

파이썬2 목차

2018년 11월 19일 월요일 오후 6:42

- 파이썬 149제(round, int, filter, max, min, 분석함수, 그룹함수)

- 68. 반올림수 구하기(round)
- 69. 실수형 자료를 정수형 자료로 변환하기(int)
- 70. 정수 리스트에서 소수만 걸러내기(filter)
- 71. 최대값, 최소값 구하기(max, min)
- 73. 판다스와 오라클 데이터 분석 함수 비교
- 72. 판다스와 오라클 그룹함수 비교

- 파이썬 149제(문자열, Comprehension)

- 74. 문자열에서 특정위치 문자 얻기
- 75. 문자열에서 지정한 구간의 문자열 얻기
- 76. 문자열에서 홀수 번째 문자만 추출하기
- 77. 문자열을 거꾸로 만들기
- 78. 두 개의 문자열 합치기
- 79. 문자열을 반복해서 새로운 문자열로 만들기
- 80. 문자열에서 특정 문자가 있는지 확인하기(in)
- 81. 파이썬의 Comprehension
 - 1. list Comprehension
 - 2. set Comprehension
 - 3. Dictionary Comprehension
- 82. 문자열이 알파벳인지 검사하기(isalpha)
- 83. 문자열에 특수문자가 있는지 검사하기
- 84. 문자열에서 대소문자 변환하기 (upper,lower)
- 85. 문자열에서 좌우 공백 제거하기(lstrip, rstrip, strip)
- 86. 문자열에서 있는 문자(열) 개수 구하기(count)
- 87. 문자열에서 특정 문자(열) 위치 찾기(find)
- 88. 문자열을 특정 문자(열로) 분리하기(split)
- 89. 문자열을 특정 문자(열)로 결합하기(join)

- [90. 문자열에서 특정 문자\(열\)을 다른 문자\(열\)로 바꾸기\(replace\)](#)
- [91. 문자열을 바이트 객체로 바꾸기\(encode\)](#)
- [92. 바이트 객체를 문자열로 바꾸기\(decode\)](#)

- [파이썬 149제\(리스트\)](#)

- [93. 순차적인 정수 리스트 만들기\(range\)](#)
- [94. 리스트에서 특정 위치의 요소 얻기](#)
- [95. copy와 deepcopy 모듈의 차이](#)
- [96. 리스트에서 특정 요소의 위치 구하기\(index\)](#)
- [97. 리스트에서 특정 위치의 요소를 변경하기](#)
- [98. 리스트에서 특정 구간에 있는 요소 추출하기](#)
- [99. 리스트 요소 순서를 역순으로 만들기1\(reverse\)](#)
- [100. 리스트 요소 순서를 역순으로 만들기2\(reversed\)](#)
- [101. 리스트 합치기\(+\)](#)
- [102. 리스트 반복하기\(*\)](#)
- [103. 리스트에 요소 추가하기\(append\)](#)
- [104. 리스트의 특정 위치에 요소 삽입하기\(insert\)](#)
- [105. 리스트의 특정 위치\(index\)의 요소 제거하기\(del\)](#)
- [106. 리스트에서 특정 요소 제거하기\(remove\)](#)
- [107. 리스트 요소 정렬하기 1\(sort\)](#)
- [108. 리스트 요소 정렬하기 2 \(sorted\)](#)
- [109. 리스트 요소 무작위로 섞기\(shuffle\)](#)
- [110. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기\(enumerate\)](#)
- [111. 리스트의 모든 요소의 합 구하기\(sum\)](#)
- [112. 리스트 요소가 전부 참인지 확인하기\(all, any\)](#)

- [파이썬 149제\(딕셔너리\)](#)

- [113. 파이썬에서 _ \(언더바\)를 사용하는 경우](#)
- [114. 사전에 요소 추가하기](#)
- [115. 딕셔너리의 특정 요소값 변경하기](#)
- [116. 딕셔너리의 모든 요소 제거하기\(clear\)](#)
- [117. 딕셔너리에서 키만 추출하기\(keys\)](#)
- [118. 딕셔너리에서 값만 추출하기\(values\)](#)
- [119. 딕셔너리에 있는 모든 요소 추출하기\(items\)](#)
- [120. 딕셔너리 정렬하기\(sorted\)](#)

- [파이썬 149제\(입,출력 및 etc\)](#)

- [121. 문자 코드값 구하기\(ord\)](#)
- [122. 코드값에 대응하는 문자 얻기\(chr\)](#)

- [123. 문자열로 된 식을 실행하기\(eval\)](#)
- [124. 이름없는 한줄짜리 함수 만들기\(lambda\)](#)
- [125. 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기\(map\)](#)
- [126. 텍스트 파일을 읽고 출력하기\(read\)](#)
- [127. 텍스트 파일을 한줄씩 읽고 출력하기\(readlines\)](#)
- [128. 화면에서 사용자 입력을 받고 파일로 쓰기\(write\)](#)
- [129. 텍스트 파일 복사하기\(read,write\)](#)

- [파이썬 149제\(파일 복사\)](#)

- [130. 바이너리 파일 복사하기\(read, write\)](#)
- [131. 파일을 열고 자동으로 닫기\(with~as\)](#)
- [132. 파일의 특정 부분만 복사하기\(seek, read, write\)](#)
- [133. 파일크기 구하기\(ospathgetsize\)](#)
- [134. 파일 삭제하기\(osremove\)](#)
- [135. 파일 이름 바꾸기\(osrename\)](#)
- [136. 파일을 다른 디렉터리로 이동하기\(osrename\)](#)
- [137. 디렉토리에 있는 파일목록 얻기\(oslistdir, globglob\)](#)
- [138. 현재 디렉터리 확인하고 바꾸기\(osgetcwd, oschdir\)](#)
- [139. 디렉토리 생성하기\(osmkdir\)](#)
- [140. 디렉토리 제거하기 \(osrmdir\)](#)
- [141. 하위 디렉토리 및 파일 전체 삭제하기\(shutilrmtree\)](#)
- [142. 파일이 존재하는지 체크하기\(ospathexist\)](#)
- [143. 파일인지 디렉터리인지 확인하기\(ospathisfile, ospathisdir\)](#)
- [144. 현재 시간을 년-월-일 시:분:초로 출력하기\(localtime, strftime\)](#)
- [145. 올해 경과된 날짜수 계산하기\(localtime\)](#)
- [146. 오늘의 요일 계산하기\(localtime\)](#)
- [147. 프로그램 실행 시간 계산하기\(datetimenow\)](#)
- [148. 주어진 숫자를 천 단위로 구분하기](#)
- [149. 문자열의 각 문자를 그 다음 문자로 변경하기](#)

- [파이썬 149제\(웹 스크롤링\)](#)

- [웹 스크롤링](#)
- [n사 이미지 스크롤링](#)
- [g사 이미지 스크롤링](#)
- [d사 이미지 웹스크롤링](#)
- [b사 이미지 웹스크롤링](#)
- [서울시 응답소 게시판](#)

- [파이썬으로 게임\(Ping Pong\) 만들기](#)

- [ping pong](#)
- [numpy 함수 명령어](#)
- [리스트 함수 명령어](#)
- [문자형 함수 명령어](#)
- [숫자형 함수 명령어](#)
- [필수 알고리즘](#)
 - [이진탐색](#)
 - [재귀 알고리즘](#)
 - [버블정렬](#)
 - [탐욕 알고리즘](#)
 - [비밀 지도](#)
 - [다트게임](#)
 - [LRU 알고리즘](#)
 - [셔틀버스](#)
 - [자카드 유사도](#)
 - [다이나믹 프로그래밍\(최단거리\)](#)
- [통계](#)
 - [2장. 데이터 수치 요약](#)
 - [numpy 사용법](#)
 - [파이썬으로 그래프 그리기](#)
 - [1. 라인 그래프 그리기](#)
 - [2. 원형 그래프 그리기](#)
 - [3. 막대 그래프 그리기](#)
 - [4. 히스토그램 그리기](#)
 - [5\) plot 그래프](#)
 - [6\) 사분위수 그래프](#)
 - [3장. 확률의 정의](#)
 - [4장. 확률분포와 표본분포 \(총12문제\)](#)
 - [4-1. 이산 확률 분포](#)
 - [4-2 연속 확률 분포](#)
 - [5장. 통계적 추정의 이해](#)
 - [모비율의 추정](#)
 - [6장. 가설검정의 이해](#)

파이썬 149제(round, int, filter, max, min, 분석함수, 그룹함수)

2018년 7월 27일 금요일 오후 6:09

68. 반올림수 구하기(round)

```
ret1 = round(1118)    # 소수점 첫째자리에서 반올림해줌
ret2 = round(16.554)    # 소수점 첫째자리에서 반올림해줌
ret3 = round(1118, -1) # 1자리에서 반올림해줌
ret4 = round(16.554, 2) # 소수점 셋째자리에서 반올림해줌
print(ret1)    # 1118
print(ret2)    # 17
print(ret3)    # 1120
print(ret4)    # 16.55
```

```
1 6 . 5 5 4
-2 -1 0 1 2
```

문제183. 이름, 월급*0.15를 출력하시오.(판다스 이용해서)

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
emp['sal']=emp['sal']*0.15
print(emp[['ename','sal']])
```

	ename	sal
0	KING	750.00
1	BLAKE	427.50
2	CLARK	367.50
3	JONES	446.25
4	MARTIN	187.50
5	ALLEN	240.00
6	TURNER	225.00
7	JAMES	142.50
8	WARD	187.50
9	FORD	450.00
10	SMITH	120.00
11	SCOTT	450.00
12	ADAMS	165.00
13	MILLER	195.00

문제184. 위의 결과를 다시출력하는데 소수점 첫번째 자리에서 반올림 해서 출력 하시오.

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
emp['sal']=round(emp['sal']*0.15)
```

```
print(emp[['ename','sal']])
```

	ename	sal
0	KING	750.0
1	BLAKE	428.0
2	CLARK	368.0
3	JONES	446.0
4	MARTIN	188.0
5	ALLEN	240.0
6	TURNER	225.0
7	JAMES	142.0
8	WARD	188.0
9	FORD	450.0
10	SMITH	120.0
11	SCOTT	450.0
12	ADAMS	165.0
13	MILLER	195.0

답2)

```
import pandas as pd
```

```
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
emp['sal']=emp['sal']*0.15
```

```
emp['sal']=emp['sal'].round(0)
```

```
print(emp[['ename','sal']])
```

설명 :

MARTIN 187.50 ---> 188

JAMES 142.50 ---> 142

"R 처럼 파이썬도 짝수를 좋아한다"

문제185. (생각해야할 문제)2.5가 3이 되게 하려면 어떻게 해야하는가?

답)

```
import pandas as pd
```

```
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
emp['sal']=emp['sal']*0.15+0.5
```

```
emp['sal']=emp['sal'].round(0)
```

```
print(emp[['ename','sal']])
```

	ename	sal
0	KING	750.0
1	BLAKE	428.0
2	CLARK	368.0
3	JONES	447.0
4	MARTIN	188.0
5	ALLEN	240.0
6	TURNER	226.0
7	JAMES	143.0
8	WARD	188.0

	ename	sal
0	KING	750.0
1	BLAKE	428.0
2	CLARK	368.0
3	JONES	447.0
4	MARTIN	188.0
5	ALLEN	240.0
6	TURNER	226.0
7	JAMES	143.0
8	WARD	188.0
9	FORD	450.0
10	SMITH	120.0
11	SCOTT	450.0
12	ADAMS	166.0
13	MILLER	196.0

설명 :

파이썬에서는 기본적으로 제공하는 올림함수는 없습니다. 따라서 기존 round 함수를 응용하여 올림의 효과를 만들어내야 합니다.

round(a + 0.5) 를 사용해서 올림의 효과를 만들어 냅니다.

완벽답)

```
import csv
from decimal import Decimal,ROUND_HALF_UP

#print(Decimal(2.3).quantize(0,ROUND_HALF_UP))

file=open('C:\python_data\emp2.csv','r')
emp=csv.reader(file)
for i in emp:
    print([i[1],Decimal(int(i[5])*0.15).quantize(0,ROUND_HALF_UP)])
```

문제186. 위의 결과를 출력하는데 일의 자리에서 바로 반올림되게 하시오.

답)

```
import pandas as pd

emp=pd.read_csv("C:\python_data\emp.csv")
emp['sal']=emp['sal']*0.15
emp['sal']=emp['sal'].round(-1)
print(emp[['ename','sal']])
```

	ename	sal
0	KING	750.0
1	BLAKE	430.0
2	CLARK	370.0
3	JONES	450.0
4	MARTIN	190.0
5	ALLEN	240.0
6	TURNER	226.0
7	JAMES	143.0
8	WARD	188.0
9	FORD	450.0
10	SMITH	120.0
11	SCOTT	450.0
12	ADAMS	166.0
13	MILLER	196.0

	ename	sal
0	KING	750.0
1	BLAKE	430.0
2	CLARK	370.0
3	JONES	450.0
4	MARTIN	190.0
5	ALLEN	240.0
6	TURNER	220.0
7	JAMES	140.0
8	WARD	190.0
9	FORD	450.0
10	SMITH	120.0
11	SCOTT	450.0
12	ADAMS	160.0
13	MILLER	200.0

69. 실수형 자료를 정수형 자료로 변환하기(int)

ex)

```
idata1 = int(-5.4)
```

```
idata2 = int(1.78e1)
```

```
idata3 = int(171.56)
```

```
print(idata1) # -5가 출력됨
```

```
print(idata2) # 17이 출력됨
```

```
print(idata3) # 171이 출력됨
```

• 파이썬의 변환함수 정리

	Data type	변환함수
숫자형	int	int()
문자형	str	str()
실수형	float	float()
리스트형	list	list()
튜플형	tuple	tuple()

문제187. 판다스를 이용하지 말고 이름과 연봉을 출력하시오.(sal*12)

답)

```
import csv
```

```
file=open("C:\python_data\emp2.csv",'r')
```

```
emp_csv=csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    print(emp_list[1], int(emp_list[5])*12)
```

```
KING 60000
BLAKE 34200
CLARK 29400
JONES 35700
MARTIN 15000
ALLEN 19200
TURNER 18000
JAMES 11400
WARD 15000
FORD 36000
SMITH 9600
SCOTT 36000
ADAMS 13200
MILLER 15600
```

문제188. 문제187번을 다시 수행하는데 이름과 연봉을 출력하는 아래의 예제에서 int()를 사용

하지 않고 float()를 사용해도 결과가 출력 되겠는가?

답)

네

```
import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)
```

```
for emp_list in emp_csv:
    print(emp_list[1], float(emp_list[5])*12)
```

```
KING 60000.0
BLAKE 34200.0
CLARK 29400.0
JONES 35700.0
MARTIN 15000.0
ALLEN 19200.0
TURNER 18000.0
JAMES 11400.0
WARD 15000.0
FORD 36000.0
SMITH 9600.0
SCOTT 36000.0
ADAMS 13200.0
MILLER 15600.0
```

70. 정수 리스트에서 소수만 걸러내기(filter)

ex)

```
def getPrime(x):
    for i in range(2, x-1):
        if x%i == 0:
```

```

        break
    else:
        return x

listdata = [117, 119, 1113, 11113, 11119]
ret = filter(getPrime, listdata)
print(list(ret))      # [11113, 11119] 가 출력됨

```

문제189. 이름의 첫글자가 S로 시작하는 직원들의 이름을 출력하시오.

답)

```

import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
result=emp['ename'].apply(lambda x:x[0]=='S')
result2=emp[['ename']] [result==True]
print(result2)

```

문제190. 위의 문제를 다시 수행하는데 lambda 표현식을 사용하지 말고 함수를 직접 생성해서

수행하시오.

답)

```

def emp_first(x):
    return x[0]=='S'

import pandas as pd

emp=pd.read_csv("C:\python_data\emp.csv")
result=emp['ename'].apply(emp_first)
result2=emp[['ename']] [result==True]
print(result2)

```

	ename
10	SMITH
11	SCOTT

71. 최대값, 최소값 구하기(max, min)

ex)

```

listdata = [9.96, 1.27, 5.07, 6.45, 8.38, 9.29, 4.93, 7.73, 3.71, 0.93]
maxval = max(listdata)
minval = min(listdata)
print(maxval)      # 9.96이 출력됨
print(minval)      # 0.93이 출력됨

```

```
txt = 'Alotofthingsoccureachday'
maxval = max(txt)
minval = min(txt)
print(maxval);    # 'y'가 출력됨
print(minval)    # 'A'가 출력됨
```

```
maxval = max(2+3, 2*3, 2**3, 3**2)
minval = min('abz', 'a12')
print(maxval)    # 9가 출력됨
print(minval)    # 'a12'가 출력됨
```

설명 : 숫자보다 문자가 더 우선순위 이다.

그룹함수 비교

오라클	vs	파이썬
max		max
min		min
count		count
sum		sum
avg		사용자 정의 함수 (pandas는 mean)

문제191. 판다스를 이용하지 않고 사원 테이블의 최대 월급을 출력하시오.

```
답)
import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)
sal_list=[]

for emp_list in emp_csv:
    sal_list.append(int(emp_list[5]))

print(max(sal_list))
5000
```

문제192. 판다스를 이용해서 사원 테이블의 최소월급을 출력하시오.

```
답)
import pandas as pd
```

```
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
print(emp['sal'].min())
```

```
800
```

73. 판다스와 오라클 데이터 분석 함수 비교

순위를 출력하는 분석함수를 판다스에서 구현

문제196. 이름, 월급, 순위를 출력하시오.

답)

```
import pandas as pd
```

```
emp=pd.read_csv("C:/python_data/emp.csv")
```

```
emp['rnk']=emp['sal'].rank(ascending=0).apply(int) #0=False, 1=True
```

```
print(emp[['ename','sal','rnk']])
```

	ename	sal	rnk
0	KING	5000	1
1	BLAKE	2850	5
2	CLARK	2450	6
3	JONES	2975	4
4	MARTIN	1250	10
5	ALLEN	1600	7
6	TURNER	1500	8
7	JAMES	950	13
8	WARD	1250	10
9	FORD	3000	2
10	SMITH	800	14
11	SCOTT	3000	2
12	ADAMS	1100	12
13	MILLER	1300	9

정렬하기

```
import pandas as pd
```

```
emp=pd.read_csv("C:/python_data/emp.csv")
```

```
emp['rnk']=emp['sal'].rank(ascending=0).apply(int)
```

```
emp=emp.sort_values('rnk',ascending=1)
```

```
print(emp[['ename','sal','rnk']])
```

	ename	sal	rnk
0	KING	5000	1
9	FORD	3000	2
11	SCOTT	3000	2
3	JONES	2975	4
1	BLAKE	2850	5
2	CLARK	2450	6
5	ALLEN	1600	7
6	TURNER	1500	8
13	MILLER	1300	9
4	MARTIN	1250	10
8	WARD	1250	10
12	ADAMS	1100	12
7	JAMES	950	13
10	SMITH	800	14

문제197. (점심시간)직업, 직업별 토탈월급, 순위를 출력하시오.

(순위는 직업별 토탈월급이 높은 순으로)

답)

```
import pandas as pd
```

```
emp=pd.read_csv("C:/python_data/emp.csv")
```

```
result=emp.groupby(['job'])['sal'].sum().reset_index()
```

```
result['rnk']=result['sal'].rank(ascending=0).apply(int)
```

```
result=result.sort_values('rnk',ascending=1)
```

```
print(result[['job','sal','rnk']])
```

	job	sal	rnk
2	MANAGER	8275	1
0	ANALYST	6000	2
4	SALESMAN	5600	3
3	PRESIDENT	5000	4
1	CLERK	4150	5

72. 판다스와 오라클 그룹함수 비교

문제193. 직업, 직업별 최대월급을 출력하시오.

