
    result1=soup.find_all('div',class_='elementor-post__title')
    for i in result1:
        params.append(i.find_all('a')[0].get("href"))
my_result = set(params) # set 함수를 이용해 중복이 제거된 요소들만 남는다.
my_result2 = list(my_result) # 중복이 제거된 요소들을 다시 리스트로 감싸서 리스트화 시킨다.
```

```
print(len (my_result2) ) # 36개
```

###

13. 위의 코드들을 가지고 bs\_scroll() 라는 함수로 생성하시오.

```
from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,4):
```

```

list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
stuff = BeautifulSoup( result, "html.parser")

res1 = stuff.find_all('div',class_="elementor-post__title")

for i in res1:
    params1.append(i.find_all('a')[0].get('href'))

return(params1)

print(len(bs_scroll()))

```

###

14. 더 나은 미래의 상세기사 url 하나를 가지고 본문 기사를 출력하시오.

<https://futurechosun.com/archives/52491> -> 기사 본문의 div & class

```

<div class="elementor-widget-container"><p><strong>
<div class="elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content" data-id="24e82692"
data-element_type="widget" data-widget_type="theme-post-content.default"><div class="elementor-widget-container"><p><strong>[우리사회 체주민] 박시은 ‘다빛나’ 대
표</strong></p><p>사람은 타인과 사회로부터 상처를 받으면 주눅 들게 된다. 상처를 성장의 발판으로 삼은 사람들의 이야기가 영웅담처럼 읽히는 것도 그만큼
그런 일이 드물고 어렵다는 방증이다. 결혼 이후 여성들의 모임 ‘다빛나’도 그런 노력의 결과물 중 하나다. 다빛나는 중국·베트남·네팔 등에서 온 결혼 이후 여성 26
명이 참여하는 자조 모임이다. 기댈 곳 없는 이주 여성끼리 마음을 나누던 모임이 사회봉사를 통해 이주 여성에 대한 인식을 바꾸는 단체로 자라났다.</p><p>다
빛나를 이끌어온 사람은 중국 엔지 출신 박시은(41) 대표다. 지난 23일 서울 광장동에서 만난 박 대표는 “봉사는 우리 자신을 위한 일이기도 하다”고 했다. 그는 “결
혼 이후 여성은 사회적으로 소외되고 경제적으로 어려운 경우가 많은데, 다른 사람을 도우면서 스스로 ‘무언가 할 수 있는 사람’이라는 자존감이 높아지기 때문”
이라고 했다. 박 대표는 “봉사를 통해 이주 여성 스스로 역량을 강화하고, 한국 내 이주민에 대한 인식을 개선하는 게 목표”라고 설명했다.</p><figure
id="attachment_52527" aria-describedby="caption-attachment-52527" style="width: 500px" class="wp-caption aligncenter"><noscript></noscript><figcaption id="caption-attachment-52527" class="wp-caption-text">지난달 23일 서울 광장
동에서 만난 박시은 다빛나 대표는 “이주민도 존중받아야 하는 평범한 사람이라는 걸 알리기 위해 시작한 봉사가 지금은 이주 여성들의 ‘자존감 지킴이’ 역할을
하고 있다”고 했다. / 이신영 c영상미디어 기자</figcaption></figure><p><strong>“아이들이 차별받을 때 가장 마음 아팠어요”</strong></p><p>박 대표는 대학을 졸업
하고 베이징에서 직장생활을 하다 남편을 만났다. 2006년 가족이 함께 한국으로 이주했다. 한국 생활은 녹록지 않았다. 박 대표는 “대학도 나왔고 직장생활도 해서
자신감이 있었지만 한국에서는 차별이 심해 상처를 많이 받았다”고 했다.</p><p>“가장 마음이 아플 땐 아이들이 차별받을 때였죠. ‘저 애 엄마가 중국인이니까 놀
지 마라’고 하는 사람도 많았어요. 서울 말씨를 익혔더니 사람들이 제가 중국 출신인 걸 모르고 ‘저 동네엔 중국인이 많아 더럽고 위험하다’고 서슴없이 말했어요.
거기에 또 상처를 받았어요.”</p><p>주저앉아 있을 순 없었다. 할 수 있는 일부터 하자는 생각으로 심리상담사 자격증반을 수강했다. 꼬박 3년이 걸렸다.</p><p>
“자격증을 따면서 스스로 내공이 쌓이니 가족은 물론 주변 사람들의 시선이 달라지는 걸 느꼈어요. 그때 저와 비슷한 고민을 가진 다른 결혼 이후 여성들이 자신
감을 되찾을 수 있도록 도와야겠다고 결심했습니다.”</p><p>여러 이주 여성들이 그의 뜻에 동참했고 2018년 함께 다빛나를 설립했다. “처음엔 다문화센터에서 떡
케이크 만들기 교육을 받은 후, 혼자 사는 어르신께 떡 케이크를 만들어 드렸어요. 예상보다 훨씬 좋은 반응에 탄력이 붙었죠. ‘그간 너무 외로웠는데 고맙다’며 눈
물을 보이는 어르신도 있었어요.”</p><p><strong>‘다문화’라는 단어에 담긴 편견 깨고 싶어</strong></p><p>봉사활동은 그동안 주눅 들어 있던 결혼 이후 여성들
에게 변화를 가져왔다. 박 대표는 “많은 회원이 ‘봉사를 통해 자존감이 올라간다’며 무척 행복해한다”고 했다. “오랜 기간 한국에서 차별받다 보니 부당한 일을 당
해도 항의조차 못 할 정도로 주눅이 든 사람들이 많았어요. 그런데 봉사를 하니 ‘나도 누군가에게 감사 인사를 받는 사람’이라며 얼굴이 확 밝아졌죠.”</p><p>봉사
활동의 범위도 넓어졌다. 독거 노인들에게 반려 식물을 선물하고, 지역 취약 계층 아동을 대상으로 세계 문화 체험 교육도 진행했다. 수공예품을 만들어 판 뒤 아
동센터 등에 기부하기도 했다. 박 대표는 “요즘은 복지단체에서 먼저 ‘좀 도와줄 수 있느냐’며 연락이 올 정도”라며 자랑했다. 최근에는 천주교에서 준비 중인 이주
민 동료 상담 기관 ‘엔피코’ 설립에도 참여하고 있다. 박 대표는 “말과 문화도 통하고, 서로의 상황을 깊이 공감하는 이주민끼리 서로를 지지해주는 모델”이라고 했
다.</p><p>최근 다빛나는 ‘다문화가 빛나는 나눔’에서 ‘다 함께 빛나는 나눔’으로 단체의 뜻을 바꿨다. 박 대표는 “‘다문화’라는 단어에 들어 있는 차별적인 시선 때
문”이라고 했다. “호주 출신 방송인 샘 해밍턴 가족은 ‘다문화’가 아니라 ‘글로벌 가족’이라고 부르잖아요. 사전적 의미로는 그 가족도 다문화 가족인데도요. 다문화
라는 단어에는 한국보다 못사는 나라에서 온 도움이 필요한 사람들이란 시선이 담겨서가 아닐까요? 다문화라는 단어에 갇히지 않고, ‘한국 출신 한국인’들과도 손
을 잡는다는 의미로 ‘다 함께 빛나는 나눔’이라고 부르기로 했습니다.”</p><p>다빛나의 목표는 두 가지다. 첫째는 결혼 이후 여성들의 당당한 자립. 둘째는 봉사활
동이다. 궁극적으로는 결혼 이후 여성에 대한 편견을 없애고 싶다고 했다. 최근엔 일자리 창출 사업도 시작했다. 재단법인 ‘밴드’의 도움으로 지난달부터 사회적기
업 ‘화유 플라워’에 이주 여성 2명을 파견하고 있다.</p><p>“대표님이 ‘이주 여성 분들이 아주 손이 빨라서 일에 큰 도움이 된다’며 놀라워하세요. 이렇게 한번 저희
에게 기회를 주면, 제 몫을 충분히 한다는 걸 보여줄 수 있어요. 당당한 한 명의 주민으로 결혼 이후 여성이 존중받을 수 있게 꾸준히 활동을 이어갈 계획입니다.”
</p><p>박선하 더 나은미래 기자 sona@chosun.com</p></div></div>

```



```

from bs4 import BeautifulSoup
import urllib.request
list_url = 'https://futurechosun.com/archives/52491'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup=BeautifulSoup(result,"html.parser")
result=soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
for i in result:
    print(i.find_all('p'))

```

###

15. (점심시간 문제) 위의 코드를 가지고 bs\_detail\_scroll() 함수를 생성하시오!

링크가 'http://futurechosun.com/archives/52491' 인 세부기사 1개의 내용만 출력한 것

###

```

from bs4 import BeautifulSoup
import urllib.request
def bs_detail_scroll():
    list_url = 'http://futurechosun.com/archives/52491'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup=BeautifulSoup(result,"html.parser")
    result=soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
    for i in result:
        return i.find_all('p')

print(bs_detail_scroll())
###

```

16. 지금 생성한 be\_detail\_scroll() 함수의 코드안에 이전에 만든 bs\_scroll() 함수를 호출하는 코드를 추가해서

bs\_scroll() 함수가 리턴하는 36개의 상세기사 url 에 대한 기사 본문이 전부 출력되게 하시오.

###

```

from bs4 import BeautifulSoup
import urllib.request
import time

def bs_scroll():
    params1 = []

    for i in range(1,4):
        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")

        result1 = soup.find_all('div',class_="elementor-post__title")

        for i in result1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)

def bs_detail_scroll():
    list_url = bs_scroll()
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup=BeautifulSoup(result,"html.parser")
    result=soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
    for i in result:
        time.sleep(1)
        return i.find_all('p')

print(bs_detail_scroll())

```

```

#%%

from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,4):

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)

def bs_detail_scroll():
    list_url = bs_scroll()
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result=soup.find_all('div', class_ = 'elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
        for i in result:
            print(i.find_all('p'))

print( bs_detail_scroll())

```

#%%

17. bs\_detail\_scroll() 함수의 params2 리스트를 추가해서 params2 리스트에 16번에서 출력하고 있는 24개의 기사가 append되게 하시오.

#%%

```

from bs4 import BeautifulSoup
import urllib.request

def bs_scroll():
    params1 = []

    for i in range(1,3):

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

        res1 = stuff.find_all('div',class_="elementor-post__title")

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

    return(params1)

def bs_detail_scroll():
    list_url = bs_scroll()
    params2=[]

```

```

for i in list_url:
    url = urllib.request.Request(i)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup=BeautifulSoup(result,"html.parser")
    result=soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
    for i in result:
        params2.append(i.find_all('p'))
    return(params2)

```

```
print(bs_detail_scroll())
```

```
###
```

18. 위에서 수집한 기사가 들어있느 params2 리스트의 내용을 c 드라이브 밑에 data 폴더 밑에 bongsa.txt로 생성되게 하시오.

```
###
```

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def bs_scroll():
    params1 = []

```

```
    for i in range(1,2):
```

```

        list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        stuff = BeautifulSoup( result, "html.parser")

```

```
        res1 = stuff.find_all('div',class_="elementor-post__title")
```

```

        for i in res1:
            params1.append(i.find_all('a')[0].get('href'))

```

```
    return(params1)
```

```
def bs_detail_scroll():
```

```

    list_url = bs_scroll()
    f = open('c:\\data\\bongsa.txt', 'w', encoding="UTF-8")

```

```

    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result=soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')
        for i in result:
            for j in i.find_all('p'):
                f.write(str(j.get_text()) + "\n")

```

```
f.close()
```

```
print( bs_detail_scroll()) #none 가 나오지만 bongsa.txt 에는 제대로 저장이 되어 있다.
```

```
# 그냥 bs_detail_scroll() 로 해도 된다!!! 파이썬에 출력될게 없으니까!
```

```
###
```

tip. 웹스크롤링은 알바도 많이 있다. 크몽 등

다른 학생이 실패한 사이트도 실습해보기!

이데일리 사이트 웹 스크롤링 하기

1. 이데일리에서 "첫눈" 으로 기사들을 검색하시오.

page 1 url : <https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1>

page 2 url : <https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=2>

page 3 url : <https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=3>

2. 위 page 1 url 의 html 코드를 BeautifulSoup 으로 파싱하시오.

```
###
```

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")
print(soup)
```

```
###
```

3. 상세기사 url을 찾기위한 태그 이름과 클래스 이름을 알아내시오.

F12 눌러서 ctrl+shift + c 누르고 기사제목 부분 클릭

```
<div class="newsbox_04">
```

4. 위의 div 태그와 newsbox\_04 클래스를 가지고 href 상세기사 url 을 출력하시오.

```
###
```

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")
```

```
result1 = soup.find_all('div', class_='newsbox_04')
for i in result1:
    print(i.find_all('a'))
```

를 하면

```
[<a href="/news/read?newsId=04008166625996880&mediaCodeNo=257" title="10일 새벽 서울에 첫 눈 내리나">
<span class="newsbox_visual">

</span>
<ul class="newsbox_texts">
<li>10일 새벽 서울에 첫 눈 내리나</li>
<li>[이데일리 송주오 기자] 10일 새벽에 서울에 비 또는 눈이 내릴 전망이다. 지난 2일 올 겨울 첫눈이 내린 강원 평창군 대관령에서 주민이 눈길을 조심스럽게 걸
어가고 있다.(사진=연합뉴스)기상청은 9일 오후 9시부터 경기 북부와 강원 영서 중·북부에, 오는 10일 오전 3시부터는 서울과 경기 남부, 충남 북부에 비 또는 눈이
올 것으로 예상했다. 서울에 눈이 내릴 경우 이번 겨울 첫 눈으로 기록된다. 이날까지 첫눈이 기록된 곳은 수원, 대전, 인천, 청주 등이다. 기상청은 서울과 인근 지
역에서 기온에 따라 눈이 내리거나 눈과 비가 섞인 진눈깨비 형태의 눈이 내릴 것으로 내다봤다. 예상 강수량은 5mm 미만, 적설량은 1~3cm다.10일 낮부터 밤 사이
에는 중부지방과 전북 서해안에 산발적으로 빗방울이 떨어지거나 눈이 날리는 곳이 있겠다. 지형적인 영향을 받는 곳은 1mm 내외의 강수량이 기록될 예정이다.10
일 아침기온은 이날보다 5~6도 높아 평년보다 따뜻할 것으로 보인다. 또 주말까지 일부 중부내륙지역과 남부 산지를 제외한 대부분 지역의 아침 기온은 영상을
기록할 전망이다. 0일 아침 최저기온은 영하 4~5도, 낮 최고기온은 6~14도로 예상된다.</li>
</ul>
</a>, <a href="/jroom/main?jid=juoh413">
<span class="author">송주오 기자</span>
</a>]
```

이런 포맷으로 기사마다 하나씩 되어서 리스트로 담겨있다.

그래서 i.find\_all('a')[0] 을 쓰면 get\_text() 를 쓸수있게 된다.


리스트는 get\_list 를 쓸 수없기 때문.

그리고 a href 로 출력된 부분도 완벽한 세부기사의 url이 아니기 때문에

"<http://edaily.co.kr/>" + a href ~ 부분이 되어야 한다.

(실제 세부기사의 url 은 <https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258> 인데

a href 뒤에 있는 부분은 <a href="/news/read?newsId=01423526625998520&mediaCodeNo=258"이다. )

이것을 반영해 수정된 버전 (by 윤모) 

#%%

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")
```

```
result1 = soup.find_all('div', class_='newsbox_04')
for i in result1:
    print("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))
```

#%%

결과물!

```
http://edaily.co.kr//news/read?newsId=01423526625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01439926625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01230006625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01157846625998192&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01758086625997208&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01174246625998848&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01207046625998848&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=02843766625998848&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=04057366625997208&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01243126625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01302166625999176&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01187366625998520&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01079126625999176&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=02499366625998520&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01289046625997536&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01453046625997536&mediaCodeNo=257
http://edaily.co.kr//news/read?newsId=01216886625997864&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01098806625997864&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=01387446625997536&mediaCodeNo=258
http://edaily.co.kr//news/read?newsId=04008166625996880&mediaCodeNo=257
```

#%%

5. 위의 상세기사 url 을 params 라는 비어있는 리스트에 담으시오.

#%%

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")
```

```
result1 = soup.find_all('div', class_='newsbox_04')
```

```
params = []
for i in result1:
    params.append("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))
```

```
print(params)
```

#%%

```
['http://edaily.co.kr//news/read?newsId=01423526625998520&mediaCodeNo=258',
```

```
'http://edaily.co.kr//news/read?newsId=01439926625998520&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01230006625998520&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01157846625998192&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01758086625997208&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=01174246625998848&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01207046625998848&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=02843766625998848&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=04057366625997208&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=03076646625999504&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=01243126625998520&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01302166625999176&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=01187366625998520&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01079126625999176&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=02499366625998520&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=01289046625997536&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01453046625997536&mediaCodeNo=257',
'http://edaily.co.kr//news/read?newsId=01216886625997864&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01098806625997864&mediaCodeNo=258',
'http://edaily.co.kr//news/read?newsId=01387446625997536&mediaCodeNo=258']
```

로 [ ] 안에 4번에서 출력한 url들이 append 되었다.

6. 위의 코드들을 가지고 eda\_scroll() 함수를 생성하시오.

```
#####

from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
    list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("UTF-8")
    soup = BeautifulSoup( result,"html.parser")

    result1 = soup.find_all('div', class_='newsbox_04')

    params=[]
    for i in result1:
        params.append("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))

    return(params)

print(eda_scroll() )
```

#####  
(상세기사 스크롤링하는 eda\_detail\_scroll() 만들기 밑작업!)

7. 위의 상세기사 url 을 가지고 기사 본문을 출력하시오.

```
#####

from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")

print(soup)
```

# 세부기사의 html 코드가 출력된다.

```
#####

8. 7번의 세부기사에서 태그와 클래스 이름을 찾고, 해당 기사의 텍스트를 출력하시오.
```

```
<div class="news_body" itemprop="articleBody">
```

```

####

from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")

result1 = soup.find_all('div', class_ = "news_body")

for i in result1:
    print(i)

####

<div class="news_body" itemprop="articleBody">
<table align="CENTER" bgcolor="ffffff" border="0" cellpadding="0" cellspacing="5" style="margin:5px 5px 5px 5px" width="670"><tr><td bgcolor="ffffff" style="PADDING-
RIGHT: 2px; PADDING-LEFT: 2px; PADDING-BOTTOM: 2px; PADDING-TOP: 2px;"><table bgcolor="ffffff" border="0" cellpadding="0" cellspacing="5"><tr><td></td></tr></table></td></tr></table><div class="ad_textrbanner"></div>[이데일리
스타in 김현식 기자] 그룹 인피니트 멤버 김성규가 솔로 컴백 소감을 밝혔다. <br><br>김성규는 14일 오후 4시에 연 세 번째 미니앨범 ‘인사이드 미’(INSIDE ME) 발
매 기념 온라인 미디어 쇼케이스에서 “오랜만에 앨범으로 인사드리게 돼 떨리고 설렌다”고 말했다.<br><br><div class="view_ad02" id="view_ad02"><div data-
widget_id="xlzQ0A7Z" id="dablewidget_xlzQ0A7Z">
<script>
(function(d,a,b,l,e,_) {
if(d[b]&&d[b].q)return;d[b]=function(){{d[b].q=d[b].q||[]}.push(arguments)}};e=a.createElement();
e.async=1;e.charset='utf-8';e.src="//static.dable.io/dist/plugin.min.js";
_ =a.getElementsByTagName(l)[0];_ .parentNode.insertBefore(e,_);
})(window,document,'dable','script');
dable('setService', 'edaily.co.kr');
dable('renderWidget', 'dablewidget_xlzQ0A7Z', {ignore_items: true});
</script>
</div></div>이어 “어제 첫눈을 보고 이젠 하늘의 계시이자 운명이구나 싶었다. 신곡 뮤직비디오에 눈을 맞는 장면이 많기 때문”이라면서 “덕분에 기분 좋게 앨범
활동을 시작할 수 있겠다는 생각이 든다”며 미소 지었다.<br><br><div class="view_ad01" id="view_ad01"><iframe allowfullscreen="true" allowtransparency="true"
frameborder="0" height="250" marginheight="0" marginwidth="0" scrolling="no" src="//tm.interworksmedia.co.kr/ads.sx/4F97F58D" width="300"></iframe></div>앨범명에
대해선 “매번 앨범을 낼 때마다 드리는 말씀이지만, 좀 더 성숙된 모습을 보여드리고 싶었다”며 “지금의 제 모습과 제 안에 있는 것들을 보여드리자는 의미에서
앨범명을 ‘인사이드 미’로 정했다”고 설명했다.<br><br>김성규는 이날 오후 6시 새 앨범 전곡 음원을 공개한다. 앨범에는 타이틀곡 ‘아임 콜드’(I’m Cold)를 포함해
총 6곡이 담겼다.<br><br>
<script>document.write('<scr' + 'ipt type="text/javascript" src="https://movideo.ai/sc/aiScript?from=' + escape(document.querySelector('meta[property="og:url"]').content) +
'&type=script&a=78&movideo_id=' + document.querySelector('meta[property="movideo:news_key"]').getAttribute('content') + "'></scr' + 'ipt>');</script>
</div>

로 나오니까 텍스트들만 추출하기 위해 get.text() 를 활용
####

from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup( result,"html.parser")

result1 = soup.find_all('div', class_ = "news_body")

for i in result1:
    print(i.get_text() )

####

[이데일리 스타in 김현식 기자] 그룹 인피니트 멤버 김성규가 솔로 컴백 소감을 밝혔다.
김성규는 14일 오후 4시에 연 세 번째 미니앨범 ‘인사이드 미’(INSIDE ME) 발매 기념 온라인 미디어 쇼케이스에서 “오랜만에 앨범으로 인사드리게 돼 떨리고 설렌
다”고 말했다.

```

이어 “어제 첫눈을 보고 이건 하늘의 계시이자 운명이구나 싶었다.

신곡 뮤직비디오에 눈을 맞는 장면이 많기 때문”이라면서 “덕분에 기본 좋게 앨범 활동을 시작할 수 있겠다는 생각이 든다”며 미소 지었다.

앨범명에 대해선 “매번 앨범을 낼 때마다 드리는 말씀이지만, 좀 더 성숙된 모습을 보여드리고 싶었다”며

“지금의 제 모습과 제 안에 있는 것들을 보여드리자는 의미에서 앨범명을 ‘인사이드 미’로 정했다”고 설명했다.김성규는 이날 오후 6시 새 앨범 전곡 음원을 공개한다.

앨범에는 타이틀곡 ‘아임 콜드’(I’m Cold)를 포함해 총 6곡이 담겼다.

이런식으로 텍스트만 뽑혀 나옴.

\* 간혹 크롬 옵션에서 자바스크립트 허용 등으로 해주면 위와 같은 코드를 써도 신문기사 페이지의 광고 등에 활용되는 자바 코드들도 같이 출력되기도 한다.

#%%

9. 위의 결과물을 가지고 기사의 세부내용을 출력하는 eda\_detail\_scroll 함수를 생성하시오.

#%%

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def eda_detail_scroll():
    list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'

    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("UTF-8")
    soup = BeautifulSoup( result,"html.parser")

    result1 = soup.find_all('div', class_ = "news_body")

    for i in result1:
        return i.get_text()
```

#%%

10. 처음에 만들었던 함수인 eda\_scroll() 를 수정하는데, 페이지 1개가 아니라 페이지 3개의 상세기사 url 이 params 리스트에 담겨지게 수정하시오.

#%%

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def eda_scroll():
    params=[]
    for i in range(1,4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page='+str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("UTF-8")
        soup = BeautifulSoup( result,"html.parser")

        result1 = soup.find_all('div', class_ ='newsbox_04')

        for i in result1:
            params.append("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))

    return(params)
```

print( len( eda\_scroll()) # 60개가 나온다.

#%%

11. eda\_detail\_scroll() 함수안에 eda\_scroll() 함수를 호출해 상세기사 url 60개를 가져와서 기사를 출력할 수 있도록 하시오.

#%%



```

from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
    params=[]
    for i in range(1, 4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=' + str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")
        result1=soup.find_all('div', class_='newsbox_04')

        for i in result1:
            params.append("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))
    return params

```

```

def eda_detail_scroll():
    list_url= eda_scroll()
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result1 = soup.find_all('div', class_='news_body')
        for i in result1:
            print( i.get_text() ) # return은 1번만 결과값을 내기 때문에 print를 해야 전체 가 다 나온다.