보기)

```
SQL> select job, max(sal)
      from emp
      group by job;
```

답)

판다스o

```
import pandas as pd
```

```
emp=pd.read_csv("C:/python_data/emp.csv")
result=emp.groupby('job')['sal'].max()
print(result)
```

문제194. 부서번호, 직업, 부서번호별 직업별 토탈월급을 출력하시오.

보기)

```
SQL> select deptno, job, max(sal)
      from emp
      group by deptno,job;
```

답)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result=emp.groupby(['job','deptno'])['sal'].sum()
print(result)
```

job	deptno	sal
ANALYST	20	6000
CLERK	10	1300
	20	1900
	30	950
MANAGER	10	2450
	20	2975
	30	2850
PRESIDENT	10	5000
SALESMAN	30	5600

Name: sal, dtype: int64

문제195. 직업, 직업별 토탈월급을 출력하는데 직업이 SALESMAN은 제외하고 출력하고, 직업별 토탈월급이 6000이상인 것만 출력하시오.

답)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
emp['job']=emp['job'] [emp['job']!='SALESMAN']
```

```
result=emp.groupby('job')['sal'].sum()
```

```
result2=result>=6000
```

```
print(result[result2==True])
```

job	sal
ANALYST	6000
MANAGER	8275

Name: sal, dtype: int64

파이썬 149제(문자열, Comprehension)

2018년 7월 27일 금요일 오후 6:12

74. 문자열에서 특정위치 문자 얻기

```
ex)
txt1 = 'A tale that was not right'
txt2 = '이 또한 지나가리라.'
print(txt1[5])    # 'e'가 출력됨
print(txt2[-2])    # '다'가 출력됨
```

75. 문자열에서 지정한 구간의 문자열 얻기

```
ex)
txt1 = 'A tale that was not right'
txt2 = '이 또한 지나가리라.'
print(txt1[3:7])    # 'ale '이 출력됨
print(txt1[:6])    # 'A tale' 이 출력됨
print(txt2[-4:])    # '가리라.'가 출력됨
```

문제198. 리스트로 위의 코드를 연습하시오.

```
보기)
a=[1,2,3,4,5,6,7,8]
숫자 5이후는 다 출력되게 해보시오.
```

```
답)
print(a[4:])
[5, 6, 7, 8]
5포함해서 이전 숫자 출력
print(a[:5])
[1, 2, 3, 4, 5]
```

문제199. 아래의 리스트에서 중간 숫자를 출력하시오.

```
보기)
a=[1,2,3,4,5,6,7,8,9]
```

```
답)
a=[1,2,3,4,5,6,7,8,9]
b=int(len(a)/2)
print(a[b])
```


문제200. 아래의 리스트에서 중간에 해당하는 4, 5가 출력되게 하시오.

보기)

```
a=[1,2,3,4,5,6,7,8]
```

답)

```
a=[1,2,3,4,5,6,7,8]
```

```
b=int(len(a)/2)
```

```
print(a[b-1:b+1])
```

76. 문자열에서 홀수 번째 문자만 추출하기

ex)

```
txt = 'aAbBcCdDeEfFgGhHiJjKkK'
```

```
ret = txt[::2]
```

```
print(ret)
```

'abcdefghijk' 가 출력됨

문제201. 위의 문자열에서 짝수 번째만 출력하시오.

답)

```
txt = 'aAbBcCdDeEfFgGhHiJjKkK'
```

```
ret = txt[1::2]
```

```
print(ret)
```

77. 문자열을 거꾸로 만들기

ex)

```
txt = 'abcdefghijk'
```

```
ret = txt[::-1]
```

```
print(ret)
```

'kjihgfedcba' 가 출력됨

설명 : "파이썬의 리스트가 딱 스택구조를 가지고 있다.

문제202. 파이썬의 리스트가 스택구조임을 확인하시오.

답)

```
a=[]
```

```
a.append(1)
```

```
a.append(2)
```

```
print(a)
```

```
a.append(3)
```

```
print(a)
```

```
[1, 2]
```

```
[1, 2, 3]
```

문제203. 다시 숫자를 빼내시오. (나중에 들어온 숫자가 먼저 나가도록)

```
답)
a=[]
a.append(1)
a.append(2)
print(a)
a.append(3)
print(a)
a.pop()
print(a)
```

```
[1, 2]
[1, 2, 3]
[1, 2]
```

78. 두 개의 문자열 합치기

```
ex)
filename = input('저장할 파일 이름을 입력하세요: ')
filename = filename + '.jpg'
display_msg = '당신이 저장한 파일은 <' + filename + '> 입니다.'
print(display_msg)
저장할 파일 이름을 입력하세요: asd
당신이 저장한 파일은 <asd.jpg> 입니다.
```

79. 문자열을 반복해서 새로운 문자열로 만들기

```
ex)
msg1 = '여러분'
msg2 = '파이팅!'
display_msg = msg1 + ', ' + msg2*3 + '~!'
print(display_msg)
여러분, 파이팅!파이팅!파이팅!~!
```

80. 문자열에서 특정 문자가 있는지 확인하기(in)

```
ex)
msg = input('임의의 문장을 입력하세요: ')
if 'a' in msg:
    print('당신이 입력한 문장에는 a가 있습니다.')
else:
    print('당신이 입력한 문장에는 a가 없습니다.')
임의의 문장을 입력하세요: aqweqwea
당신이 입력한 문장에는 a가 있습니다.
```

문제204. 겨울왕국에서 elsa를 검색해서 몇건이 나오는지 출력하는 코드를 클래스로 만들어서

실행되게 하시오.

코드)

```
file_name=str( input('분석할 스크립트를 입력 : ') )
```

```
file = open("C:/python_data/"+file_name, 'r')
```

```
word=str( input('찾을 단어 입력 : ') )
```

```
from time import sleep
```

```
sum=0
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
    for b in a:
```

```
        sum+=b.lower().count(word)
```

```
        msg = '₩r 찾은 개수 : %d' %(sum)
```

```
        print(msg, end="")
```

```
        sleep(0.00001)
```

보기)

```
tm=Text_mining()
```

```
print(tm.find_word())
```

답)

```
class Text_mining():
```

```
    def find_word(self):
```

```
        from time import sleep
```

```
        file_name=str( input('분석할 스크립트를 입력 : ') )
```

```
        file = open("C:/python_data/"+file_name, 'r')
```

```
        word=str( input('찾을 단어 입력 : ') )
```

```
        sum=0
```

```
        for winter_list in file:
```

```
            a=winter_list.split(' ')
```

```
            for b in a:
```

```
                sum+=b.lower().count(word)
```

```

msg = 'Wr 찾은 개수 : %d' %(sum)
print(msg, end='')
sleep(0.00001)

```

```

tm=Text_mining()
tm.find_word()

```

문제205. 겨울왕국 대본에 단어들만 출력하시오

답)

```

def find_word():
    file=open("C:\python_data\winter.txt", "r")
    sum=0
    for winter_list in file:
        a=winter_list.split(' ')
        for b in a:
            print(b.lower())

```

```

find_word()

```

```

" final
shooting
draft

by

jennifer
lee

9/23/13

property
of
the
walt
disney
animation
studios

```

문제206. 위의 출력된 결과중에 엔터를 정제하시오.

답)

```

import re
def find_word():
    file=open("C:\python_data\winter.txt", "r")
    sum=0
    for winter_list in file:

```

```

a=winter_list.split(' ')
for b in a:
    print(re.sub("\Wn","",b).lower())

```

find_word()

```

"final
shooting
draft
by
jennifer
lee
9/23/13
property
of
the
walt
disney
animation
studios
open
on:
ice.
we' re
underwater

```

문제207. 긍정단어 텍스트 파일인 **positive-word.txt**에서 엔터 정제하고 단어만 추출하는 소문자로 출력하시오.

답)

```

import re
def find_word():
    file=open("C:\python_data\positive-words.txt", "r")
    sum=0
    for winter_list in file:
        a=winter_list.split(' ')
        for b in a:
            print(re.sub("\Wn","",b).lower())

```

find_word()

```

accolade
accolades
accommodative
accomodative
accomplish
accomplished
accomplishment
accomplishments
accurate
accurately
achievable
achievement
achievements
achievable
acumen
adaptable
adaptive
adequate
adjustable
admirable

```

문제208. 위의 긍정 단어들을 **positive**라는 비어있는 리스트를 만들고 **positive**라는 비어있는

리스트에 추가해서 담아 놓으시오.

답)

```
import re
```

```

def find_word():
    positive=[]
    file=open("C:\python_data\positive-words.txt", "r")
    sum=0
    for winter_list in file:
        a=winter_list.split(' ')
        for b in a:
            positive.append(re.sub("\Wn","",b).lower())
    print(positive)

```

```
find_word()
```

★★★★★★★★★★★★★★중요★★★★★★★★★★★★★★

문제209. 겨울왕국 대본에 긍정단어가 몇건이 있는지 출력하는 함수를 생성하시오.

보기)

```

print(positive_word() )
319

```

힌트)

for i in 겨울왕국 단어리스트:

```

for j in positive:
    if i == j:
        sum+=1

```

답)

```
import re
```

```
def positive_word():
```

```

    winter_list=[]
    file1=open("C:\python_data\winter.txt", "r")

```

```

    for winter in file1:
        a = winter.split(' ')
        for b in a :
            winter_list.append(re.sub('\Wn', '', b).lower())

```

```

    positive=[]
    file2=open("C:\python_data\positive-words.txt", "r")
    for winter in file2:
        a = winter.split(' ')
        for b in a :
            positive.append(re.sub('\Wn', '', b).lower())

```

```

    sum=0
    for i in winter_list:
        for j in positive:
            if i == j:
                sum+=1
    return sum

```

```
print(positive_word())
```

419

문제210. 셜록 홈즈 영화 대본에서 알파벳 외에 문자나 숫자는 제외 하시오.

답)

```
import re
```

```

for j in open("c:/python_data/Sherlock.txt", 'r'): # 셜록홈즈 로딩
    kq = re.sub('[^A-z ]', '', j).lower() #re의 sub를 써서 A-z아닌 것들을 "(공백)으로 바

```

꿔라

```
print (kq)

text on screen

so far on sherlock

sherlock unzips the body bag in a study in pink

sherlock at the door to the barts lab the names sher
he clickwinks at john
sherlock afternoon
he leaves the lab
mike stamford to john yeah hes always like that

brief shot of sherlock in his security mans uniform

sherlock flogs the dead body in asip
molly bad day was it

in the warehouse in asip
mycroft since yesterday youve moved in with him
```

문제211. 위의 결과를 리스트에 담으시오.

답)

```
import re
```

```
for j in open("c:/python_data/Sherlock.txt", 'r'): # 셜록홈즈 로딩
```

```
    kq = re.sub('[^A-z ]', '', j).lower().split(' ') #re의 sub를 써서 A-z아닌 것들을 "(공백)
```

으로 바꿔라

```
    print (kq)

['text', 'on', 'screen']
['']
['so', 'far', 'on', 'sherlock']
['']
['']
['sherlock', 'unzips', 'the', 'body',
['']
['sherlock', 'at', 'the', 'door', 'to
'sherlock', 'holmes', 'and', 'the', '
['he', 'clickwinks', 'at', 'john']
['sherlock', 'afternoon']
['he', 'leaves', 'the', 'lab']
['mike', 'stamford', 'to', 'john', 'y
['']
['brief', 'shot', 'of', 'sherlock', '
'at', 'the', 'hickman', 'gallery', 'i
['']
['sherlock', 'flogs', 'the', 'dead',
['molly', 'bad', 'day', 'was', 'it']
['']
```


문제212. 위의 결과를 공백을 없애라.

답)

1.단어들을 뽑아낸다.

```
import re

for j in open("c:/python_data/Sherlock.txt", 'r'):
    kq = re.sub('[^A-z ]', '', j).lower().split(' ')
    for k in kq:
        print(k)

text
on
screen

so
far
on
sher lock
```

2. if문을 넣어준다

```
import re

for j in open("c:/python_data/Sherlock.txt", 'r'):
    kq = re.sub('[^A-z ]', '', j).lower().split(' ')

    for k in kq:
        if k != "":
            print(k)

text
on
screen
so
far
on
sher lock
sher lock
unzips
the
body
bag
in
a
study
```

문제213. 정제 작업한 셴록홈즈 대본을 단어별로 분리해서 k_list에 담으시오.

답)

```
import re
```

```
k_list=[]
```

```
for j in open("c:/python_data/Sherlock.txt", 'r'):
```

```
    kq = re.sub('[^A-z ]', '', j).lower().split(' ')
```

```
    for k in kq:
```

```
        if k != "":
```

```
            k_list.append(k)
```

```
print(k_list)
```

```
['text', 'on', 'screen', 'so', 'far', 'on', 'sherlock', 'sherlock', 'unzips',  
'the', 'body', 'bag', 'in', 'a', 'study', 'in', 'pink', 'sherlock', 'at', 'th  
e', 'door', 'to', 'the', 'barts', 'lab', 'the', 'names', 'sherlock', 'holmes',  
'and', 'the', 'address', 'is', 'b', 'baker', 'street', 'he', 'clickwinks', 'a  
t', 'john', 'sherlock', 'afternoon', 'he', 'leaves', 'the', 'lab', 'mike', 'sta  
mford', 'to', 'john', 'yeah', 'hes', 'always', 'like', 'that', 'brief', 'shot',  
'of', 'sherlock', 'in', 'his', 'security', 'mans', 'uniform', 'at', 'the', 'hic  
kman', 'gallery', 'in', 'the', 'great', 'game', 'sherlock', 'flogs', 'the', 'de  
ad', 'body', 'in', 'asip', 'molly', 'bad', 'day', 'was', 'it', 'in', 'the', 'wa  
rehouse', 'in', 'asip', 'mycroft', 'since', 'yesterday', 'youve', 'moved', 'i  
n', 'with', 'him', 'theres', 'a', 'brief', 'shot', 'of', 'the', 'door', 'to',  
'b', 'closing', 'mycroft', 'and', 'now', 'youre', 'solving', 'crimes', 'togethe  
r', 'in', 'the', 'hallway', 'of', 'b', 'in', 'asip', 'sherlock', 'kisses', 'mr  
s', 'hudsons', 'cheek', 'mrs', 'hudson', 'look', 'at', 'you', 'all', 'happy',  
'its', 'not', 'decent', 'sherlock', 'who', 'cares', 'about', 'decent', 'the',  
'game', 'mrs', 'hudson', 'is', 'on', 'brief', 'shot', 'of', 'the', 'houses', 'o  
f', 'parliament', 'exploding', 'in', 'the', 'empty', 'hearse', '[which', 'is',  
'out', 'of', 'context', 'when', 'so', 'far', 'this', 'is', 'meant', 'to', 'be',  
'a', 'summary', 'of', 'the', 'season', 'episodes]', 'bs', 'living', 'room', 'i  
n', 'tgg', 'sherlock', 'dont', 'make', 'people', 'into', 'heroes', 'john', 'her
```

문제214. positive_words.txt 텍스트도 위에 셜록홈즈 처럼 단어별로 다 리스트화해서 p_list에

담으시오.

답)

```
import re
```

```
p_list=[]
```

```
for j in open("c:/python_data/positive-words.txt", 'r'):
```

```
    pq = re.sub('[^A-z ]', '', j).lower().split(' ')
```

```
    for p in pq:
```

```
        if p != "":
```

```
            p_list.append(p)
```

```
print(p_list)
```

```

['a', 'abound', 'abounds', 'abundance', 'abundant', 'accessible', 'accessible',
'acclaim', 'acclaimed', 'acclamation', 'accolade', 'accolades', 'accommodativ
e', 'accomodative', 'accomplish', 'accomplished', 'accomplishment', 'accomplish
ments', 'accurate', 'accurately', 'achievable', 'achievement', 'achievements',
'achievable', 'acumen', 'adaptable', 'adaptive', 'adequate', 'adjustable', 'adm
irable', 'admirably', 'admiration', 'admire', 'admirer', 'admiring', 'admirabl
y', 'adorable', 'adore', 'adored', 'adorer', 'adoring', 'adoringly', 'adroit',
'adroitly', 'adulate', 'adulation', 'adulatory', 'advanced', 'advantage', 'advan
tageous', 'advantageously', 'advantages', 'adventuresome', 'adventurous', 'adv
ocate', 'advocated', 'advocates', 'affability', 'affable', 'affably', 'affectat
ion', 'affection', 'affectionate', 'affinity', 'affirm', 'affirmation', 'affirm
ative', 'affluence', 'affluent', 'afford', 'affordable', 'affordably', 'afordab
le', 'agile', 'agilely', 'agility', 'agreeable', 'agreeableness', 'agreeably',
'allaround', 'alluring', 'alluringly', 'altruistic', 'altruistically', 'amaze',
'amazed', 'amazement', 'amazes', 'amazing', 'amazingly', 'ambitious', 'ambitiou
sly', 'ameliorate', 'amenable', 'amenity', 'amiability', 'amiably', 'amiable',
'amicability', 'amicable', 'amicably', 'amity', 'ample', 'amply', 'amuse', 'amu
sing', 'amusingly', 'angel', 'angelic', 'apotheosis', 'appeal', 'appealing', 'a
pplaud', 'appreciable', 'appreciate', 'appreciated', 'appreciates', 'appreciati
ve', 'appreciatively', 'appropriate', 'approval', 'approve', 'ardent', 'ardentl

```

문제215. 위의 스크립트의 결과를 보면 'a'가 포함되어있다. 원래 a+였는데 특수문자를 제거하

면서 'a'로 변경되었다. 그래서 다시 한 철자 이상인 것만 p_list에 담기게 하시오.

답)

```
import re
```

```
p_list=[]
```

```
for j in open("c:/python_data/positive-words.txt", 'r'):
```

```
    pq = re.sub('[^A-z ]', '', j).lower().split(' ')
```

```
    for p in pq:
```

```
        if p != " and len(p)>=2:
```

```
            p_list.append(p)
```

```
print(p_list)
```

['abound', 'abounds', 'abundance', 'abundant', 'accessible', 'accessible', 'acclaim', 'acclaimed', 'acclamation', 'accolade', 'accolades', 'accommodative', 'accommodative', 'accomplish', 'accomplished', 'accomplishment', 'accomplishments', 'accurate', 'accurately', 'achievable', 'achievement', 'achievements', 'achievable', 'acumen', 'adaptable', 'adaptive', 'adequate', 'adjustable', 'admirable', 'admirably', 'admiration', 'admire', 'admirer', 'admiring', 'admiringly', 'adorable', 'adore', 'adored', 'adorer', 'adoring', 'adoringly', 'adroit', 'adroitly', 'adulate', 'adulation', 'adulatory', 'advanced', 'advantage', 'advantageous', 'advantageously', 'advantages', 'adventuresome', 'adventurous', 'advocate', 'advocated', 'advocates', 'affability', 'affable', 'affably', 'affectation', 'affection', 'affectionate', 'affinity', 'affirm', 'affirmation', 'affirmative', 'affluence', 'affluent', 'afford', 'affordable', 'affordably', 'affordable', 'agile', 'agilely', 'agility', 'agreeable', 'agreeableness', 'agreeably', 'allaround', 'alluring', 'alluringly', 'altruistic', 'altruistically', 'amaze', 'amazed', 'amazement', 'amazes', 'amazing', 'amazingly', 'ambitious', 'ambitiously', 'ameliorate', 'amenable', 'amenity', 'amiability', 'amiability', 'amiable', 'amicability', 'amicable', 'amicably', 'amity', 'ample', 'amply', 'amuse', 'amusing', 'amusingly', 'angel', 'angelic', 'apotheosis', 'appeal', 'appealing', 'applaud', 'appreciable', 'appreciate', 'appreciated', 'appreciates', 'appreciative', 'appreciatively', 'appropriate', 'approval', 'approve', 'ardent', 'ardently', 'ardor', 'articulate', 'aspiration', 'aspirations', 'aspire', 'assurance', 'assurances', 'assure', 'assuredly', 'assuring', 'astonish', 'astonished', 'astonishing', 'astonishingly', 'astonishment', 'astound', 'astounded', 'astounding', 'astoundingly', 'astutely', 'attentive', 'attraction', 'attractive', 'attractively', 'attune', 'audible', 'audibly', 'auspicious', 'authentic', 'authoritative', 'autonomous', 'available', 'aver', 'avid', 'avidly', 'award', 'awarded', 'awards', 'awe', 'awed', 'awesome', 'awesomely', 'awesomeness', 'awestruck', 'awso

문제216. k_list에 p_list에 단어가 얼마나 포함되어 있는지 카운트 하시오.