```

```

eda_detail_scroll()

```

```

#%%%

```

12. 위에 출력되고 있는 기사 본문이 c 드라이브 밑에 data 밑에 eda.txt 로 저장되게 하시오.

```

#%%%

```

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def eda_scroll():
    params=[]
    for i in range(1, 4):
        list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=' + str(i)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result, "html.parser")
        result1=soup.find_all('div', class_='newsbox_04')

        for i in result1:
            params.append("http://edaily.co.kr/"+i.find_all('a')[0].get("href"))
    return params

```

```

def eda_detail_scroll():
    f = open("c:\\data\\eda.txt","w",encoding="UTF-8")
    list_url= eda_scroll()
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup=BeautifulSoup(result,"html.parser")
        result1 = soup.find_all('div', class_='news_body')
        for i in result1:
            f.write( str( i.get_text() ) + '\n' )
    f.close() # 저장만 할것이기 때문에 return은 필요없다.

```

```

eda_detail_scroll()

```

#####

(오늘의 마지막 문제)

문제 422. 네이버 영화페이지의 평가리뷰를 스크롤링 하시오. (레이디 버그때처럼!)

영화 선택은 자율. 영화의 영어 제목.txt 로하여 파일과 코드를 업로드

1. 네이버 영화 '이터널 선샤인' 의 평점 사이트를 접속해 url을 따온다.

page 1 url :<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1>

page 2 url :<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=2>

page 3 url :<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=3>

맨뒤의 &page=3 부분의 숫자만 페이지별로 바뀐다.

2. 게시판의 url을 가지고 html 코드들을 출력한다.

#####

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("CP949")
```

```
print(result) # html 코드들이 출력되었다.
```

#####

3. 위에서 긁어온 html 문서를 BeautifulSoup 함수로 파이썬이 인식할 수 있도록 파싱

#####

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("CP949")
soup = BeautifulSoup(result, "html.parser")
```

```
print(soup) # 2와 같은 결과가 출력. but 파이썬이 인식될 수있게 파싱된 버전
```

#####

4. 위의 url 의 게시판 글 내용에 해당하는 부분의 태그와 클래스 파악하고, 그 부분을 스크롤링하기

```
<td class="title">
```

#####

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("CP949")
soup = BeautifulSoup(result, "html.parser")
```

```
result2 = soup.find_all('td', class_="title")
```

```
print(result2)
```

#####

```
[<td class="title">
<a class="movie color_b" href="/movie/bi/mi/basic.nhn?code=38444">이터널 선샤인</a>
<div class="list_netizen_score">
```

```
<span class="st_off"><span class="st_on" style="width:100%">별점 - 총 10점 중</span></span><em>10</em>
</div>
```

<br/>내게도 그런 사람이 있었다. 아니, 영화를 보는 내내 떠오른 걸 보니 여전히 있는 건가 보다. 공교롭게도 그가 가장 좋아하는 영화 역시 이것이었고 우리가 함께 이 영화를 볼 때면 그는 늘 극중 케이트 윈슬렛이 나 같다고 하곤 했다. 한참 전 일이지만 실제 당시 만남의 전개와 상당히 흡사하기도 했다. 그와 나는 끝내 okay를 말할 수 없었지만 나는 마치 현실에서 클렘을 잊어야 했기에 기억 속에 숨겨두려 이리저리 도망 다니던 조엘처럼 이 영화에 그의 기억을 몽땅 숨겨둔 듯 하다. 한 번씩 이렇게 꺼내보며 그를 쫓으면 그걸로 되겠다.

```

<a class="report" href="javascript:report('elwl****', 'Lxo5brZQyBsqty0nKmy+ISO3ueVihTApN2/tTVttTno=', '내게도 그런 사람이 있었다. 아니, 영화를 보
는 내내 떠오른 걸 보니 여전히 있는 건가 보다. 공교롭게도 그가 가장 좋아하는 영화 역시 이것이었고 우리가 함께 이 영화를 볼 때면 그는 늘
극중 케이트 윈슬렛이 나 같다고 하곤 했다. 한참 전 일이지만 실제 당시 만남의 전개와 상당히 흡사하기도 했다. 그와 나는 끝내 okay를 말할 수
없었지만 나는 마치 현실에서 클렘을 잊어야 했기에 기억 속에 숨겨두려 이리저리 도망 다니던 조엘처럼 이 영화에 그의 기억을 몽땅 숨겨둔 듯
하다. 한 번씩 이렇게 꺼내보며 그를 쫓으면 그걸로 되겠다.');" style="color:#8F8F8F" title="새 창">신고</a>
</td>, <td class="title">
<a class="movie_color_b" href="/movie/bi/mi/basic.nhn?code=38444">이터널 선사인</a>
<div class="list_netizen_score">
<span class="st_off"><span class="st_on" style="width:100%">별점 - 총 10점 중</span></span><em>10</em>
</div>
<br/>끝을 잇는것은 시작이다
```

```

<a class="report" href="javascript:report('kapd****', 'ee0L2Il0njZ4B9bk7PPcLBQfDRai+HtEz1QRfY7pcol=', '끝을 잇는것은 시작이다', '17294016',
'point_after');" style="color:#8F8F8F" title="새 창">신고</a>
</td>, <td class="title">
<a class="movie_color_b" href="/movie/bi/mi/basic.nhn?code=38444">이터널 선사인</a>
<div class="list_netizen_score">
<span class="st_off"><span class="st_on" style="width:100%">별점 - 총 10점 중</span></span><em>10</em>
</div>
<br/> ~~~~ ]
```

이런식으로 [ ] 에 담겨져 있음.

###

5. 위의 결과에서 html 문서가 아닌 텍스트만 출력하기

###

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("CP949")
soup = BeautifulSoup(result, "html.parser")
```

```
result2 = soup.find_all('td', class_="title")
```

```
for i in result2:
    print(i.get_text())
```

###

이터널 선사인

별점 - 총 10점 중10

내게도 그런 사람이 있었다. 아니, 영화를 보는 내내 떠오른 걸 보니 여전히 있는 건가 보다. 공교롭게도 그가 가장 좋아하는 영화 역시 이것이었고 우리가 함께 이 영화를 볼 때면 그는 늘 극중 케이트 윈슬렛이 나 같다고 하곤 했다. 한참 전 일이지만 실제 당시 만남의 전개와 상당히 흡사하기도 했다. 그와 나는 끝내 okay를 말할 수 없었지만 나는 마치 현실에서 클램을 잊어야 했기에 기억 속에 숨겨두려 이리저리 도망 다니던 조엘처럼 이 영화에 그의 기억을 몽땅 숨겨둔 듯하다. 한번씩 이렇게 꺼내보며 그를 쫓으면 그걸로 되겠다.

신고

이터널 선사인

별점 - 총 10점 중10

끝을 잇는것은 시작이다

신고

이터널 선사인

별점 - 총 10점 중10

너가 영화를 본다면 떠올리는 사람이 나일까?

신고

이런식으로 결과가 출력됨.

###

6. 위에서 출력된 텍스트들을 params 라는 빈 리스트에 담기

###

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("CP949")
soup = BeautifulSoup(result, "html.parser")
```

```
result2 = soup.find_all('td', class_="title")
params=[]
```

```
for i in result2:
    params.append(i.get_text(" ", strip=True) )
```

```
print(params)
```

###

[ '이터널 선사인 별점 - 총 10점 중 10 내게도 그런 사람이 있었다. 아니, 영화를 보는 내내 떠오른 걸 보니 여전히 있는 건가 보다. 공교롭게도 그가 가장 좋아하는 영화 역시 이것이었고 우리가 함께 이 영화를 볼 때면 그는 늘 극중 케이트 윈슬렛이 나 같다고 하곤 했다. 한참 전 일이지만 실제 당시 만남의 전개와 상당히 흡사하기도 했다. 그와 나는 끝내 okay를 말할 수 없었지만 나는 마치 현실에서 클램을 잊어야 했기에 기억 속에 숨겨두려 이리저리 도망 다니던 조엘처럼 이 영화에 그의 기억을 몽땅 숨겨둔 듯하다. 한 번씩 이렇게 꺼내보며 그를 쫓으면 그걸로 되겠다. 신고, '이터널 선사인 별점 - 총 10점 중 10 끝을 잇는것은 시작이다 신고', '이터널 선사인 별점 - 총 10점 중 10 너가 영화를 본다면 떠올리는 사람이 나일까? 신고', '이터널 선사인 별점 - 총 10점 중 10 일단 짐캐리의 연기를 보고 박수를 짹짹! 짐캐리가 이런 절절한 로맨스에 어울린다고 생각을 못해봤다. 진짜 연기변신. 연인과 헤어진후 본거여서 더 마음에 와 닿았다. 나도 기억을 지우고싶다라고 생각했는데 나를 더 잃어가는 현상이라고 생각해 나에게 교훈을 준 영화였다. 또 영화를 보고 눈 쌓인 겨울 바다 보러 가야겠다라고 생각했다ㅎㅎ 놀러갈 핑계가 생겼다. 야호 신고, '이터널 선사인 별점 - 총 10점 중 9 침부터 긴가민가했는데 정말 짐캐리였다고맙다~ 신고', '이터널 선사인 별점 - 총 10점 중 10 너무.. 가슴이 아팠다. 보는 내내 신고', '이터널 선사인 별점 - 총 10점 중 8 여러 사랑을 거쳐 진정한 사랑을 알게 되는 것처럼 이 영화도 한번은 부족하다. 더더더 보고 보면 볼수록 느끼는게 달라지는 영화. 묘하고 독특하면서도 충분히 친근하다. 신고, '이터널 선사인 별점 - 총 10점 중 10 신고', '이터널 선사인 별점 - 총 10점 중 10 인간의 보이지 않는 의식속에 모든 감정의 기록이 존재한다면 얼마든지 공명하는 성질끼리 어떤 형식으로든 만나게 될수도 있음을.. 존재를 사랑할 때 과연 표면적인 것이 좋아 사랑할까? 그 존재가 존재할 수 있는 보이지 않는 사랑의 힘이 너무 자비로워서가 아닐까 싶다. 신고, '이터널 선사인 별점 - 총 10점 중 9 좋았던 기억과 평생을 함께 하자는 그 다짐이 기억나는 영화. 신고']

기타 코드들이 보이지 않고 글자만 선별되어 출력이 되었다.

####

7. 위의 코드를 적용해서 여러개의 게시판에서 리뷰를 가져오기 (글자수가 적어 20페이지 정도 해보기로.)

page 1 url : <https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=1>  
 page 2 url : <https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=2>  
 page 3 url : <https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page=3>

맨뒤의 &page=3 부분의 숫자만 페이지별로 바뀐다.

####

```
from bs4 import BeautifulSoup
import urllib.request

for i in range(1,21):
    list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=38444&target=&page='+str(i)

    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("CP949")
    soup = BeautifulSoup(result, "html.parser")

    result2 = soup.find_all('td', class_ = "title")
    params=[]

    for i in result2:
        params.append(i.get_text(" ", strip=True) )

    print(params)
```

####

8. 위의 글들을 eternal\_sunshine.txt 에 저장하기.

####

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def review_scroll(num):
```

```
f = open('c:\\data\\eternal_sunshine.txt', 'w', encoding="UTF-8")
for i in range(1,num+1):
    list_url = '

```

```
review_scroll(100)
```

```
###
```

결이가 한것 처럼 10점중의 10점 이런것잘라내고 출력하기

```
###
```

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def review_scroll(num):
    f = open('c:\\data\\eternal_sunshine.txt', 'w', encoding="UTF-8")
    for i in range(1,num+1):
        list_url = '

```

```
review_scroll(100)
```

```
###
```

20.12.17 파이썬 강의 Day 16(문제423~문제439)

144 파이썬에서 워드 클라우드 그리기

웹스크롤링한 데이터를 분석해야 하는 방법

1. 우리회사의 신제품이 출시되었을때 소비자들의 반응을 살펴보고자 할때 (기업)
2. 시기별 사회현상을 파악하고자 할때(국가)
3. 인공지능 상담원(딥러닝의 RNN) 을 만들기 위한 자연어 처리 학습 데이터로 웹스크롤링한 데이터를 활용
4. 인공지능의 눈이라고 할 수 있는 딥러닝의 CNN 의 신경망의 학습자료로 이미지 데이터 활용

\* 빅데이터 기사 문제 : 데이터의 종류 3가지

1. 정형데이터 : 정형화된 스키마구조. DBMS에 저장될 수 있는 구조  
e.g) 오라클의 emp 테이블, MySQL, MSSQL 등의 테이블 등
2. 반정형 데이터 : 데이터 내부의 데이터 구조에 대한 메타정보가 포함된 구조의 데이터  
e.g) html 문서, xml문서  
html 에 반(정형)했다. 외우기 ㅋㅋㅋ
3. 비정형 데이터 : 웹스크롤링 기술로 수집해서 모은 데이터  
e.g) text문서, 이미지, 동영상  
가수 비(정형)의 동영상 키워드로 외우기 ㅋㅋㅋ

문제 423. 어제 조선일보에서 '봉사'로 검색했을때 다운 받았던 기사에서 가장 많이 나오는 단어(어절)가 무엇인가?  
bongsa.txt 읽어와서 활용!

###

1. 어제 저장한 bongsa.txt를 불러오고 어절단위로 출력 & 정렬한다.

###

stev = open("c:\\data\\bongsa.txt", encoding='UTF8', errors='ignore') # 샘은 그냥 되는데 나는 UTF8 인코딩 에러 나서 ignore 사용

stev2=stev.read().split() # split() 괄호 안에 아무것도 안넣으면 '어절' 단위로 읽어온다.

stev2.sort() # 리스트안의 요소들을 정렬

print(stev2)

###

2. collections 모듈의 Counter 메소드를 사용해서 stev2에서 사용된 단어들의 빈도를 카운트한다.

###

from collections import Counter

count\_list = Counter(stev2)

print(count\_list)

###

```
Counter({'수': 33, '있다.': 28, '등': 22, '했다.': 22, '더 나은 미래': 18,
'비자': 17, '봉사': 16, '-': 15, '마스크': 15, '있는': 15, '봉사활동': 14, '&': 13,
'Copyrights': 13, 'futurechosun.com.': 13, '©': 13, '이주': 13, '금지': 12, '대표는': 12,
'하는': 12, '대한': 11, '된다.': 11, '지난': 11, '함께': 11, '활동': 11, '활동을': 11,
'위해': 10, '해외': 10, '더': 9, '말했다.': 9, ~~~~ ]
```

이런식으로 어절별로 각각 몇건 씩 있는지 정리 작업을 해서 [] 안에 담아서 출력해준다.

###

3. 출력된 count\_list에서 가장 많은 빈도로 나온 어절 30개를 추출한다.

most\_common(출력할 숫자) 활용

###

from collections import Counter

count\_list = Counter(stev2)

result = count\_list.most\_common(30) # top 10 만 추출

print(result)

###

결과

```
[('수', 33), ('있다.', 28), ('등', 22), ('했다.', 22),
('더 나은 미래', 18), ('비자', 17), ('봉사', 16), ('-', 15),
('마스크', 15), ('있는', 15), ('봉사활동', 14), ('&', 13),
```

```
('Copyrights', 13), ('futurechosun.com,', 13), ('©', 13),  
('이주', 13), ('금지', 12), ('대표는', 12), ('하는', 12),  
('대한', 11), ('된다.', 11), ('지난', 11), ('함께', 11), ('활동', 11),  
('활동을', 11), ('위해', 10), ('해외', 10), ('더', 9), ('말했다.', 9),  
('밝혔다.', 9)]
```

문제 424. 위의 결과를 for loop 문을 이용해서 아래와 같이 출력하시오.

괄호없이!

```
수 33  
있다 28  
등 22  
.  
.  
.
```

```
###
```

```
from collections import Counter  
stev = open("c:\\data\\bongsa.txt",encoding='UTF8', errors = 'ignore') # 샘은 그냥 되는데 나는 UTF8 인코딩 에러나서 ignore 사용
```

```
stev2=stev.read().split() # split() 괄호 안에 아무것도 안넣으면 '어절' 단위로 읽어온다.
```

```
stev2.sort() # 리스트안의 요소들을 정렬
```

```
count_list = Counter(stev2)
```

```
result = count_list.most_common(30) # top 10 만 추출
```

```
for i in result:  
    print(i[0],i[1])
```

```
#그냥 for i in result:  
#     print(i)  
# 는 괄호가 있는 상태로 나온다.
```

```
###
```

문제 425. 어제 마지막으로 스크롤링 했던 영화 평가 게시글들에서 위와 같이 가장 많이 나오는 어절이 무엇인지 출력하시오.

```
###
```

```
from collections import Counter
```

```
sunshine = open("c:\\data\\eternal_sunshine.txt",encoding='UTF8')
```

```
sunshine2 = sunshine.read().split()
```

```
sunshine2.sort()
```

```
count_list = Counter(sunshine2)
```

```
result = count_list.most_common(30)
```

```
for i in result:  
    print(i[0],i[1])
```

```
###
```

다희님의 결과처럼(레이디버그때 날자따로 출력, 리뷰따로 출력 -> 날짜+리뷰 로 나오게했던 것처럼) 날짜와 리뷰가 나오도록 리스트를 만들었어야 ...  
-> 점심먹고 수정해보기!



이터널 1008  
중 1007  
10점 1002  
선샤인 1002  
- 1001  
별점 1001  
신고 1000  
총 1000  
10 798  
다시 162  
영화 143  
이 140  
수 111  
그 85  
너무 85  
영화를 85  
9 69  
기억을 57  
더 52  
사랑을 48  
보고 47  
정말 45  
내 41  
내가 38  
영화. 38  
그냥 37  
또 37  
1 35  
나는 35  
8 34

###

문제 426. (감정분석) 웹스크롤링한 데이터로 감정분석(긍정적, 부정적) 을 하기 위해 신성님의 라라랜드 영화 평가를 다운로드해서 c 밑에 data폴더에 두고 가장 많이 나오는 단어가 무엇인지 확인하시오.

sample6\_laland\_review.txt

###

from collections import Counter

lala = open("c:\\data\\sample6\_laland\_review.txt",encoding='UTF8')

lala2 = lala.read().split()

lala2.sort()

count\_list = Counter(lala2)

result = count\_list.most\_common(30)

for i in result:  
 print(i[0],i[1])

###

10점 100  
중 100  
10 79  
영화 22

이 12  
다시 10  
수 9  
본 7  
9 6  
그 6  
그때 6  
내 6  
너무 6  
보고 6  
봐도 6  
8 5  
내가 5  
더 5  
번 5  
않았다면. 만약에 5  
있는 5  
잘 5  
한 5  
것 4  
다 4  
마지막 4  
만약에, 4  
아주 4  
영화가 4

####

문제 427. 영화 라라랜드에서 긍정적인 단어가 몇건 나오는지 출력하시오.(문제 276의 스티브 잡스 긍정단어 사용 코드 가져오기)  
####

```
lala = open("c:\\data\\sample6_laland_review.txt", encoding='UTF8')
lala2 = lala.read().split() # 스티브 잡스 연설문을 어절별로 분리해 lala2에 입력하라. lala2는 리스트이고, 잡스의 연설문이 어절별로
positive = open("c:\\data\\positive-words2.txt") #저장되어 있음.
pos = positive.read().split('\n') #positive 에서 '행' 별로 분리해서 pos 에 담아라.
    # pos는 리스트이고, 긍정단어들이 들어있는 곳

cnt=0

for i in lala2: #stev2 에서 단어를 하나씩 빼온다.
    if i.lower() in pos: #만약 stev2 에서 빼온 단어중에 pos 와 일치하는 단어가 있다면 (lala2는 한글이라lower필요없지만 그냥둘러줌)
        cnt+=1 #cnt 를 1개 증가시키시오.
        print(i,cnt)
print('긍정단어의 수는' , cnt , '개 입니다.')
```

# 37개 출력

####

문제 428. 라라랜드 영화평가의 부정단어는 몇개가 있고 그 단어들은 무엇인지 출력하시오.

####

```
lala = open("c:\\data\\sample6_laland_review.txt", encoding='UTF8')
lala2 = lala.read().split() # 스티브 잡스 연설문을 어절별로 분리해 lala2에 입력하라. lala2는 리스트이고, 잡스의 연설문이 어절별로
negative = open("c:\\data\\negative-words2.txt") #저장되어 있음.
neg = negative.read().split('\n') #negative 에서 '행' 별로 분리해서 pos 에 담아라.
```

```

# pos는 리스트이고, 긍정단어들이 들어있는 곳

cnt=0

for i in lala2: #stev2 에서 단어를 하나씩 빼온다.
    if i.lower() in neg: #만약 stev2 에서 빼온 단어중에 pos 와 일치하는 단어가 있다면 (lala2는 한글이라lower필요없지만 그냥둘러줌)
        cnt+=1 #cnt 를 1개 증가시키시오.
        print(i,cnt)
print('부정단어의 수는' , cnt , '개 입니다.')

###

문제 429. 영화 라라랜드의 평가글들을 시각화하시오. (워드클라우드로)
###

# 1. 아나콘다 프롬프트 창을 열고 wordcloud 패키지 설치

conda install wordcloud

또는

pip install wordcloud

# 2. c 드라이브 밑에 project 폴더를 생성

# 3. project 폴더 밑에 4가지 파일을 둡니다.

- usa_im.png
- s_korea.png
- word.txt
- 중앙일보 스크롤링했던 기사 파일 my_text21.txt

###
# 텍스트마이닝 데이터 정제

from wordcloud import WordCloud, STOPWORDS #STOPWORDS 구두점 데이터를 정제하기 위해 불러옴(콤마,점 등 정제)
import matplotlib.pyplot as plt # 그래프 그리는 모듈
from os import path # os 에 있는 파일을 파이썬에서 인식하기 위해서 import // 윈도우 os 의미 (c:\project~) 를 불러오기 위해 사용
import re # 데이터 정제를 위해서 필요한 모듈 (매우유명)
import numpy as np # 행렬연산을 빠르게 하기위해서 사용함. wordcloud 그릴때 numpy array로 변경할때 사용
from PIL import Image # 이미지 시각화를 위한 모듈

# 1. 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:\\project\\usa_im.png")) #Image 의 i 는 반드시 대문자!

# 2. 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
#script = input( 'input file name : ' )
# or 코드에서 직접 파일명 입력
script = 'sample6_laland_review.txt'

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
d = path.dirname("c:\\project\\") # 혹은 ("c:\\project\\") for window // 근데 그냥해도 된다.

# 기사 스크립트(script)와 os 의 위치(d)를 연결하여 utf8로 인코딩해서 한글 텍스트를 text 변수로 리턴한다.
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()

# %s means -> 문자열에 대응됨 -> 즉 script 에서 불러오는 파일의 문자열에 대응한다.

#print(text) # sample6_laland_review.txt 파일에 있는 텍스트가 쪽 나옴.

```

```
# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
file = open('c:\\project\\word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') # 어절별로 분리해서 word 에 담아 리스트로 구성한다.
for i in word:
    text = re.sub(i, "", text)
# re.sub('있다', "", '있다') 를 예시로 위의 코드 풀이
# 만약 지정된 txt파일(지금은 script = 에 있는 파일) 에 '있다'
# 라는 문자가 있다면 그것을 " 으로 대체하겠다.

# 일반적으로 자주 나오는 단어들을 전부 " (" 아님주의) 로 아래와 같이 대체하려고 하는데
# 아래처럼 일일이 다 써주면 너무 코딩이 많아지니까 for 문을 이용해서 쉽게 하는 것
# re.sub('있다', "", '있다'), re.sub('하지만', "", '하지만') 등
# 지금 프로젝트 폴더 안에 word.txt 파일에 지우고 싶은 단어들을 추가하면 된다.
# 이르면 결이가 한것 처럼 특정 위치 인덱스를 찾아서 지우고 파일에 담지 않아도 된다.
```

```
# 워드 클라우드를 그린다.
wordcloud = WordCloud(font_path='C:\\Windows\\Fonts\\gulim', # 글씨체가 있는 디렉토리에서 gulim 이란 글씨체를 사용하겠다.
    stopwords=STOPWORDS, # 구두점(마침표, 느낌표, 싱글 쿼테이션 등)을 정제
    max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
    background_color='white', # 배경색깔
    max_font_size = 100, # 최대 글씨 크기
    min_font_size = 1, # 최소 글씨
    mask = usa_mask, # 배경 모양
    colormap='jet').generate(text).to_file('c:\\project\\lala_cloud.png')
# c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름
# jet 는 wordcloud 모양의 명칭중 하나

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear') # 글씨가 퍼지는 스타일
plt.axis("off") # plt의 축(axis)를 나오는 기능을 off해서 나오지 않도록 하겠다.
```

#####

(점심시간문제)

문제 430. 어제 마지막 문제로 받았던 영화 평가글 또는 어제 스크롤링했던 기사들중 하나를 선택해서 wordcloud 로 그리고 그림을 첨부하여 제출하시오.