답)

```
import re
```

```
k_list=[]
```

```
for i in open("c:/python_data/Sherlock.txt", 'r'):
```

```
    kq = re.sub('[^A-z ]', '', i).lower().split(' ')
```

```
    for k in kq:
```

```
        if k != "":
```

```
            k_list.append(k)
```

```
p_list=[]
```

```
for j in open("c:/python_data/positive-words.txt", 'r'):
```

```
    pq = re.sub('[^A-z ]', '', j).lower().split(' ')
```

```
    for p in pq:
```

```
        if p != "" and len(p)>=2:
```

```
            p_list.append(p)
```

```
cnt=0
```

```
for x in k_list: #셜록홈즈 리스트에서 하나씩 뽑는것
```

```
    for y in p_list: #긍정단어 리스트에서 하나씩 뽑는것
```

```

        if x==y:
            cnt+=1

print(cnt)
529

```

문제217. 위의 스크립트가 수행된 시간을 확인하시오.

```

답)
import re
import time

start_time = time.time() #현재 실행할 스크립트 시간 담기

k_list=[]
for i in open("c:/python_data/Sherlock.txt", 'r'):
    kq = re.sub('[^A-z ]', '', i).lower().split(' ')
    for k in kq:
        if k != "":
            k_list.append(k)

p_list=[]
for j in open("c:/python_data/positive-words.txt", 'r'):
    pq = re.sub('[^A-z ]', '', j).lower().split(' ')
    for p in pq:
        if p != "" and len(p)>=2:
            p_list.append(p)

cnt=0
for x in k_list: #셜록홈즈 리스트에서 하나씩 뽑는것
    for y in p_list: #긍정단어 리스트에서 하나씩 뽑는것
        if x==y:
            cnt+=1

end = time.time() #끝나는 시간 담기

print(cnt, '걸린시간: ', round(end-start_time,1))
529 걸린시간:  3.6

```

문제218. (점심시간) mydata3.txt를 m_list라는 비어있는 변수에 단어별로 쪼개서 담으시오.

답)

```
m_list=[]
```

```
for i in open("c:/python_data/mydata3.txt", 'r', encoding='UTF8'):
```

```
    mq = re.sub('[-=, #/?:$}▲△◇"₩'WWn"]', '', i).split(' ')
```

```
    for m in mq:
```

```
        if m != "":
```

```
            m_list.append(m)
```

```
print(m_list)
```

```
['신한은행', '클라우드', '기반', '인공지능', '코어', '플랫폼', '금융권',  
'로', '상용화신한은행이', '클라우드', '기반의', '인공지능(AI)', '코어',  
'을', '금융권', '최초로', '상용화했다', '빅데이터센터', '직원이', 'AI',  
'플랫폼을', '이용한', '분석', '화면을', '모니터링하고', '있다', '박지호',  
'jihopress@etnews.comBIM인공지능', '건설자동화', '기술', '2025년까지',  
'건설', '사망자', '30%정부가', '인공지능(AI)과', '빌딩정보시스템(BIM)을',  
'한', '건설', '자동화', '기술을', '2025년까지', '개발한다', 'BIM', '활용',  
'도하기', '위해', '2020년까지', '500억원', '이상', '도로', '사업에', 'BI',  
'계를', '의무화한다', '첨단', '기술로', '건설현장', '노동생산성을', '40%',  
'시키고', '안전사고로', '인한', '사망자', '수를', '30%', '줄인다', '국토',  
'는', '향후', '5년간(2018~2022년)', '건설기술정책', '로드맵인', '제6차',  
'기술진흥', '기본계획을', '수립했다고', '2일', '밝혔다', '국내', '건설신',  
'전통산업', '이미지를', '벗어나지', '못하고', '있다', '생산성이', '20년
```

문제219. 위의 기사에서 조사를 빼버리고 단어만 출력될 수 있도록 하시오.

(리눅스 환경에서만 된다..... 생각 해야할 문제)

답)

1. 아나콘다 프롬프트에서

```
pip install konlpy
```

```
pip install JPYe1
```

을 깔아준다.

```
'python -m pip install --upgrade pip'
```

```
from konlpy.tag import Twitter
```

```
str_file = input('분석 할 스크립트 파일 입력: ')
```

```
file = open("C:\python_data\%s" %str_file, mode="r", encoding="UTF-8").read()
```

```
npl = Twitter()
```

```
file_word = npl.nouns(file)
```

```
print(file_word)
```

문제220. mydata3.txt를 가지고 워드 클라우드를 그리시오.

답)

```
from wordcloud import WordCloud, STOPWORDS    # 워드 클라우딩 모듈
import matplotlib.pyplot as plt              # 시각화 모듈
from os import path    # 텍스트 파일을 불러오기 위한 open, path 하기 위해 os 임
포트
import re

d = path.dirname("c://python_data//")    # 텍스트 파일이 있는 상위 디렉토리를 path
로 지정
text = open(path.join(d, "mydata3.txt"), mode="r", encoding="UTF-8").read()
# 텍스트파일을 open 하는데 reading만 되게 (mode="r"), UTF-8 방식으로 불러옴
(UTF-8)

text = re.sub("있다", "", text)
text = re.sub("있는", "", text)
text = re.sub("하지만", "", text)
text = re.sub("것이다", "", text)
text = re.sub("대한", "", text)
text = re.sub("통해", "", text)
text = re.sub("함께", "", text)
text = re.sub("인공지능", "", text)

text = re.sub("hani", "", text)
text = re.sub("한다", "", text)
text = re.sub("하는", "", text)
text = re.sub("위해", "", text)
text = re.sub("co", "", text)
text = re.sub("kr", "", text)
text = re.sub("위한", "", text)
text = re.sub("했다", "", text)
text = re.sub("같은", "", text)
text = re.sub("것은", "", text)

wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs_ttf',
# 폰트 위치(거의 기본적으로 C://Windows//Fonts 안에 들어있습니다)
                        stopwords=STOPWORDS, background_color='white',
# STOPWORDS 옵션은 공백/줄바꾸기 기준으로 단어를 추출해 냅니다
                        width=1000,
```

```
# background_color는 워드클라우드 배경색을 나타냅니다. 'black'으로하면 검은색
height=800,
# width와 height는 워드클라우드의 크기를 지정해 줍니다.
colormap='jet').generate(text)
# colormap은 워드 색깔을 지정해주는데 첨부한 색감표를 사용하시면 됩니다.
generate() 메소드는
```

```
plt.figure(figsize=(13,13)) # matplotlib의 pyplot을 figsize로 생성합니다
plt.imshow(wordcloud) # 워드 클라우드 이미지를 pyplot에 띄웁니다
plt.axis("off") # pyplot에 x, y축 표시를 없앱니다.
plt.show()
```



문제221. 위의 코드를 수정해서 워드 클라우드 그릴 텍스트 파일을 물어보게하고 입력하면 바로 워드 클라우드가 그려질 수 있게 코드를 수정하시오.

보기)

워드 클라우드를 그릴 텍스트를 입력하세요~ mytest3.txt

ebs레이디 버그 게시판 글들 : ladybug4.txt

안철수 연설문 : ann.txt

답)


```

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from os import path
import re

d = path.dirname("c://python_data//")
file = input('워드 클라우드를 그릴 텍스트를 입력하세요 : ')
text = open(path.join(d, file), mode="r", encoding="UTF-8").read()

text = re.sub("있다", "", text)
text = re.sub("있는", "", text)
text = re.sub("하지만", "", text)
text = re.sub("것이다", "", text)
text = re.sub("대한", "", text)
text = re.sub("통해", "", text)
text = re.sub("함께", "", text)
text = re.sub("인공지능", "", text)

text = re.sub("hani", "", text)
text = re.sub("한다", "", text)
text = re.sub("하는", "", text)
text = re.sub("위해", "", text)
text = re.sub("co", "", text)
text = re.sub("kr", "", text)
text = re.sub("위한", "", text)
text = re.sub("했다", "", text)
text = re.sub("같은", "", text)
text = re.sub("것은", "", text)

wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs_ttf',
                      stopwords=STOPWORDS, background_color='white',
                      width=1000,
                      height=800,
                      colormap='jet').generate(text)

plt.figure(figsize=(13,13))
plt.imshow(wordcloud)
plt.axis("off")

```

```
plt.show()
```

문제222. 워드클라우드를 그리는 함수를 클래스로

생성해서 아래와 같이 실행되게 하시오.(한글)

보기)

```
word=Word_cloud()
```

```
word.paint()
```

워드 클라우드를 그릴 텍스트를 입력하세요~ mydata3.txt

답)

```
class Word_cloud():
```

```
    def paint(self):
```

```
        from wordcloud import WordCloud, STOPWORDS
```

```
        import matplotlib.pyplot as plt
```

```
        from os import path
```

```
        import re
```

```
        d = path.dirname("c://python_data//")
```

```
        file = input('워드 클라우드를 그릴 텍스트를 입력하세요 : ')
```

```
        encoding_n=input('인코딩 옵션 입력 ex) : UTF-8')
```

```
        text = open(path.join(d, file), mode="r", encoding=encoding_n).read()
```

```
        text = re.sub("있다"," ",text)
```

```
        text = re.sub("있는"," ",text)
```

```
        text = re.sub("하지만"," ",text)
```

```
        text = re.sub("것이다"," ",text)
```

```
        text = re.sub("대한"," ",text)
```

```
        text = re.sub("통해"," ",text)
```

```
        text = re.sub("함께"," ",text)
```

```
        text = re.sub("인공지능"," ",text)
```

```
        text = re.sub("hani"," ",text)
```

```
        text = re.sub("한다"," ",text)
```

```
        text = re.sub("하는"," ",text)
```

```
        text = re.sub("위해"," ",text)
```

```
        text = re.sub("co"," ",text)
```

```
        text = re.sub("kr"," ",text)
```

```
        text = re.sub("위한"," ",text)
```

```
        text = re.sub("했다"," ",text)
```

```
        text = re.sub("같은"," ",text)
```


ex) 이렇게하면 파이썬 못배운사람들

```
oldlist=[1,2,'A',False,3]
newlist=[]
for i in oldlist:
    if type(i)==int:
        newlist.append(i*i)
```

```
print(newlist)
[1, 4, 9]
```

문제223. 위의 코드를 **Comprehension** 코드로 변경하시오.

답)

```
oldlist=[1,2,'A',False,3]
newlist=[i*i for i in oldlist if type(i)==int]
print(newlist)
```

2. set Comprehension

{ 출력 표현식 for 요소 in 입력시퀀스 if 조건식 } #if 조건식은 생략가능

ex)

```
oldlist=[1,1,2,3,3,4]
newlist={i*i for i in oldlist}
print(newlist)
```

```
{1,1,4,9,9,16}
```

↓

```
{16, 1, 4, 9}
```

설명: 중복이 제거된다

문제224. 위의 코드를 **comprehension** 을 안쓴 코드로 작성하시오.

답)

```
oldlist=[1,1,2,3,3,4]
newlist=[]
for i in oldlist:
    newlist.append(i*i)
newlist2=set(newlist)
print(newlist2)
```

3. Dictionary Comprehension

{ Key:Value for 요소 in 입력시퀀스 if 조건식 } #if 조건식은 생략가능

ex)

```
id_name={1:'신현수', 2:'김광록', 3:'김원섭'}
a={val:key for key, val in id_name.items()}
print(a)

{'신현수': 1, '김광록': 2, '김원섭': 3}
```

문제225. 위의 코드를 comprehension을 안쓴 코드로 작성하시오.

답)

```
id_name={1:'신현수', 2:'김광록', 3:'김원섭'}
a={}
for key,val in id_name.items():
    a[val]=key
print(a)

{'신현수': 1, '김광록': 2, '김원섭': 3}
```

문제226. 판다스를 이용하지말고 comprehension을 활용해서 사원 테이블의 토탈월급을 출력하시오.

답)

```
import csv
file = open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)

sumsal=sum(int(emp_list[5]) for emp_list in emp_csv)
print(sumsal)
29025
```

max, min도 되는데 따로따로 실행해야함

문제227. 판다스를 이용하지말고 comprehension을 활용해서 사원테이블의 최대 월급을 출력

하시오.

답)

```
import csv
file = open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)

maxsal=max(int(emp_list[5]) for emp_list in emp_csv)
print(maxsal)
```

82. 문자열이 알파벳인지 검사하기(isalpha)

- 설명 : 1. 알파벳이 맞는지 확인 : isalpha
2. 숫자가 맞는지 확인 : isdigit

3. 공백이 맞는지 확인 : isspace

문제228. 겨울왕국에 알파벳 문자의 개수가 어떻게 되는가?

답)

```
text_file=open('C:\python_data\winter.txt','r')
lines=text_file.readlines()
total=0
```

```
for s in lines:
    cnt=sum(i.isalpha() for i in s)
    print(s,cnt)
    total+=cnt
print(total)
```

```
"Final Shooting Draft
18
by
2
Jennifer Lee
11
9/23/13
0
Property of the Walt Di
39
OPEN ON: ICE.
9
We' re underwater looki
42
... ..
```

| 82180

문제229. 겨울왕국에 숫자의 개수가 몇 개인가?

답)

```
text_file=open('C:\python_data\winter.txt','r')
lines=text_file.readlines()
```

```
total=0
for s in lines:
    cnt=sum(i.isdigit() for i in s)
    print(s,cnt)
    total+=cnt
print(total)
```

```
Anna rolls onto her b
0
FROZEN - J. Lee 2
1
```

```

~
Anna rolls onto her k
0
FROZEN - J. Lee 2
1
YOUNG ANNA
0
(drama queen-ish)
0
I just can' t. The sk
0

```

83. 문자열에 특수문자가 있는지 검사하기

ex)

겨울왕국에는 특수문자가 몇 개나 있는지 확인

특수문자 = 전체문자 - (문자+숫자+공백)

```

text_file=open('C:\python_data\winter.txt','r')
lines=text_file.readlines()

```

```

total1=0

```

```

total2=0

```

```

for s in lines:

```

```

    word_cnt = sum(i.isalpha() for i in s)

```

```

    num_cnt=sum(i.isdigit() for i in s)

```

```

    space_cnt=sum(i.isspace() for i in s)

```

```

    cnt = len(s)

```

```

    total1+=cnt

```

```

    total2+=(word_cnt+num_cnt+space_cnt)

```

```

print(total1 - total2)

```

문제230. text_mining클래스에 위의 스크립트를 special_cnt라는 함수로 만들어서 추가하고 아

래와 같이 실행되게 하시오.

보기)

```

from my_loc import text_mining

```

```

tm=Text_mining()

```

```

tm.special_cnt()

```

답)

```

def special_cnt(self):

```

```

    file_name=str( input('특수문자 찾을 파일 이름 입력 : ') )

```

```

    file = open("C:/python_data/"+file_name, 'r')

```

```

lines=file.readlines()

total1=0
total2=0

for s in lines:
    word_cnt = sum(i.isalpha() for i in s)
    num_cnt=sum(i.isdigit() for i in s)
    space_cnt=sum(i.isspace() for i in s)
    cnt = len(s)
    total1+=cnt
    total2+=(word_cnt+num_cnt+space_cnt)
print(total1 - total2)

```

스크립트를 입력하세요~ Sherlock.txt
8685

```

from my_loc import text_mining
tm=Text_mining()
tm.special_cnt()
특수문자 찾을 파일 이름 입력 : Sherlock.txt
8685

```

84. 문자열에서 대소문자 변환하기 (upper,lower)

```

ex)
txt = 'A lot of Things occur each day.'
ret1 = txt.upper()
ret2 = txt.lower()
print(ret1)      # 'A LOT OF THINGS OCCUR EACH DAY.'가 출력됨
print(ret2)      # 'a lot of things occur each day.'가 출력됨

```

85. 문자열에서 좌우 공백 제거하기(lstrip, rstrip, strip)

```

ex)
txt = ' 양쪽에 공백이 있는 문자열입니다. '
ret1 = txt.lstrip()
ret2 = txt.rstrip()
ret3 = txt.strip()
print('<'+txt+'>')
print('<'+ret1+'>')
print('<'+ret2+'>')
print('<'+ret3+'>')

```



```
< 양쪽에 공백이 있는 문자열입니다. >  
<양쪽에 공백이 있는 문자열입니다. >  
< 양쪽에 공백이 있는 문자열입니다.>  
<양쪽에 공백이 있는 문자열입니다.>
```

86. 문자열에서 있는 문자(열) 개수 구하기(count)

ex)

```
txt = 'A lot of things occur each day, every day.'  
word_count1 = txt.count('o')  
word_count2 = txt.count('day')  
word_count3 = txt.count(' ')  
print(word_count1) # 3이 출력됨  
print(word_count2) # 2가 출력됨  
print(word_count3) # 8이 출력됨
```

87. 문자열에서 특정 문자(열) 위치 찾기(find)

ex)

```
txt = 'A lot of things occur each day, every day.'  
offset1 = txt.find('e')  
offset2 = txt.find('day')  
offset3 = txt.find('day', 30)  
print(offset1) # 22가 출력됨  
print(offset2) # 27이 출력됨  
print(offset3) # 38이 출력됨
```

88. 문자열을 특정 문자(열로) 분리하기(split)

ex)

```
url = 'http://www.naver.com/news/today=20160831'  
log = 'name:홍길동 age:17 sex:남자 nation:조선'  
  
ret1 = url.split('/')  
print(ret1)  
  
ret2 = log.split()  
for data in ret2:  
    d1, d2 = data.split(':')  
    print('%s -> %s' %(d1, d2))
```

```
['http:', '', 'www.naver.com', 'news', 'today=20160831']  
name -> 홍길동  
age -> 17  
sex -> 남자  
nation -> 조선
```

89. 문자열을 특정 문자(열)로 결합하기(join)

ex)

```
loglist = ['2016/08/26 10:12:11', '200', 'OK', '이 또한 지나가리라']
```

```
bond = ','
```

```
log = bond.join(loglist)
```

```
print(log)
```

```
2016/08/26 10:12:11,200,OK,이 또한 지나가리라
```

90. 문자열에서 특정 문자(열)을 다른 문자(열)로 바꾸기(replace)

ex)

```
txt = 'My password is 1234'
```

```
ret1 = txt.replace('1', '0')
```

```
ret2 = txt.replace('1', 'python')
```

```
print(ret1) # 'My Password is 0234'가 출력됨
```

```
print(ret2) # 'My Password is python234'가 출력됨
```

```
txt = '매일 많은 일들이 일어납니다.'
```

```
ret3 = txt.replace('매일', '항상')
```

```
ret4 = ret3.replace('일', '사건')
```

```
print(ret3) # '항상 많은 일들이 일어납니다.'가 출력됨
```

```
print(ret4) # '항상 많은 사건들이 사건어납니다.'가 출력됨
```

91. 문자열을 바이트 객체로 바꾸기(encode)

encoding이란?

"사람이 인지할 수 있는 형태의 데이터를 약속된 규칙에 의해 컴퓨터가 사용하는 0과 1로

변환하는 과정"

컴퓨터의 발명, 발전이 미국의 학계와 기업을 중심으로 이루어지다보니 문자집합도 미국을 기준으로 재정되었다.

미국에서 재정된 ASCII(미국 정보 교환 표준 부호)는 1960년대에 재정된 문자집합으로 이후 개발된 문자집합의 토대를 이루고 있다.