#####

# 텍스트마이닝 데이터 정제

```
from wordcloud import WordCloud, STOPWORDS #STOPWORDS 구두점 데이터를 정제하기 위해 불러옴(콤마,점 등 정제)
import matplotlib.pyplot as plt # 그래프 그리는 모듈
from os import path # os 에 있는 파일을 파이썬에서 인식하기 위해서 import // 윈도우 os 의미 (c:\project~) 를 불러오기 위해 사용
import re # 데이터 정제를 위해서 필요한 모듈 (매우유명)
import numpy as np # 행렬연산을 빠르게 하기위해서 사용함. wordcloud 그릴때 numpy array로 변경할때 사용
from PIL import Image # 이미지 시각화를 위한 모듈

# 1. 워드 클라우드의 배경이 되는 이미지 모양을 결정
usa_mask = np.array(Image.open("c:\\project\\s_korea.png")) #Image 의 I 는 반드시 대문자!
```

# 2. 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.

```
#script = input('input file name : ')
```

# or 코드에서 직접 파일명 입력

```
script = 'eternal_sunshine.txt'
```

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정

```
d = path.dirname("c:\\project\\") # 혹은 ("c:\\project\\") for window // 근데 그냥해도 된다.
```

# 기사 스크립트(script)와 os 의 위치(d)를 연결하여 utf8로 인코딩해서 한글 텍스트를 text 변수로 리턴한다.

```
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()
```

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업

```
file = open('c:\\project\\word.txt', 'r', encoding = 'utf-8')
```

word = file.read().split(' ') # 어절별로 분리해서 word 에 담아 리스트로 구성한다.

```
for i in word:
```

```
    text = re.sub(i,",",text)
```

# 워드 클라우드를 그린다.

```
wordcloud = WordCloud(font_path='C:\\Windows\\Fonts\\gulim', # 글씨체가 있는 디렉토리에서 gulim 이란 글씨체를 사용하겠다.
```

```
    stopwords=STOPWORDS, # 구두점(마침표,느낌표,싱글 쿼테이션 등)을 정제
```

```
    max_words=2000, # 워드 클라우드에 그릴 최대 단어갯수
```

```
    background_color='white', # 배경색깔
```

```
    max_font_size = 150, # 최대 글씨 크기
```

```
    min_font_size = 5, # 최소 글씨
```

```
    mask = usa_mask, # 배경 모양
```

```
    colormap='jet').generate(text).to_file('c:\\project\\eternal_cloud.png')
```

```
# c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름
```

```
# jet 는 wordcloud 모양의 명칭중 하나
```

```
plt.figure(figsize=(15,15))
```

```
plt.imshow(wordcloud, interpolation='bilinear') # 글씨가 퍼지는 스타일
```

```
plt.axis("off") # plt의 축(axis)를 나오는 기능을 off해서 나오지 않도록 하겠다.
```

##%

145 네이버 영화평점을 긍정과 부정으로 분류해서 시각화 하기

데이터 분석가가 하는 일?

웹스크롤링 -----> 스크롤링한 데이터 시각화(워드클라우드)

감정 분석

비정형 데이터를 활용

기업 : 기업의 의사결정의 도움을 주기 위해

-> 감정 분석으로 신제품에 대한 선호도 파악 등

정부 : 현 사회 현상 파악

e.g) 팀장님이 시킨 일

지금 우리회사 게시판에 올라온 신제품(화장품)에 대한 반응을 긍정적인 반응과 부정적인 반응을 조사해서 보고 하라!

라고 시켰다면?

문제 431. 신성이가 받은 라라랜드 리뷰 게시글에서 평가 점수만 출력하시오.

(6점 이상은 긍정, 6점이하는 부정으로 기준 잡아 감정분석을 하려는 것)

sample6\_laland\_review.txt 에서 뽑아내기

파일 내용을 보면

10점 중 10 아까운 게 하나도 없는 솜사탕

10점 중 10 2020 최고의영화 라라랜드와 함께한 하루 최고였다

10점 중 9 감동 그자체 만족스럽습니다..

10점 중 2 지기스

라고 되어있는데 이중 '10점 중 10' 이 부분만 뽑고 싶은 것.

```
###
```

```
lala = open("c:\\data\\sample6_laland_review.txt",encoding='UTF8')
lala2 = lala.readlines()
```

```
# print(lala2) 리스트에 담겨서 출력되어 있다. for 문을 사용해 1줄씩 출력
```

```
#for i in lala2:
```

```
# print(i, end="") # end="" 하면 글씨사이 과도한 간격이 정렬되어 줄여져 출력된다.
```

```
for i in lala2:
```

```
# print( i[6:8] ) # 로하면 점수들만 출력됨.
```

```
# print(type(i) ) # str 형태 ->int 화 시켜야 함
```

```
print( int(i[6:8])) # 결과를 숫자형으로 출력하기 위함
```

```
###
```

문제 432. 위의 라라랜드 평가글 중에 6점이상인 글들만 출력하시오.

```
###'
```

```
lala = open("c:\\data\\sample6_laland_review.txt",encoding='UTF8')
lala2 = lala.readlines()
```

```
# print(lala2) 리스트에 담겨서 출력되어 있다. for 문을 사용해 1줄씩 출력
```

```
#for i in lala2:
```

```
# print(i, end="") # end="" 하면 글씨사이 과도한 간격이 정렬되어 줄여져 출력된다.
```

```
for i in lala2:
```

```
if int(i[6:8]) >= 6:
```

```
print(i, end="") #6점 이상 점수를 준 리뷰들만 출력된다.
```

```
###
```

문제 433. 라라랜드 평가글 중에 평점이 6점이상이면 pos 라는 비어있는 리스트에 해당 평가글을 append 되게 하고, 6점 미만이면 neg 라는 비어있는 리스트에 해당 평가글이 append 되게 하시오.'

```
###'
```

```
lala = open("c:\\data\\sample6_laland_review.txt",encoding='UTF8')
lala2 = lala.readlines()
```

```
# print(lala2) 리스트에 담겨서 출력되어 있다. for 문을 사용해 1줄씩 출력
```

```
#for i in lala2:
```

```
# print(i, end="") # end="" 하면 글씨사이 과도한 간격이 정렬되어 줄여져 출력된다.
```

```
pos=[]
```

```
neg=[]
```

```

for i in lala2:
    if int(i[6:8]) >= 6:
        pos.append(i[8:]) #6점 이상 점수를 준 리뷰들만 출력된다.
    else:
        neg.append(i[8:]) # [9:] 을 하는 이유는 10점 중 숫자(점수) 를 빼기위함.

# pos.append(i, end="") 를 하면 TypeError: append() takes no keyword arguments

print(pos) #확인용 \n (엔터) 가 같이 들어가 있음.
# 만약 이게 보기 싫으면 append할때 i[9:].strip("\n") 하면 된다.
# 하지만 엔터 빼면 나중에 txt로 저장해서 결과물 출력할때 이쁘게 출력되지 않음

#print(neg) #확인용

###
문제 434. 위의 pos 에 들어있는 긍정글들을 c 드라이브 밑에 project 폴더 하에 poslala.txt 로 저장하시오.
###

lala = open("c:\\data\\sample6_laland_review.txt",encoding='UTF8')
lala2 = lala.readlines()

# print(lala2) 리스트에 담겨서 출력되어 있다. for 문을 사용해 1줄씩 출력

#for i in lala2:
#    print(i, end="") # end="" 하면 글씨사이 과도한 간격이 정렬되어 줄여져 출력된다.

pos=[]
neg=[]
for i in lala2:
    if int(i[6:8]) >= 6:
        pos.append(i[8:]) #6점 이상 점수를 준 리뷰들만 출력된다.
    else:
        neg.append(i[8:])
f = open("c:\\project\\pos_lala.txt","w")
f.writelines(pos)
f.close()

###

문제 435. 6점 이하의 부정글들을 neg_lala.txt 로 하여 문제 434와 같은 경로에 저장되게 하시오.

###

lala = open("c:\\data\\sample6_laland_review.txt",encoding='UTF8')
lala2 = lala.readlines()

# print(lala2) 리스트에 담겨서 출력되어 있다. for 문을 사용해 1줄씩 출력

#for i in lala2:
#    print(i, end="") # end="" 하면 글씨사이 과도한 간격이 정렬되어 줄여져 출력된다.

pos=[]
neg=[]
for i in lala2:
    if int(i[6:8]) >= 6:
        pos.append(i[8:]) #6점 이상 점수를 준 리뷰들만 출력된다.
    else:
        neg.append(i[8:])
f2 = open("c:\\project\\neg_lala.txt","w")
f2.writelines(neg)
f2.close()

```

```
###
```

문제 436. 위의 pos\_lala.txt 의 글과 neg\_lala.txt 의 글을 워드 클라우드로 시각화 하시오.'

```
###
```

```
# 1 긍정
```

```
# 텍스트마이닝 데이터 정제
```

```
from wordcloud import WordCloud, STOPWORDS #STOPWORDS 구두점 데이터를 정제하기 위해 불러옴(콤마,점 등 정제)
```

```
import matplotlib.pyplot as plt # 그래프 그리는 모듈
```

```
from os import path # os 에 있는 파일을 파이썬에서 인식하기 위해서 import // 윈도우 os 의미 (c:\project~) 를 불러오기 위해 사용
import re # 데이터 정제를 위해서 필요한 모듈 (매우유명)
```

```
import numpy as np # 행렬연산을 빠르게 하기위해서 사용함. wordcloud 그릴때 numpy array로 변경할때 사용
```

```
from PIL import Image # 이미지 시각화를 위한 모듈
```

```
# 1. 워드 클라우드의 배경이 되는 이미지 모양을 결정
```

```
usa_mask = np.array(Image.open("c:\\project\\s_korea.png")) #Image 의 l 는 반드시 대문자!
```

```
# 2. 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.
```

```
#script = input( 'input file name : ' )
```

```
# or 코드에서 직접 파일명 입력
```

```
script = 'pos_lala.txt'
```

```
# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정
```

```
d = path.dirname("c:\\project\\") # 혹은 ("c:\\project\\") for window // 근데 그냥해도 된다.
```

```
# 기사 스크립트(script)와 os 의 위치(d)를 연결하여 utf8로 인코딩해서 한글 텍스트를 text 변수로 리턴한다.
```

```
text = open(path.join(d, "%s"%script), mode="r", encoding="CP949", errors='ignore').read()
```

```
#UTF8 도 예러나고 UTF8도 예러나서 encoding="CP949", errors='ignore' 를 써줬다.
```

```
# 혹은 맨처음 원본이 되는 txt파일 만들때 유형에 맞는 파일 형식으로 encoding 해주면 된다.
```

```
# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업
```

```
file = open('c:\\project\\word.txt', 'r', encoding = 'utf-8')
```

```
word = file.read().split(' ') # 어절별로 분리해서 word 에 담아 리스트로 구성한다.
```

```
for i in word:
```

```
    text = re.sub(i, "", text)
```

```
# 워드 클라우드를 그린다.
```

```
wordcloud = WordCloud(font_path='C:\\Windows\\Fonts\\gulim', # 글씨체가 있는 디렉토리에서 gulim 이란 글씨체를 사용하겠다.
```

```
    stopwords=STOPWORDS, # 구두점(마침표, 느낌표,싱글 쿼테이션 등)을 정제
```

```
    max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
```

```
    background_color='white', # 배경색깔
```

```
    max_font_size = 110, # 최대 글씨 크기
```

```
    min_font_size = 5, # 최소 글씨
```

```
    mask = usa_mask, # 배경 모양
```

```
    colormap='jet').generate(text).to_file('c:\\project\\pos_lala_cloud.png')
```

```
# c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름
```

```
# jet 는 wordcloud 모양의 명칭중 하나
```

```
plt.figure(figsize=(15,15))
```

```
plt.imshow(wordcloud, interpolation='bilinear') # 글씨가 퍼지는 스타일
```

```
plt.axis("off") # plt의 축(axis)를 나오는 기능을 off해서 나오지 않도록 하겠다.
```

```
# 2 부정
```

```
# 텍스트마이닝 데이터 정제
```



```

from wordcloud import WordCloud, STOPWORDS #STOPWORDS 구두점 데이터를 정제하기 위해 불어옴(콤마,점 등 정제)
import matplotlib.pyplot as plt # 그래프 그리는 모듈
from os import path # os 에 있는 파일을 파이썬에서 인식하기 위해서 import // 윈도우 os 의미 (c:\project~) 를 불러오기 위해 사용
import re # 데이터 정제를 위해서 필요한 모듈 (매우유명)
import numpy as np # 행렬연산을 빠르게 하기위해서 사용함. wordcloud 그릴때 numpy array로 변경할때 사용
from PIL import Image # 이미지 시각화를 위한 모듈

```

# 1. 워드 클라우드의 배경이 되는 이미지 모양을 결정

```

usa_mask = np.array(Image.open("c:\project\s_korea.png")) #Image 의 l 는 반드시 대문자!

```

# 2. 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.

```

#script = input('input file name : ')
# or 코드에서 직접 파일명 입력
script = 'neg_lala.txt'

```

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정

```

d = path.dirname("c:\project\") # 혹은 ("c:\project\") for window // 근데 그냥해도 된다.

```

# 기사 스크립트(script)와 os 의 위치(d)를 연결하여 utf8로 인코딩해서 한글 텍스트를 text 변수로 리턴한다.

```

text = open(path.join(d, "%s"%script), mode="r", encoding="CP949", errors='ignore').read()

```

#UTF8 도 에러나고 UTF8도 에러나서 encoding="CP949", errors='ignore' 를 써줬다.

# 혹은 맨처음 원본이 되는 txt파일 만들때 유형에 맞는 파일 형식으로 encoding 해주면 된다.

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업

```

file = open('c:\project\word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ') # 어절별로 분리해서 word 에 담아 리스트로 구성한다.
for i in word:
    text = re.sub(i,",",text)

```

# 워드 클라우드를 그린다.

```

wordcloud = WordCloud(font_path='C:\Windows\Fonts\gulim', # 글씨체가 있는 디렉토리에서 gulim 이란 글씨체를 사용하겠다.

```

```

    stopwords=STOPWORDS, # 구두점(마침표, 느낌표,싱글 쿼테이션 등)을 정제

```

```

    max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수

```

```

    background_color='white', # 배경색깔

```

```

    max_font_size = 110, # 최대 글씨 크기

```

```

    min_font_size = 5, # 최소 글씨

```

```

    mask = usa_mask, # 배경 모양

```

```

    colormap='jet').generate(text).to_file('c:\project\neg_lala_cloud.png')

```

```

# c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름

```

```

# jet 는 wordcloud 모양의 명칭중 하나

```

```

plt.figure(figsize=(15,15))

```

```

plt.imshow(wordcloud, interpolation='bilinear') # 글씨가 퍼지는 스타일

```

```

plt.axis("off") # plt의 축(axis)를 나오는 기능을 off해서 나오지 않도록 하겠다.

```

```

#%

```

\* 빅데이터 기사 문제 : 데이터의 종류 3가지

1. 정형데이터 : 정형화된 스키마구조. DBMS에 저장될 수 있는 구조

e.g) 오라클의 emp 테이블, MySQL, MSSQL 등의 테이블 등 + 판다스로도 가능

2. 반정형 데이터 : 데이터 내부의 데이터 구조에 대한 메타정보가 포함된 구조의 데이터

e.g) html 문서, xml문서

3. 비정형 데이터 : 웹스크롤링 기술로 수집해서 모은 데이터  
e.g) text문서, 이미지, 동영상

146. 웹에 있는 사진을 스크롤링하는 방법(구글이미지)

딥러닝 기술 ?

1. cnn : 인공지능의 눈

e.g) 관련된 학습 빅데이터 ? 이미지 파일!

2. rnnL: 인공지능의 입과 귀

e.g) 관련된 학습 빅데이터 ? 자연어 처리를 위한 문장들

###

인공지능의 눈!

셀레니움을 써서 마치 손으로 클릭해서 이미지를 저장하듯이 저장을 하는데 컴퓨터에게 시켜서 자동화가 되도록 할 것

☐ 구글에서 이미지 스크롤링하기 <5단계> (내컴퓨터는 E드라이브에 데이터 관련 수업 다 몰아뒀음) 꼭 C드라이브 아니어도 괜찮음

1. 크롬 웹브라우저가 설치 되어져 있어야 한다.
2. E 드라이브 밑에 chromdriver 폴더를 생성하고 거기에 chromdriver.exe를 넣어놔야 한다.
3. E 드라이브 밑에 gimages 폴더를 생성한다. (내려받기용)
4. 다운받을 이미지 키워드를 결정한다.
5. 아나콘다 프롬프트 창을 열고 selenium을 설치한다.  
-> conda install selenium  
-> conda list selenium (으로 잘 설치되었는지 확인)  
실패하면  
pip install selenium

키워드 : 떡볶이

6. 구글에서 '떡볶이' 로 검색했을때 웹스크롤링 하는 코드

###

import urllib.request # 웹 url을 파이썬이 인식 할 수 있게하는 패키지

from bs4 import BeautifulSoup # html에서 데이터 검색을 용이하게 하는 패키지

from selenium import webdriver

# selenium : 1) 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크

# 2) 손으로 클릭하는것을 컴퓨터가 대신하면서 스크롤링하게 하는 패키지

from selenium.webdriver.common.keys import Keys

import time # 중간중간 sleep 을 걸어야 해서 time 모듈 import

##### url 받아오기 #####

# 웹브라우저로 크롬을 사용할거라서 크롬 드라이버를 다운받아 아래 파일경로의 위치에 둔다

# 팬텀 js로 하면 백그라운드로 실행할 수 있음 -> 클릭하면서 보여주는 것을 보지 않고 컴퓨터가 자동으로 수행하도록 하는 기능

binary = 'E:\chromedriver/chromedriver.exe'

# 브라우저를 인스턴스화

browser = webdriver.Chrome(binary)

# browser 라는 class에 binary 를 상속받아서 인스턴스화(실행) 한다.

# 구글의 이미지 검색 url 받아옴(아무것도 안 쳤을때의 url) --> 구글 이미지 사이트 url 처음 들어갔을때의 주소

browser.get("https://www.google.co.kr/imghp?hl=ko&tab=wi&ei=l1AdWbegOcr8QXvtr-4Cw&ved=0EKouCBUoAQ")

# 구글의 이미지 검색에 해당하는 input 창의 id 가 ' ? ' 임 -> ? 를 알아내야 한다는 뜻

# 검색창에 해당하는 html코드를 찾아서 elem이 사용할 수 있도록 설정

# input창 찾는 방법은 원노트에 있음

# elem = browser.find\_elements\_by\_class\_name('gLfyf gsfi') # Tip : f12누른후 커서를 검색창에 올려두고 id를 찾으면 best

elem = browser.find\_element\_by\_xpath("//\*[@class='gLfyf gsfi']") # 위의 코드대로 하거나 이렇게 하거나 둘 중 하나 select

# 혹시나 그 사이에 바뀔 수도있어 다시 확인해 보았으나 (20.12.18일자 기준으로 동일함)

##### 검색어 입력 #####

# elem 이 input 창과 연결되어 스스로 띄워옴을 검색

elem.send\_keys("떡볶이") # 여기에 스크롤링하고싶은 검색어를 입력

# 웹에서의 submit 은 엔터의 역할을 함

elem.submit()

# 현재 결과 더보기는 구현 되어있지 않은상태 -> 구글의 경우 400개 image가 저장됨.

##### 반복할 횟수 #####

# 스크롤을 내리려면 브라우저 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번 하고 End키를 눌러야함

```

for i in range(1, 6): # 5번 스크롤 내려가게 구현된 상태 range(1,5)

    browser.find_element_by_xpath("//body").send_keys(Keys.END)

    time.sleep(10)    # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 씬

time.sleep(10)        # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)

html = browser.page_source    # 크롬브라우저에서 현재 불러온 소스 가져옴

soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


browser.find_element_by_xpath("//*[@class='mye4qd']").click() # 여기가 결과 더보기 코드!! 그 더보기를 클릭하라는 것!


for i in range(1, 5): # 4번 스크롤 내려가게 구현된 상태 range(1,5)

    browser.find_element_by_xpath("//body").send_keys(Keys.END)

    time.sleep(10)    # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 씬

time.sleep(10)        # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)

html = browser.page_source    # 크롬브라우저에서 현재 불러온 소스 가져옴

soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


##### 그림파일 저장 #####

### 검색한 구글 이미지의 url을 따오는 코드 ###

def fetch_list_url():

    params = []

    imgList = soup.find_all("img", class_="rg_i Q4LuWd") # 구글 이미지 url 이 있는 img 태그의 _img 클래스에 가서 (f12로 확인가능.)

    for im in imgList:

        try :

            params.append(im["src"])    # params 리스트에 image url 을 담음.

        except KeyError:

            params.append(im["data-src"])

    return params


# except부분

# 이미지의 상세 url 의 값이 있는 src 가 없을 경우

```

# data-src 로 가져오시오 ~

```
def fetch_detail_url():
```

```
    params = fetch_list_url()
```

```
    for idx,p in enumerate(params,1):
```

```
        # 다운받을 폴더경로 입력
```

```
        urllib.request.urlretrieve(p, "E:/gimages/" + str(idx) + "_google.jpg")
```

# enumerate 는 리스트의 모든 요소를 인덱스와 쌍으로 추출

# 하는 함수 . 숫자 1은 인덱스를 1부터 시작해라 ~

```
fetch_detail_url()
```

# 끝나면 브라우저 닫기

```
browser.quit()
```

```
###
```

문제 438. 이번에는 E드라이브 밑에 gimages2 라는 폴더를 만들고 거기에 다른 이미지를 다운받아 보시오.

이번 키워드 'porto'

```
###
```

import urllib.request # 웹 url을 파이썬이 인식 할 수 있게하는 패키지

from bs4 import BeautifulSoup # html에서 데이터 검색을 용이하게 하는 패키지

from selenium import webdriver

# selenium : 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크

# 손으로 클릭하는것을 컴퓨터가 대신하면서 스크롤링하게 하는 패키지

from selenium.webdriver.common.keys import Keys

import time # 중간중간 sleep 을 걸어야 해서 time 모듈 import

```
##### url 받아오기 #####
```

# 웹브라우저로 크롬을 사용할거라서 크롬 드라이버를 다운받아 아래 파일경로의 위치에 둔다

# 팬텀 js로 하면 백그라운드로 실행할 수 있음

```
binary = 'E:\chromedriver/chromedriver.exe'
```

```
# 브라우저를 인스턴스화
```

```
browser = webdriver.Chrome(binary)
```

```
# 구글의 이미지 검색 url 받아옴(아무것도 안 쳤을때의 url)
```

```
browser.get("https://www.google.co.kr/imghp?hl=ko&tab=wi&ei=l1AdWbegOcra8QXvtr-4Cw&ved=0EKouCBUoAQ")
```

```
# 구글의 이미지 검색에 해당하는 input 창의 id 가 ' ? ' 임(검색창에 해당하는 html코드를 찾아서 elem 사용하도록 설정)
```

```
# input창 찾는 방법은 원노트에 있음
```

```
# elem = browser.find_elements_by_class_name('gLfyf gsfi') # Tip : f12누른후 커서를 검색창에 올려두고 id를 찾으면 best
```

```
elem = browser.find_element_by_xpath("//*[@class='gLfyf gsfi']") # 위의 코드대로 하거나 이렇게 하거나 둘 중 하나 select
```

```
##### 검색어 입력 #####
```

```
# elem 이 input 창과 연결되어 스스로 햄버거를 검색
```

```
elem.send_keys("porto") # 여기에 스크롤링하고싶은 검색어를 입력
```

```
# 웹에서의 submit 은 엔터의 역할을 함
```

```
elem.submit()
```

```
# 현재 결과 더보기는 구현 되어있지 않은상태 -> 구글의 경우 400개 image가 저장됨.
```

```
##### 반복할 횟수 #####
```

```
# 스크롤을 내리려면 브라우저 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번 하고 End키를 눌러야함
```

```
for i in range(1, 6): # 5번 스크롤 내려가게 구현된 상태 range(1,5)
```

```
browser.find_element_by_xpath("//body").send_keys(Keys.END)
```

```
time.sleep(10) # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 쉼
```

```

time.sleep(10)          # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)

html = browser.page_source    # 크롬브라우저에서 현재 불러온 소스 가져옴

soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


browser.find_element_by_xpath("//*[@class='mye4qd']").click() # 여기가 결과 더보기 코드입니다


for i in range(1, 5): # 4번 스크롤 내려가게 구현된 상태 range(1,5)

    browser.find_element_by_xpath("//body").send_keys(Keys.END)

    time.sleep(10)      # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5번 누르는데 end1번누르고 10초 쉼


time.sleep(10)          # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지가 다운받아질 수 있음.)

html = browser.page_source    # 크롬브라우저에서 현재 불러온 소스 가져옴

soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정


##### 그림파일 저장 #####

### 검색한 구글 이미지의 url을 따오는 코드 ###

def fetch_list_url():

    params = []

    imgList = soup.find_all("img", class_="rg_i Q4LuWd") # 구글 이미지 url 이 있는 img 태그의 _img 클래스에 가서 (f12로 확인가능.)

    for im in imgList:

        try :

            params.append(im["src"])          # params 리스트에 image url 을 담음.

        except KeyError:

            params.append(im["data-src"])

    return params


# except부분

# 이미지의 상세 url 의 값이 있는 src 가 없을 경우

# data-src 로 가져오시오 ~


def fetch_detail_url():

```

```

params = fetch_list_url()

for idx,p in enumerate(params,1):

    # 다운받을 폴더경로 입력

    urllib.request.urlretrieve(p, "E:/gimages2/" + str(idx) + "_google.jpg")


# enumerate 는 리스트의 모든 요소를 인덱스와 쌍으로 추출

# 하는 함수 . 숫자 1은 인덱스를 1부터 시작해라 ~


fetch_detail_url()


# 끝나면 브라우저 닫기

browser.quit()

#%%%

딥러닝으로 인공지능을 학습시키려면 한 클래스당 2000장 정도가 있으면 적당하다.
e.g) 개와 고양이를 분류(알아보는) 하는 인공지능을 만들고 싶다고 할때, 개(class1) 2000장 과 고양이(class2) 2000장의 사진이 있어야
둘을 구분하는 인공지능을 만들 수 있다.

147. 웹에 있는 사진을 스크롤링하는 방법(네이버이미지)

#%%%

■ 147. 이미지 스크롤링 하기 ( 네이버 이미지 검색)

#%%%

import urllib.request # 웹 url을 파이썬이 인식할 수 있게 하는 패키지
from bs4 import BeautifulSoup # html 코드에서 원하는 지점을 빠르게 찾을 수 있도록 만든 모듈
from selenium import webdriver # 손으로 클릭하는 것을 컴퓨터가 하게 만드는 모듈
from selenium.webdriver.common.keys import Keys
import time # 중간 중간 sleep 을 주려고 -> 그래야 사진이 완벽하게 load된 상태에서 완벽한 이미지를 다운 받을 수 있음. (네트워크 속도가 느릴수록 sleep 시간을
더 줘야함)

binary = 'E:\chromedriver/chromedriver.exe' # chromedriver 위치 명시

browser = webdriver.Chrome(binary) # browser 객체 생성
#충(객체)      #충설계도 처럼
#
browser.get("https://search.naver.com/search.naver?where=image&sm=stb_nmr&") #네이버의 이미지 검색창에 아무것도 입력하지 않은 상태의 url
elem = browser.find_element_by_id("nx_query") # 일반적으로 하듯이 class 로 찾는데 아니라, id명으로 찾았음.
# elem -> 이미지 검색을 위해 입력해야 하는 검색창 부분을 알아내는 코드를 elem 에 담은 것

# 20.12.18 기준 id("nx_query") 는 아직 바뀌지 않음 (4년째)
# 만약 이 코드 돌려서 안되면 이부분 제일 먼저 확인!

#find_elements_by_class_name("") #클래스 이름으로 찾으려고 할때 필요한 코드

```



# 검색어 입력

elem.send\_keys("블랙팬서") # 컴퓨터가 위에서 지정한 elem의 위치에 '아이언맨'이라는 글씨를 직접 타이핑

elem.submit() # 엔터!