ASCII는 7비트만을 이용해서 음이 아닌수(0~127)에 문자 집합내의 문자를 할당을 한 약속.

ex) 숫자 65 = 'A', 97 = 'a' 할당

10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자
64	0x40	@	80	0x50	P	96	0x60	`	112	0x70	p
65	0x41	A	81	0x51	Q	97	0x61	a	113	0x71	q
66	0x42	B	82	0x52	R	98	0x62	b	114	0x72	r
67	0x43	C	83	0x53	S	99	0x63	c	115	0x73	s
68	0x44	D	84	0x54	T	100	0x64	d	116	0x74	t
69	0x45	E	85	0x55	U	101	0x65	e	117	0x75	u
70	0x46	F	86	0x56	V	102	0x66	f	118	0x76	v
71	0x47	G	87	0x57	W	103	0x67	g	119	0x77	w
72	0x48	H	88	0x58	X	104	0x68	h	120	0x78	x
73	0x49	I	89	0x59	Y	105	0x69	i	121	0x79	y
74	0x4A	J	90	0x5A	Z	106	0x6A	j	122	0x7A	z
75	0x4B	K	91	0x5B	[107	0x6B	k	123	0x7B	{
76	0x4C	L	92	0x5C	\	108	0x6C	l	124	0x7C	
77	0x4D	M	93	0x5D]	109	0x6D	m	125	0x7D	}
78	0x4E	N	94	0x5E	^	110	0x6E	n	126	0x7E	~
79	0x4F	O	95	0x5F	_	111	0x6F	o	127	0x7F	DEL

ex)

```
u_txt = 'A'
b_txt = u_txt.encode()
print(u_txt)
print(b_txt[0])
A
65
```

아스키코드의 두가지 문제점

1. 포스코 더~~샵~~ 아파트 -> 포스코 더? 아파트
2. EUC-JP 를 지원하는 일본어용 소프트웨어를 한글 윈도우에서 실행하면 텍스트가 깨진다.

이 문제를 해결하기 위해 나온게 **유니코드** 이다.

UTF-7, UTF-8, UTF-16, UTF-32

이중에 **UTF-8**를 많이 사용한다.

92. 바이트 각체를 문자열로 바꾸기(decode)

ex)

```
b_txt = b'A lot of things occur each day.'
print(b_txt)
u_txt = b_txt.decode()
print(u_txt)
```

```
b'A lot of things occur each day.'  
A lot of things occur each day.
```

파이썬 149제(리스트)

2018년 8월 1일 수요일 오후 7:26

93. 순차적인 정수 리스트 만들기(range)

문제231. 아래의 리스트로 아래와 같은 결과를 출력하시오.

보기)

[0,12,3,4,5,6,7,8,9]

[10,11,12,13,14,15,16,17,18,19]

답)

```
list1=[]
```

```
for i in range(0,10):
```

```
    list1.append(i)
```

```
print(list1)
```

```
list2=[]
```

```
for i in range(10,20):
```

```
    list2.append(i)
```

```
print(list2)
```

문제232. Wt(탭)을 이용해서 아래의 결과를 출력하시오.

(카카오 공채 시험 문제 풀기위해 알아야함)

보기)

10 11 12 13 14 15 16 17 18 19

답)

```
for i in range(10,20):
```

```
    print (i, end='Wt')
```

10 11 12 13 14 15 16 17 18 19

문제233. 아래의 탭으로 구분된 단어를 리스트에 넣으시오.

(카카오 공채 시험 문제 풀기위해 알아야함)

보기)

```
song = 'YesterdayWtLet it beWtGee(지)Wt소원을 말해봐Wt다시 만난 세계'.split('Wt')
```

답)

```
k_pop=[]
```

```
k_pop=song
```

```

song = 'YesterdayWtLet it beWtGee(지)Wt소원을 말해봐Wt다시 만난 세계'.split('Wt')
print(k_pop)

['Yesterday', 'Let it be', 'Gee(지)', '소원을 말해봐', '다시 만난 세계']

```

94. 리스트에서 특정 위치의 요소 얻기

문제234. 아래의 리스트에서 숫자 2와 알파벳 b와 숫자5를 출력하시오.

```

보기)
listdata=[1,2,'a','b','c',[4,5,6]]

```

```

답)
listdata=[1,2,'a','b','c',[4,5,6]]
print(listdata[5][1])

```

95. copy와 deepcopy 모듈의 차이

1. 리스트 변수에 값을 복사하는 방법

```

a=[1,2,3]
b=a
print(b)

[1, 2, 3]

```

```

print(a[1]) #2

```

```

a[1]=4

```

```

print(a) #[1,4,3]

```

```

print(b) #[1,4,3]

```

#설명: 왜 1,4,3 이 나오는가? b=a 이 뜻은 a가 가리키는 리스트 객체를 b도 가리키겠다.

a가 가지고 있는 값으로 b변수를 생성할때 같은 값으로 복사하는 방법은?

1. [:]를 이용한다

```

a=[1,2,3]
a=[1,2,3]
b=a[:]
print(b) #[1,2,3]
a[1]=4
print(a) #[1,4,3]
print(b) #[1,2,3]

```

2. copy 모듈을 이용한다.

```

from copy import copy

```

```

a=[1,2,3]
b=copy(a)
print(b) #[1,2,3]
a[1]=6
print(a) #[1,6,3]
print(b) #[1,2,3]

```

3. deepcopy 모듈을 이용한다.

```

from copy import copy, deepcopy
a=[1,2,[3,4]]
b=copy(a)
print(b) #[1,2,[3,4]]
a[2][0]= 5

```

```

print(a)
print(b)

```

```

[1, 2, [3, 4]]
[1, 2, [5, 4]]
[1, 2, [5, 4]]

```

```

from copy import copy, deepcopy
a=[1,2,[3,4]]
b=deepcopy(a)
print(b) #[1,2,[3,4]]
a[2][0]= 5

```

```

print(a)
print(b)

```

```

[1, 2, [3, 4]]
[1, 2, [5, 4]]
[1, 2, [3, 4]]

```

96. 리스트에서 특정 요소의 위치 구하기(index)

문제235. 아래의 리스트를 가지고 아래의 결과를 출력하시오.

보기)

```

a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성', '지구']
planet = '지구'

```

결과)

지구는 태양계에서 3번째에 위치하고 있습니다.

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성', '지구']
planet = '지구'
print(planet,'는 태양계에서 ',a.index(planet),'번째에 위치하고 있습니다.')
print(planet,'는 태양계에서 ',a.index(planet,4),'번째에 위치하고 있습니다.')
```

```
지구 는 태양계에서 3 번째에 위치하고 있습니다.
지구 는 태양계에서 9 번째에 위치하고 있습니다.
```

a.index(planet,4)= 4번째 이후에 있는 인덱스를 가져옴

97. 리스트에서 특정 위치의 요소를 변경하기

문제236. 아래의 리스트에서 화성을 영어로 Mars라고 변경하시오.

(리스트의 index 메소드를 활용 해서 수행)

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성', '지구']
```

답)

```
mar_index=a.index("화성")
a[mar_index]='Mars'
print(a)
```

```
['태양', '수성', '금성', '지구', 'Mars', '목성', '토성', '천왕성', '해왕성']
```

98. 리스트에서 특정 구간에 있는 요소 추출하기

문제237. 아래의 리스트에서 아래의 결과를 출력하시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성', '지구']
```

결과)

태양계의 암석형 행성 : ['수성', '금성', '지구']

태양계의 가스형 행성 : ['화성', '목성', '토성', '천왕성', '해왕성']

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
am=a[1:4]
gas=a[4:]
print('태양계의 암석형 행성 : ',am)
```



```
print('태양계의 가스형 행성 : ', gas)
태양계의 암석형 행성 : ['수성', '금성', '지구']
태양계의 가스형 행성 : ['화성', '목성', '토성', '천왕성', '해왕성']
```

문제238. 아래의 리스트에서 짝수번째 요소만 출력하시오.

(카카오 블라인드 시험문제 풀기 위해 알아야 할 내용)

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

결과)

```
['수성', '지구', '목성', '천왕성']
```

답1)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
print(a[1::2])
```

```
['수성', '지구', '목성', '천왕성']
```

답2)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
for i in range(len(a)):
```

```
    if i%2 != 0:
```

```
        print(a[i], end=' ')
```

문제239. 이번에는 아래의 리스트에서 홀수 번째를 출력하시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
print(a[::2])
```

```
['태양', '금성', '화성', '토성', '해왕성']
```

99. 리스트 요소 순서를 역순으로 만들기1(reverse)

문제240. 아래의 리스트를 역순으로 만드시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

결과)

```
['해왕성', '천왕성', '토성', '목성', '화성', '지구', '금성', '수성', '태양']
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
a.reverse()
```

```
print(a)
```

100. 리스트 요소 순서를 역순으로 만들기2(reversed)

문제241. 이번에는 아래의 리스트를 역순으로 만들어서 b라는 변수에 저장해서 출력하시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
b=list(reversed(a))
```

```
print(b)
```

```
['해왕성', '천왕성', '토성', '목성', '화성', '지구', '금성', '수성', '태양']
```

101. 리스트 합치기(+)

문제242. 아래 두개의 리스트를 하나로 합쳐서 출력하시오.

보기)

```
listdata1 = ['a', 'b', 'c', 'd', 'e']
```

```
listdata2 = ['f', 'g', 'h', 'i', 'j']
```

답)

```
listdata1 = ['a', 'b', 'c', 'd', 'e']
```

```
listdata2 = ['f', 'g', 'h', 'i', 'j']
```

```
listdata3=listdata1+listdata2
```

```
print(listdata3)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

102. 리스트 반복하기(*)

문제243. 아래의 결과를 출력하시오.

```
보기)
[0,1,2,0,1,2,0,1,2]
```

```
답)
a = [0,1,2]
print(a*3)
[0, 1, 2, 0, 1, 2, 0, 1, 2]
```

103. 리스트에 요소 추가하기(append)

문제244. 아래와 같이 리스트에 추가할 값을 물어보게하고 입력하면 리스트의 내용이 출력 되게

하시오.

```
보기)
print(add_list())
리스트에 추가할 값을 입력 : a
['a']
리스트에 추가할 값을 입력 : b
['a','b']
리스트에 추가할 값을 입력 : c
['a','b','c']

답)
def add_list():
    a_list=[]
    while True:
        s=input('리스트에 추가할 값을 입력 : ')
        a_list.append(s)
        if len(a_list)==3:
            break
    return a_list
print(add_list())
리스트에 추가할 값을 입력 : a
리스트에 추가할 값을 입력 : b
리스트에 추가할 값을 입력 : c
['a', 'b', 'c']
```

104. 리스트의 특정 위치에 요소 삽입하기(insert)

문제245. 아래의 리스트에 목성 전에 소행성을 입력하시오.

```
보기)
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

결과)

```
a= ['태양', '수성', '금성', '지구', '화성', '소행성', '목성', '토성', '천왕성', '해왕성']
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
a_index=a.index('목성')
a.insert(a_index,'소행성')
print(a)
```

간단하게 이렇게 해도된다.

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
#a_index=a.index('목성')
a.insert(5,'소행성')
print(a)
```

105. 리스트의 특정 위치(index)의 요소 제거하기(del)

문제246. 아래의 리스트에서 del 메소드를 이용해서 토성을 지우시오.(숫자로 없애기)

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del(a[-3])
print(a)
```

문제247. 아래의 리스트에서 목성 이후의 요소를 다 지우시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del(a[-4:])
print(a)
['태양', '수성', '금성', '지구', '화성']
```

문제248. 아래의 리스트의 모든 요소를 다 삭제하시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
print(a)
[]
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del(a[:])
print(a)
```

106. 리스트에서 특정 요소 제거하기(remove)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']

a.remove('천왕성')
print(a)
['태양', '수성', '금성', '지구', '화성', '목성', '토성', '해왕성']
```

del 과 remove의 차이점?

: del은 리스트 **인덱스 번호**로 삭제, remove는 **요소명**으로 삭제

문제249. len과 count의 차이는 무엇인가?

```
답)
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
print(len(a))
print(a.count('지구'))
```

9
1

설명: len은 **리스트의 개수**를, count는 **요소의 개수**를

107. 리스트 요소 정렬하기 1(sort)

문제250. 아래의 리스트를 ABCD순으로 정렬하시오.

```
보기)
namelist = ['Mary', 'Sams', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']

답)
namelist = ['Mary', 'Sams', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']
namelist.sort()
print(namelist)
```

['Aimy', 'Bob', 'Kelly', 'Mary', 'Michale', 'Sams', 'Tom']

108. 리스트 요소 정렬하기 2 (sorted)

문제251. 아래의 리스트를 abcd순으로 정렬해서 b라는 변수에 담고 출력하시오.

보기)

```
namelist = ['Mary', 'Sams', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']
```

답)

```
namelist = ['Mary', 'Sams', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']
```

```
b=sorted(namelist,reverse=0)
```

```
print(b)
```

```
['Aimy', 'Bob', 'Kelly', 'Mary', 'Michale', 'Sams', 'Tom']
```

109. 리스트 요소 무작위로 섞기(shuffle)

(카카오 블라인드 테스트를 풀기 위해 알아야 할 내용4)

"순서용 자료를 뒤죽박죽 섞어놓은 함수"

ex)

```
from random import shuffle
```

```
listdata=list(range(1,11))
```

```
for i in range(3):
```

```
    shuffle(listdata)
```

```
    print(listdata)
```

```
[1, 6, 8, 9, 3, 7, 4, 2, 10, 5]
```

```
[9, 6, 2, 3, 1, 4, 7, 8, 10, 5]
```

```
[7, 8, 9, 2, 3, 4, 10, 1, 5, 6]
```

문제252. 아래의 리스트를 무작위로 섞어서 3번 출력하시오.

보기)

```
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee', '다만세']
```

답)

```
from random import shuffle
```

```
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee', '다만세']
```

```
for i in range(3):
```

```
    shuffle(music)
```

```
    print(music)
```

110. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate)

"열거하다라는 뜻인데 이 함수는 순서가 있는 자료형(리스트, 튜플, 문자형)을 입력받아 인덱스 값을 포함하는 enumerate 객체를 리턴한다."

```
ex)
for i,name in enumerate(['body','foo','bar']):
    print(i,name)
0 body
1 foo
2 bar
```

문제253. 아래의 리스트를 enumerate를 이용해서 아래의 결과로 출력하시오.

보기)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

결과)

```
[ (0,'태양'),(1,'수성'),(2,'금성'), ... (8,'해왕성') ]
```

답)

```
a= ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```
b=[]
```

```
for i, name in enumerate(a):
```

```
    b.append((i,name))
```

```
print(b)
```

```
[(0, '태양'), (1, '수성'), (2, '금성'), (3, '지구'), (4, '화성'), (5, '목성'), (6, '토성'), (7, '천왕성'), (8, '해왕성')]
```

111. 리스트의 모든 요소의 합 구하기(sum)

문제254. 아래 리스트의 요소를 전부 더한 값을 출력하시오.

보기)

```
listdata = [2, 2, 1, 3, 8, 5, 7, 6, 3, 6, 2, 3, 9, 4, 4]
```

답)

```
listdata = [2, 2, 1, 3, 8, 5, 7, 6, 3, 6, 2, 3, 9, 4, 4]
```

```
print(sum(listdata))
```

112. 리스트 요소가 전부 참인지 확인하기(all, any)

문제255. 아래의 리스트 요소중에 하나라도 True가 있으면 True를 출력하시오.

보기)

```
a=[True,False,True,True]
```

답)

```
print(any(a)) #하나라도 True이면 True
```

```
print(all(a)) #전부 True이어야지 True
```

```
True
False
```


파이썬 149제(딕셔너리)

2018년 8월 1일 수요일 오후 7:27

113. 파이썬에서 _(언더바)를 사용하는 경우

1. 인터프리터에서 마지막 값을 저장할 경우
2. 값을 무시하고 싶을 경우
3. 변수나 함수 명에 특별한 의미를 부여하고 싶을 경우
4. 숫자 또는 문자값의 자릿수 구분을 위한 구분자로써 사용할 경우

1. 인터프리터에서 마지막 값을 저장할 경우

```
10
-
_*3
_30
10
-
_*3
_*20
_600
```

2-1. 값을 무시하고 싶을 경우

```
x,y=(1,2)
x #1
y #2
x,_y=(1,2,3)
x #1
y #3
```

2-2. 여러 개 값을 무시하고 싶을 경우

```
x,*_y=(1,2,3,4,5)
print(x)
print(y)

1
5
```

2-3. for loop문 사용시 index 무시하고 싶을 경우

```
for _ in range(10):
    print('a')
```

a
a
a
a
a
a
a
a
a
a

3. 변수나 함수명에 특별한 의미를 부여하고 싶을 경우

예) `__init__` : 클래스가 객체화 될때 반드시 실행되는 메소드

예) 폴더안에 어떠한 파일이 있어야 이 폴더가 패키지가 될 수 있다고 했는데 그 파일이?

`__init__.py`

4. 숫자 또는 문자값의 자릿수 구분을 위한 구분자로써 사용할 경우

```
dec_base=1_000_000_000
print(dec_base)
1000000000
```

114. 사전에 요소 추가하기

```
ex)
sol={}
sol['태양']='sun'

print(sol)
{'태양': 'sun'}
```

문제256. 아래의 두개의 리스트를 이용해서 아래 딕셔너리를 생성하시오.

(카카오 블라인드 알아야할 내용)

보기)

```
solar1 = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
solar2 = ['Sun', 'Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']
```

결과)

```
{'태양': 'Sun', '수성': 'Mercury', '금성': 'Venus', '지구': 'Earth', '화성': 'Mars', '목성': 'Jupiter', '토성': 'Saturn', '천왕성': 'Uranus', '해왕성': 'Neptune'}
```

답)

```
solar1 = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
```

```

solar2 = ['Sun', 'Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus',
'Neptune']
solar3={}
for i,j in enumerate(solar1):
    val=solar2[i]
    solar3[j]=val

print(solar3)
{'태양': 'Sun', '수성': 'Mercury', '금성': 'Venus', '지구': 'Earth', '화성': 'Mars',
'목성': 'Jupiter', '토성': 'Saturn', '천왕성': 'Uranus', '해왕성': 'Neptune'}

```

문제257. (카카오 디러닝 파트 수시모집 문제를 풀기위한 문제)

아래 두개의 리스트를 아래의 딕셔너리로 출력하시오.

보기)

```
artist = ['Beatles', '소녀시대']
```

```
music = ['Yesterday', '소원을 말해봐']
```

결과)

```
{'Beatles': 'Yesterday', '소녀시대': '소원을 말해봐'}
```

답)

```
artist = ['Beatles', '소녀시대']
```

```
music = ['Yesterday', '소원을 말해봐']
```

```
dic={}

```

```
for i,j in enumerate(artist):

```

```
    val=music[i]

```

```
    dic[j]=val

```

```
print(dic)

```

- **default 딕셔너리를 구성하는 방법**

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

결과)

```
[ ('a',2), ('b',3), ('c',2) ]
```

답1)

setdefault 이용

```
for k in s:

```

```
    d.setdefault(k,0) # 기본 딕셔너리의 초기값을 0으로 설정

```

```
    print(d)

```

```
d[k]+=1
```

```
print(list(d.items()))
{'a': 0}
{'a': 1, 'b': 0}
{'a': 1, 'b': 1, 'c': 0}
{'a': 1, 'b': 1, 'c': 1}
{'a': 1, 'b': 2, 'c': 1}
{'a': 2, 'b': 2, 'c': 1}
{'a': 2, 'b': 3, 'c': 1}
[('a', 2), ('b', 3), ('c', 2)]
```

답2)

Counter 이용

```
import collections
s=['a','b','c','b','a','b','c']
c=collections.Counter(s)
print(list(c.items()))
[('a', 2), ('b', 3), ('c', 2)]
```

답3)

defaultdict 이용 (dictionary의 기본값을 정의하고 값이 없더라도 에러를 출력하지 않고

기본값을 출력한다.)