# 반복할 횟수 end 키를 1번씩 누르면서 내리는 작업을 5번 반복 하는 것. + sleep을 5초간 주면서! (완벽하게 로드된 이미지를 가져오기 위해)

for i in range(1,5):

browser.find\_element\_by\_xpath("//body").send\_keys(Keys.END)

time.sleep(5)

time.sleep(5) # 1번 end 눌러서 내려온뒤에 5초 sleep,

html = browser.page\_source # 현페이지의 html 코드를 불러와서

soup = BeautifulSoup(html,"xml") # BeautifulSoup 을 이용할 수 있도록 파싱!

#print(soup) # 중간 절차 확인차 출력해본 것

#print(len(soup))

def fetch\_list\_url(): # 이미지의 상세 url 가져오는 함수 생성

params = []

imgList = soup.find\_all("img", class="\_img") # img 태그의 class 이름 \_img 로 접근해서

for im in imgList:

params.append(im["src"]) #src의 값(=각 이미지의 url)을 가져와서 params에 append

return params

def fetch\_detail\_url(): # 상세 이미지 url 가져와서 그 이미지를 다운로드하는 함수

params = fetch\_list\_url()

#print(params)

a = 1

for p in params:

# 다운받을 폴더경로 입력

urllib.request.urlretrieve(p, "E:/naverImages/"+ str(a) + ".jpg" ) #cnt개념처럼 이미지의 저장이름을 1부터 1개늘어날때마다 1씩 더 늘어나게 1,2,3으로 저장하는 것  
# E 드라이브 밑에 naverImages라는 폴더를 만들어서 저장하라는 것

a = a + 1

fetch\_detail\_url() #

browser.quit()

###

보통 한클래스당 필요한 사진 2000장은 구글에서 1번, 네이버에서 1번, bing까지 3개정도 돌리면 한클래스를 돌릴 분량은 충분히 수집가능

페이스북, 인스타그램도 가능(나중에 시도해보기)

딥러닝 수업 대비해서 학습하고 싶은 이미지 2가지 각각 2000장씩 사진 수집 해두기

###

(오늘의 마지막 문제)

문제 439. 마이크로 소프트에서 만든 검색페이지인 bing에서 이미지 검색을 할 수 있도록 하는 웹스크롤링 코드를 작성하시오

(네이버 코드를 조금 수정하면 충분히 가능)

###

import urllib.request # 웹 url을 파이썬이 인식할 수 있게 하는 패키지

from bs4 import BeautifulSoup # html 코드에서 원하는 지점을 빠르게 찾을 수 있도록 만든 모듈

from selenium import webdriver # 손으로 클릭하는 것을 컴퓨터가 하게 만드는 모듈

from selenium.webdriver.common.keys import Keys

import time # 완벽한 이미지 수집위해 중간 중간 sleep으로 로딩시간 확보

```

binary = 'E:\chromedriver/chromedriver.exe' # chromedriver 위치 명시

browser = webdriver.Chrome(binary) # browser 객체 생성

browser.get("https://www.bing.com/?scope=images&nr=1&FORM=NOFORM") # bing의 이미지 검색창에 아무것도 입력하지 않은 상태의 url
elem = browser.find_element_by_id("sb_form_q") # 위의 url에서 검색어를 입력하는 창의 위치를 elem에 할당

# 검색어 입력
elem.send_keys("harry potter") # 컴퓨터가 위에서 지정한 elem의 위치에 'harry potter' 이라는 글씨를 직접 타이핑

elem.submit() # 검색창에 입력후 엔터키 입력

bing은 왜인지 모르겠으나 이미지 창에서 검색후 엔터를 치면 종합검색 창으로 넘어감
파이썬 코드사용없이 똑같은 url에서 손으로 직접 키워드 입력후 엔터치면 이미지 검색으로 넘어감
[↓][↓] 그래서 종합검색페이지로 넘어간 후에 이미지탭을 마우스가 클릭하도록 하는 함수 작성 [↓][↓]

browser.find_element_by_id("b-scopeListItem-images").click() # 키워드에 대한 전체검색결과 출력된 페이지에서 '이미지' 부분을 클릭하는 함수

time.sleep(2) # 혹시나 이미지탭 넘어갈 시 로딩속도가 오래걸릴 것 대비

# 반복할 횟수 end 키를 1번씩 누르면서 내리는 작업을 5번 반복 하는 것. + sleep을 5초간 주면서! (완벽하게 로드된 이미지를 가져오기 위해)
for i in range(1,16):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5)

time.sleep(5) # 1번 end 눌러서 내려온뒤에 5초 sleep,

html = browser.page_source # 현페이지의 html 코드를 불러와서
soup = BeautifulSoup(html,"lxml") # BeautifulSoup 을 이용할 수 있도록 파싱!

#print(soup) # 중간 절차 확인차 출력해본 것
#print(len(soup))

def fetch_list_url(): # 이미지의 상세 url 가져오는 함수 생성
    params = []
    imgList = soup.find_all("img", class_="mimg") # img 태그의 class 이름 _img 로 접근해서
    for im in imgList:
        params.append(im["src"]) #src 의 값(=각 이미지의 url) 을 가져와서 params 에 append
    return params

def fetch_detail_url(): # 상세 이미지 url 가져와서 그 이미지를 다운로드하는 함수
    params = fetch_list_url()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "E:/bingImages/"+ str(a) + ".jpg" ) #cnt 개념처럼 이미지의 저장이름을 1부터 1개늘어날때마다 1씩 더 늘어나게 1,2,3 으로 저장하는 것
        # E 드라이브 밑에 naverImages 라는 폴더를 만들어서 저장하라는 것
        a = a + 1

fetch_detail_url() #

browser.quit()

###

```

딥러닝 수업 대비해서 학습하고 싶은 이미지 2가지 각각 2000장씩 사진 수집 해두기

20.12.21 파이썬 강의 Day 17(문제440~문제475)

```
# -*- coding: utf-8 -*-  
"""
```

Created on Mon Dec 21 09:32:52 2020

```
@author: noste  
"""
```

연동

오라클 DB ----- 파이썬 (통계구현, 시각화, 머신러닝, 업무자동화)



비즈니스 데이터            e.x) emp2.csv  
(정형화된 데이터)            emp1222.csv

실시간으로 데이터가 수시로 업데이트 되는 경우 오라클에서 저장하고 파이썬에서 불러오는 것은 비효율적  
실시간으로 두 프로그램이 연동되어 사용할 수 있도록 하는 것이 더 좋음

빅데이터 기사 시험 내용중..

데이터 구조에 따른 종류 3가지

1. 정형화 데이터 : rdbms에 저장된 데이터

rdbms = relational database management system (관계형 데이터 베이스 관리 시스템)

2. 반정형화 데이터 : html, web log data

3. 비정형화 데이터 : 동영상, 이미지 (SNS 등)

오라클과 파이썬을 연동하는 이유?

1. 오라클 데이터베이스에서 실시간으로 변하는 데이터를 csv파일로 매번 내릴려면 수시로 해야 하기때문에 바로 연동해서 볼 수있도록 하기 위해

오라클과 파이썬을 연동하는 방법

1. 리스너의 상태를 확인

도스창 열고

lsnrctl status 입력

3) 'HOST = ' 라고 뜨는 것이 건물주소 (아이피 주소)

4) PORT = 라고 뜨는 것이 건물 통로 번호

5) "orcl" 서비스는 라고 뜨는 것이 회사 이름. CLRExtProc 서비스는~~

(1) ip 주소 (건물 주소)

localhost

(2) 포트번호 (건물안의 복도)

1521

(3) 서비스 이름 (회사 이름)

sql

2. 위의 정보들을 이용해서 SQL에 접속한다.

sql 접속 시도

sqlplus system/oracle\_4U@localhost:1521/orclXDB as sysdba 로했음

3. 아나콘다 프롬프트 창을 열고 cx\_Oracle 모듈을 설치한다.

conda install cx\_Oracle

4. 파이썬(스파이더) 에서 오라클과 연동하는 코드를 작성한다.

```
###
```

```
import cx_Oracle
import pandas as pd
```

```
dsn = cx_Oracle.makedsn("localhost",1521,'orcl')
#오라클 주소를 써주는 것. (로컬호스트,포트번호, 서비스 이름)
```

```
print(dsn)
###
db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보
```

```
cursor = db.cursor() #데이터를 담을 메모리 이름을 cursor 선언하는 것
cursor.execute("""select*from emp""") #괄호안에 작성한 쿼리문의 결과가 cursor메모리에 담긴다.
```

```
row = cursor.fetchall() # cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
```

```
emp = pd.DataFrame(row)
```

```
print(emp)
```

```
###
```

상태: 실패 -테스트 실패: Listener refused the connection with the following error:  
ORA-12505, TNS:listener does not currently know of SID given in connect descriptor

<https://stackoverflow.com/questions/56119490/cx-oracle-error-dpi-1047-cannot-locate-a-64-bit-oracle-client-library>  
사이트의 5번 답변을 기반으로 해결 했다.

```
###
```

문제 위에~~~

문제 443. 월급이 1200 이상인 직원들의 이름과 월급을 출력하시오.

#%%

```
cursor.execute("""select ename, sal from emp where sal>=1200""")
row = cursor.fetchall()
```

```
emp = pd.DataFrame(row)
```

```
print(emp)
```

#%%

문제 444. 사원 테이블의 월급을 막대그래프로 그리시오.

#%%

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

```
dsn = cx_Oracle.makedsn("localhost",1521,'orcl')
```

```
db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보
```

```
cursor = db.cursor() #데이터를 담을 메모리 이름을 cursor 선언하는 것
```

```
cursor.execute("""select ename, sal from emp""") #괄호안에 작성한 쿼리문의 결과가 cursor메모리에 담긴다.
```

```
row = cursor.fetchall() # cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
```

```
emp = pd.DataFrame(row)
```

```
print(list(emp[0])) # 왜 0을 썼는가? 현재 cursor에 담긴 데이터가 column 형태로 되어있지 않아서!
```

```
# 컬럼으로 바꾸는 법은 오후에 다시 알려주시기로!
```

```
emp.index = list(emp[0])
```

```
emp.plot(kind='bar') # 막대그래프가 형성되어 뜬다.
```

#%%

(점심시간 문제)

문제 445. 위의 그래프 색깔을 변경하고, 그래프를 카페에 업로드 하시오.

#%%

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

```
dsn = cx_Oracle.makedsn("localhost",1521,'orcl')
```

```
db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보
```

```
cursor = db.cursor() #데이터를 담을 메모리 이름을 cursor 선언하는 것
```

```
cursor.execute("""select ename, sal from emp""") #괄호안에 작성한 쿼리문의 결과가 cursor메모리에 담긴다.
```

```
row = cursor.fetchall() # cursor 메모리에 담긴 결과를 한번에 row변수에 담는다.
```

```
emp = pd.DataFrame(row)
```

```
# print(list(emp[0])) # sql 명령어로 입력된 결과값의 컬럼 x 벨류 형태로 모든 값이 나옴. (표가 1개의 값으로서 0번째 값으로 인식됨.)
```

```
# 왜 0을 썼는가? 현재 cursor에 담긴 데이터가 column 형태로 되어있지 않아서!
```

```
# 컬럼으로 바꾸는 법은 오후에 다시 알려주시기로!
```

```
emp.index = list(emp[0])
```

```
emp.plot(kind='bar',color='purple') # 막대그래프가 형성되어 뜬다.
```

#%%

문제 446. 직업, 직업별 최대월급을 출력하시오.

```
select job, max(sal)
  from emp
 group by job;
```

###

cursor.execute("""select job,max(sal) from emp group by job""") # 출력하고 싶은 sql 명령어를 입력한다

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.

emp.index = list(emp[0]) # emp의 결과를 list화 한다.

emp.plot(kind='bar',color='purple')

print(emp) # 출력

###

문제 447. 직업, 직업별 토달월급을 출력하는데, 직업별 토달월급이 높은 것부터 출력하시오.

```
select job, sum(sal)
  from emp
 group by job
 order by sum(sal) desc;
```

###

cursor.execute("""select job, sum(sal)

from emp

group by job

order by sum(sal) desc""") # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.

emp.index = list(emp[0]) # emp의 결과를 list화 한다.

emp.plot(kind='bar',color='purple')

print(emp) # 출력

# 결국은 SQL만 잘알면, 판다스 문법없이도 SQL 로 모두 가져올 수 있다.

# 물론 위처럼 SQL과 파이썬을 연동하는 코드는 알아야함(구글링하면 다 얻을 수 있음.)

###

문제 448. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 '사원' 순으로 출력하시오.

```
select ename, sal, dense_rank() over(order by sal desc) rnk
  from emp;
```

###

cursor.execute("""select ename, sal,

dense\_rank() over (order by sal desc) rnk

from emp """) # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.

emp.index = list(emp[0]) # emp의 결과를 list화 한다.

emp.plot(kind='bar',color='purple')

print(emp) # 출력

###

문제 449. 부서번호, 부서번호별 토달월급을 출력하시오.

```
select deptno, sum(sal)
  from emp
 group by deptno;
```

###

```
cursor.execute("""select deptno, sum(sal)
  from emp
  group by deptno""") # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!
```

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

```
emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
emp.index = list(emp[0]) # emp의 결과를 list화 한다.
emp.plot(kind='bar',color='purple')
print(emp) # 출력
```

###

문제 450. 위의 결과를 막대 그래프로 시각화 하시오. (이전문제도 혼자 그냥 계속 만들어왔어서 답이 449와 같음)

###

```
cursor.execute("""select deptno, sum(sal)
  from emp
  group by deptno""") # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!
```

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

```
emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
emp.index = list(emp[0]) # emp의 결과를 list화 한다.
emp.plot(kind='bar',color='purple')
print(emp) # 출력
```

###

문제 451. emp 테이블 전체를 출력하는데 컬럼명이 붙어서 출력되게 하시오.

###

```
cursor.execute("""select *
  from emp""") # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!
```

row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

```
emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
colname = cursor.description # emp 테이블의 컬럼명 가져오기
col = []

for i in colname:
    col.append(i[0].lower())
print(col) # ['empno', 'ename', 'job', 'mgr', 'hiredate', 'sal', 'comm', 'deptno'] # 소문자로 하는게 나중에 작업하기 편하다.
```

###

문제 452. 위의 col 리스트에 담긴 컬럼명을 위의 데이터 매핑시키시오.

###

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

```
dsn = cx_Oracle.makedsn("localhost",1521,'orcl')
```

```
db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보
cursor = db.cursor()
```

```

cursor.execute(""" select *
                from emp
                """) # emp의 전체 테이블 내용을 불러온다.

row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description # 컬럼명을 가져온다.
col = [ ]
for i in colname:
    col.append( i[0].lower() )
emp = pd.DataFrame (list(row), columns=col)
print (emp)

###
문제 453. 위의 컬럼명을 이용해서 판다스 문법으로 이름, 월급을 출력하시오.
###

```

```

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

dsn = cx_Oracle.makedsn("localhost",1521,'orcl')

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보
cursor = db.cursor()
cursor.execute(""" select *
                from emp
                """) # emp의 전체 테이블 내용을 불러온다.

row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description # 컬럼명을 가져온다.
col = [ ]
for i in colname:
    col.append( i[0].lower() )
emp = pd.DataFrame (list(row), columns=col)
print (emp[['ename','sal']])

###

```

```

문제 454. 월급이 3000이상인 사원들의 이름과 월급을 출력하시오.
###
import cx_Oracle
import pandas as pd
import matplotlib.pyplot as plt

dsn = cx_Oracle.makedsn("localhost",1521,'orcl')

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보

cursor = db.cursor()
cursor.execute(""" select *
                from emp
                """) # emp의 전체 테이블 내용을 불러온다.

row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description # 컬럼명을 가져온다.
col = [ ]
for i in colname:
    col.append( i[0].lower() )
emp = pd.DataFrame (list(row), columns=col)
print (emp[['ename','sal']] [emp['sal'] >=3000] )

###

```

문제 455. 이름과 부서위치를 작성하시오

```

select e.ename, d.loc

```



```

from emp e, dept d
where e.deptno = d.deptno;

###
import cx_Oracle
import pandas as pd
import matplotlib.pyplot as plt

dsn = cx_Oracle.makedsn("localhost",1521,'orcl')

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보

cursor = db.cursor()
cursor.execute("""select e.ename, d.loc
                  from emp e, dept d
                  where e.deptno = d.deptno
                  """) # 출력하고 싶은 sql 명령어를 입력한다 sql 쓰듯이 이런 형태로 써도 된다.!
row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
emp.index = list(emp[0]) # emp의 결과를 list화 한다.
emp.plot(kind='bar',color='purple')
print(emp) # 출력

```

###

문제 456. 부서위치, 부서위치별 토달월급을 출력하시오.

```

select d.loc, sum(e.sal)
  from emp e, dept d
 where e.deptno = d.deptno
 group by d.loc

###

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

dsn = cx_Oracle.makedsn("localhost",1521,'orcl')

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보

cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
                  from emp e, dept d
                  where e.deptno = d.deptno
                  group by d.loc
                  """) # 출력하고 싶은 sql 명령어를 입력한다 sql 쓰듯이 이런 형태로 써도 된다.!
row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
# 아래 2개는 그래프화를 위해 필요한 것들
# emp.index = list(emp[0]) # emp의 결과를 list화 한다.
# emp.plot(kind='bar',color='purple')

print(emp) # 출력

```

###

문제 457. 위의 결과를 막대그래프로 시각화 하시오.

###

```

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

```

```

dsn = cx_Oracle.makedsn("localhost",1521,'orcl')

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보

cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
                from emp e, dept d
                where e.deptno = d.deptno
                group by d.loc
                """) # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.
row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.

emp.index = list(emp[0]) # emp의 결과를 list화 한다.
emp.plot(kind='bar',color='purple')
# print(emp) # 출력

#%%

```

149 파이썬과 mySQL 연동 (연동 시작은 훨씬 아래에!)

현업에서는 오라클도 많이 쓰지만 mySQL도 많이 사용한다.

오라클은 가격이 좀 비싼편인지라 중요한 데이터(Business data)를 오라클에 저장하고 상대적으로 덜 중요한 데이터는 mySQL에 저장한다.

## ■ mySQL 설치

아래의 링크 참조해서 설치  
[m.blog.naver.com/bjh7007/221829548634](https://m.blog.naver.com/bjh7007/221829548634)

아이디는 default로 root  
비번은 tiger로 했음. (샘은 oracle로 하라 하셨는데 그전에 이미 설치해버림)

## ■ mySQL에서 데이터 베이스 생성하기

시작 -> mySQL 폴더 -> MySQL 8.0 Command Line Client 실행

password에 tiger 입력 후 엔터

```

create database orcl; 입력 후 엔터
>> Query OK, 1 row affected (0.01 sec)

```

```

use orcl;
>> Database changed

```

## ■ MySQL에서 emp 테이블 만들기

해당 스크립트 카페에 올려둬. 다운받아서 사용!

MySQL 8.0 Command Line Client에 복사한 스크립트 삽입(우클릭)

SQL 때 처럼 한줄씩 실행되면서 Query OK, #rows affected ~~~ 이런식으로 나옴.

```

select * from emp;
하면 테이블이 나옴.

```

```

mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+-----+
| empno | Ename | Job   | Mgr | Hiredate | Sal | Comm | Deptno |
+-----+-----+-----+-----+-----+-----+-----+
| 7839  | KING  | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL | 10 |

```

```

| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | NULL | 30 |
| 7782 | CLARK | MANAGER | 7839 | 1981-05-09 | 2450 | NULL | 10 |
| 7566 | JONES | MANAGER | 7839 | 1981-04-01 | 2975 | NULL | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 1981-09-10 | 1250 | 1400 | 30 |
| 7844 | TURNER | SALESMAN | 7698 | 1981-08-21 | 1500 | 0 | 30 |
| 7900 | JAMES | CLERK | 7698 | 1981-12-11 | 950 | NULL | 30 |
| 7521 | WARD | SALESMAN | 7698 | 1981-02-23 | 1250 | 500 | 30 |
| 7902 | FORD | ANALYST | 7566 | 1981-12-11 | 3000 | NULL | 20 |
| 7369 | SMITH | CLERK | 7902 | 1980-12-09 | 800 | NULL | 20 |
| 7788 | SCOTT | ANALYST | 7566 | 1982-12-22 | 3000 | NULL | 20 |
| 7876 | ADAMS | CLERK | 7788 | 1983-01-15 | 1100 | NULL | 20 |
| 7934 | MILLER | CLERK | 7782 | 1982-01-11 | 1300 | NULL | 10 |
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)

```

문법은 대부분 SQL 과 같음

문제 458. 월급이 3200이상인 직원들의 이름, 월급을 출력하시오.

```

select ename, sal
  from emp
 where sal >=3200;

mysql> select ename, sal
  -> from emp
  -> where sal >=3200;
+-----+-----+
| ename | sal |
+-----+-----+
| KING | 5000 |
+-----+-----+
1 row in set (0.00 sec)

```

문제 459. 직업이 SALESMAN인 직원들의 이름, 월급, 직업을 출력하는데 월급이 높은 순으로 출력하시오.

```

mysql> select ename, sal, job
  -> from emp
  -> order by sal desc;
+-----+-----+-----+
| ename | sal | job |
+-----+-----+-----+
| KING | 5000 | PRESIDENT |
| FORD | 3000 | ANALYST |
| SCOTT | 3000 | ANALYST |
| JONES | 2975 | MANAGER |
| BLAKE | 2850 | MANAGER |
| CLARK | 2450 | MANAGER |
| TURNER | 1500 | SALESMAN |
| MILLER | 1300 | CLERK |
| MARTIN | 1250 | SALESMAN |
| WARD | 1250 | SALESMAN |
| ADAMS | 1100 | CLERK |
| JAMES | 950 | CLERK |
| SMITH | 800 | CLERK |
+-----+-----+-----+
13 rows in set (0.00 sec)

```

문제 460. 이름과 부서위치를 출력하시오.

문제 461. JONES보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오.

```

select ename, sal
  from emp
 where (select ename, sal
        from emp
        where ename='JONES');

```

문제 462. 이름, 월급, 순위를 출력하시오.

```
mysql> select ename, sal, dense_rank() over(order by sal desc)
-> from emp;
+-----+-----+
| ename | sal | dense_rank() over(order by sal desc) |
+-----+-----+
| KING | 5000 | 1 |
| FORD | 3000 | 2 |
| SCOTT | 3000 | 2 |
| JONES | 2975 | 3 |
| BLAKE | 2850 | 4 |
| CLARK | 2450 | 5 |
| TURNER | 1500 | 6 |
| MILLER | 1300 | 7 |
| MARTIN | 1250 | 8 |
| WARD | 1250 | 8 |
| ADAMS | 1100 | 9 |
| JAMES | 950 | 10 |
| SMITH | 800 | 11 |
+-----+-----+
13 rows in set (0.01 sec)
```

## ■ Oracle 과 MySQL 함수 비교

Oracle	vs	MySQL
1. nvl		ifnull
2. sysdata		sysdata()
3. months_between		period_add
4. decode		if
5. rollup		with rollup
6. listagg		group_concat

문제 463. 이름, 커미션출력하는데 커미션이 null인 사람들은 0으로 출력하시오.