```
ex)
d={}
d['one']='a'
print(d)
```

결과)

```
{'one': 'a'}
```

위의 코드를 다시 수행하는데 아래처럼 해보시오.

```
d={}
d['one']
print(d)
```

```
Traceback (most recent call last):
  <ipython-input-177-b86603211b0c> in <module>()
    1 d={}
--> 2 d['one']
    3 print(d)

KeyError: 'one'
```

에러가 난다!!!

그런데 defaultdict()을 사용하면 key만 생성하고 값이 없어도 에러가 나지 않는다.

```

from collections import defaultdict

d3=defaultdict(list)

d3['one']
d3['two']

print(d3)

for item in d3:
    print(item)

defaultdict(<class 'list'>, {'one': [], 'two': []})
one
two

```

문제258. 아래 두개의 리스트로 아래의 딕셔너리를 생성하시오.

```

보기)
artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee']

결과)
{ 'Beatles' : ['Yesterday', 'let it be'], '소녀시대' : ['소원을 말해봐', 'Gee'] }

```

```

답)
artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대']

music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee']

```

```

box=defaultdict(list)

for i,ar in enumerate(artist):
    box[ar].append(music[i])
    print(list(box.items()))

print(box)
[('Beatles', ['Yesterday'])]
[('Beatles', ['Yesterday', 'let it be'])]
[('Beatles', ['Yesterday', 'let it be']), ('소녀시대', ['소원을 말해봐'])]
[('Beatles', ['Yesterday', 'let it be']), ('소녀시대', ['소원을 말해봐', 'Gee'])]
defaultdict(<class 'list'>, {'Beatles': ['Yesterday', 'let it be'], '소녀시대': ['소원을 말해봐', 'Gee']})

```

문제259. (카카오 딥러닝 파트 알고리즘 문제)

문제 설명

- 음악명으로 구성된 플레이리스트를 해당 음악의 아티스트 명을 참고해서 동일한 아티스트의 곡이 인접하지 않도록 플레이리스트를 셔플하는 알고리즘을 구현하시오
- 아래의 아티클을 보고 적합한 알고리즘을 구현하시오

(<http://labs.spotify.com/2014/02/28/how-to-shuffle-songs/>)

참고

- 반드시 셔플 기능을 구현해야 합니다.
- 매 실행마다 임의로 섞인 결과를 반환해야 합니다.
- 10000개 샘플 기준 3초 이내에 동작해야 합니다.
- 아티클 내용 이외에도 더 좋은 알고리즘이 있으면 그것을 구현해도 좋습니다.
- 아티클을 주의 깊게 읽어보시고 글의 목적을 파악하셔서 코드를 작성하시는 것이 필요합니다.
- Expected Output의 결과는 임의로 셔플한 결과이므로 반드시 Expected Output과 같을 필요는 없습니다.
- 한글처리에 주의하세요.
- 시간제한은 240분 입니다.

입력

- 입력의 첫 줄에는 테스트 케이스의 개수 T가 주어진다 ($T \leq 10000$)
- 각 테스트 케이스는 2줄로 구성되어 있다.
- 첫 줄에는 플레이리스트가 TSV(Tab Separated Value) 형태로 주어지고 이어서 다음 줄에는 아티스트가 TSV 형태로 주어집니다.
- 각 플레이리스트, 아티스트의 최대 길이는 4096을 넘지 않는다.

출력

- 각 테스트 케이스 별로 한줄로 적당히 셔플된 플레이리스트를 TSV 포맷으로 출력한다.

ex)

입력

1

Yesterday Let it be Gee 소원을 말해봐

Beatles Beatles 소녀시대 소녀시대

출력

소원을 말해봐 Let it be Gee Yesterday

아래코드 참고)

```
song = 'YesterdayWtLet it beWtGee(지)Wt소원을 말해봐.split('Wt')
```

```
artist = 'BeatlesWtBeatlesWt소녀시대Wt소녀시대Wt소녀시대'.split('Wt')
```

[카카오 딥러닝 파트 수시 모집의 문제 요약]

1. 아래의 리스트를 이용해서 아래의 결과를 출력하는데 출력할때 마다 shuffle 이 되게 하시오 !
동일한 아티스트의 곡이 인접해 있으면 안된다

```
from collections import defaultdict
from random import shuffle
```

```
artist = ['Beatles','Beatles','소녀시대','소녀시대']
music=['Yesterday','let it be','소원을 말해봐','Gee']
```

결과 :

Yesterday Gee let it be 소원을 말해봐

문제260. 아래의 두개의 리스트를 딕셔너리에 담아서 출력하시오.

보기)

```
artist = ['Beatles','Beatles','소녀시대','소녀시대']
music=['Yesterday','let it be','소원을 말해봐','Gee']
```

답)

```
artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대']
```

```
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee']
```

```
box=defaultdict(list)
```

```
for i,ar in enumerate(artist):
```

```
    box[ar].append(music[i])
```

```
    print(list(box.items()))
```

```
print(box)
```

문제261. 문제260번을 defaultdict가 아니라 setdefault를 이용해서 수행하시오.

답)

```
artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대']
```

```
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee']
```

```
box={} #그냥 딕셔너리로 선언
```

```
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])
    print(list(box.items()))
```

```
print(box)
```

```
[('Beatles', ['Yesterday'])]
[('Beatles', ['Yesterday', 'let it be'])]
[('Beatles', ['Yesterday', 'let it be']), ('소녀시대', ['소원을 말해봐'])]
[('Beatles', ['Yesterday', 'let it be']), ('소녀시대', ['소원을 말해봐', 'Gee'])]
{'Beatles': ['Yesterday', 'let it be'], '소녀시대': ['소원을 말해봐', 'Gee']}
```

문제262. 아래의 곡을 더 추가해서 결과가 출력되게 하시오.

보기)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']
```

답)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']
```

```
box={}

```

```
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])

```

```
print(box)
```

```
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

115. 딕셔너리의 특정 요소값 변경하기

문제263. 아래의 딕셔너리 값중에 let it be를 Women으로 변경하시오.

보기)

```
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

답)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
```



```

music = ['Yesterday', 'let it be', 'hey jude','소원을 말해봐', 'Gee', '다만세']

box = {} # 일반 딕셔너리로 선언
for i, arti in enumerate(artist):
    box.setdefault(arti, []).append(music[i])
print(box)

box['Beatles'][1] = 'women'
print(box)
{'Beatles': ['Yesterday', 'let it be', 'hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
{'Beatles': ['Yesterday', 'women', 'hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}

```

문제264. 아래의 딕셔너리에서 소녀시대의 키와 값을 전부 삭제하시오.

```

보기)
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}

답)
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']

box={}
for i,ar in enumerate(artist):
    box.setdefault(ar, []).append(music[i])

del(box['소녀시대'])
print(box)

```

116. 딕셔너리의 모든 요소 제거하기(clear)

문제265. 아래의 리스트에 모든 요소를 다 삭제하시오.

```

보기)
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
결과)
{}

답)
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']

```

```

box={}
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])

box.clear()
print(box)

```

117. 딕셔너리에서 키만 추출하기(keys)

문제266. 아래의 딕셔너리에서 key에 해당하는 부분만 추출하시오.

보기)
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}

결과)
Beatles, 소녀시대

답)
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']

```

box={}
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])
print(box)
print(box.keys())
| dict_keys(['Beatles', '소녀시대'])

```

118. 딕셔너리에서 값만 추출하기(values)

문제267. 아래의 딕셔너리에서 값만 추출하시오.

보기)
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}

결과)
dict_values(['Yesterday', 'let it be', 'Hey jude'], ['소원을 말해봐', 'Gee', '다만세'])

답)
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']

```

box={}
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])
#print(box)
print(box.values())
dict_values(['Yesterday', 'let it be', 'Hey jude'], ['소원을 말해봐', 'Gee', '다만세'])

```

문제268. 아래 딕셔너리에서 아래의 노래만 추출하시오.

보기)

```
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

결과)

Yesterday, 소원을 말해봐

답)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
```

```
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']
```

```
box={}
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])
```

```
print(type(box))
```

```
box2=list(box.values())
```

```
print(box2[0][0],box2[1][0])
```

```
<class 'dict'>
Yesterday 소원을 말해봐
```

문제269. 아래의 리스트의 값들을 아래와 같이 출력하시오.

보기)

```
{'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

결과)

Yesterday 소원을 말해봐 let it be Gee Hey jude 다만세

답)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대']
```

```
music = ['Yesterday', 'let it be', 'Hey jude', '소원을 말해봐', 'Gee', '다만세']
```

```

box={}
for i,ar in enumerate(artist):
    box.setdefault(ar,[]).append(music[i])

print(type(box))
box2=list(box.values())
print(len(box2))
print(len(box.keys()))
print(box2[0][0],box2[1][0])

for j in range(len(box2)):
    for k in range(len(box.keys())+1):
        print(box2[k][j], end='Wt')
<class 'dict'>
2
2
Yesterday 소원을 말해봐
Yesterday 소원을 말해봐 let it be Gee Hey Jude 다만세

```

아래의 셔플 모듈을 사용한 코드를 돌려보시오

```
music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee', '다만세']
```

```

from random import shuffle
for i in range(3):
    shuffle(music)
    print(music)

```

문제270. 문제269코드에 shuffle 기능을 추가해서 코드가 수행될때 마다 곡이 섞여서 출력 되게

하십시오.(단, 동일한 아티스트 노래가 겹치지 않도록)

답)

```
from collections import defaultdict
```

```

artist = ['Beatles', 'Beatles', 'Beatles','소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey Jude','소원을 말해봐', 'Gee', '다만세']

```

```
box = {} # 그냥 딕셔너리로 선언
```

```

for i, k in enumerate(artist):
    box.setdefault(k,[]).append(music[i])

```

```

        shuffle(box[k])

from random import shuffle

b = list(box.values())
shuffle(b)

for i in range(len(b)+1):
    for j in range(len(b)):
        res = b[j][i]
        print(res, end=' ')
    Yesterday kissing you Hey Jude 소원을 말해봐 let it be Gee

```

문제271. (점심시간) 아래와 같이 곡이 소녀시대가 한곡이 더 있는 상태에서 구현해보시오.

보기)

```

artist = ['Beatles', 'Beatles', 'Beatles','소녀시대', '소녀시대', '소녀시대','소녀시대']
music = ['Yesterday', 'let it be', 'Hey Jude','소원을 말해봐', 'Gee', '다만세','kissing
you']

```

답)

```

from collections import defaultdict

```

```

artist = ['Beatles', 'Beatles', 'Beatles','소녀시대', '소녀시대', '소녀시대','소녀시대']
music = ['Yesterday', 'let it be', 'Hey Jude','소원을 말해봐', 'Gee', '다만세','kissing
you']

```

```

box = {} # 그냥 딕셔너리로 선언

```

```

for i, k in enumerate(artist):
    box.setdefault(k,[]).append(music[i])
    shuffle(box[k])

```

```

from random import shuffle

```

```

b = list(box.values())
shuffle(b)

```

```

if len(b[0])%2 == 1:
    print(b[1][len(b[1])-1], end=' ')

```

```

        for i in range(len(b)+1):
            for j in range(len(b)):
                res = b[j][i]
                print(res, end=' ')
    else:
        for i in range(len(b)+1):
            for j in range(len(b)):
                res = b[j][i]
                print(res, end=' ')
        print(b[0][len(b[0])-1], end=' ')
| Gee   Hey Jude   다만세   Yesterday   kissing you   let it be   소원을 말해봐

```

119. 딕셔너리에 있는 모든 요소 추출하기(items)

보기)

```
box={'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

• 딕셔너리 자료형에서 데이터를 추출 하는 방법

1. key 추출 : box.keys()
2. 값 추출 : box.values()
3. 둘다 추출 : box.items()

문제272. 아래 딕셔너리의 모든 아이템을 다 추출하시오.

보기)

```
box={'Beatles': ['Yesterday', 'let it be', 'Hey jude'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}
```

답)

```
artist = ['Beatles', 'Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대', '소녀시대']
music = ['Yesterday', 'let it be', 'Hey Jude', '소원을 말해봐', 'Gee', '다만세', 'kissing you']
```

```
box = {} # 그냥 딕셔너리로 선언
```

```
for i, k in enumerate(artist):
    box.setdefault(k, []).append(music[i])
```

```
print(box.items())
```

120. 딕셔너리 정렬하기(sorted)

문제273. 아래 딕셔너리 데이터의 키값을 ABCD 순으로 정렬해서 출력하시오.

보기)

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245, 'Michale':27115, 'ob':5887, 'Kelly':7855}
```

답)

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245, 'Michale':27115, 'ob':5887, 'Kelly':7855 }
```

```
print(sorted(names.keys()))
```

문제274. 위의 사전에 key를 정렬할때 아래와 같이 값도 같이 출력되게 하시오.

결과)

```
[('Aimy', 9778), ('Kelly', 7855), ('Mary', 10999), ('Michale', 27115), ('Sams', 2111), ('Tom', 20245), ('ob', 5887)]
```

답)

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245, 'Michale':27115, 'ob':5887, 'Kelly':7855 }
```

```
print(sorted(names.items() )
```

파이썬 149제(입,출력 및 etc)

2018년 8월 1일 수요일 오후 7:28

121. 문자 코드값 구하기(ord)

아스키 코드값

ex)

```
print(ord('A')) # 65
```

문제275. 알파벳 소문자 A~Z까지의 아스키 코드값을 출력하시오.

결과)

A->65

B->66

.

.

Z->90

답)

```
import string
```

```
a=string.ascii_uppercase
```

```
for i in string.ascii_uppercase:
```

```
    print(i,'->',ord(i))
```

```
A -> 65
```

```
B -> 66
```

```
C -> 67
```

```
D -> 68
```

```
E -> 69
```

122. 코드값에 대응하는 문자 얻기(chr)

ex)

```
print(chr(65)) # A
```

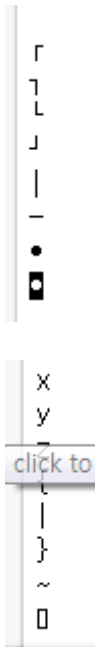
문제276. 아스키 코드값 1~127번까지의 문자가 무엇인지 전부 출력하시오.

답)

```
import string
```

```
for i in range(128):
```

```
    print(chr(i))
```

123. 문자열로 된 식을 실행하기(eval)

" 문자열을 명령문으로 인식하게 하는 함수"

```
ex)
a='2+3'
print(a) # '2+3'
print(eval(a)) # 5
```

문제277. eval을 이용해서 구구단 2단을 출력하시오.

```
보기)
2 * 2 = 4
2 * 3 = 6
.
.
2 * 9 = 18
```

```
답)
a='2*i'
for i in range(2,10):
    print('2*%d=' %i,eval(a))
2*2= 4
2*3= 6
2*4= 8
2*5= 10
2*6= 12
2*7= 14
2*8= 16
2*9= 18
```

124. 이름없는 한줄짜리 함수 만들기(lambda)

ex)

```
add=lambda x, y : x+y
```

입력값 : 실행문

```
print (add(1,3)) #4
```

문제278. 아래의 리스트의 결과가 아래와 같이 수행되도록 하시오.(multi를 lambda로)

보기)

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

```
multi(a)
```

```
[1,4,9,16,25]
```

답1)

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

```
res=[]
```

```
multi=lambda b : b*b
```

```
for i in a:
```

```
    res.append(multi(i))
```

```
print (res)
```

답2)

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

```
multi = lambda x : x*x
```

```
print([multi(i) for i in a])
```

```
[1, 4, 9, 16, 25]
```

125. 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map)

답3)

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

```
multi=lambda x : x*x
```

```
b=map(multi,a)
```

```
print(list(b))
```

문제279. R수업때 만들었던 normalize 함수를 파이썬으로 구현하시오.

R코드)

```
normalize<-function(x){
```

```
    return((x-min(x)) / (max(x)-min(x)) )
```

```
}
```

답)

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

```

res=[]
def normalize(a):
    for i in a:
        res.append((i-min(a))/(max(a)-min(a)))
    return res

print(normalize(a))

```

126. 텍스트 파일을 읽고 출력하기(read)

```

ex)
f=open("C:\python_data\stockcode.txt",'r')
data=f.read()
print(data)
f.close()

```

```

000020 동화약품
000040 S&T모터스
000050 경방
000060 메리츠화재
000070 삼양사
000071 삼양사우
000100 유한양행
000101 유한양행우
000120 대한통운
000140 하이트홀딩스
000141 하이트홀딩스우
000150 두산
000151 두산우
000152 두산2우B
000180 성창기업지주
000210 대림산업
000211 대림산업우
000220 유유제약
000221 유유제약1우
-----

```

구분	설명	예시
readline()	한 줄씩 읽어들이도록 한다 만약 모든 라인을 읽어서 화면에 출력하고 싶다면 다음과 같이 작성하면 된다. # readline_all.py f = open("C:/doit/새파일.txt", 'r') while True: line = f.readline() if not line: break print(line) f.close()	# readline.py f = open("C:/doit/새파일.txt", 'r') line = f.readline() print(line) f.close() #새파일.txt의 가장 첫 번째 줄이 화면에 출력
readlines()	파일의 모든 라인을 읽어서 각각의 줄을 요소로 갖는 리스트로 리턴한다.	f = open("C:/doit/새파일.txt", 'r') lines = f.readlines() for line in lines: print(line) f.close()
read()	파일의 내용 전체를 문자열로 리턴한다.	f = open("C:/doit/새파일.txt", 'r') data = f.read() print(data) f.close()

127. 텍스트 파일을 한줄씩 읽고 출력하기(readlines)

```
f=open("C:\python_data\stockcode.txt",'r')
data=f.read()
print(data)
f.close()
```

```
[ '000020 동화약품\n', '000040 S&T모터스\n', '000050 경방\n', '000060
'000070 삼양사\n', '000071 삼양사우\n', '000100 유한양행\n', '000101
'000120 대한통운\n', '000140 하이트홀딩스\n', '000141 하이트홀딩스우
산\n', '000151 두산우\n', '000152 두산2우B\n', '000180 성창기업지주
산업\n', '000211 대림산업우\n', '000220 유유제약\n', '000221 유유제
유유제약2우B\n', '000230 일동제약\n', '000240 한국타이어\n', '000270
300 대유신소재\n', '000320 디피아이홀딩스\n', '000321 디피아이홀딩스
디피아이홀딩스2B\n', '000360 삼환기업\n', '000361 삼환기업우\n', '00
험\n', '000390 삼화페인트\n', '000400 롯데손해보험\n', '000420 로케트
1 로켓트전우\n', '000430 대원강업\n', '000470 그린손해보험\n', '0004
'000490 대동공업\n', '000500 가온전선\n', '000520 삼일제약\n', '0005
\n', '000541 흥국쌍용화재우\n', '000542 흥국쌍용화재2우B\n', '000590
00610 제일화재\n', '000640 동아제약\n', '000650 천일고속\n', '000660
00670 영풍\n', '000680 LSN네트웍스\n', '000681 LSN네트웍스우\n', '0007
'000720 현대건설\n', '000721 현대건설우\n', '000760 이화산업\n', '00
'000800 경남기업\n', '000810 삼성화재\n', '000811 삼성화재우\n', '00
\n', '000831 삼성물산우\n', '000850 화천기공\n', '000860 건설화학
\n', '000881 한화우\n', '000890 보해양조\n', '000891 보해양조우\n',
\n', '000950 전방\n', '000970 한국주철관\n', '000990 동부하이텍\n',
'001000 동부하이텍2우B\n', '001020 파이핀테크산업\n', '00104
```

문제280. 위의 예제를 이용해서 기업명을 물어보게 하고 기업명을 입력하면 주식코드가 출력

되게 하시오.

보기)

기업명을 입력하세요~ 엔씨소프트

036570

답1)

```
f = open('C:/python_data/stockcode.txt', 'r')
```

```
data=f.read()
```

```
data=list(data.split('\n'))
```

```
l=[]
```

```
for i in data:
```

```
    l.append(i.split(" "))
```

```
name=input('기업명 입력 : ')
```

```
for i in l:
```

```
    if i[1]==name:
```

```
        print(i[0])
```

```
        break
```

```
기업명 입력 : 엔씨소프트
```

```
036570
```

답2)

```
def stock_code():
```

```
    c_n = input('코드명을 보고싶은 기업명을 입력하세요 : ')
```

```
    f = open('C:/python_data/stockcode.txt', 'r')
```

```
    data = f.read()
```

```
    for i in range(len(data.split('\n'))):
```

```
        if c_n in data.split('\n')[i]:
```

```
            print(data.split('\n')[i])
```

```
stock_code()
```

```
코드명을 보고싶은 기업명을 입력하세요 : 엔씨소프트
```

```
036570 엔씨소프트
```

128. 화면에서 사용자 입력을 받고 파일로 쓰기(write)

ex)

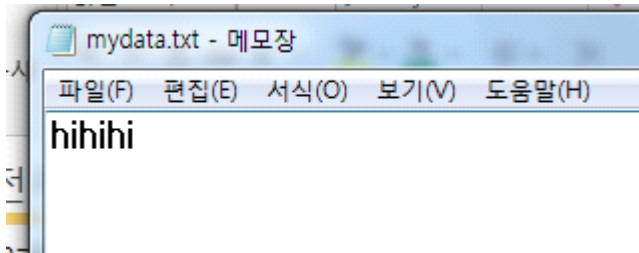
```
text=input('파일에 저장할 내용 입력 : ')
```

```
f = open('C:/python_data/mydata.txt', 'w')
```

```
f.write(text)
```

```
f.close()
```

파일에 저장할 내용 입력 : hihhi



문제281. 위의 스크립트는 한줄 쓰고 엔터치면 바로 종료가 되는데, 한줄쓰고 엔터쳐도 계속

물어보게해서 계속 쓸 수 있게 하되 아무것도 안쓰고 엔터를 치면 종료되게 하시오.

답)

```
f = open('C:/python_data/mydata.txt', 'w')
```

```
while True:
```

```
    text=input('파일에 저장할 내용 입력 : ')
```

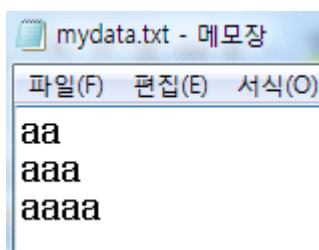
```
    f.write(text+'\n')
```

```
    if text == "":
```

```
        break
```

```
f.close()
```

파일에 저장할 내용 입력 : aa
파일에 저장할 내용 입력 : aaa
파일에 저장할 내용 입력 : aaaa
파일에 저장할 내용 입력 :



129. 텍스트 파일 복사하기(read,write)

ex)

```
f = open('C:/python_data/stockcode.txt', 'r')
```

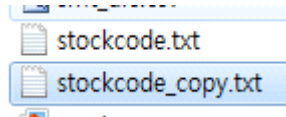
```
h = open('C:/python_data/stockcode_copy.txt', 'w')
```

```
data=f.read()
```

```
h.write(data)
```

```
f.close()
```

```
h.close()
```



- 특정 폴더 밑에 있는 파일들을 파이썬에서 읽어오는 방법

```
import os
```

```
path_dir='C:\\python_data'
```

```
file_list=os.listdir(path_dir)
```

```
file_list.sort()
```

```
for items in file_list:
```

```
    print(items)
```

```
1cust.txt
```

```
5memb.txt
```

```
NIV.txt
```

```
Sherlock.txt
```

```
a.png
```

```
aa.csv
```

```
aaaa.png
```

```
ann.txt
```

```
dept.csv
```

```
emp.csv
```

```
emp2.csv
```

```
emp3.csv
```

```
ladybug4.txt
```

```
mydata.txt
```

```
mydata3.txt
```

```
negative-words.txt
```

```
positive-words.txt
```

```
smt_dic.csv
```

```
stockcode.txt
```

```
stockcode_copy.txt
```

```
sun.jpg
```

```
winter.txt
```

문제282. C:\\python 밑에 있는 파일들 중에 csv파일은 C:\\csv_backup폴더로 복사하고,

tet 파일은 text_backup파일로 복사하시오.

(C드라이브 밑에 폴더 두개를 먼저 생성하고 시작)

답)

```
import os
```

```
import csv
```

```
path_dir='C:\\python_data'
```

```

file_list=os.listdir(path_dir)
file_list.sort()

for items in file_list:
    try:
        if items[-4:]=='.txt':
            f = open('C:\\python_data\\'+items[:-4]+'.txt', 'r', encoding='UTF-8')
            h = open('C:\\text_backup\\'+items[:-4]+'_copy.txt', 'w',
encoding='UTF-8')
            data=f.read()
            h.write(data)

            f.close()
            h.close()
        except:
            print('카피 못한 파일 : ',items)

    try:
        if items[-4:]=='.csv':
            f2=open('C:\\python_data\\'+items[:-4]+'.csv','r')
            h2=open('C:\\csv_backup\\'+items[:-4]+'_copy.csv','w')
            data_csv=f2.read()
            h2.write(data_csv)

            f2.close()
            h2.close()
        except:
            print('카피 못한 파일: ', items)
카피 못한 파일 :  Sherlock.txt
카피 못한 파일:  emp3.csv
카피 못한 파일:  smt_dic.csv
카피 못한 파일 :  stockcode.txt
카피 못한 파일 :  stockcode_copy.txt
카피 못한 파일 :  winter.txt

```


파이썬 149제(파일 복사)

2018년 8월 2일 목요일 오전 10:02

130. 바이너리 파일 복사하기(read, write)

"이미지 파일을 파이썬으로 복사하는 방법"

ex)

```
bufsize=1024 #이미지를 저장할 최소 기본 메모리 사이즈
f = open("C:\python_data\whanse.jpg",'rb') #바이너리 파일 읽기
h = open("C:\python_data\whanse_copy.jpg",'wb') #바이너리 파일 쓰기
data = f.read(bufsize)

while data:
    h.write(data)
    data = f.read(bufsize)

f.close()
h.close()
```

문제283. 아래의 개사진을 전부 C드라이브 밑에 image라는 폴더를 만들고 거기에 복사하는

파이썬 코드를 작성하시오.

C:\datdog\dog_image

↓ copy

C:\image

답)

```
import os
path_dir = 'C:\catdog\dog_images'
bufsize=1024
```

```
file_list=os.listdir(path_dir)
file_list.sort()
```

```
for items in file_list:
    f = open('C:\catdog\dog_images'+items, 'rb')
    h = open('C:\image'+items,'wb')
    data = f.read(bufsize)
```

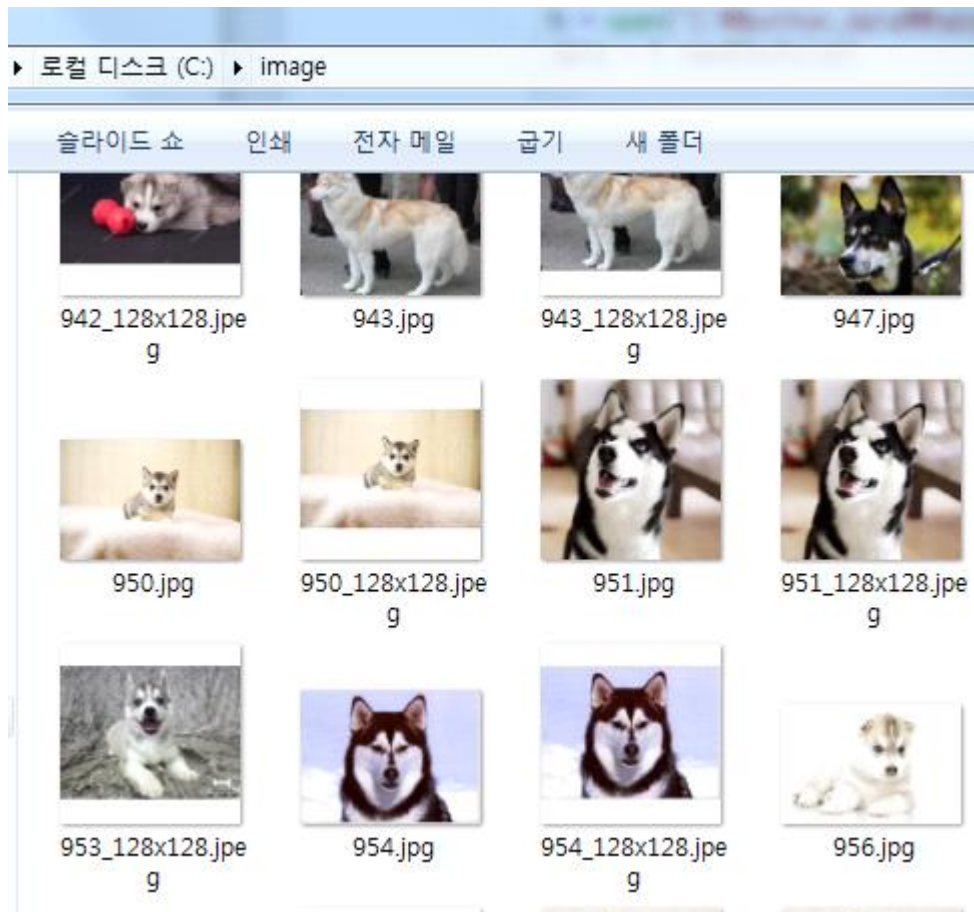
```

while data:
    h.write(data)
    data = f.read(bufsize)

f.close()
h.close()

```

결과)



131. 파일을 열고 자동으로 닫기(with~as)

```

ex)
f = open("C:\\python\\stockcode.txt", 'r')
data = f.readlines()
print(data)
f.close() #파일을 닫는다

```

이렇게 해줬었는데.... 만약 close()를 안해주다면 메모리가 꽉차서 프로그램 실행이 안 됨.

문제284. 위의 예제를 파일을 닫는 f.close()를 명시하지 않고 자동으로 닫히게 하시오.

답)

```

with open("C:\\python_data\\stockcode.txt", 'r') as f:

```

```
for i in f.readlines():
```

```
    print(i)
```

```
000020 동화약품
```

```
000040 S&T모터스
```

```
000050 경방
```

```
000060 메리츠화재
```

```
000070 삼양사
```

```
000071 삼양사우
```

```
000100 유한양행
```

```
000101 유한양행우
```

설명 : with ~ as 절을 사용하면, f.close()를 명시하지 않아도, 프로그래머가 실수로 f.close() 를 작성하지 않아서 코드 실행할 때 메모리 부족 오류를 예방.

132. 파일의 특정 부분만 복사하기(seek, read, write)

ex)

```
spos = 105    # 파일을 읽는 위치 지정 105 바이트 위치부터
```

```
size = 500    # 읽을 크기를 지정 500 바이트까지
```

```
f = open('d:\data\stockcode.txt', 'r')
```

```
h = open('d:\data\stockcode_part.txt', 'w')
```

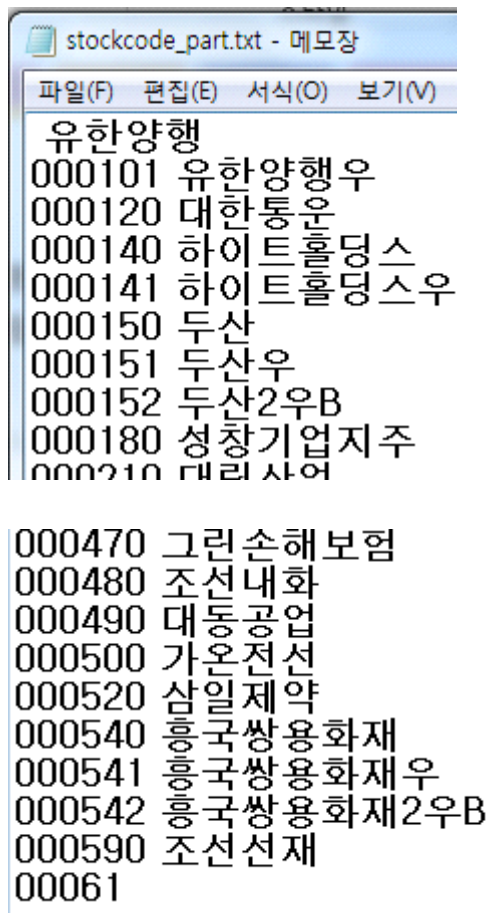
```
f.seek(spos)
```

```
data = f.read(size)
```

```
h.write(data)
```

```
h.close()
```

```
f.close()
```



133. 파일크기 구하기(ospath.getsize)

ex)

```
from os.path import getsize
```

```
file1 = 'C:\python_data\stockcode.txt'
```

```
file2 = 'C:\python_data\hanse.jpg'
```

```
file_size1 = getsize(file1)
```

```
file_size2 = getsize(file2)
```

```
print('File Name: %s \tFile Size: %d' %(file1, file_size1))
```

```
print('File Name: %s \tFile Size: %d' %(file2, file_size2))
```

```
File Name: C:\python_data\stockcode.txt      File Size: 16339
File Name: C:\python_data\hanse.jpg          File Size: 7699
```

134. 파일 삭제하기(os.remove)

ex)

```
from os import remove
```

```
target_file = 'C:\python_data\stockcode_copy.txt'
```

```
k = input('[%s] 파일을 삭제하겠습니까? (y/n)' %target_file)
if k == 'y':
    remove(target_file)
    print('[%s]를 삭제했습니다.' %target_file)
```

```
[C:\python_data\stockcode_copy.txt] 파일을 삭제하겠습니까? (y/n)y
[C:\python_data\stockcode_copy.txt]를 삭제했습니다.
```

설명 : shift + del 과 같은 작동(휴지통에 안들어간다)

문제285. 한새를 위한 코드를 작성하시오. 위의 코드를 가지고 파일명을 물어보게 하고 파일명

을 입력하면 삭제가 되는 코드를 작성하시오.

보기)

```
print(delete_file())
```

삭제하고 싶은 파일 입력 : C:\python_data\mydata.txt

답)

```
def delete_file():
```

```
    from os import remove
```

```
    target_file = input('삭제 파일 입력 : ')
    k = input('[%s] 파일을 삭제하겠습니까? (y/n)' %target_file)
```

```
    if k == 'y':
```

```
        remove(target_file)
```

```
        print('[%s]를 삭제했습니다.' %target_file)
```

```
    print(delete_file())
```

135. 파일 이름 바꾸기(os.rename)

ex)

```
from os import rename
```

```
target_file = 'd:\data\stockcode.txt'
```

```
newname = input('[%s]에 대한 새로운 파일 이름을 입력하세요: ' %target_file)
```

```
rename(target_file, newname)
```

```
print('[%s] -> [%s] 로 파일이름이 변경되었습니다.' %(target_file, newname))
```

[C:\python_data\stockcode.txt]에 대한 새로운 파일 이름을 입력하세요: stockcode2
[C:\python_data\stockcode.txt] -> [stockcode2] 로 파일이름이 변경되었습니다.

이러면 파일 날라감....

파일 경로와 바꿀 이름을 전부 써줘야함!!!!

136. 파일을 다른 디렉터리로 이동하기(os.rename)

ex)

```
from os import rename
```

```
target_file = 'd:\python_data\stockcode_part.txt'
```

```
newpath = input('[%s]를 이동할 디렉터리의 절대경로를 입력하세요: ' %target_file)
```

```
if newpath[-1] == '/':
```

```
    newname = newpath + target_file
```

```
else:
```

```
    newname = newpath + '/' + target_file
```

```
try:
```

```
    rename(target_file, newname)
```

```
    print('[%s] -> [%s]로 이동되었습니다.' %(target_file, newname))
```

```
except FileNotFoundError as e:
```

```
    print(e)
```

137. 디렉토리에 있는 파일목록 얻기(os.listdir, glob.glob)

- 딥러닝때 사용할 데이터 전처리 코드 작성!!!!!!!!

1. 이미지 데이터 로드하는 함수(직접 스스로 해야함) ★★★★★★

2. 신경망 구현 함수

문제286. cifar10 폴더 밑에 test 폴더에 있는 이미지를 불러오는 함수를 생성하시오.

보기)

cifar10 >> 개, 고양이, 비행기, 배, 자동차 ...(10개)의 이미지에 대한 데이터와 라벨에 대한

데이터

정확도가 80% 를 넘기면 논문에 등재 가능!!

정확도 : 이미지 --> 컴퓨터 --> 비행기

답)

```
import os
```

```
test_image='C:\\cifar10\\test\\'
```

```
def image_load(path):
```

```
    file_list = os.listdir(path)
```

```
    return file_list
```

```
print(image_load(test_image))
```

```
['1.png', '10.png', '100.png', '1000.png', '10000.png', '1001.png', '1002.png', '1003.png', '1004.png', '1005.png', '1006.png', '1007.png', '1008.png', '1009.png', '101.png', '1010.png', '1011.png', '1012.png', '1013.png', '1014.png', '1015.png', '1016.png', '1017.png', '1018.png', '1019.png', '102.png', '1020.png', '1021.png', '1022.png', '1023.png', '1024.png', '1025.png', '1026.png', '1027.png', '1028.png', '1029.png', '103.png', '1030.png', '1031.png', '1032.png', '1033.png', '1034.png', '1035.png', '1036.png', '1037.png', '1038.png', '1039.png', '104.png', '1040.png', '1041.png', '1042.png', '1043.png', '1044.png', '1045.png', '1046.png', '1047.png', '1048.png', '1049.png', '105.png', '1050.png', '1051.png', '1052.png', '1053.png', '1054.png', '1055.png', '1056.png', '1057.png', '1058.png', '1059.png', '106.png', '1060.png', '1061.png', '1062.png', '1063.png', '1064.png', '1065.png', '1066.png', '1067.png', '1068.png', '1069.png', '107.png', '1070.png', '1071.png', '1072.png', '1073.png', '1074.png', '1075.png', '1076.png', '1077.png', '1078.png', '1079.png', '108.png', '1080.png', '1081.png', '1082.png', '1083.png', '1084.png', '1085.png', '1086.png', '1087.png', '1088.png', '1089.png', '109.png', '1090.png', '1091.png', '1092.png', '1093.png', '1094.png', '1095.png', '1096.png', '1097.png', '1098.png', '1099.png', '11.png', '110.png', '1100.png', '1101.png', '1102.png', '1103.png', '1104.png', '1105.png', '1106.png', '1107.png', '1108.png', '1109.png', '111.png', '1110.png', '1111.png', '1112.png', '1113.png', '1114.png', '1115.png', '1116.png', '1117.png', '1118.png', '1119.png', '112.png', '1120.png', '1121.png', '1122.png', '1123.png', '1124.png', '1125.png', '1126.png', '1127.png', '1128.png', '1129.png', '113.png', '1130.png', '1131.png', '1132.png', '1133.png', '1134.png', '1135.png', '1136.png', '1137.png', '1138.png', '1139.png', '114.png', '1140.png', '1141.png', '1142.png', '1143.png', '1144.png', '1145.png', '1146.png', '1147.png', '1148.png', '1149.png', '115.png', '1150.png', '1151.png', '1152.png', '1153.png', '1154.png', '1155.png', '1156.png', '1157.png', '1158.png', '1159.png', '116.png', '1160.png', '1161.png', '1162.png', '1163.png', '1164.png', '1165.png', '1166.png', '1167.png', '1168.png', '1169.png', '117.png', '1170.png', '1171.png', '1172.png', '1173.png', '1174.png', '1175.png', '1176.png', '1177.png', '1178.png', '1179.png', '118.png', '1180.png', '1181.png', '1182.png', '1183.png', '1184.png', '1185.png', '1186.png', '1187.png', '1188.png', '1189.png', '119.png', '1190.png', '1191.png', '1192.png', '1193.png', '1194.png', '1195.png', '1196.png', '1197.png', '1198.png', '1199.png']
```

불러오는 순서가 이상하다!!(1->10->100->1000->10000)

문제287. (점심시간) 위의 스크립트를 수정해서 png때버리고 아래와 같이 출력되게 하시오.