\* nvl-> ifnull이다.

```
select ename, ifnull(comm,0)
from emp;
```

```
mysql> select ename, ifnull(comm,0)
-> from emp;
+-----+-----+
| ename | ifnull(comm,0) |
+-----+-----+
| KING | 0 |
| BLAKE | 0 |
| CLARK | 0 |
| JONES | 0 |
| MARTIN | 1400 |
| TURNER | 0 |
| JAMES | 0 |
| WARD | 500 |
| FORD | 0 |
| SMITH | 0 |
| SCOTT | 0 |
| ADAMS | 0 |
| MILLER | 0 |
+-----+-----+
13 rows in set (0.00 sec)
```

문제 464. 오늘 날짜를 출력하시오.

```
SQL> select sysdate
from dual;
```

```
mysql> select sysdate()
-> ;
+-----+
| sysdate() |
+-----+
```

| 2020-12-21 15:12:23 |

+-----+

1 row in set (0.00 sec)

\* MySQL 은 오라클의 dual 이란 개념이 없다.

그냥 한칸 내리고 ; 로 마무리

문제 465. 부서번호, 부서번호별 토달월급

```
mysql> select deptno, sum(sal)
```

```
-> from emp
```

```
-> group by deptno;
```

+-----+

| deptno | sum(sal) |

+-----+

| 10 | 8750 |

| 30 | 7800 |

| 20 | 10875 |

+-----+

3 rows in set (0.00 sec)

문제 466. 위의 결과를 다시 출력하는데 맨 아래에 전체 토달월급이 출력되게 하시오.

\* rollup -> with rollup 이다.

```
SQL> select deptno, sum(sal)
```

```
from emp
```

```
group by rollup(deptno);
```

```
mysql> select deptno, sum(sal)
```

```
-> from emp
```

```
-> group by deptno with rollup;
```

+-----+

| deptno | sum(sal) |

+-----+

| 10 | 8750 |

| 20 | 10875 |

| 30 | 7800 |

| NULL | 27425 |

+-----+

4 rows in set (0.00 sec)

☆ 선배기수들이 선생님에게 수료 후에도 자주 물어보는 문제

문제 467. scott 의 월급을 0으로 변경하시오. (update 하라는 것)

```
SQL>update emp
```

```
set sal =0
```

```
where ename ='SCOTT';
```

```
mysql> update emp
```

```
-> set sal =0
```

```
-> where ename ='SCOTT';
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

실행 결과를 되돌리고 싶어서 rollback 을 했는데..

```
mysql> rollback;
```

Query OK, 0 rows affected (0.00 sec) -> 바뀌는게 없다!!

이래서 선배기수들이 연락이 많이옴.

설명 : 오라클과는 다르게 MySQL은 기본이 자동 커밋이 활성화 되어 있다.

그래서 업데이트를 하든, 삭제를 하든, 변경을 하든 실행하는 순간 바로 커밋이 되어 rollback 을 할수가 없다.

MySQL 사용자들이 많이 하는 실수중 하나.

문제 468. 자동 커밋이 활성화 되어져 있는지 확인하는 법?

```
mysql> select @@autocommit;
+-----+
| @@autocommit |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)
```

설명:

숫자 1이면 autocommit이 활성화 된 것이다.

그래서 rollback 을 쓰고 싶다면 이 기능을 비활성화 해야 한다.

문제 469. 자동 커밋 기능을 비활성화(Off) 하시오.

```
mysql> set autocommit = FALSE;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select @@autocommit;
+-----+
| @@autocommit |
+-----+
|          0 |
+-----+
1 row in set (0.00 sec)
```

설명 : 재확인한 결과, 결과값이 0으로 나온다. 0 = 비활성화 되었다.

문제 470. KING의 월급을 0으로 바꿔보시오.

```
mysql> update emp
-> set sal = 0
-> where ename ='KING';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

rollback 시도하고 복구 되었는지 확인.

```
mysql> rollback;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select ename, sal
-> from emp
-> where ename = 'KING';
+-----+-----+
| ename | sal |
+-----+-----+
| KING | 5000 |
+-----+-----+
1 row in set (0.01 sec)
```

문제 471. 오라클의 listagg 함수와 같은 기능인 group\_concat 을 이용해서 부서번호, 부서번호별로 속한 직원들의 이름 을 가로로 출력하시오.

```
mysql> select deptno, group_concat(ename)
-> from emp
-> group by deptno;
+-----+-----+
| deptno | group_concat(ename) |
+-----+-----+
| 10 | KING,CLARK,MILLER |
| 20 | JONES,FORD,SMITH,SCOTT,ADAMS |
| 30 | BLAKE,MARTIN,TURNER,JAMES,WARD |
+-----+-----+
3 rows in set (0.00 sec)
```

문제 472. MySQL 과 파이썬을 연동해서 MySQL 의 emp 테이블을

1. 아나콘다 cmd prompt창을 연다.

2. pymysql 모듈을 설치한다.

```
conda install pymysql
```

3. MySQL 과 파이썬을 연동하는 코드를 작성하여 문제를 푼다.

```
###
```

```
import pymysql,pandas as pd # 모듈 2가지를 한번에 import 할수도있다!
```

```
conn = pymysql.connect(host="localhost", user="root",password="tiger", db="orcl",charset="utf8")
```

```
#          호스트명      MySQL설치시 기본 유저명, 비밀번호, db명
```

```
# 이하는 SQL때와 같음
```

```
curs = conn.cursor()
```

```
sql = "select * from emp"
```

```
curs.execute(sql)
```

```
rows = curs.fetchall()
```

```
colname = curs.description
```

```
col = []
```

```
for i in colname:
```

```
    col.append(i[0].lower())
```

```
emp = pd.DataFrame(list(rows),columns=col)
```

```
print(emp[['ename', 'sal']])
```

```
###
```

문제 473. 직업, 직업별 토달월급을 출력하시오.

```
select job, sum(sal)
```

```
from emp
```

```
group by job;
```

```
###
```

```
import pymysql,pandas as pd # 모듈 2가지를 한번에 import 할수도있다!
```

```
conn = pymysql.connect(host="localhost", user="root",password="tiger", db="orcl",charset="utf8")
```

```
#          호스트명      MySQL설치시 기본 유저명, 비밀번호, db명
```

```
# 이하는 SQL때와 같음
```

```
curs = conn.cursor()
```

```
sql = "select * from emp"
```

```
curs.execute(sql)
```

```
rows = curs.fetchall()
```

```
colname = curs.description
```

```
col = []
```

```
for i in colname:
```

```
    col.append(i[0].lower())
```

```
emp = pd.DataFrame(list(rows),columns=col)
```

```
result = emp.groupby('job')['sal'].sum().reset_index()
```

```
print(result)
```

```
# or
```

```
###
```

```
import pymysql
```

```
import pandas as pd
```

```
conn = pymysql.connect(host="localhost", user="root",password="tiger", db="orcl",charset="utf8")
```

```
curs=conn.cursor()
```

```
sql = "select job, sum(sal) from emp group by job"
```

```
curs.execute(sql)
```

```
rows = curs.fetchall()
```

```
emp = pd.DataFrame(rows)
```

```
print(emp)
```

```
###
```

문제 474. 위의 결과를 막대그래프로 시각화 하시오.

```
###
```

```
import pymysql,pandas as pd # 모듈 2가지를 한번에 import 할수도 있다!
```

```
conn = pymysql.connect(host="localhost", user="root",password="tiger", db="orcl",charset="utf8")
```

```
#          호스트명      MySQL설치시 기본 유저명, 비밀번호, db명
```

```
# 이하는 SQL때와 같음
```

```
curs = conn.cursor()
```

```
sql = "select * from emp"
```

```
curs.execute(sql)
```

```
rows = curs.fetchall()
```

```
colname = curs.description
```

```
col = []
```

```
for i in colname:
```

```
    col.append(i[0].lower())
```

```
emp = pd.DataFrame(list(rows),columns=col)
```

```
result = emp.groupby('job')['sal'].sum().reset_index()
```

```
emp.plot(kind='bar')
```

```
# or
```

```
###
```

```
import pymysql
```

```
import pandas as pd
```

```
conn = pymysql.connect(host="localhost", user="root",password="tiger", db="orcl",charset="utf8")
```

```
curs=conn.cursor()
```

```
sql = "select job, sum(sal) from emp group by job"
```

```
curs.execute(sql)
```

```
rows = curs.fetchall()
```

```
emp = pd.DataFrame(rows)
```

```
# emp.index = list(emp)
```

```
# emp.plot(kind='bar')
```

```
###
```

(오늘의 마지막 문제)

문제 475. 오라클과 파이썬을 연동하여 아래의 직원들을 검색하시오.

이름, 월급, 부서번호와 자기가 속한 부서번호의 평균 월급을 출력하는데

자신의 월급이 자기가 속한 부서번호의 평균 월급보다 더 큰 직원들만 출력하시오

```
select ename, sal, deptno, (select avg(sal) from emp) 평균월급
```

```
from emp
```

```
where sal > (select avg(sal) from emp);
```

```
###
```

```
import cx_Oracle
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
dsn = cx_Oracle.makedsn("localhost",1521,'orcl')
```



```

db = cx_Oracle.connect('scott','tiger',dsn) #오라클 접속 유저 정보

cursor = db.cursor()
cursor.execute("""select ename, sal, deptno, (select avg(sal) from emp) 평균월급
                from emp
                where sal > (select avg(sal) from emp)
                """) # 출력하고 싶은 sql 명령어를 입력한다 SQL 쓰듯이 이런 형태로 써도 된다.!
row = cursor.fetchall() # sql의 결과를 '한번에' row변수에 담는다.

emp = pd.DataFrame(row) # row 변수에 담은 결과를 판다스로 dataframe화 한다.
# 아래 2개는 그래프화를 위해 필요한 것들
# emp.index = list(emp[0]) # emp의 결과를 list화 한다.
# emp.plot(kind='bar',color='purple')

print(emp) # 출력

```

20.12.22 파이썬 강의 Day 18(문제476~문제501)

150 이미지를 숫자로 변환하는 방법 (폐사진)

딥러닝 수업때 이미지를 인공 신경망에 넣어서 학습을 시키는데 인공신경망에 사진을 넣을때 숫자로 변환을 해서 넣어야 한다.

```

사진 -----> 인공신경망 -----> 폐결절
                정상폐
                폐암
                .
                .
                .

```

공항검색대에 입국할때 위반되는 물건을 반입하는 것을 방지하기 위해 사람을 모니터링하는데 이를 딥러닝의 기술로 알아내게끔 하는 기능을 구현하려면 다양한 사진들이 필요!

이러한 기능을 구현하는 기초가 되는 것이 이미지를 숫자로 변환하는 것

e 드라이브 밑에 images 라는 폴더 밑에 폐사진 (1~20.jpg) 가 있어야 한다.

e.g 1) e드라이브 밑에 images 폴더에 있는 사진들의 이름을 불러오는 함수를 생성  
#%%

```

import os
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    return file_list

```

print(image\_load(test\_image)) # 파일명.png 로 1~20까지 총 20장이 출력되었다.

```

# 리스트가 정렬되어 있지 않고, ' ' 로 둘러져 문자형으로 출력되어 있다.
# 이를 정제해서 숫자 순서대로 정렬해야 한다.
#%%

```

e.g 2) 위의 결과에서 숫자만 나오게 하시오.  
#%%

```

import os

```

```

import re # 데이터 정제를 전문으로 하는 모듈
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]',",",i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
        # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
        file_name.append(a)
    return file_name

print(image_load(test_image))

###

```

문제 476. 위에서 출력되고 있는 리스트안의 요소들은 문자이다. 이것들을 문자가 아니라 숫자가 되게 하시오.

```

###

import os
import re # 데이터 정제를 전문으로 하는 모듈
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]',",",i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
        # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
        file_name.append(int(a) )
    return file_name

print(image_load(test_image))

###

```

문제 477. 위의 리스트의 요소가 ascending하게 정렬되게 하시오.

```

###

import os
import re # 데이터 정제를 전문으로 하는 모듈
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]',",",i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
        # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
        file_name.append(int(a) )
        file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False
    return file_name

print(image_load(test_image))

###

```

문제 478. 위의 함수의 코드를 추가해서 아래와 같이 출력되게 하시오!

```

['1.png', '2.png', '3.png', '4.png', '5.png', '6.png', '7.png',
'8.png', '9.png', '10.png', '11.png', '12.png', '13.png', '14.png',
'15.png', '16.png', '17.png', '18.png', '19.png', '20.png']

```

```

#%%

import os
import re # 데이터 정제를 전문으로 하는 모듈
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]', '', i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
        # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
        file_name.append(int(a) )
        file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False

    file_res=[]
    for k in file_name:
        file_res.append( str(k)+'\\.png') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.
    return file_res

print(image_load(test_image))

```

#%%

문제 479. 위의 함수 코드를 수정해서 아래와 같이 이름 앞에 절대경로가 붙게 하시오.

```
['e:\\images\\1.png', ['e:\\images\\2.png', ['e:\\images\\3.png', ... ]
```

#%%

```

import os
import re # 데이터 정제를 전문으로 하는 모듈
test_image = 'e:\\images'

def image_load(path):
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
    file_name = []
    for i in file_list:
        a = re.sub('[^0-9]', '', i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
        # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
        file_name.append(int(a) )
        file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False

    file_res=[]
    for k in file_name:
        file_res.append('e:\\images\\' + str(k)+'\\.png') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.

    return file_res

```

```
print(image_load(test_image))
```

# list안의 각 요소이기 때문에 양옆의 ' ' 는 자연스럽게 둘러진다.

#%%

e.g 3) 폐사진 이미지를 숫자로 변환하기 위해 필요한 파이썬 모듈을 install 하시오.

아나콘다 prompt창을 열고,

conda install opencv 를 입력해 설치

에러가 날 경우

```
pip install opencv-python
```

을 입력하면 된다.

설명 : opencv 모듈은 이미지를 파이썬에서 숫자로 변환하고 다양한 이미지처리를 할수있는 기술을 제공해주는 함수이다.

(모자이크 등)

e.x) 구글, 카카오, 네이버 지도등에서 street view라는 기능이 있는데, 거기서 사람얼굴이나 자동차 번호판을 모자이크 처리해줘야함  
이때 딥러닝 기술을 이용해서 사람얼굴, 번호판등을 판별하도록 하고 자동으로 모자이크 처리되게 할때 사용된다.

e.g 4) 위에서 설치한 opencv 모듈을 임요해 폐사진을 숫자로 변환한다.

```
#%%
```

```
import cv2 # opencv 모듈을 import 하겠다.
```

```
import os
```

```
import re # 데이터 정제를 전문으로 하는 모듈
```

```
test_image = 'e:\\images'
```

```
def image_load(path):
```

```
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
```

```
    file_name = []
```

```
    for i in file_list:
```

```
        a = re.sub('[^0-9]',",",i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것) 를 붙이고, 그 값을 null로 변경해라.
```

```
        # ^ 는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
```

```
        file_name.append(int(a) )
```

```
        file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False
```

```
    file_res=[]
```

```
    for k in file_name:
```

```
        file_res.append('e:\\images\\' + str(k)+'.png') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.
```

```
    image=[]
```

```
    for h in file_res:
```

```
        img = cv2.imread(h) # 이미지를 숫자로 변환하는 코드
```

```
        image.append(img)
```

```
    return image
```

```
print(image_load(test_image))
```

# 폐사진들이 숫자형태로 변환되어 [] 안에 담겨서 출력된다.

```
#%%
```

e.g 5) 위에 숫자로 변환한 리스트를 신경망에 넣기 위해서는 numpy 모듈의 array 형태로 제공을 해줘야 한다.

위의 리스트를 numpy array 로 변환하시오.

```
#%%
```

```
import numpy as np # 행렬연산을 쉽고 빠르게 할 수 있게 해주는 모듈
```

```
import cv2 # opencv 모듈을 import 하겠다.
```

```
import os
```

```
import re # 데이터 정제를 전문으로 하는 모듈
```

```
test_image = 'e:\\images'
```

```
def image_load(path):
```

```
    file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
```

```
    file_name = []
```

```
    for i in file_list:
```

```
        a = re.sub('[^0-9]',",",i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것) 를 붙이고, 그 값을 null로 변경해라.
```

```
        # ^ 는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
```

```
        file_name.append(int(a) )
```

```
        file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False
```

```

file_res=[]
for k in file_name:
    file_res.append('e:\\images\\' + str(k) + '.png') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.

image=[]
for h in file_res:
    img = cv2.imread(h) # 이미지를 숫자로 변환하는 코드
    image.append(img)

return np.array(img,dtype=object) # dtype=object 를 입력하라는 조언이 떠서 입력했더니 해당 메세지 사라짐.

```

```

print(image_load(test_image))

```

```

###

```

151 이미지를 숫자로 변환하는 방법 (개와 고양이)

개와 고양이 사진을 분류하는 인공지능망을 만들려면 개와 고양이 사진을 각각 2000장씩 내려 받아서 숫자로 변환하는 작업을 해줘야 한다.

사진들을 스크롤링을 해서 모아도 되지만

머신러닝 데이터 분석의 세계대회인 케글에서 개와 고양이 사진을 내려 받아서 사용하는게 좋다.(빠름.)

구글에서 kaggle 검색 -> 사이트 접속 -> register 로 가입 -> search -> cat and dog 검색 -> 파일 다운로드

파일 받아서 e:\\images2에 사진 압축 풀어서 옮겨두기

e.g 2) 개사진 30장만 e:\\images2 폴더에 넣어두시오

e.g 3) e:\\images2 라는 폴더에 있는 이미지이름을 가져오는 함수를 image\_load2라는 함수로 생성하시오.

```

###

```

```

import os
path = "e:\\images2"

def image_load2(path):
    file_list = os.listdir(path)
    return file_list

```

```

print( image_load2(path))

```

```

###

```

import numpy as np # 행렬 연산을 쉽고 빠르게 할 수 있게해주는 모듈

import cv2 # opencv 모듈을 임포트 하겠다.

import os

import re # 데이터 정제를 전문으로 하는 모듈

```

path = "e:\\images2"

```

```

def image_load2(path):
    file_list = os.listdir(path)
    file_name=[]
    for i in file_list:
        a=re.sub('[^0-9]',"",i)
        file_name.append( int(a))
    return file_name

```

```

print( image_load2(path))

```

```

###

```

e.g 5) file\_name 리스트 안의 숫자를 정렬해서 출력되게 하시오.

###

```
import numpy as np # 행렬 연산을 쉽고 빠르게 할 수 있게해주는 모듈
```

```
import cv2 # opencv 모듈을 임포트 하겠다.
```

```
import os
```

```
import re # 데이터 정제를 전문으로 하는 모듈
```

```
path = "e:\\images2"
```

```
def image_load2(path):
```

```
    file_list = os.listdir(path)
```

```
    file_name=[]
```

```
    for i in file_list:
```

```
        a=re.sub('[^0-9]',"",i)
```

```
        file_name.append( int(a))
```

```
    file_name.sort() # 요소를 정렬하는 명령어 sort() 를 쓴다.
```

```
    return file_name
```

```
print( image_load2(path))
```

###

e.g 6) 아래와 같이 절대경로와 확장자가 붙어서 출력되게 하시오.

```
['e:\\images2\\dog1.jpg', 'e:\\images2\\dog2.jpg', 'e:\\images2\\dog3.jpg', .....
```

###

```
import numpy as np # 행렬 연산을 쉽고 빠르게 할 수 있게해주는 모듈
```

```
import os
```

```
import re # 데이터 정제를 전문으로 하는 모듈
```

```
path = "e:\\images2"
```

```
def image_load2(path):
```

```
    file_list = os.listdir(path)
```

```
    file_name=[]
```

```
    for i in file_list:
```

```
        a=re.sub('[^0-9]',"",i)
```

```
        file_name.append(int(a))
```

```
    file_name.sort() # 요소를 정렬하는 명령어 sort() 를 쓴다.
```

```
# 위는 이전 문제와 동일 코드
```

```
# 이하 e.g 6) 의 답으로 추가되는 부분
```

```
    file_res=[]
```

```
    for k in file_name:
```

```
        file_res.append('e:\\images2\\dog'+ str(k)+' .jpg') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.
```

```
    return file_res # file_name -> file_res
```

```
print( image_load2(path))
```

###

e.g 7) 위의 개사진 이미지들을 opencv와 numpy 를 이용해서 숫자로 변환하고 numpy array 로 변환되게 하시오.

###

```
import numpy as np # 행렬연산을 쉽고 빠르게 할 수 있게 해주는 모듈
```

```
import cv2 # opencv 모듈을 import 하겠다.
```

```
import os
```

```
import re # 데이터 정제를 전문으로 하는 모듈
```

```
path = "e:\\images2"
```

```
def image_load2(path):
```

```

file_list = os.listdir(path) # 해당 디렉토리에 파일명을 추출한다.
file_name = []
for i in file_list:
    a = re.sub('[^0-9]', '', i) # i 값중에서 숫자가 아닌 것들은 "(싱글두개 붙인것)를 붙이고, 그 값을 null로 변경해라.
    # ^는 '아니다' 라는 뜻 // 0-9는 모든 숫자 표현가능. 그외에 다른 문자열을 null로 바꾼것
    file_name.append(int(a))
    file_name.sort() # sort 명령어의 기본값은 ascending 이다. 바꾸려면 괄호안에 ascending=False

file_res=[]
for k in file_name:
    file_res.append('e:\\images2\\dog.' + str(k) + '.jpg') # .png가 str형이기 때문에 k의 형태도 맞춰줘야 한다.

image =[]
for h in file_res:
    img = cv2.imread(h) # 이미지를 숫자로 변환하는 코드
    image.append(img)

return np.array(img, dtype=object) # dtype=object 를 입력하라는 조언이 떠서 입력했더니 해당 메세지 사라짐.

```

```

print(image_load2(path))

```

```

###

```

(점심시간 문제)

문제 480 지난번에 직접 스크롤링한 사진 20장을 e 드라이브 밑에 images3에 넣고 숫자로 변환하는 함수를 image\_load3로 생성하시오.

스크롤링 사진을 활용한 작업물을 이력서에 쓰려면 독특한 사진을 쓰는 것이 좋다.

ex) 스킨스쿠버가 취미인 학생이 있었음.

바다속 자연보호를 위해서 비닐사진과 해파리 사진을 다운받아 비닐 사진과 해파리 사진을 분류하는 인공지능망 생성하는 포트폴리오를 만들었음.

거북이가 해파리를 먹어야 되는데 비닐을 먹고 죽는 경우가 많아서 이것에 착안해서 만들었다고 함.

지금 이 파트 수업 다 끝나면 끝에 홈페이지를 만드는데, 그곳에 사진을 하나 올리면 인공지능망으로 분류가 가능하도록!

```

###

```

```

import cv2 # opencv 모듈을 임포트 하겠다.
import os
import re # 데이터 정제를 전문으로 하는 모듈

```

```

path = "e:\\images3" # 파일볼러를 위치 path 에 할당

```

```

def image_load3(path):
    file_list = os.listdir(path) # os에서 path경로의 파일을 불러와 file_list에 할당
    file_name=[]
    for i in file_list:
        a=re.sub('[^0-9]', '', i) # 0-9(=숫자)를 제외만 모든 값을 null값으로 변환해 a에 할당
        file_name.append( int(a)) # a 에 담긴 자료들을 file_name에 append( int형태로 변환하여)
    return file_name

```

```

print( image_load3(path))

```

```

###

```

152. 필수 알고리즘1 (합성곱 연산)

알고리즘 문제를 풀어야 하는 이유?

1. 지금까지 배운 파이썬 문법을 완성하는 단계 - by 알고리즘 문제 해결
2. 구글, 네이버, 카카오와 같이 대기업들에 지원하고자하면 알고리즘 문제를 많이 풀어봐야 한다.  
(바로 대기업가려할것 아니면 웹스크롤링, 데이터 분석만 잘해도 충분하긴 하다..)

1

프로그래머스 사이트에 여러 알고리즘 문제들이 올라와 있음.

하루에 하나씩만 꾸준히 풀면서 실력을 천천히 쌓아올리면 된다.

\* 합성곱 연산 알고리즘?

딥러닝의 필수 알고리즘(퍼셉트론, 합성곱 연산 알고리즘)- 매우 자주 쓰이고 중요! 반드시 알아야함.

이미지의 형상을 무시하지 않고 이미지를 그대로 인공신경망이 학습할 수 있게 해준 수학적행렬 연산.

합성곱에서 원본이미지는 학습해야할 데이터(사진) 이고, filter 는 원본이미지에서 특징을 잡아내는데 사용되는 행렬이다.

문제 481. 아래의 두 행렬을 만들고 덧셈연산을 하시오.

```
행렬 a      행렬 b
[ [ 1 2 3]   [ [ 2 0 1]
  [ 0 1 2]   [ [ 0 1 2]
  [ 3 0 1] ] [ [ 1 0 2] ]
```

###

```
import numpy as np
```

```
a = [ [ 1,2,3],[0,1,2],[3,0,1] ]
a2= np.array(a)
b = [ [ 2,0,1],[0,1,2],[1,0,2] ]
b2= np.array(b)
```

```
print(a2)
print(b2)
```

```
print(a2+b2)
```

###

문제 482. 아래의 두 행렬을 만들고 두 행렬의 원소들의 곱을 구하시오.

```
a = [ [ 1,2,3],[0,1,2],[3,0,1] ]
```

```
b = [ [ 2,0,1],[0,1,2],[1,0,2] ]
```

###

```
import numpy as np
```

```
a = [ [ 1,2,3],[0,1,2],[3,0,1] ]
a2= np.array(a)
b = [ [ 2,0,1],[0,1,2],[1,0,2] ]
b2= np.array(b)
```

```
print(a2*b2)
```

###

문제 483. 위에서 출력된 결과(3x3 행렬의 요소들) 을 모두다 더하시오.

결과 : 15가 나와줘야 함.

###



```
import numpy as np
```

```
a = [ [1,2,3],[0,1,2],[3,0,1] ]
a2= np.array(a)
b = [ [2,0,1],[0,1,2],[1,0,2] ]
b2= np.array(b)
```

```
result = a2*b2
```

```
print( np.sum(result) ) # 행렬한의 원소들의 합을 출력한다.
```

```
# 15가 나온다.
```

```
###
```

numpy 란?

python 언어에서 기본적으로 지원하지 않는 배열(array) 혹은 행렬(matrix) 의 계산을 쉽게 해주는 라이브러리이다.

머신러닝에서 많이 사용하는 선형대수학에 관련된 수식들을 python에서 쉽게 프로그래밍 할 수 있게 해준다.

문제 484. 아래의 4x4 행렬을 만드시오.

```
###
```

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
print(a2)
```

```
###
```

문제 485. 아래의 4x4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]
```

중에

```
[[1 2 3]
 [0 1 2]
 [3 0 1]]
```

만!!

즉,

행이 1부터 3까지

열도 1부터 3까지

```
###
```

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
print(a2[0:3,0:3]) # 행을 0이상에서 3미만까지 (원소 index 번호)
```

```
      # 열을 0이상에서 3미만까지 (원소 index 번호)
```

```
# numpy 사용하지 않았으면 적어도 for 문을 2개는 썼어야 함
```

```
###
```

문제 486. 아래의 4x4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

```
[[1 2 3 0]
 [0 1 2 3]
```

```
[3 0 1 2]
[2 3 0 1]]
```

에서

```
[[2 3 0]
 [1 2 3]
 [0 1 2]]
```

을 출력!

즉,

행이 2부터 0까지

열이 2부터 0까지

###

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
print(a2[0:3,1:4]) # 행을 0이상에서 3미만까지 (원소 index 번호)
                # 열을 1이상에서 4미만까지 (원소 index 번호)
```

###

문제 487. 아래의 4x4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

열

```
[[1 2 3 0] 1줄 = 1행
 [0 1 2 3] 1줄 = 2행
 [3 0 1 2] 1줄 = 3행
 [2 3 0 1]] 1줄 =4행
```

에서

```
[[0 1 2]
 [3 0 1]
 [2 3 0]]
```

을 출력해야 함.

즉, 행이 0부터 2까지

열이 0부터 2까지

###

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
print(a2[1:4,0:3]) # 행을 1이상에서 4미만까지 (원소 index 번호)
                # 열을 0이상에서 3미만까지 (원소 index 번호)
```

###

문제 488. 아래의 4x4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

열

```
[[1 2 3 0] 1줄 = 1행
 [0 1 2 3] 1줄 = 2행
 [3 0 1 2] 1줄 = 3행
 [2 3 0 1]] 1줄 =4행
```

에서

```
[[1 2 3]
 [0 1 2]
 [3 0 1]]
```

을 출력해야 함.

###

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
print(a2[1:4,1:4]) # 행을 1이상에서 4미만까지 (원소 index 번호)
                # 열을 1이상에서 4미만까지 (원소 index 번호)
```

###

문제 489. 아래의 4x4 행렬에서 빨간색으로 지정된 4개의 영역의 숫자들을 for loop문을 이용해서 모두 출력하시오.

즉, 4x4행렬에서 위에서 푼 4개의 3x3 행렬을 출력하는 것.

###

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        print(a2[i:i+3, j:j+3] )
```

###

순으로 출력이 되어야 하는데

```
[0:3],[0:3]
[0:3],[1:4]
[1:4],[0:3]
[1:4],[1:4]
```

a2[ $\begin{smallmatrix} \neg \\ \neg \end{smallmatrix}$ : $\begin{smallmatrix} \neg \\ \neg \end{smallmatrix}$ ],[ $\begin{smallmatrix} \neg \\ \neg \end{smallmatrix}$ : $\begin{smallmatrix} \neg \\ \neg \end{smallmatrix}$ ] 라고 한다면

1set

$\neg$ 에 나와야 하는것 0 또는 1 // 0 0 1 1 순으로 나와야함 -> i

$\neg$ 에 나와야 하는것 0 또는 1 // 0 1 0 1 순으로 나와야함 -> j

2set

$\neg$ 에 나와야 하는것 3 또는 4 // 3 3 4 4 순으로 나와야함 -> i

$\neg$ 에 나와야 하는것 3 또는 4 // 3 4 3 4 순으로 나와야함 -> j

i 가 0, j가 0일때 a2[0:0+3, 0:0+3] -> a2[0:3,0:3]

i 가 0, j가 1일때 a2[0:0+3, 1:1+3] -> a2[0:3,1:4]

i 가 1, j가 0일때 a2[1:1+3, 0:0+3] -> a2[1:4,0:3]

i 가 1, j가 1일때 a2[1:1+3, 1:1+3] -> a2[1:4,1:4]

j가 더 밑에 있는 for 문이기 때문에 j의 range가 모두 반복되어야 i가 1번 돌아간다.

###

문제 490. 위에서 선택한 4개의 행렬 (3x3) 과 아래의 filter 행렬(3x3)과의 원소의 곱을 출력하시오.

```

[1, 2, 3]
[0, 1, 2]
[3, 0, 1]
-----
[2, 3, 0]
[1, 2, 3]
[0, 1, 2]
-----
[0, 1, 2]
[3, 0, 1]
[2, 3, 0]
-----
[1, 2, 3]
[0, 1, 2]
[3, 0, 1]
-----

```

각 4개의 행렬 x filter의 행렬 총 4개가 나와야 함.

```

#####

```

```

import numpy as np

a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)

f = [ [2,3,4],[1,2,3],[2,0,1]]
filter = np.array(f)

# print(filter)

a3 = []
for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        a3.append(a2[i:i+3, j:j+3])

print(a3*filter)

#####

```

```

import numpy as np

a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)

f = [ [2,3,4],[1,2,3],[2,0,1]]
filter = np.array(f)

for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        print(a2[i:i+3, j:j+3]*filter )

```

이 훨씬 간단!

```

#####

```

문제 491. 위에서 출력된 3x3 행렬 4개에 대한 원소들의 합이 각각 출력되게 하시오.

```

[[ 2  6 12]
 [ 0  2  6]
 [ 6  0  1]]
의 원소들의 합 1개 -> 35

[[4 9 0]
 [1 4 9]
 [0 0 2]]

```

의 원소들의 합 1개 -> 29

```
[[0 3 8]
 [3 0 3]
 [4 0 0]]
```

의 원소들의 합 1개 -> 21

```
[[ 2  6 12]
 [ 0  2  6]
 [ 6  0 11]]
```

의 원소들의 합 1개 -> 35

```
###
```

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
f = [ [2,3,4],[1,2,3],[2,0,1]]
filter = np.array(f)
```

```
for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        print(np.sum(a2[i:i+3, j:j+3]*filter ) ) #기존 코드에 np.sum 만 앞에 붙여 준다.
```

```
###
```

문제 492. 위에서 출력된 4개의 숫자로 아래의 행렬을 만드시오.

```
[ 35 29
 21 35 ]
```

합성곱 연산의 최종 코드!

```
###
```

```
import numpy as np
```

```
a = [ [1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1] ]
a2= np.array(a)
```

```
f = [ [2,3,4],[1,2,3],[2,0,1]]
filter = np.array(f)
```

```
result=[]
for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        result.append(np.sum(a2[i:i+3, j:j+3]*filter ) )
```

```
#print(result) 4개의 값이 나옴.
```

```
result2 = np.array(result).reshape(2,2) # numpy array 의 2x2 행렬로 만드는 명령어
```

```
print(result2)
```

```
###
```

설명 : a2라는 원본 이미지(e.g 개사진) 에 filter(랜덤으로 생성한 이미지) 를 가지고 원본이미지를 스트라이드(양옆,위아래로 스캔) 하면서 특징을 잡아내 이미지를 추출(result2) 하는 것을 합성 곱이라고 한다.

문제 493. 아래의 원본 이미지 행렬(b2,5x5) 에서 filter 행렬(4x4)로 스트라이딩해서 합성곱해서 특징을 추출한 feature(3x3)를 추출하시오.

```
###
```

```
import numpy as np
```

```
a = [ [2,3,1,4,7],[3,1,6,4,3],[2,1,5,3,1],[6,2,4,1,2],[7,3,1,2,3] ]
a2= np.array(a)
```

```
f = [ [3,1,4,1],[2,3,3,4],[5,1,2,1],[6,1,3,4] ]
filter = np.array(f)
```

```
result=[]
for i in range(0,2): # i는 0,1 2개
    for j in range(0,2): # j 는 0,1 2개
        result.append(np.sum(a2[i:i+4, j:j+4]*filter ) )
```

```
result2 = np.array(result).reshape(2,2)
print(result2)
```

###

[0:4] 는 인덱스 번호 0번부터 시작해서 인덱스 번호 4이전 까지 (즉, 0,1,2,3 까지) 인덱스로 0~3이면 1번째부터 4번째

```
[0:4],[0:4]
[0:4],[1:5]
[1:5],[0:4]
[1:5],[1:5]
```

순으로 출력이 되어야 하는데

a2[⌈:⌋],[⌋:⌋] 라고 한다면

1set

⌈에 나와야 하는것 0 또는 1 // 0 0 1 1 순으로 나와야함 -> i so for i in range(0,2)

⌋에 나와야 하는것 0 또는 1// 0 1 0 1 순으로 나와야함 -> i 1번 실행마다 j 가 in range만큼 수행

2set

⌋에 나와야 하는것 4 또는 5 // 4 4 5 5 순으로 나와야함 -> i+4

⌋에 나와야 하는것 4 또는 5 // 4 5 4 5 순으로 나와야함 -> j+4

i 가 0, j가 0일때 a2[0:0+4, 0:0+4] -> a2[0:4,0:4]

i 가 0, j가 1일때 a2[0:0+4, 1:1+4] -> a2[0:4,1:5]

i 가 1, j가 0일때 a2[1:1+4, 0:0+4] -> a2[1:5,0:5]

i 가 1, j가 1일때 a2[1:1+4, 1:1+4] -> a2[1:5,1:5]

j가 더 밑에 있는 for 문이기 때문에 j의 range가 모두 반복되어야 i가 1번 돌아간다.

###

153 필수 알고리즘2 (이진 탐색)

이진탐색?

정렬된 데이터를 좌우 둘로 나눠서 원하는 값의 탐색 범위를 좁혀가며 찾는 방법

문제 494. 아래의 리스트에서 숫자 3이 있는지 순차 탐색으로 구현하시오!

있으면 숫자 3이 있습니다. 라는 메시지가 뜨게 하시오.

\* 순차탐색이란 주어진 데이터를 처음부터 차례대로 비교하면서 찾는 방법

###

```
a = [15,11,1,3,8]
```

```
for i in a:
    if i == 3:
        print('숫자 3이 있습니다.')
        break
```

```
else:
    print('숫자 3이 없습니다.')
```

###

문제 495. 위의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자가 있는지 없는지 출력되게 하시오.

###

```
num= int(input(' 검색할 숫자를 입력하세요~')) #나는 int 넣으면 에러남.. 'int' object is not iterable
a = [15,11,1,3,8]
```

```
for a in num:
    if a == num:
        print('숫자',num, '이 있습니다.')
        break
    else:
        print('숫자',num, '가 없습니다.')
```

###

문제 496. 아래의 a list에서 중앙값(median)을 찾으시오.

###

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
import numpy as np
```

```
a_n = np.array(a)
```

```
print(np.median(a_n)) #실수로 나옴. 정수로 나오게 하고 싶으면 int() 돌려주면 된다.
print(len(a)) # 15
###
```

문제 497. a list에서 첫번째 숫자부터 중앙값에 해당하는 숫자 까지만 검색하시오.

\* 중앙값 : 값을 크기순으로 나열하였을때 중앙에 위치하는 값

힌트 : a.index(요소명) 을 치면 index 번호가 나온다.

중앙값은 47 , 인덱스 순서는 7번째

So, a.index(47) 을 하면 7이 나옴

```
###
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
a_n=np.array(a)
a_m= np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력
# print(a.index(a_m)) 7이 나온다. 0부터 시작하기 때문!

print(a [ : a.index(a_m)]+1 )
```

```
# 맨처음부터 시작할거니 숫자입력 없이 바로 [: (끝지점)]
# 중앙값의 인덱스가 7까지 출력되면 [:7] 인데 이 경우 맨처음~6번째 인덱스값까지만 출력됨.
# 그래서 +1 해줌
```

# median은 리스트안의 수의 중앙값을 출력한 것

```
###
```

문제 498. 위의 a list 에서 문제 497 번에서 선택된 숫자들을 중앙값 까지 포함해서 다 지우고 아래의 결과만 출력되게 하시오.

결과 : [51,64,67,77,139,672,871]

```
###
```

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
a_n=np.array(a)
```

```
a_m= np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력
```

```
del (a [ : a.index(a_m)+1] )  
print(a)
```

```
###
```

문제 499. 위의 결과로 출력된 아래의 리스트에서 중앙값을 출력하시오.

```
###
```

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
a_n=np.array(a)
```

```
a_m= np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력
```

```
del (a [ : a.index(a_m)+1] )  
a_m2 = np.median(a)
```

```
print(a_m2)
```

```
###
```

문제 500. 지금 위에서 출력한 중앙값 77이 내가 검색하고자 하는 67보다 크다면 아래의 결과 리스트만 출력되게 하시오.

```
###
```

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
a_n=np.array(a)
```

```
a_m= np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력
```

```
del (a [ : a.index(a_m)+1] ) # 1부터 기존a 의 중앙값 까지를 삭제  
a_m2 = np.median(a) # 새로운 리스트 a 의 중앙값 구하기
```

```
if a_m2 > 67: # 만약 새로운 중앙값이 67보다 크다면
```

```
    del ( a[a.index(a_m2):] ) # 67 이후의 숫자들을 지워라
```

```
else:
```

```
    del (a[ : a.index(a_m2) ] ) # 67보다 크지 않다면 67 이전의 숫자들을 지워라
```

```
print(a) # if문을 수행하고 난 뒤의 값을 출력
```

```
###
```

(오늘의 마지막 문제)

문제 501 EBS에서 나온 영상대로 이진탐색을 구현하시오.



## binary search

a 리스트에서 검색할 숫자를 입력하세요 ~ 67

```
num= int(input(' a리스트에서 검색할 숫자를 입력하세요~'))
```

결과 : 67은 이진탐색 3번만에 검색되었습니다.

```
print(num,'은 이진탐색',cnt, '만에 검색되었습니다.')
```

a 리스트에서 검색할 숫자를 입력하세요 ~ 68

결과 : 68은 a 리스트에 없습니다.

```
###
```

1트! 67이 아닌 다른 숫자 입력하려고 할때 에러가 보기 싫게 나옴.

좀더 정확한 코드가 필요할 듯

```
###
```

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
target = int(input('찾고싶은 숫자를 입력하세요~'))
```

```
a_n = np.array(a)
```

```
a_m = np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력 #47
```

```
def binary_search(target):
```

```
    cnt= 0
```

```
    #좌측탐색 [1~47]
```

```
    if a_m < target: # 만약 중앙값(47)이 67보다 작다면
```

```
        cnt=cnt+1
```

```
        del(a[ : a.index(a_m)+1 ]) #왼쪽 [1~47삭제], 새로운 a는 [51~871]
```

```
        a_m2 = np.median(a) #새로운 a의 중앙값은 77
```

```
        print(target, '은 a 리스트에 없습니다.',cnt,'회 탐색')
```

```
    else:
```

```
        cnt=cnt+1
```

```
        print(target,'은 이진탐색',cnt,'만에 검색되었습니다.')
```

```
    #우측탐색 [77~871]
```

```
    if a_m2 > target: # 만약 중앙값(77)이 67보다 크다면
```

```
        cnt=cnt+1
```

```
        del ( a[a.index(a_m2): ] ) # 오른쪽 77(포함) 이후의 숫자들을 지워라
```

```
        a_m3 = np.median(a) # 새로운 a[51~67]의 중앙값은 64
```

```
        print(target, '은 a 리스트에 없습니다.',cnt,'회 탐색')
```

```
    else:
```

```
        cnt=cnt+1
```

```
        print(target,'은 이진탐색',cnt,'만에 검색되었습니다.')
```

```
    #좌측탐색 [51~67]
```

```
    if a_m3 < target: #만약 중앙값(64) 가 67보다 작다면
```

```
        cnt=cnt+1
```

```
        del(a[ a.index(a_m3)+1 ]) # [51,64] 삭제, 새로운 a는 [67]
```

```
        print(target,'은 이진탐색',cnt,'만에 검색되었습니다.')
```

```
    else:
```

```
        cnt=cnt+1
```

```
        print(target, '은 a 리스트에 없습니다.',cnt,'회 탐색')
```

```
print(binary_search(target))
```

```
###
```

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
target = 67 #int(input('찾고싶은 숫자를 입력하세요~'))
```

```

cnt=0
while True:
    a_n = np.array(a)
    a_m = np.median(a_n) # array 한 a list 의 숫자중 중앙값을 출력
    #좌측탐색 [1~47]
    if a_m < target: # 만약 중앙값(47)이 67보다 작다면
        cnt=cnt+1
        del(a[ : a.index(a_m)+1 ] ) #왼쪽 [1~47삭제]
        print('이진탐색',cnt,'회 검색했습니다.')
        #print(a) # [51,64,67,77,139,672,871] , a_m = 77
        #print(a_m)

    #우측탐색 [51~871] # 새로운 중앙값은 77
    elif a_m > target: # 만약 중앙값(77)이 target(67)보다 크다면
        cnt=cnt+1
        del ( a[a.index(a_m): ] ) # 77부터 끝까지(871) 삭제 a=[51,64,67] , 중앙값은 64
        print('이진탐색',cnt,'회 검색했습니다.')
        #print(a) # [51,64,67,77,139,672,871] , a_m = 77
        #print(a_m)

    #좌측탐색 [51,64,67] , a_m = 64
    elif a_m == target:
        print(target,'은 이진탐색',cnt,'만에 검색되었습니다.')
        break
    else:
        print(target, '은 a 리스트에 없습니다.',cnt,'회 탐색')
        break

```

###

while 문의 무한루프문 (while True) 이 실행된 것.:

```

while True:
    수행할 문장1
    수행할 문장2
    .
    .
    .
    break # 을 해줘야 멈춘다.

```

의 경우 True이기 때문에 while 문 이하의 실행문이 무한반복되어 실행된다.

```
import numpy as np
```

```
a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
```

```
target = int(input('찾고싶은 숫자를 입력하세요~')) # 리스트의 숫자중 이진탐색법으로 찾으려는 숫자.
```

```
cnt=0
```

```
while True: # 무한루프문
```

```
    a_n = np.array(a) # a list 값을 array 하게 변형
```

```
    a_m = np.median(a_n) # a 값중에 median(중앙값) 을 출력해 a_m이라 할당
```

```
    #조건1
```

```
    if a_m < target: # 만약 중앙값(47)이 67보다 작다면
```

```
        cnt=cnt+1
```

```
        del(a[ : a.index(a_m)+1 ] ) #왼쪽 [1~47삭제]
```

```
        print('이진탐색',cnt,'회 검색했습니다.')

```

```
    중앙값이 < target(67) 이면 (즉, 중앙값이 67보다 왼쪽에 위치하면)
```

```
    cnt 에 1을 더하고
```

```
    0번째 인자값의 수부터 중앙값의 인덱스+1까지(= 1부터 중앙값까지) 삭제한다.
```

```
    a = [51,64,67,77,139,672,871] 이 된다.

```

#조건2

elif a\_m > target: # 만약 중앙값(77)이 target(67)보다 크다면

cnt=cnt+1

del ( a[a.index(a\_m):] )

print('이진탐색',cnt,'회 검색했습니다.')

조건1을 만족하지 않으면 조건2로 넘어와서 시행

중앙값 > target(67) 이면 (즉, 중앙값이 target보다 우측에 위치한다면)

cnt 에 1을 더하고

중앙값 우측에 있는 모든 값을 삭제한다.

a = [51,64,67] 이 된다.

#조건3

elif a\_m == target:

print(target,'은 이진탐색',cnt,'만에 검색되었습니다.')

break

중앙값과 target의 값이 동일하다면 (즉, target값이 중앙값이라면) 1:1로 비교해서 일치해야함

주어진 문장을 출력하고

while 무한루프문을 멈춘다.

else:

print(target, '은 a 리스트에 없습니다.',cnt,'회 탐색')

break

중앙값과 target의 값이 동일하지 않다면 // 1:1로 비교해서 일치하지 않아야함

주어진 문장을 출력하고

while 무한루프문을 끝낸다.:

조건2에서 넘겨진 a의 값은 1:1비교를 할 수 있는 1개의 값이 아닌 3개의 값이다.

따라서 조건3의 elif 문이나 else 문 모두 수행 조건에 부합하지 않으므로

다시 조건 1로 넘어간다.

index 번호 0 1 2

☐ ☐ ☐

a = [51, 64, 67] 일때 조건1을 다시 적용시키면

중앙값이 < target 이면 (즉, 중앙값이 67보다 왼쪽에 위치하면)

cnt 에 1을 더하고

0번째 인자값의 수부터 중앙값까지 삭제 (즉, 중앙값 좌측의 수는 모두 삭제)

a = [67] 이 된다.

이 결과는 조건2 ( 67 > 67이면 ) 에 적용되지 않으므로

조건 3으로 넘어온다.

그래서 조건 3에 부합 ( 67 == 67 ) 하므로

지정된 문구를 출력하고

while 무한 루프문을 break한다.:

###

한결이 코드

###

import numpy as np

a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]

b = (input('a 리스트에서 검색할 숫자를 입력하세요 ~'))

cnt = 1

finish=0

```

if int(b[-1]) in [1,3,6,7,8,0]: # '은/는' 불편해서 만들
    d = '은'
else:
    d = '는'
b = int(b)

```

```

while True:
    a_m = round((len(a)-1)/2)
    if b == a[a_m]:
        print(f'{b}{d} 이진탐색 {cnt}번만에 검색되었습니다.')
        break

    elif b > a[a_m]:
        cnt +=1
        del(a[:a_m+1])

    else:
        cnt +=1
        del(a[a_m:])

    if len(a) == 0:
        print(f'{b}{d} 검색되지 않았습니다.')
        break

```

###  
코드 이해를 위한 해석

```
import numpy as np
```

```

a = [1,7,11,12,14,23,33,47,51,64,67,77,139,672,871]
b = (input('a 리스트에서 검색할 숫자를 입력하세요 ~'))

```

```

cnt = 1
finish=0

```

```

if int(b[-1]) in [1,3,6,7,8,0]: # '은/는' 불편해서 만들
    d = '은'
else:
    d = '는'

```

# b[-1] 은 1자리수일 경우 0번째 인자값 = -1번째 인자값  
 # 2자리수 이상일 경우 -1번째 값은 맨뒤의 숫자  
 # 만약 b 에입력한 숫자의 끝자리가 1,3,6,7,8,0 이면 d = '은' 이라고하고  
 # 해당 범위안에 없는 수이면 d = '는' 이라 한다.  
 # 여기서 d에 문자를 할당하는 것은 파이썬의 문자열 포매팅 기능중 f 문자열 포매팅에 사용하기 위함  
 # f문자열 포매팅은 print 할때 한글 쓸때 번거롭게 '' 둘러주고 ,침표 쓰고 하는일없이 원활하게 작성하기 위해 사용

```
b = int(b) #int 절을 돌려 숫자화 시켜준다.
```

```

while True:
    a_m = round((len(a)-1)/2) # a가 15개이므로 중앙값을 구하려면 전체 개수 -1 /2를 해줘야 한다.

    #조건1
    if b == a[a_m]:
        print(f'{b}{d} 이진탐색 {cnt}번만에 검색되었습니다.')
        break
    # 검색할 숫자 = 중앙값 이면 (리스의 숫자가 딱 1개 ) 지정한 문자열 출력

```

```

#조건2
elif b > a[a_m]:
    cnt +=1
    del(a[a_m+1])
# 조건1을 수행하지 못하면 조건2 수행
# 검색할 숫자 > 중앙값 (즉, 검색할 숫자가 중앙값 우측에 있으면)
# 중앙값(포함) 왼쪽의 모든 수를 지운다.

#조건3
else:
    cnt +=1
    del(a[a_m:])
#조건 2도 수행하지 못하면 조건3 수행
# 검색할 숫자 < 중앙값 (즉, 검색할 숫자가 중앙값 좌측에 있으면)
# 중앙값(포함) 오른쪽의 모든 수를 지운다.

# 조건 1~4를 계속 반복해서 조건 1을 만족하면
# 지정된 문자열을 출력하고 break

# 조건4
if len(a) == 0:
    print(f'{b}{d} 검색되지 않았습니다.')
```

64

```

    break
# a 리스트에 없는 값을 찾으려고 입력했다면, 조건1~3을 무한 수행해서 a 의 모든 값이 삭제된다.
# a list에 든 숫자가 0개가 되면 지정된 문장을 출력하고 break
```

20.12.23 파이썬 강의 Day 19 (문제502~문제519)

#### 154. 필수 알고리즘3 (버블정렬)

버블정렬이란

서로 인접한 두 요소의 크기를 서로 비교하여 순서에 맞지 않는 요소를 인접한 요소와 서로 교환하여 정렬하는 정렬방법을 bubble sort (버블정렬) 이라 한다.