보기)

```
[1, 10, 100, 1000, 10000 ....]
```

힌트)

```
re.sub('[^0-9]', '', i)
```

답)

```
import os
```

```
import re
```

```
test_image = 'c:/cifar10/test/'
```

```
def image_load(path):
```

```
    file_list = os.listdir(path)
```

```
    file_list2=[]
```

```
    for i in file_list:
```



```

file_list2=[]

for i in file_list:
    file_list2.append(int(re.sub('[^0-9]', '', i)))
file_list2.sort()

a=[]
for i in file_list2:
    a.append(str(i)+'.png')
return a

print(image_load(test_image))
['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', '21.png', '22.png', '23.png', '24.png', '25.png', '26.png', '27.png', '28.png', '29.png', '30.png', '31.png', '32.png', '33.png', '34.png', '35.png', '36.png', '37.png', '38.png', '39.png', '40.png', '41.png', '42.png', '43.png', '44.png', '45.png', '46.png', '47.png', '48.png', '49.png', '50.png', '51.png', '52.png', '53.png', '54.png', '55.png', '56.png', '57.png', '58.png', '59.png', '60.png', '61.png', '62.png', '63.png', '64.png', '65.png', '66.png', '67.png', '68.png', '69.png', '70.png', '71.png', '72.png', '73.png', '74.png', '75.png', '76.png', '77.png', '78.png', '79.png', '80.png', '81.png', '82.png', '83.png', '84.png', '85.png', '86.png', '87.png', '88.png', '89.png', '90.png', '91.png', '92.png', '93.png', '94.png', '95.png', '96.png', '97.png', '98.png', '99.png']

```

문제290. 이미지 이름 앞에서 절대 경로가 아래처럼 붙게 하시오.

```

보기)
['C:\WWcifar10\WWtest\WW1.png', 'C:\WWcifar10\WWtest\WW2.png', ..... ]

```

```

답)
import os
import re

test_image = 'C:/cifar10/test/'
def image_load(path):
    file_list = os.listdir(path)
    file_list2=[]

    for i in file_list:
        file_list2.append(int(re.sub('[^0-9]', '', i)))
    file_list2.sort()

    a=[]
    for i in file_list2:

```

```

        a.append(test_image + str(i) + '.png')
    return a
print(image_load(test_image))

```

문제291. 위의 이미지들을 딥러닝 신경망에 넣기 전의 데이터인 숫자 배열로 만들기 위해서 **cv2.imread** 함수를 이용해서 list로 변환하시오.

ex)
list = cv2.imread(이미지)

#opencv 설치

```

(base) C:\Users\wit>pip install OpenCV-python
Collecting OpenCV-python

```

답)

```

import cv2 # opencv 설치
import numpy
import os
import re

test_image = 'c:\\cifar10\\test\\'

def image_load(path):
    file_list = os.listdir(path)
    a = []
    for i in file_list:
        a.append(int(re.sub('[^0-9]', '', i)))
    a.sort()

    b = []
    for i in a:
        b.append('c:\\cifar10\\test\\' + str(i) + '.png')
    image = []
    for j in b:
        img = cv2.imread(j)
        image.append(img)
    return image

print(image_load(test_image))

```

138. 현재 디렉터리 확인하고 바꾸기(os.getcwd, os.chdir)

ex)
import os

```

pdir = os.getcwd(); print(pdir)
os.chdir('.'); print(os.getcwd())
os.chdir(pdir); print(os.getcwd())
C:\python_data\

```

C:\W
C:\Wpython_data

139. 디렉토리 생성하기(osmkdir)

```
ex)
import os

newfolder = input('새로 생성할 디렉터리 이름을 입력하세요: ')
try:
    os.mkdir(newfolder)
    print('[%s] 디렉터리를 새로 생성했습니다.' %newfolder)
except Exception as e:
    print(e)
```

문제292. 디렉토리를 생성할 때 이름이 동일한 디렉터리가 있으면 동일한 디렉터리가 있습니다.

라는 메시지가 출력되게 하시오.

```
답)
import os

newfolder = input('새로 생성할 디렉터리 이름을 입력하세요: ')
try:
    os.mkdir(newfolder)
    print('[%s] 디렉터리를 새로 생성했습니다.' %newfolder)
except Exception as e:
    print('동일한 폴더가 있습니다.')
```

새로 생성할 디렉터리 이름을 입력하세요: myfolder
동일한 폴더가 있습니다.

140. 디렉토리 제거하기 (osrmdir)

```
ex)
import os

target_folder = 'C:\Users\Wit\Anaconda3\myfolder' #조심!!
k = input('[%s] 디렉터를 삭제하겠습니까? (y/n)' %target_folder)
if k == 'y':
    try:
        os.rmdir(target_folder)
```

```

    print('[%s] 디렉토리를 삭제했습니다.' %target_folder)
except Exception as e:
    print(e)

```

문제293. data7 폴더에 다시 만들고 data7폴더에 파일을 몇 개 넣은 후에 삭제해보시오.
 답)

141. 하위 디렉토리 및 파일 전체 삭제하기(shutil.rmtree)

```

ex)
import shutil
import os

target_folder = 'd://data7'
print('[%s] 하위 모든 디렉토리 및 파일들을 삭제합니다.' %target_folder)
for file in os.listdir(target_folder):
    print(file)
k = input('[%s]를 삭제하겠습니까? (y/n) ' %target_folder)
if k == 'y':
    try:
        shutil.rmtree(target_folder)
        print('니다.%s' %target_folder)
    except Exception as e:
        print(e)

```

142. 파일이 존재하는지 체크하기(os.path.exists)

```

ex)
import os
from os.path import exists

dir_name = input('새로 생성할 디렉토리 이름을 입력하세요: ')
if not exists(dir_name):
    os.mkdir(dir_name)
    print('[%s] 디렉토리를 생성했습니다.' %dir_name)
else:
    print('[%s]은(는) 이미 존재합니다.' %dir_name)

```

143. 파일인지 디렉토리인지 확인하기(os.path.isfile, os.path.isdir)

```

ex)
import os
from os.path import exists, isdir, isfile

```

```

files = os.listdir()
for file in files:
    if isdir(file):
        print('DIR: %s' %file)

for file in files:
    if isfile(file):
        print('FILE: %s' %file)

```

144. 현재 시간을 년-월-일 시:분:초로 출력하기(localtime, strftime)

```

ex)
from time import localtime, strftime

print( strftime('%Y-%m-%d %X%t', localtime()) )
2018-08-02 15:37:27

```

145. 올해 경과된 날짜수 계산하기(localtime)

```

ex)
from time import localtime

t = localtime()
start_day = '%d-01-01' %t.tm_year
elapsed_day = t.tm_yday

print('오늘은 [%s]이후 [%d]일째 되는 날입니다.' %(start_day, elapsed_day))
오늘은 [2018-01-01]이후 [214]일째 되는 날입니다.

```

문제294. 우리반이 시작된 이후에 몇일 지났는지 출력하시오.

```

답)
from datetime import date
d0 = date(2018, 3, 26)
d1 = date(2018, 8, 2)
delta = d1 - d0
print('오늘은 [%s]이후 [%d]일째 되는 날입니다.'%(d0, delta.days))
오늘은 [2018-03-26]이후 [129]일째 되는 날입니다.

```

146. 오늘의 요일 계산하기(localtime)

```

ex)
from time import localtime

weekdays = ['월요일', '화요일', '수요일', '목요일', '금요일', '토요일', '일요일']

t = localtime()
today = '%d-%d-%d' %(t.tm_year, t.tm_mon, t.tm_mday)
print(t.tm_mday)
week = weekdays[t.tm_wday]

print('[%s] 오늘은 [%s]입니다.' %(today, week))
2
[2018-8-2] 오늘은 [목요일]입니다.

```

147. 프로그램 실행 시간 계산하기(datetimeow)

```

ex)
from datetime import datetime

start = datetime.now()
print('1에서 백만까지 더합니다.')
ret = 0
for i in range(1000000):
    ret += i
print('1에서 백만까지 더한 결과: %d' %ret)
end = datetime.now()
elapsed = end - start
print('총 계산 시간: ', end='');print(elapsed)
elapsed_ms = int(elapsed.total_seconds()*1000)
print('총 계산 시간: %dms' %elapsed_ms)
1에서 백만까지 더합니다.
1에서 백만까지 더한 결과: 499999500000
총 계산 시간: 0:00:00.143009
총 계산 시간: 143ms

```

148. 주어진 숫자를 천 단위로 구분하기

```

ex)
num = input('아무 숫자를 입력하세요: ')

if num.isdigit():
    num = num[::-1]
    ret = ''

```


파이썬 149제(웹 스크롤링)

2018년 8월 2일 목요일 오후 4:20

웹 스크롤링

"beautiful soup 모듈"

html 이란?

Hyper Text Markup Language 의 약자이고 여러 개의 태그(tag)를 연결해서 모아놓은 문서

웹스크롤링

1. ebs 레이디 버그
2. 한겨레 신문
3. 중앙일보
4. 동아일보
5. 네이버 이미지
6. 구글 이미지
7. 다음 이미지
8. Bing 이미지
9. 서울시 응답소 게시판(java script)
10. 트위터 웹 스크롤링

문제295. 아래의 간단한 html 문서를 만들어 보시오.

보기)

```
<html>
```

```
<head>
```

```
<title>
```

신현수님의 일정

```
</title>
```

```
</head>
```

```
<body>
```

```
<p class="tittle">
```

신현수님은 오늘 머리를 하셨습니다.

```
</body>
```

```
</html>
```


문제296. 문제295 번에 나온 글씨를 진하게 하시오.

답)

```
<html>
```

```
<head>
```

```
<title>
```

```
신현수님의 일정
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<p class="tittle"><b>
```

```
신현수님은 오늘 머리를 하셨습니다.
```

```
</b>
```

```
</p>
```

```
</body>
```

```
</html>
```

문제297. 위의 글씨에 밑줄을 그어보시오.

답)

```
<html>
```

```
<head>
```

```
<title>
```

```
신현수님의 일정
```

```
</title>
```

```
</head>
```

```
<body>
```

```
<p class="tittle"><b><u>
```

```
신현수님은 오늘 머리를 하셨습니다.
```

```
</b>
```

```
</u>
```

```
</p>
```

```
</body>
```

```
</html>
```

문제298. 위의 글씨체를 이탤릭체로 변경하시오.

답)

```
<html>
```

```

<head>
<title>
신현수님의 일정
</title>
</head>

<body>
<p class="tittle"><b><u><i>
신현수님은 오늘 머리를 하셨습니다.
</b></u></i>

</p>
</body>

</html>

```

신현수님은 오늘 머리를 하셨습니다.

문제299. p tag를 추가해서 제목과 내용을 나누시오.

```

답)
<html>

<head>
<title>
신현수님의 일정
</title>
</head>

<body>
<p class="tittle"><b><u><i>
신현수님은 오늘 머리를 하셨습니다.
</b></u></i>

<p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만원 파마는 10
만원, 파마를 하고 싶었지만 돈이 없었다. 그래서 커트로 머리를 올렸다.
</p>

</body>

```

</html>

신현수님은 오늘 머리를 하셨습니다.

오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만원 파마는 10만원

문제300. 위의 html 문서에 링크를 거시오.

답)

<html>

<head>

<title>

신현수님의 일정

</title>

</head>

<body>

<p class="tittle"><u><i>

신현수님은 오늘 머리를 하셨습니다.

</u></i>

<p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만원 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다. 그래서 커트로 머리를 올렸다.

 다음카페

</p>

</body>

</html>

이 없었다. 그래서 커트로 머리를 올렸다. 다음카페

문제301. BeautifulSoup 모듈을 이용해서 aa.html의 title을 검색하시오.

답)

pip install bs4

from bs4 import BeautifulSoup

with open("C:\python_data\wwaa.html") as aa:

soup = BeautifulSoup(aa,'lxml')

```
print(soup.title)
<title>
신현수님의 일정
</title>
```

문제302. 위의 결과에서 html코드는 나오지 않고 텍스트만 출력되게 하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:\python_data\Waa.html") as aa:
    soup = BeautifulSoup(aa, 'html.parser')
```

```
print(soup.title.string)

신현수님의 일정
```

파싱 : 사람이 알 수 있는 언어를 컴퓨터가 알 수 있는 코드로 변환 하는 것

문제303. aa.html 문서에서 p태그에 대한 html을 검색하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:\python_data\Waa.html") as aa:
    soup = BeautifulSoup(aa, 'html.parser')
```

```
print(soup.p)
print(soup.find('p'))

<p class="tittle"><b><u><i>
신현수님은 오늘 머리를 하셨습니다.
</i></u></b></p>
<p class="tittle"><b><u><i>
신현수님은 오늘 머리를 하셨습니다.
</i></u></b></p>
```

문제304. aa.html 문서에서 p태그에 대한 html을 모두 검색하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:\python_data\Waa.html") as aa:
    soup = BeautifulSoup(aa, 'html.parser')
```

```
#print(soup.p)
print(soup.find_all('p'))
```

```
[<p class="title"><b><u><i>
신현수님은 오늘 머리를 하셨습니다.
</i></u></b></p>, <p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만
원 파마는 10만원,<br/> 파마를 하고 싶었지만 돈이 없었다. 그래서 커트로 머리를 올렸다.

<a class="cafe1" href="http://cafe.daum.net/oracleoracle" id="link1"> 다음카페 </a></p
>]
```

문제305. aa.html문에서 a태그에 대한 html을 모두 검색하시오.

```
답)
from bs4 import BeautifulSoup
with open("C:₩₩python_data₩₩aa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')

#print(soup.p)
print(soup.find_all('a'))

[<a class="cafe1" href="http://cafe.daum.net/oracleoracle" id="link1"> 다음카페<br/>
</a>, <a class="cafe2" href="http://www.naver.com" id="link2">네이버<br/></a>, <a clas
s="cafe3" href="http://www.google.com" id="link3">구글<br/></a>]
```

문제306. a.html문서에서 a태크에 href링크의 url만 긁어 오시오.

```
답)
1. for 문을 써서 돌려본다.
from bs4 import BeautifulSoup
with open("C:₩₩python_data₩₩aa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
for i in soup.find_all('a'):
    print(i)
```

```
<a class="cafe1" href="http://cafe.daum.net/oracleoracle" id="link1"> 다음카페<br/> </
a>
<a class="cafe2" href="http://www.naver.com" id="link2">네이버<br/></a>
<a class="cafe3" href="http://www.google.com" id="link3">구글<br/></a>
```

2. get('href')를 써준다

```
from bs4 import BeautifulSoup
with open("C:₩₩python_data₩₩aa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
for i in soup.find_all('a'):
    print(i.get('href'))
```

문제307. aa.html 문서에서 html 소스 말고 text만 출력하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:python_dataa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
print(soup.get_text())
```

신현수님의 일정

신현수님은 오늘 머리를 하셨습니다.

오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만원 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다. 그래서 커트로 머리를 올렸다.

다음카페

네이버

구글

문제308. 위의 텍스트를 한줄로 나오게 하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:python_dataa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
print(soup.get_text("",strip=True))
```

신현수님의 일정신현수님은 오늘 머리를 하셨습니다.오늘 나는 강남의 더킹 미용실에 갔다. 커트는 2만원 파마는 10만원,파마를 하고 싶었지만 돈이 없었다. 그래서 커트로 머리를 올렸다.다음카페네이버구글

꿀팁!!!!

```
from bs4 import BeautifulSoup
with open("C:python_dataa.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
#print(soup.get_text("",strip=True))
```

```
print(soup.select('p.content > a'))
```

```
[<a class="cafe1" href="http://cafe.daum.net/oracleoracle" id="link1"> 다음카페<br/>
</a>, <a class="cafe2" href="http://www.naver.com" id="link2">네이버<br/></a>, <a clas
s="cafe3" href="http://www.google.com" id="link3">구글<br/></a>]
```

설명:

p속성의 content 라는 클래스 안에 있는 a 속성들을 select 하겠다

문제309. ecologicalpyramid.html 문서에서 text만 출력하시오.

답)

```
from bs4 import BeautifulSoup
with open("C:\\python_data\\ecologicalpyramid.html") as aa:
    soup = BeautifulSoup(aa,'html.parser')
```

```
print(soup.get_text())
```

```
plants
100000
```

```
algae
100000
```

```
deer
1000
```

```
rabbit
2000
```

```
fox
100
```

```
bear
100
```

```
lion
80
```

```
tiger
50
```

문제310. ecologicalpyramid.html 문서에서 number 클래스에 있는 모든 텍스트를 가져오
시

오.

답)

```
from bs4 import BeautifulSoup
with open("C:\\python_data\\ecologicalpyramid.html") as aa:
```

```
soup = BeautifulSoup(aa,'html.parser')
```

```
res=soup.find_all(class_="number")
```

```
print(res)
```

```
[<div class="number">100000</div>, <div class="number">100000</div>, <div class="number">1000</div>, <div class="number">2000</div>, <div class="number">100</div>, <div class="number">100</div>, <div class="number">80</div>, <div class="number">50</div>]
```

```
from bs4 import BeautifulSoup
```

```
with open("C:\wwwpython_data\wwwecologicalpyramid.html") as aa:
```

```
    soup = BeautifulSoup(aa,'html.parser')
```

```
res=soup.find_all(class_="number")
```

```
print(res)
```

```
for i in res:
```

```
    print(i.get_text())
```

```
[<div class="number">100000</div>, <div class="number">100000</div>, <div class="number">1000</div>, <div class="number">2000</div>, <div class="number">100</div>, <div class="number">100</div>, <div class="number">80</div>, <div class="number">50</div>]
```

```
100000  
100000  
1000  
2000  
100  
100  
80  
50
```

문제311. 위의 결과에서 숫자 1000만 가져오시오.

답)

```
from bs4 import BeautifulSoup
```

```
with open("C:\wwwpython_data\wwwecologicalpyramid.html") as aa:
```

```
    soup = BeautifulSoup(aa,'html.parser')
```

```
res=soup.find_all(class_="number")
```

```
print(res[2].get_text())
```

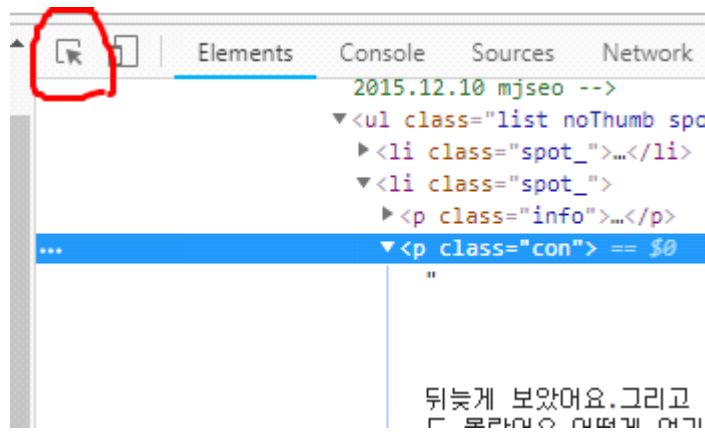
```
1000
```

문제312. ebs의 레이디 버그 시청자 게시판의 html 문서 전체를 스크롤링하는 함수 생성.

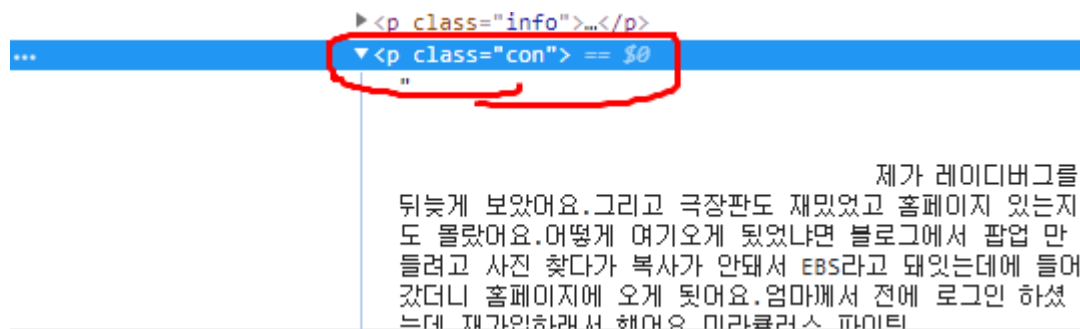
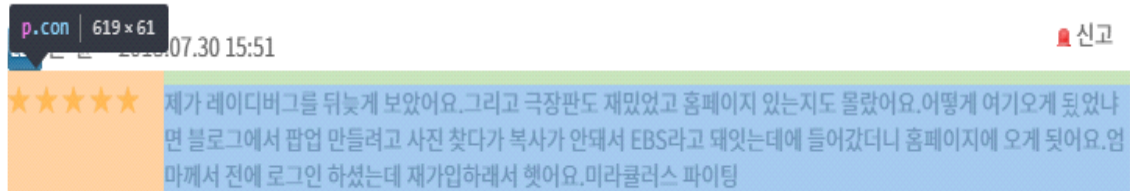
보기)

<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106>

홈페이지 들어가서 F12



을 누르고, 보고싶은 댓글을 누르고 보면된다.



답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
def ebs_crawling():
```

```
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    return result
```

```
print(ebs_crawling())
```

문제313. 위의 html 문서를 BeautifulSoup 모듈로 파싱하고 이 html문서중에 p태그에 해당하는 부분을 모두 가져오시오.