즉,

(오름차순 기준) 서로 인접한 두 요소의 크기를 비교해서 큰 수를 뒤로 보내는 것

문제 502. 아래의 리스트를 만들고 첫번째 요소와 두번째 요소의 순서를 변경하시오.

```
a = [10,5,20,9,8]
```

결과 = [5,10,20,9,8] 즉, 0번째 인자와 2번째 인자의 위치가 바뀌어야 함.

```
###
```

```
a = [10,5,20,9,8]
```

```
temp = a[1] # 1번째 요소(5) 을 temp에 할당(임시저장)
```

```
a[1] = a[0] # 1번째 요소를 0번째요소(10) 라고 정의 (변경)
```

```
a[0] = temp # 0번째 요소를 임시저장했던 1번째 요소로 변경
```

```
print(a)
```

```
###
```

문제 503. 아래의 a 리스트의 첫번째 요소와 두번째 요소의 크기를 비교해서 첫번째 요소의 숫자가 두번째 요소의 숫자보다 크다면 두개를 바꿔치기 해라

결과 : [5, 10, 20, 9, 8]

```
###
```

```

a = [10,5,20,9,8]

if a[0] > a[1]:
    temp = a[1] # 1번째 요소(5) 을 temp에 할당(임시저장)
    a[1] = a[0] # 1번째 요소를 0번째요소(10) 라고 정의 (변경)
    a[0] = temp # 0번째 요소를 임시저장했던 1번째 요소로 변경

print(a)

#%%%

```

문제 504. 문제 503번 코드에 for loop 문을 넣어서 버블정렬 하시오!

결과 = [4, 3, 2, 1, 5, 7, 8, 10] 나올때 까지만!

```

#%%%

a = [5,4,3,2,1,8,7,10]

for i in range(0,6):
    if a[i] > a[i+1]:
        temp = a[i+1] # 1번째 요소(5) 을 temp에 할당(임시저장)
        a[i+1] = a[i] # 1번째 요소를 0번째요소(10) 라고 정의 (변경)
        a[i] = temp # 0번째 요소를 임시저장했던 1번째 요소로 변경
print(a)

#%%%

```

(버블정렬 완성 코드)

문제 505. 위의 코드를 이용해서 버블정렬하는 함수를 아래와 같이 생성하시오.

```

print(bubble_sort(a))

#%%%
a = [5,4,3,2,1,8,7,10]

def bubble_sort(a):
    length = len(a) -1
    for i in range(length):
        for j in range(length-1):
            if a[j] > a[j+1]:
                temp = a[j+1]
                a[j+1] = a[j]
                a[j] = temp
        return a

print(bubble_sort(a))

#%%%

```

#좀더 간결하게 코드 정리 + 소라코드 참조해서 break 추가

```

a = [5,4,3,2,1,8,7,10]

def bubble_sort(a):
    length = len(a) -1
    cnt = 0
    for i in range(length):
        for j in range(length-i):
            if a[j] > a[j+1]:
                a[j],a[j+1] = a[j+1],a[j]
                cnt +=1
        if cnt ==0:
            break
    return a

```

```

print(bubble_sort(a) )
#bubble_sort(a)
#%%%

a[j],a[j+1] = a[j+1],a[j]
는 좌우가 대칭 하여 매칭 됨
a[j] = a[j+1]
a[j+1] = a[j]
라고 2줄에 쓸 것을 한 줄로 줄인 것.

#%%%
# 한결이 코드

def bubble_sort(a):
    n = 1 # while loop문 실행시키기 위해 n에 1 할당
    j = -1 # 한번 정렬한 숫자는 다시 계산하지 않게 하기 위해 -1 할당
    while n!=0:
        n = 0 # 실행문 1
        j += 1 # 실행문 2
        for i in range(len(a)-1): # len(a)-1 = 7 --> 0~6 범위
            for j in range(len(a)-i-1): # 과도한 계산을 막기 위해서 하나씩 빼줌 정렬된 수는 안봐도 되니까
                if a[j] > a[j+1]:
                    a[j],a[j+1] = a[j+1],a[j] #핵심 코드! 1줄로 줄인 것
                    n +=1
            return a

a=[5,4,3,2,1,8,7,10]

print(bubble_sort(a) )

# 정렬이 완성되었을때 멈추는 코드는 아닌 듯
#%%%

# 세원이형 코드 // 결이것 참조해서 수정 보완하여 완성!
# 정렬이 완료되는 시점에 추가적인 작업 없이 멈춘다.

a = [5,4,3,2,1,8,7,10]

def bubble_sort(a):
    n=1 # 아래의 while 루프문 조건이 실행되게 하기 위해 1이라 설정
    while n>0: # n = 0 이 되면 정렬이 완료된것임
        n=0
        b=0
        for i in range(len(a)-1):
            b+=1
            if a[i] >= a[i+1]:
                temp = a[i+1]
                a[i+1] = a[i]
                a[i] = temp
            n+=1 # while문이어서 if 조건을 실행할 조건이 되지 않으면 처음으로 돌아가 n=0이됨 -> 멈춤
        print(a,n,b)
    return a

print(bubble_sort(a))

# 과정이 다 보여서 좋다! + 쓸데없이 반복되지 않고 정렬이 완료된 시점에 정지됨.

#%%%

# for 문으로 구현할 경우 (소라 코드)

a = [5, 4, 3, 2, 1, 8, 7, 10]
def bubble_sort(num):
    n = len(num) # 8

    for k in range(n): # == in range(0,8) == 0 ~ 7

```

```

cnt = 0 # 실행문 1
for i in range(n-1): # 실행문 2(교환작업)
    if num[i] > num[i+1]:
        temp = num[i+1]
        num[i+1] = num[i]
        num[i] = temp
    cnt = 1 # 실행문2가 시행되면 cnt=1, 시행되지 않으면 실행문 1로 돌아가 cnt=0
if cnt == 0: # 실행문 3
    break
return num

print(bubble_sort(a))

###

```

#### 155. 필수 알고리즘4 (탐욕알고리즘) - Greedy algorithm

머신 러닝 배울때 의사결정트리를 구현할 때 사용하는 알고리즘

- \* 의사결정트리는 과정에 대한 설명이 너무 잘 구현되기 때문에 데이터 분석가들이 애용함.
- \* 인공지능망은 고객입장에서 이해하고 납득하기가 어려움(결과 자체는 너무 좋음)

탐욕 알고리즘이란

매 순간마다 최선의 선택을 하는 것이다.  
즉 선택에 기로에 놓였을때마다 가장 좋다고 생각되는 것을 선택해 나가며 최종적인 해답을 구하는 알고리즘

이 알고리즘을 설계할때 주의할 점은 '전체'를 고려할 것이 아니라 문제를 부분적으로 나누어, 나누어진 문제에 대한 '최적의 해답'을 구하도록 해야 한다는 점이다. (국지전? 느낌)

e.g)  
14원의 잔돈을 나눠줘야 하는데 잔돈의 종류가 10원, 7원, 1원이 있다.  
잔돈을 가장 빨리 줄 수 있는 방법은 무엇인가?  
-> 7원 2개 주는 것 (7원 1번, 7원 1번 씩 총 2번주는 것)

탐욕알고리즘은 10원 1개 / 7원 0개 / 1원 4개로 주는게 가장 빨리주는 법이라고 함.

1트  
가장 큰돈(10원) 먼저 주고(fin), 4원남음

2트  
그 다음 줄 수 있는 돈 중에 가장 빨리 줄 수 있는 것. (7원>4원 so, 7원 지급불가, 1원 4번이 최선)

If, 잔돈을 딱 1번만 받을 수 있는것이라면 당연히 7원보다는 10원을 받는게 더 최선!

###

문제 506. 14를 10으로 나눈 몫을 출력하시오.  
###

```
print(14/10) # 1.4
```

```
result1, result2 = divmod(14,10)
```

```
print(int(14/10)) # 실수로 나오게 하기 위함
print(14//10) # 실수로 나눔!
```

```
# * // 기억하기!
```

###  
문제 507. 14를 10으로 나눈 나머지 값을 출력하시오.  
###

```
print(14%10) # 4가 나옴.
```



```
###
```

문제 508. 숫자를 물어보게 하고, 숫자를 10으로 나눈 몫과 숫자를 10으로 나눈 나머지값을 출력하게 하시오.

```
###
```

```
num = int( input ('숫자를 입력하세요~') )
```

```
print('몫은', num // 10)
```

```
print('나머지는', num % 10)
```

```
###
```

문제 509. 아래에 잔돈리스트를 만들고 그 리스트의 첫번째 요소로 나눈 몫과 나머지 값이 출력되게 하시오.

```
###
```

```
coin = [10,7,1]
```

```
num = int( input ('숫자를 입력하세요~') )
```

```
print('몫은', num // coin[0])
```

```
print('나머지는', num % coin[0])
```

```
###
```

문제 510. 위의 코드를 함수로 만들어서 실행되게 하시오.

greedy() 라고 함수 입력하면

잔돈을 입력하세요~ 14<-입력

몫은 1

나머지는 4

라고 나오도록!

```
###
```

```
def greedy():
```

```
    coin = [10,7,1]
```

```
    num = int( input ('잔돈을 입력하세요~') )
```

```
    print('몫은', num // coin[0])
```

```
    print('나머지는', num % coin[0])
```

```
greedy()
```

```
###
```

(점심시간 문제)

필수알고리즘 4 탐욕 알고리즘

문제 511. 탐욕알고리즘을 파이썬으로 구현하시오.

```
greedy()
```

잔돈을 입력하세요~14

10원 동전 1개, 7원동전 0개, 1원동전 4개로 줍니다.

잔돈을 입력하세요~107

10원 동전 10개, 7원동전 1개, 1원동전 0개로 줍니다.

라고 출력되도록!

```
###
```

#내가 하고 싶었던 코드

```
def greedy():
```

```

coin = [ 10, 7, 1 ]
num = int(input('숫자를 입력하세요 ~'))
a = num // coin[0]    # 잔돈을 10으로 나눈 몫
b = num % coin[0]     # 잔돈을 10으로 나눈 나머지

if b >= 7:    # 잔돈을 10으로 나눈 나머지가 7보다 크면
    print('10원 동전', a, '개, 7원 동전', b // coin[1], '개, 1원 동전', b % coin[1], '개로 줍니다.')
        # 나머지를 7로 나눈 몫 : 7원 동전 개수,
        # 나머지를 7로 나눈 나머지 : 1원 동전 개수
else:        # 잔돈을 10으로 나눈 나머지가 7보다 작으면
    print('10원 동전', a, '개, 7원 동전', 0, '개, 1원 동전', b, '개로 줍니다.')
        # 나머지를 1원 동전에 할당

greedy()

#####

def greedy() :
    coin = [10,7,1]
    coin = sorted(coin, reverse = True) # 큰값을 맨 앞으로 오게 정렬! (그리드는 가장 큰값부터 선택)

    num = int(input('숫자를 입력하시오 : '))
    a = []
    for i in range(3): # 0 ~ 2      i=0 -> 10      i=1 -> 7      i=2 -> 1
        result = num//coin[i] # 14//10 = 1 -> result // num=4, 4//7 = 0 -> result // num=4, 4//1 = 4 -> result
        num = num%coin[i] # 14%10 = 4 -> num // 4%7 = 4 -> num // num=4, 4%1 = 0 -> num
        a.append(result) # a = result = 1 이 됨. // a = result = 4 가 됨 /// a = result = 0 이 됨.

    print('10원 동전', a[0], '개, 7원 동전', a[1], '개, 1원 동전', a[2], '개로 줍니다.')
greedy()

#####

```

## 156. 필수 알고리즘5 (재귀알고리즘)

재귀 알고리즘은 처음에는 이해하기가 어려운 알고리즘이지만 많이 연습해서 잘 알아두면 loop문을 최소화하면서 코드를 간단하게 작성할 수 있는 알고리즘이다.

### 1. 재귀 함수란?

복문 + 스택구조가 결합된 함수



후입선출 (가장 나중에 추가된 데이터가 가장 먼저 삭제되는 구조)

### 2. 재귀 함수의 특징?

재귀 함수는 함수내에서 다시 자기 자신을 호출한 후 그 함수가 끝날때까지 함수 호출 이후의 명령문을 수행하지 않는다.

### 3. 함수내에서 다른 함수를 호출 하는 예제

```

#####

def hap(a,b):
    return (a+b)

def gob(a,b):
    return (a*b)

def hap_gob(a,b):
    k = hap(a,b)
    m = gob(a,b)
    return k+m

print(hap_gob(2,3))

```

# 함수내에서 함수를 불러와서 쓸 수 가 있다.

###

4. 숫자를 입력하면 1부터 해당 숫자까지의 합을 출력하는 함수를 생성하시오.

print( add\_func(5)) # 15가 나오도록.

###

```
def add_func(n)    :  
    cnt = 0  
    for i in range(1,n+1):  
        cnt = cnt+i  
    return cnt
```

print(add\_func(5))

###

5. 위의 add\_func 함수를 재귀함수로 구현하시오.

(재귀함수를 사용하면 loop문을 상용하지 않아도 된다.)

###

```
def add_func(n):  
    if n == 0:  
        return 0  
    else:  
        return n+ add_func(n-1) # 재귀: 자기자신의 함수가 그 함수속에 들어가 있음.
```

print(add\_func(5)) # 15 가 나옴.

###

if 문의 조건이 while 쓸때 break 불도록 if문을 만드는 부분과 같은듯?

```
def add_func(n): # 5가 n에 입력  
    if n == 0:  
        return 0  
    else: # 5 != 0 이므로 else 조건 수행  
        return n+ add_func(n-1)
```

print(add\_func(5)) # 15 가 나옴.

```
# 5+ add_func(4) 가 실행  
# add_func(4)도 if문 해당없음. 4+add_func(3)실행  
# 3 + add_func(2) 실행  
# 2 + add_func(1) 실행 -> 리턴받은 0  
# 1 + add_func(0) 실행 -> 1 + 리턴받은 add_func(0) == 0 을 연산하면 1 -> add_func(1) 에 값을 리턴..  
# add_func(0) -> 0이 바로 직전 함수에 리턴
```

###

문제 512. factorial (!) 을 수행하는 함수를 만드시오.

print(factorial(5)) -> 5 \* 4 \* 3 \* 2 \* 1 = 120이 나오도록 하시오.

###

```
def factorial(n):  
    if n ==1:  
        return 1  
    else:  
        return n*factorial(n-1)
```

```
print(factorial(5))
```

```
###
```

5 \* factorial(4) -> 12 \* 5 == 120 -> 결과값

↓

4\*factorial(3) -> 4\*3

↓

3\*factorial(2) -> 3\*1

↓

2\*factorial(1) #1! 은 1

나중에 들어온 것이 먼저 실행된다.

문제 513. 위의 factorial() 을 재귀함수 사용하지 않고 for loop문으로 구현하시오.

```
###
```

```
def factorial(n):
```

```
    cnt = 1
```

```
    for i in (1,n+1):
```

```
        cnt = cnt*i
```

```
    return cnt
```

```
print(factorial(5))
```

```
###
```

```
n=5,
```

```
for i in (1,6) -> i = 1,2,3,4,5
```

```
i =1 , cnt = 1 -> cnt*i = 1*1 = 1 , cnt = 1
```

```
i =2 , cnt = cnt*i = 1*2 = 2 , cnt = 2
```

```
i =3 , cnt = cnt*i = 2*3 = 6 , cnt = 6
```

```
i =4 , cnt = cnt*i = 6*4 = 24, cnt = 24
```

```
i =5 , cnt = cnt*i = 24*5 = 120
```

재귀를 이용할때 2가지 장점

1. loop문을 복잡하게 이용하지 않아도 된다.

2. 코드가 더 간결해진다.

```
###
```

문제 514. 오라클의 power 함수를 파이썬으로 구현하시오.

SQL> select power(2,3) from dual -> 8 (2의 3승)

```
print( power(2,3))
```

```
###
```

```
def power(n1,n2):
```

```
    cnt =1
```

```
    for i in (1,n2+1):
```

```
        cnt = cnt*n1
```

```
    return cnt
```

```
print(power(2,3))
```

```
###
```

문제 515. 위의 power 함수를 재귀 함수로 구현하시오.

```
###
```

```
def power(x,n):
```

```
    if n == 1: # 재귀를 종료시킬 코드
```

```
        return 1
```

```

else: #재귀할 코드
    return x*power(x,n-1)

print(power(2,3))
###

power (2,3)

n != 1 ->else
2*power(2,2) -> 4
↓
x=2, n=2 !=0 이므로 else:
2*power(2,1) -> 2
↓
1 == 1 이성립 , return 1 // 역으로 돌아서

###

```

문제 516. 구구단 2단을 아래와 같이 출력하는 함수 를 만드시오.

```

print(multi_table_2dan(9))
2 x 1 =2
2 x 2 = 4
.
.
.2 x 9 = 18
###

```

```

def multi_table_2dan(n):
    for i in range(1,n+1):
        print( '2x', i , '=' , 2*i )

print(multi_table_2dan(9) )

```

###

문제 517. 두 숫자를 입력해서 함수를 실행하면 두 숫자의 최대공약수가 출력되는 함수를 생성하시오.

```

print(gcd(16,24)) -> 8
###

```

```

def gcd(n1,n2):
    for i in range(min (n1,n2),0 , -1) : # n1, n2 중 작은수를 입력하고, 0까지 역순으로 입력하겠다.
        if n1 % i == 0 and n2%i == 0:
            return i

```

```

print(gcd(16,24))

```

###

(필수 알고리즘 5) 재귀함수

문제 518. 문제 517을 재귀함수로 구현하시오.

###

```

def gcd(n1,n2):
    if n2 == 0:
        return n1
    else:
        return gcd(n2, n1%n2)

```

```

print(gcd(16,24))

```

###

n1 = 16, n2 = 24

유클리드 호제법 활용

a, b에 대해서 a를 b로 나눈 나머지를 r이라 하면(단, a>b), a와 b의 최대공약수는 b와 r의 최대공약수와 같다.

즉, n1 < n2 일때, n1, n2의 최대공약수 = n2, n2/n1의 나머지 사이의 최대공약수와 같음.

예시)

1071과 1029의 최대공약수를 구하면,

1071은 1029로 나누어떨어지지 않기 때문에, 1071을 1029로 나눈 나머지를 구한다. >> 42

1029는 42로 나누어떨어지지 않기 때문에, 1029를 42로 나눈 나머지를 구한다. >> 21

42는 21로 나누어떨어진다.

$n2 = 24 \neq 0$  이므로 else 조건 적용

1)  $\text{gcd}(24, 16\%24) \rightarrow n2 = 16\%24 \rightarrow 16$

2)  $\text{gcd}(24, 16) // n2 = 16 // 16 \neq 0$  이므로 else 조건  $\text{gcd}(16, 24\%16) \rightarrow n2 = 24\%16 \rightarrow 8 \rightarrow \text{gcd}(16, 8)$

3)  $\text{gcd}(16, 8) // n2 = 8 // 8 \neq 0$  이므로 else 조건  $\text{gcd}(8, 16\%8) \rightarrow n2 = 16\%8 \rightarrow 0 \rightarrow \text{gcd}(8, 0)$

4)  $\text{gcd}(8, 0) // n2 = 0$  이므로 if 조건 만족 n1인 8 return

###

# 준환님 코드

def gcd(a,b):

big = max(a,b) # 두수 중 큰 수를 big에 할당

small = min(a,b) # 두수 중 작은 수를 small에 할당

if small == 0: # 두수중 작은수가 0이라면

return big # 두수 중 큰 수를 리턴

else:

return gcd(small, big%small)

print (gcd(16,24))

print (gcd(14,35))

내가 짠 코드의 연장선. 이해는 좀더 쉽다.

###

# 세원이형 코드

def gcd(a,b,c=True): # c값에 디폴트 일때 트루로 시작합니다.

a,b = min(a,b),max(a,b) # 앞 쪽에 항상 작은수가 되어야 최소 계산을 합니다.

i=a #24

#print(a+c-1,b,i,c)

if (a+c-1)%i == 0 and b%i ==0: # c를 추가해줘서 변화지 않게 합니다.

return i

else:

return gcd(a-1,b,c+1)

print (gcd(24,36))

###

###

157. 필수 알고리즘6 (LRU 알고리즘)

LRU = Least Recent Used

LRU 알고리즘이란 Oracle DATABASE 의 메모리 관리를 효율적으로 하기위해 고안된 대표적인 알고리즘

최신데이터를 메모리에 유지시키고, 오래된 데이터는 메모리에서 내보내게 하는 알고리즘.

SQL>

```
select sal
  from emp
 where ename = 'SCOTT'; # 3000
```

을 수행한다고 할때

메모리에서 조회를 하면 1초가 걸림

메모리에 자료가 없어서 디스크에서 조회하면 5초가 걸린다.(예시)

즉, 검색속도 = 메모리 > 디스크

오라클은 처음에 1번 읽을때만 디스크에서 조회하도록 하고,

그 이후에는 빠르게 데이터를 읽어오게 하기 위해 해당 자료들을 메모리에 올려둬.

cache hit = 메모리에서 데이터를 조회하는 것

cache miss = 메모리에 데이터가 없어서 디스크에서 데이터를 조회하는 것

한번 디스크에서 읽은 데이터를 메모리에 올려놓고 메모리에서 빠르게 데이터를 조회할 수 있도록

LRU 알고리즘을 구현해서 만든 소프트웨어 가 오라클!

근데 이 메모리 공간이 한정적이다보니 무한하게 데이터를 올릴수는 없으므로, 오래된 데이터는 새로운 데이터가 유입  
될때마다 점점 밀려나가 메모리에서 빠져나가게 되고, 최신데이터가 그 빈자리를 채운다.

최근에 내가 검색한 데이터 = 다시 검색할 확률이 높은 데이터(라 가정해). 메모리에 오래두도록 하고

예전에 검색한 데이터는 = 다시 검색할 확률이 낮은 데이터(라 가정해). 메모리에서 빠져나가게 한다.

선입 선출! 먼저 들어오면 먼저 나감! -> '큐' 구조

스택 : 후입선출

큐 : 선입선출

(오늘의 마지막 문제)

문제 519. 구구단 2단을 재귀함수로 출력하시오.

```
print( multi_table_2(9))
```

```
2 x 1 =2
```

```
2 x 2 = 4
```

```
.
```

```
.
```

```
.
```

```
2 x 9 = 18
```

```
def multi_table_2dan(n):
```

```
    for i in range(1,n+1):
```

```
        print( '2x', i , '=', 2*i )
```

```
###
```

```
def multi_table_2(n):
```

```
    if n !=0:
```

```
        multi_table_2(n-1) #스택구조 = 후입선출
```

```
        print('2x %d = %d' %(n,2*n) )
```

```
        # multi_table_2(n-1) #스택구조 = 후입선출 재귀함수가 여기오면 역순으로 9단부터 출력된다.
```

```
multi_table_2(9)
```

```
###
```

```
# 모든 구구단에 적용가능
```

```
def multi_table(n,i=1): # 출력할 단수, 함수에 쓰기 위해 필요한 i = 1 미리 입력
```

```
    if i == 10: # 1~9단만 있으므로 i == 10이면 END 라고 출력되도록!
```

```

#print('END')
return #아무것도 출력하고 싶지 않으면 return 만 작성
else:
    print(f'{n} x {i} = {n*i}') # f-string 포매팅으로 출력
    multi_table(n, i+1)

```

multi\_table(2) #함수에서 i값을 default 로 입력해두었으므로 입력하지 않아도 된다.  
multi\_table(9)

```

###
multi_table_2(2) n=2로 함수 안에서 고정

```

1) i 는 함수 시작할때 1이라고 default 값으로 지정-> if 조건에 부합하지 않음 -> else로 이동

```

n=2, i=1 이므로 2x1 = 2 출력
multi_table_2(2,i+1) i+1 = 2
multi_table_2(2,2)
└─┘

```

```

2) multi_table_2(2,2) -> n=2, i=2 // i !=10 이므로 else 조건으로 이동
n=2, i=2 이므로 2x2 = 4 출력
multi_table_2(2,i+1) i+1 = 3
multi_table_2(2,3)
└─┘

```

```

3) multi_table_2(2,3) -> n=2, i=2 // i !=10 이므로 else 조건으로 이동
n=2, i=3 이므로 2x3 = 6 출력
multi_table_2(2,i+1) i+1 = 4
multi_table_2(2,4)
└─┘

```

```

4) multi_table_2(2,4) -> n=2, i=4 // i !=10 이므로 else 조건으로 이동
n=2, i=3 이므로 2x4 = 8 출력
multi_table_2(2,i+1) i+1 = 5
multi_table_2(2,5)
.
.
.

```

```

9) multi_table_2(2,9) -> n=2, i=9 // i !=10 이므로 else 조건으로 이동
n=2, i=9 이므로 2x9 = 18 출력
multi_table_2(2,i+1) i+1 = 10
multi_table_2(2,10)
└─┘

```

```

10) multi_table_2(2,10) -> n=2, i=10 이므로 if조건 만족
'END' 출력

```

12월 28일

159 필수 알고리즘8(비밀지도)

카카오 블라인드 시험문제(비밀지도)

<http://cafe.daum.net/oracleoracle/SfZF/1238>

1. 비밀 지도(난이도: 하)

네오는 평소 프로도가 비상금을 숨겨놓는 장소를 알려줄 비밀지도를 손에 넣었다. 그런데 이 비밀지도는 숫자로 암호화되어 있어 위치를 확인하기 위해서는 암호를 해독해야 한다. 다행히 지도 암호를 해독할 방법을 적어놓은 메모도 함께 발견했다.

지도는 한 변의 길이가 n인 정사각형 배열 형태로, 각 칸은 “공백”(“ ”) 또는 “벽”(“#”) 두 종류로 이루어져 있다.

전체 지도는 두 장의 지도를 겹쳐서 얻을 수 있다. 각각 “지도 1”과 “지도 2”라고 하자. 지도 1 또는 지도 2 중 어느 하나라도 벽인 부분은 전체 지도에서도 벽이다. 지도 1과 지도 2에서 모두 공백인 부분은 전체 지도에서도 공백이다.

“지도 1”과 “지도 2”는 각각 정수 배열로 암호화되어 있다.



암호화된 배열은 지도의 각 가로줄에서 벽 부분을 1, 공백 부분을 0으로 부호화했을 때 얻어지는 이진수에 해당하는 값의 배열이다.

5x5 형태로 이뤄진 지도

지도 1

공백 # 공백 공백 # -> 01001(2) =  $9 \cdot 2^3 + 2^0 = 8 + 1 = 9$  가 된 것  
# 공백 # 공백 공백 # -> 10100(2) =  $20 \cdot 2^4 + 2^2 = 16 + 4 = 20$  이 된 것.  
### 공백 공백 -> 11100(2) = 28  
# 공백 공백 # 공백 -> 10010(2) = 18  
공백 \$ # 공백 ## -> 01011(2) = 11

지도 2

#### 공백 -> 11110 = 30  
공백 공백 공백 공백 # -> 00001 = 1  
# 공백 # 공백 # -> 10101 = 21  
# 공백 공백 공백 # -> 10001 = 17  
### 공백 공백 -> 11100 = 28

지도 1 + 지도 2 를 한 것이 결과로

#####  
# 공백 # 공백 #  
### 공백 #  
# 공백 공백 ##  
#####

네오가 프로도의 비상금을 손에 넣을 수 있도록, 비밀지도의 암호를 해독하는 작업을 도와줄 프로그램을 작성하라.

입력 형식

입력으로 지도의 한 변 크기  $n$  과 2개의 정수 배열  $arr1$ ,  $arr2$ 가 들어온다.

$1 \leq n \leq 16$

$arr1$ ,  $arr2$ 는 길이  $n$ 인 정수 배열로 주어진다.

정수 배열의 각 원소  $x$ 를 이진수로 변환했을 때의 길이는  $n$  이하이다. 즉,  $0 \leq x \leq 2^n - 1$ 을 만족한다.

출력 형식

원래의 비밀지도를 해독하여 "#", 공백으로 구성된 문자열 배열로 출력하라.

입출력 예제

매개변수	값
$n$	5
$arr1$	[9, 20, 28, 18, 11]
$arr2$	[30, 1, 21, 17, 28]
출력	["#####", "# # #", "### #", "# ##", "#####"]

알고리즘 문제 풀때 해결하는 순서

1. 문제를 2번 정독하면서 질문을 정확하게 이해하고 출력이 뭐가 되는지 정확하게 이해하기
2. 풀이를 번호 순서대로 한글로 적어본다.

1) 10진수를 2진수로 바꾸는 작업 구현

2) 두개의 이진수(0과 1)를 가지고 '#'과 '공백'을 표현하도록 한다.

3. 한글을 파이썬 코드로 번역 한다.

문제 풀이 시작!

1. 문제를 2번 정독하면서 질문을 정확하게 이해하고 출력이 뭐가 되는지 정확하게 이해하기

n = 5

arr1=[9, 20, 28, 18, 11]

arr2=[30, 1, 21, 17, 28]

secretmap(arr1,arr2,n) # secretmap 이란 함수를 만들어서 아래와 같이 결과가 나오도록 하는게 문제

결과

['#####']

['### #']

['## ##']

['#### ']

['#####']

['### #']

2. 풀이를 번호 순서대로 한글로 적어본다.

1) 10진수를 2진수로 바꾸는 작업 구현

문제 525. 숫자 9를 이진수로 변환하시오.

결과 : 0b1001

###

print( bin(9)) # 10진수를 2진수로 변환시켜주는 내장함수 bin() 사용

# 0b1001 # b = binary(이진)

###

문제 526. 위의 결과에서 0b 는 제외하고 1001만 출력하시오.

###

print(bin(9)[2:]) # 0번째와 1번째의 값만 잘라내고 출력하면 됨

###

문제 527. 숫자 30의 이진수를 아래와 같이 출력하시오.

11110

###

print( bin(30)) # 0b11110 이므로 앞의 2개를 자르면 됨

#정답

print (bin(30)[2:]) # 11110

###

문제 528. 아래와 같이 출력되는 change 함수를 생성하시오.

```
print(change(num)) # 11110
```

```
###
```

```
def change(num):  
    return bin(num)[2:]
```

```
print(change(30))
```

```
###
```

문제 529. 위에서 만든 change 함수의 숫자 9를 넣고 출력해보시오.

```
###
```

```
print( change(9)) # 1001
```

# 4자리수로 나옴. 문제에서는 9를 넣어도 5자리로 나옴. 01001 로 나옴.

```
###
```

문제 530. change 함수를 실행할때 아래와 같이 입력매개변수를 하나 더 추가해서 입력 매개변수의 길이만큼 숫자0이 채워져서 출력되게 하시오.

```
print(change(9,5))
```

```
-> 01001
```

```
###
```

```
def change(num,length):  
    return bin(num)[2:].rjust(length,'0') # rjust('채울넓이', '채울문자/숫자')
```

```
print(change(9,5)) # 01001
```

# rjust(5,'0') 은 출력될 자리수를 전체 5자리로 잡고, 만약 출력된 값이 5자리에 미치지 못하면

# 출력된 값의 왼쪽에 '0' 을 채워넣어라~ 라는 뜻.

# ljust 는 반대로 출력된 값의 오른쪽에 지정된 문자/숫자를 채워넣는 것.

print(change(30,5)) #30의 이진수는 이미 5자리여서 따로 채워지는 것 없이 원래의 값이 출력된다.

```
# 11110
```

print(change(30,6)) # 전체 자리수를 6자리로 하면 앞에 0이 하나 붙어서 출력된다. -> 011110

```
###
```

만약 두개를 한꺼번에 출력하고 싶다면?

```
###
```

```
print( change(9,5), change(30,5) )
```

01001 11110 로 나온다.

```
###
```

문제 531. 카카오의 예제를 가지고 아래의 결과를 출력하시오.

```
n = 5
```

```
arr1=[9, 20, 28, 18, 11]
```

```
arr2=[30, 1, 21, 17, 28]
```

```
###
```

```
n = 5
```

```
arr1=[9, 20, 28, 18, 11]
```

```
arr2=[30, 1, 21, 17, 28]
```

```
for i in range(len(arr1)):
```

```
    print( change(arr1[i],n), change(arr2[i],5) )
```

```

#%
# i가 0일때 arr1 = 9, arr2 = 30 change 함수에 따라 이진수가 출력
# i가 1일때 arr1 = 20, arr2 = 1 change 함수에 따라 이진수가 출력
.
.
.
# i가 4일때 arr1 = 11, arr2 = 28 change 함수에 따라 이진수가 출력

```

```

01001 11110
10100 00001
11100 10101
10010 10001
01011 11100

```

```

#%

```

2) 두개의 이진수(0과 1)를 가지고 '#'과 '공백'을 표현하도록 한다.

```

#%

```

문제 532. 위에서 출력된 아래의 결과를 이용해서 두 지도의 숫자가 둘다 0이면 공백(" ")을 출력하고 아니면(둘중 하나라도 공백이 아니면) 벽('#')을 출력하시오.

```

#%

```

```

# 선생님 코드

```

```

def change(num,length):
    return bin(num)[2:].rjust(length,'0')

```

```

n = 5
arr1=[9, 20, 28, 18, 11]
arr2=[30, 1, 21, 17, 28]
result = []

for i in range(len(arr1)):
    map1 = change(arr1[i],n)
    map2 = change(arr2[i],n)
    f = ''
    for k in range(n):
        if map1[k] == '0' and map2[k] == '0':
            fi = ''
            f = f + fi
        else:
            fi = '#'
            f = f + fi
    result.append(f)

```

```

print(result)

```

```

#%

```

문제 533. 위의 코드를 secretmap(arr1,arr2,n) 함수로 만들어서 결과가 아래와 같이 출력되도록 하시오.

```

["#####", "### #", "## ##", "#### ", "#####", "### # "]

```

```

#%

```

```

def change(num,length):
    return bin(num)[2:].rjust(length,'0')

```

```

n = 6
arr1=[46, 33, 33 ,22, 31, 50]
arr2=[27 ,56, 19, 14, 14, 10]

```

```

def secretmap(arr1,arr2,n):
    result = []
    for i in range(len(arr1)):

```

```

map1 = change(arr1[i],n)
map2 = change(arr2[i],n)
f = ""
for k in range(n):
    if map1[k] == '0' and map2[k] == '0':
        fi = ' '
        f = f + fi
    else:
        fi = '#'
        f = f + fi
result.append(f)
print(result)

```

```
secretmap(arr1,arr2,n)
```

```

###
#다른 방법 1

```

```

n = 6
arr1=[46, 33, 33, 22, 31, 50]
arr2=[27, 56, 19, 14, 14, 10]
a = []

```

```

def secretmap(arr1, arr2,n):
    answer = []
    for a1, a2 in zip(arr1, arr2): # arr1과 arr2를 zip()으로 묶기
        a = str(bin(a1 | a2))[2:] # a1 OR a2를 binary로 바꿔주는 bin() function을 통해 2진수로 바뀐 후, 불필요한 맨 앞 2자리를 버리기 위해 string으로 변환 후 index slicing
        a = '0' * (n - len(a)) + a # index slicing을 하게 되면, 0으로 시작하는 2진수의 0들이 사라지기 때문에 입력받은 n의 길이에서 숫자만큼의 수를 뺀만큼 '0'을 곱하고 그 숫자를 더해준다.
        a = a.replace('1',"#").replace('0'," ") # 1은 #으로 replace, # 0은 공백으로 replace
        answer.append(a)

    return answer

```

```

print(secretmap(arr1,arr2,n) )
###

```

## 160. 도박사 이야기

해외여행 중 어느 뒷골목에서 동전으로 내기를 하는 도박사를 만났다.

앞면이 나오면 당신에게 1달러, 뒷면이 나오면 나에게 1달러, 어때요? 이 내기 한번 해볼래요?

그런데 동전에 부정이 있는 듯해 실제로 알아봤더니 8회중 7회는 뒷면이 나왔다.

이 동전은 앞면과 뒷면이 나올 확률이 다르지 않느냐? 고 했더니 도박사는 아무렇지도 않게 "그런게 아니고 우연" 이라고 대답했다. 어떻게 반론하면 좋을까?

동전을 8번 던졌을 때 동전이 뒷면이 나오는 횟수가 신뢰구간 68% 안에 드는지 아닌지를 출력하는 함수를 생성하시오

```

print ( coin_hypo(동전 뒷면의 횟수) )

print ( coin_hypo(4) )

```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 4번이 나올 확률은 신뢰구간 68% 안에 있습니다.

```
print ( coin_hypo(8) )
```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 8번이 나올 확률은 신뢰구간 68% 안에 없습니다.

1. 문제를 2번 정독하면서 질문을 정확하게 이해하고 출력이 뭐가 되는지 정확하게 이해하기

2. 풀이를 번호 순서대로 한글로 적어본다.

3. 한글을 파이썬 코드로 번역 한다.

문제 534. 동전 1개를 10000번 던져서 뒷면이 나오는 횟수를 출력하시오.

```
hint : random.randint(0,1)
```

```
###
```

```
# 기존 방법
```

```
import random
```

```
coin = ['앞면','뒷면']
```

```
cnt = 0
```

```
for i in range(1,10001):
```

```
    result = random.choice(coin)
```

```
    if result == '뒷면':
```

```
        cnt+=1
```

```
print(cnt)
```

```
###
```

```
# 힌트 사용
```

```
import random
```

```
cnt = 0
```

```
for i in range (1,10001):
```

```
    result=random.randint(0,1) #0과 1중 하나를 랜덤으로 출력 + 정수로 뽑으려고 int를 한 것.
```

```
    if result == 0: # 0을 앞면이라 가정
```

```
        cnt+=1
```

```
print(cnt)
```

```
# 혹시나 문자열도 되나 싶어 randstr(~) 해봤지만 없는 함수!
```

```
###
```

문제 535. 동전을 8번 던져서 뒷면이 나오는 횟수를 출력하시오.

```
###
```

```
import random
```

```
cnt = 0
```

```
for i in range (1,9):
```

```
    result=random.randint(0,1) #0과 1중 하나를 랜덤으로 출력 + 정수로 뽑으려고 int를 한 것.
```

```
    if result == 0: # 0을 뒷면이라 가정
```

```
        cnt+=1
```

```
print(cnt)
```

```
###
```

문제 536. 동전 하나를 8번 던졌을때, 그중 뒷면이 5번 나오는 횟수를 출력하시오.

```
###
```

```
import random
```

```
cnt2 = 0
```

```

for i in range(1,10001):
    for k in range(1,9):
        result=random.randint(0,1)
        if result == 0:
            cnt+=1
        if cnt ==5:
            cnt2+=1

print(cnt2)

```

###

문제 537. 아래의 결과를 출력하시오.  
[0,1,2,3,4,5,6,7,8]

```

print(a)
print(b)

```

결과 :

```

[0,1,2,3,4,5,6,7,8]
[0,0,0,0,0,0,0,0,0]

```

###

```

a = (i for i in range(9))

```

```

print(list(a) )

```

```

b = [0]*9

```

```

print(type(b) )

```

###

문제 538. 동전을 8번 던졌을때 뒷면이 0번에서 8번 나오는 횟수가 아래와 같이 출력되게 하시오.

결과 :

```

[0,0,0,0,0,0,0,0]   가

```

[412,3026,107,10798,22101,27651,21682,10851,3191, 379] 이런식으로 바뀌게 replace? 는 int 절에 쓰일 수 없다.

###

```

import random

```

```

cnt2=[] # 뒷면이 나올 각각의 횟수를 담을 곳
b=[0]*9

```

```

for h in range(1,9): # 뒷면이 나오는 횟수 0부터 8까지를 의미
    cnt=0
    for i in range(1,100001): # 아래의 작업을 10000번 반복
        result=0
        for k in range(1,9): # 주사위 8번 던지는 작업
            result+=random.randint(0,1) #0(뒷면)이 나오면 1, // 1(앞면)이 나오면 2가 되는 것.
        if result == h: # 뒷면 출력이 된수가 h와 일치하면
            cnt+=1
        b[h]=cnt # b list에서 h에 해당하는 값을 cnt로 바꾼다.

```

```

print(b)

```

뭔가 조금 이상함! 복습하면서 다시 보기!!

###

# 홍비님 풀이

```

import random
a = []

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1
        a.append(count)
print(a)

```

###

문제 539. 위의 횟수를 가지고 아래와 같이 확률이 출력되게 하시오.

(횟수/100000)

###

```

import random
a = []

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1
        a.append(count)

b = [ j/100000 for j in a]

print(b)

###

```

문제 540. coin\_cnt를 x축으로 하고 확률 (b)를 y축으로 하는 막대그래프를 그리시오.

```

coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
b = [0.00418, 0.03158, 0.10941, 0.21842, 0.27136, 0.2196, 0.11055, 0.03102, 0.00378] # 각 횟수별 확률

###

```

```

import numpy as np
import random
import matplotlib.pyplot as plt

coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
a = []

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정

```



```

        cnt+=1
    if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
        count+=1
    a.append(count)

```

```

b= [ j/100000 for j in a]

```

```

plt.bar(coin_cnt, b, tick_label=coin_cnt, align='center', color='purple')
plt.show()

```

# tick\_label -> x축의 라벨을 무엇으로 할지 명시하는 것

# align = 나란히 하다 = 정렬방식

###

문제 541. coin\_cnt 를 입력값으로 해서 coin\_cnt 의 요소들의 평균값과 표준편차를 출력하시오.

###

```

import numpy as np
import random

```

```

coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
a = []

```

```

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1
        a.append(count)

```

```

b= [ j/100000 for j in a]

```

```

c_m = np.mean(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 평균
c_s = np.std(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 표준편차

```

```

print(c_m) # 4
print(c_s) # 2.581988897471611

```

###

문제 542. 위에서 구한 평균값과 표준편차를 이용해서 coin\_cnt의 요소값에 대한 확률밀도함수 값을 출력하시오.

###

```

import numpy as np
import random
from scipy.stats import norm # 확률밀도 함수를 이용하기 위한 모듈 import
#import matplotlib.pyplot as plt

```

```

coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
a = []

```

```

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1

```

```

a.append(count)

b= [ j/100000 for j in a]

c_m = np.mean(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 평균
c_s = np.std(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 표준편차

result = norm.pdf(coin_cnt, c_m, c_s ) #확률밀도함수 값을 result에 담는다.

print(result)

###

문제 543. 위에서 구한 coin_cnt를 x축으로 두고, 위의 확률밀도함수값을 y축으로 하는 확률밀도함수 그래프를 그리시오.

###

```

```

import numpy as np
import random
from scipy.stats import norm # 확률밀도 함수를 이용하기 위한 모듈 import
import matplotlib.pyplot as plt

coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
a = []

for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1
        a.append(count)

b= [ j/100000 for j in a]

c_m = np.mean(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 평균
c_s = np.std(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 표준편차

x = coin_cnt
y = norm.pdf(coin_cnt, c_m, c_s ) #확률밀도함수 값을 result에 담는다.

plt.plot(x,y, color='purple')
plt.show()

###

```

문제 544. 위의 그래프에서 신뢰구간 68%에 해당하는 부분만 노란색으로 색칠하시오.

신뢰구간 범위

1. 68% -> 평균  $\pm 1$  \* 표준편차
2. 95% -> 평균  $\pm 1.96$  \* 표준편차
3. 99% -> 평균  $\pm 2.58$  \* 표준편차

###

```

import numpy as np
import random
from scipy.stats import norm # 확률밀도 함수를 이용하기 위한 모듈 import
import matplotlib.pyplot as plt

```

```
coin_cnt = [0,1,2,3,4,5,6,7,8] # 동전8번을 던져서 동전의 뒷면이 나오는 횟수
a = []
```

```
for k in range(0,9): # 동전 뒷면이 나오는 횟수 1~8
    count=0
    for j in range(1,100001):
        cnt=0
        for i in range(1,9): #
            result = random.randint(0,1) #숫자 0과 1 중 하나를 랜덤으로 출력
            if result == 0: #0을 뒷면이라고 가정
                cnt+=1
            if cnt==k: # k = 0,1,2,3,4,5,6,7,8 을 쓰기위해 맨위에 for k~ 를 써준 것
                count+=1
    a.append(count)
```

```
b= [ j/100000 for j in a]
```

```
c_m = np.mean(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 평균
c_s = np.std(coin_cnt) # 동전을 8번 던져서 뒷면이 나오는 횟수 의 표준편차
```

```
x = coin_cnt
y = norm.pdf(coin_cnt, c_m, c_s ) #확률밀도함수 값을 result에 담는다.
```

```
plt.plot(x,y, color='purple')
plt.fill_between(coin_cnt, y, where = (coin_cnt<c_m+c_s) & (coin_cnt>=c_m-c_s), color ='yellow' ) #신뢰구간 68%
plt.show()
```

```
###
(파이썬 마지막 문제)
```

문제545. 위의 코드들을 이용해서 아래와 같이 출력되게하시오 !

답글로 반드시 달아서 유종의 미를 이루세요 ~~~~

동전을 8번 던졌을 때 동전이 뒷면이 나오는 횟수가 신뢰구간 68% 안에 드는지  
아닌지를 출력하는 함수를 생성하시오

```
print ( coin_hypo(동전 뒷면의 횟수) )
```

```
print ( coin_hypo(4) )
```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 4번이 나올 확률은 신뢰구간 68% 안에 있습니다.

```
print ( coin_hypo(8) )
```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 8번이 나올 확률은 신뢰구간 68% 안에 없습니다.

```
###
```

```
import numpy as np
import random
from scipy.stats import norm # 확률밀도 함수를 이용하기 위한 모듈 import
import matplotlib.pyplot as plt
```

```
# 1) 던진 횟수 대비 동전의 뒷면이 나오는 확률을 구하는 함수 생성(100000번 반복)
```

```
def tail_prob(x):
    cnt=0
    for i in range(1,100001):
        a = [random.randint(0,1) for i in range(1,9) ] # 0을 뒷면이라 가정, 동전을 8번 던져서 앞면 혹은 뒷면이 나오는 실험
```

```

if a.count(0) == x: # 함수에서 지정한 뒷면이 나오는 횟수와 a 의 값이 일치하면
    cnt+=1
return cnt/100000 # 입력한 횟수만큼 동전의 뒷면이 나오는 확률

```

# 2) 동전을 던져서 뒷면이 나오는 횟수의 확률이 신뢰구간 68%에 있는지 검증하는 함수 만들기

```

def coin_hypo(n):
    prob_tail = [tail_prob(i) for i in range(1,9)] # 8번 동전을 던져서 뒷면이 나오는 횟수(0번부터 8번까지)의 확률
    coin_cnt = list(i for i in range(1,9))

    mean = np.mean(coin_cnt) # 뒷면이 나오는 횟수의 평균
    std = np.std(coin_cnt) # 동전 뒷면이 나오는 횟수의 표준편차
    x = coin_cnt # 동전 뒷면 나오는 수 x에 할당
    y = norm.pdf(coin_cnt, mean, std) # 동전 뒷면 나오는 횟수의 확률 밀도 값을 y에 할당

    plt.plot(x, y, color = 'purple')
    plt.fill_between(x, y, where = (x<=mean+1*std) & (x>=mean-1*std), color = 'yellow' )
    plt.show()

    if (n >= mean - 1*std) and (n <= mean + 1*std):
        return f' 동전을 8번 던졌을 때 뒷면이 {n}번 나올 확률은 신뢰구간 68% 안에 있습니다.'
    else:
        return f' 동전을 8번 던졌을 때 뒷면이 {n}번 나올 확률은 신뢰구간 68% 안에 없습니다.'

```

# 3) 뒷면이 4번 나오는 경우 & 8번 나오는 경우에 대한 확률 출력

```

for i in (4,8):
    print(coin_hypo(i))

동전을 8번 던졌을 때 뒷면이 4번 나올 확률은 신뢰구간 68% 안에 있습니다.
동전을 8번 던졌을 때 뒷면이 8번 나올 확률은 신뢰구간 68% 안에 없습니다.

###
# 3) 뒷면이 나오는 각 횟수에 대한 신뢰도 결과 출력
for i in range(1,9):
    print(coin_hypo(i))

```