답)

```

import urllib.request
from bs4 import BeautifulSoup

def ebs_crawling():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")

    soup = BeautifulSoup(result,'html.parser')

    return soup.find_all('p')

print(ebs_crawling())

```

```

[<p class="name_program">
<a class="pro_subTit" href="/ladybug/main">미라클러스 레이디버그</a>
<a href="#none" onclick="BHP.UI.Top.addMyProgram('ladybug', '', '');return false;"><
span class="btn_myprogram">MY</span></a>
<a class="sns_pop" data-type="1" href="#none"><span class="btn_sns_pop">공유하기</sp
an></a>
<script type="text/javascript">

    $(document).ready(function(){

        $.ajax({

            url: '/' + hmpId + '/ajax/favoriteUseYn.json',

            async: false,

            data: { 'hmpId': hmpId },

            beforeSend: function(xhr) {

```

문제314. 위의 p태그 중에 class가 con에 해당하는 부분만 스크롤링 하시오.

답)

```

import urllib.request
from bs4 import BeautifulSoup

def ebs_crawling():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")

    soup = BeautifulSoup(result,'html.parser')

```

```
return soup.find_all('p', class_='con')
```

```
print(ebs_crawling())
```

```
[<p class="con">
```

미라클스톤이 많
아서 그런데(사포티편에서 많음을 볼수있음)그 미라클스톤 다 나눠줘서 다 영웅됐으면 좋
겠어요.그리고 영웅될 사람은 거의 마스터푸가 근처에 있던대여?음....클로이영웅이라....
꽃벌영웅이 로즈면...

```
</p>, <p class="con">
```

제가 레이디버그
를 뒤늦게 보았어요.그리고 극장판도 재밌었고 홈페이지 있는지도 몰랐어요.어떻게 여기오
게 되었냐면 블로그에서 팝업 만들려고 사진 찾다가 복사가 안돼서 EBS라고 돼있는데에 들
어갔더니 홈페이지에 오게 됐어요.엄마께서 전에 로그인 하셨는데 재가입하래서 했어요.미
라클러스 파이팅

문제315. 위의 결과에서 html말고 text만 가져오시오.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
def ebs_crawling():
```

```
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?  
hmpMnuld=106"
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup(result,'html.parser')
```

```
    for i in soup.find_all('p',class_='con'):  
        print (i.get_text(",strip=True) )
```

```
print(ebs_crawling())
```


문제318. 위의 게시판 글 앞에 게시날짜를 앞에 출력되게 하기위해 글올린 날짜만 가져오는 코드를 작성하시오.

보기)

2017.01.20 23:13

2017.01.12 15:59

.

.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```
def ebs_crawling():
```

```
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?  
hmpMnuld=106"
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup(result,'html.parser')
```

```
    for i in soup.find_all('span',class_='date'):
```

```
        print(i.get_text())
```

```
print(ebs_crawling())
```

2018.07.30 15:55

2018.07.30 15:51

2018.07.30 15:47

2018.05.22 22:31

2018.02.24 20:37

2018.02.02 00:45

2018.01.30 07:37

2017.10.09 13:23

2017.09.01 15:59

2017.09.01 15:57

2017.08.24 02:17

2017.08.24 02:15

2017.08.23 17:55

2017.08.09 11:36

2017.04.12 19:48

2017.04.12 15:57

2017.04.05 16:06

2017.03.29 19:34

2017.02.23 22:13

2017.01.25 22:58

...

문제319. (점심시간) 아래와 같이 결과가 출력되게 하시오.

보기)

2018.07.30 15:55 레이디 버그와 블랙캣의 히어로....

2018.07.30 15:51 □△○뱃△□△○.....

답1)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```
def ebs_crawling():
```

```
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("utf-8")
```

```
    soup = BeautifulSoup(result,'html.parser')
```

```
    res_date=soup.find_all('span',class_='date')
```

```
    res_con=soup.find_all('p',class_='con')
```

```
    date=[]
```

```
    for i in res_date:
```

```
        date.append(i.get_text() )
```

```
    for i, name in enumerate(res_con):
```

```
        print(date[i], name.get_text(",strip=True))
```

```
ebs_crawling()
```

2018. 07. 30 15:55 미라클스톤이 많아서 그런데(사포티편에서 많음을 볼수있음)그 미라클스톤 다 나눠줘서 다 영웅됐으면 좋겠어요.그리고 영웅될 사람은 거의 마스터푸가 근처에 있던대 여?음....클로이영웅이라....꽃벌영웅이 로즈면...

2018.07.30 15:51 제가 레이디버그를 뒤늦게 보았어요.그리고 극장판도 재밌었고 홈페이지 있는지도 몰랐어요.어떻게 여기오게 됐었냐면 블로그에서 팝업 만들려고 사진 찾다가 복사가 안돼서 EBS라고 돼잇는데에 들어갔더니 홈페이지에 오게 됐어요.엄마께서 전에 로그인 하셨는데 재가입하래서 했어요.미라클러스 파이팅

2018. 07. 30 15:47 레이디버그 팬이에욤.나중에 블랙캣과 레이디버그 둘이 정체를 아는 장면 이 나오면 좋겠어요.그리고 레나루즈가 계속 나오면 좋겠어요.미라클러스 많이 만들어주세요

설명 : res_date에 먼저 리스트형으로 담아주고, enumerate를 사용해서 같은 인덱스에 res_con을 넣어준다.

답2)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```

def ebs_crawling():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?hmpMnuId=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup(result,'html.parser')

    params1 = []
    params2 = []

    for link1, link2 in zip(soup.find_all('span',class_='date'),
                           soup.find_all('p',class_='con') ):
        params1.append(link1.get_text())
        params2.append(link2.get_text())

    for i1, i2 in zip(params1, params2):
        print(i1.strip(), end=' ')
        print(i2.strip() )

ebs_crawling()

```

문제320. ebs레이디 버그 게시판 전체(1~16)의 글들을 전부 스크롤링 하시오.

```

답)
import urllib.request
from bs4 import BeautifulSoup
import re

def ebs_crawling():
    for i in range(1,17):
        list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page="+str\(i\)"
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup(result,'html.parser')

        params1 = []
        params2 = []

        for link1, link2 in zip(soup.find_all('span',class_='date'),

```



```
soup.find_all('p',class_='con') ):
params1.append(link1.get_text())
params2.append(link2.get_text())
```

```
for i1, i2 in zip(params1, params2):
    print(i1.strip(), end=' ')
    print(i2.strip() )
```

ebs_crawling()

EBS 윤*수 2015.08.31 06:31

★★★★★ 무당벌레를 연상시키는 레이디버그~제가 여군출신이라 그런지...아0

EBS 장*희 2015.08.29 21:02

★★★★★ 작년부터 기대되는 작품 인기많은 작품이 되었으면 합니다.

<< < 11 12 13 14 15 **16**

댓글

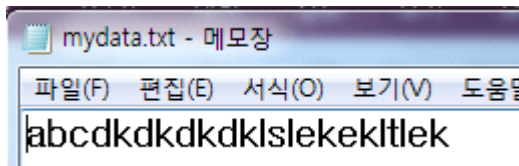
2015.08.31 06:31 무당벌레를 연상시키는 레이디버그~제가 여군출신이라 그런지...아이들보다 제가 더 기대되네요~
2015.08.29 21:02 작년부터 기대되는 작품 인기많은 작품이 되었으면 합니다.

문제321. 변수의 내용과 리스트의 내용을 파일로 저장되게 하는 방법은?

- 변수의 내용을 파일로 저장되게 하는 방법

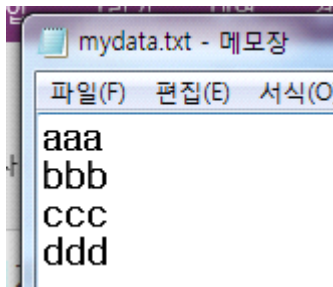
답)

```
text='abcdkdkdkdkslekekltlek'
f=open('C:\python_data\mydata.txt','w',encoding='UTF-8')
f.write(text)
f.close()
```



- 리스트의 내용을 파일에 저장하는 방법

```
text=['aaa', 'bbb', 'ccc', 'ddd']
f=open('C:\python_data\mydata.txt','w',encoding='UTF-8')
for item in text:
    f.write('%s\n' %item)
f.close()
```

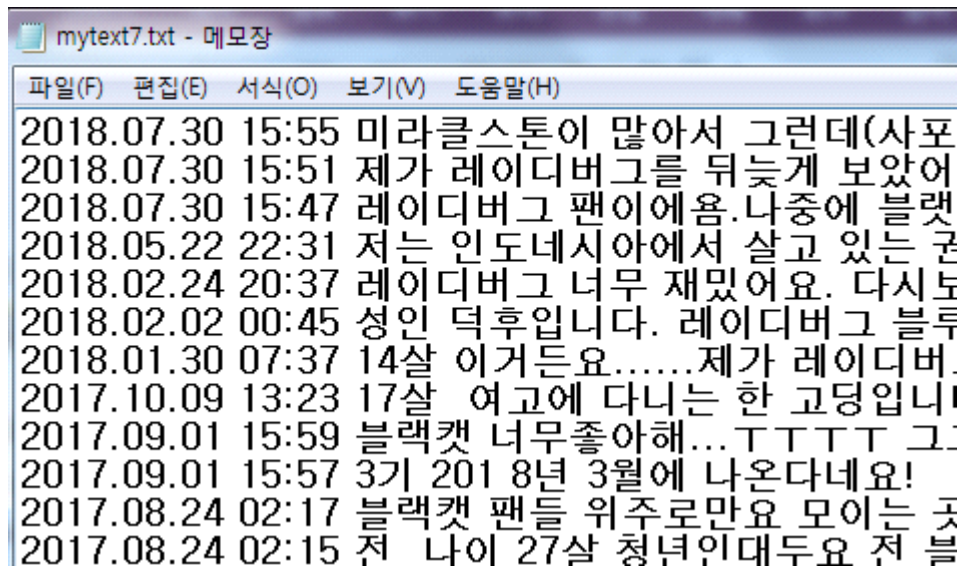


문제322. 레이디 버그 게시판의 날짜와 글을 mytext7.txt에 저장하시오.

답)

```
import urllib.request
from bs4 import BeautifulSoup
import re

def ebs_crawling():
    params1 = []    #for 문 밖으로해야 안 겹치고 들어간다
    params2 = []
    for i in range(1,17):
        list_url = "
```

문제323. 한겨레 신문자 홈페이지로 들어가서 인공지능으로 기사 검색을 한 url을 가져오시오.

(뉴스 검색 더보기)

답)

<http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.03&pageseq=0>

문제324. 한겨레 신문사에서 인공지능으로 기사 검색했을때 나오는 html문서에서 상세기사 제목을 클릭했을때 잡히는 url만 검색하시오.

힌트)

dt 태그 밑에 a태그를 가져오고 href를 가져온다.

답1)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```
def hni_crawling():
```

```
    params1 = []
```

```
    params2 = []
```

```
    for i in range(15):
```

```
        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&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.03&pageseq=0" + str(i)
```

```
        url = urllib.request.Request(list_url)
```

```
        result = urllib.request.urlopen(url).read().decode("UTF-8")
```

```

soup = BeautifulSoup(result,'html.parser')

print(soup)
hni_crawling()

```

문제325. 위의 상세 기사 url들을 params라는 빈 리스트에 담으시오.

```

답)
import urllib.request
from bs4 import BeautifulSoup
import re

def hni_crawling():
    for i in range(15):
        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&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.03&pageseq=\"+str\(i\)
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("UTF-8")
        soup = BeautifulSoup(result,'html.parser')
        params1 = []
        for j in soup.find_all('dt'):
            for k in j:
                params1.append(k.get('href'))
    return params1
hni_crawling()

```

```

['http://www.hani.co.kr/arti/economy/it/855916.html',
'http://www.hani.co.kr/arti/science/future/855895.html',
'http://www.hani.co.kr/arti/society/ngo/855833.html',
'http://www.hani.co.kr/arti/economy/economy_general/855757.html',
'http://www.hani.co.kr/arti/science/technology/855614.html',
'http://www.hani.co.kr/arti/economy/it/855595.html',
'http://www.hani.co.kr/arti/economy/economy_general/855560.html',
'http://www.hani.co.kr/arti/economy/biznews/855458.html',

```

리스트에 안담고 가져오는 방법

```

import urllib.request
from bs4 import BeautifulSoup
import re
def hani_scroll():
    params=[]
    for i in range(15):

```

```

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&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.03
&pageseq='+str(i)
url=urllib.request.Request(list_url)
result=urllib.request.urlopen(url).read().decode('utf-8') #utf-8
result=BeautifulSoup(result,'html.parser')

for i in result.select('dt > a'):
    print(i.get('href'))

```

hani_scroll()

```

http://www.hani.co.kr/arti/society/area/845934.html
http://www.hani.co.kr/arti/economy/economy_general/845768.html
http://www.hani.co.kr/arti/economy/economy_general/845707.html
http://www.hani.co.kr/arti/economy/it/845699.html
http://www.hani.co.kr/arti/society/schooling/845632.html
http://www.hani.co.kr/arti/science/science_general/845514.html
http://www.hani.co.kr/arti/science/science_general/845495.html
http://www.hani.co.kr/arti/science/future/845493.html
http://www.hani.co.kr/arti/opinion/column/845475.html
http://www.hani.co.kr/arti/economy/property/845209.html

```

문제326. 아래의 url의 기사내용을 검색하시오.

보기)

<http://www.hani.co.kr/arti/economy/it/855916.html>

답)

```

import urllib.request
from bs4 import BeautifulSoup
import re

def hni2_crawling():
    params1 = []
    params2 = []
    list_url = "http://www.hani.co.kr/arti/economy/it/855916.html"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("UTF-8")
    soup = BeautifulSoup(result,'html.parser')

    for link in soup.find_all('div', class_="text"):
        print(link.get_text())

```

hni2_crawling()

과학기술정보통신부 제공

경찰의 스마트폰 카메라로 치매 노인의 지문을 스캔하면 1분 안에 신원이 확인되는
을 줄여주는 인공지능(AI) 상담 챗봇 개발 등이 추진된다.

과학기술정보통신부와 경찰청은 '치안현장 맞춤형 연구개발 시범사업' (폴리스랩) 사
관은 국민, 현장 경찰, 연구자 등을 대상으로 아이디어를 공모하고 수요를 조사해왔다
최종 선정된 6개 과제와 연구총괄 기관은 버튼을 누르면 작동하는 접이식 초경량 방
별 및 신원확인 시스템 기술개발(에코솔루션), 신고자와 구조 요청자의 정확한 위치
시스템 개발(성균관대), 성범죄 추가피해를 막기 위한 인공지능 상담형 챗봇 개발(서울
시스템 개발(한국전자통신연구원) 등이다.

정부는 이번에 뽑힌 6개 과제에 대해 향후 3년 동안 약 100억원을 지원할 계획이다.
선 경찰관과 함께 기술개발을 추진한다.

김효실 기자 trans@hani.co.kr

**문제327. 레이디 버그 게시판 글 스크롤링 할때 처럼 기사날짜와 기사 텍스트가 아래 처럼
출력**

되게 하시오.

보기)

2018.01.01 과학기술정보통신부 제공

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```
def hni2_crawling():
```

```
    list_url = "http://www.hani.co.kr/arti/economy/it/855916.html"
```

```
    url = urllib.request.Request(list_url)
```

```
    result = urllib.request.urlopen(url).read().decode("UTF-8")
```

```
    soup = BeautifulSoup(result,'html.parser')
```

```
    params1 = []
```

```
    params2 = []
```

```
    for link1, link2 in zip(soup.find_all('p', class_="date-time"),
```

```
                           soup.find_all('div', class_="text")):
```

```
        params1.append(link1.get_text())
```

```
        params2.append(link2.get_text())
```

```
    for i1, i2 in zip(params1,params2):
```

```
        print(i1.strip(),end=' ')
```

```
        print(i2.strip() )
```

```
hni2_crawling()
```

경찰의 스마트폰 카메라로 치매 노인의 지문을 스캔하면 1분 안에 신원(을) 줄여주는 인공지능(AI) 상담 챗봇 개발 등이 추진된다.

과학기술정보통신부와 경찰청은 ‘치안현장 맞춤형 연구개발 시범사업’관은 국민, 현장 경찰, 연구자 등을 대상으로 아이디어를 공모하고 수요를 최종 선정된 6개 과제와 연구총괄 기관은 버튼을 누르면 작동하는 접이:별 및 신원확인 시스템 기술개발(에코스솔루션), 신고자와 구조 요청자의 시스템 개발(성균관대), 성범죄 추가피해를 막기 위한 인공지능 상담형 시스템 개발(한국전자통신연구원) 등이다.

정부는 이번에 뽑힌 6개 과제에 대해 향후 3년 동안 약 100억원을 지원:선 경찰관과 함께 기술개발을 추진한다.

김효실 기자 trans@hani.co.kr

문제328. 위에서 만든 함수 2개를 가지고 인공지능 전체기사가 스크롤링 되게 하시오.

보기)

1. hni_crawling() : 상세기사 url을 가져오는 함수
2. hni2_crawling() : 상세기사 url로 상세기사를 가져오는 함수

답)

```
import urllib.request
from bs4 import BeautifulSoup
import re

def hni_crawling():
    for i in range(15):
        list_url = "
```



```

url = urllib.request.Request(list_url[i])
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup(result,'html.parser')

params1 = []
params2 = []
for link1, link2 in zip(soup.find_all('p', class_="date-time"),
                        soup.find_all('div', class_="text")):
    params1.append(link1.get_text())
    params2.append(link2.get_text())
for i1, i2 in zip(params1,params2):
    print(i1.strip(),end=' ')
    print(i2.strip() )

```

hni2_crawling()

문제329. (마지막) 지금 스크롤링한 기사시간과 기사 텍스트가 레이드 버그처럼 mytext9.txt에

저장되게 하시오.

답)

```

import urllib.request
from bs4 import BeautifulSoup
import re

```

```

def hni_crawling():
    for i in range(15):
        list_url = "

```

```

def hni2_crawling():

```

```

params1 = []
params2 = []
list_url=hni_crawling()

for i in range(len(list_url)): #하나씩 파싱해줘야 한다
    url = urllib.request.Request(list_url[i])
    result = urllib.request.urlopen(url).read().decode("UTF-8")
    soup = BeautifulSoup(result,'html.parser')

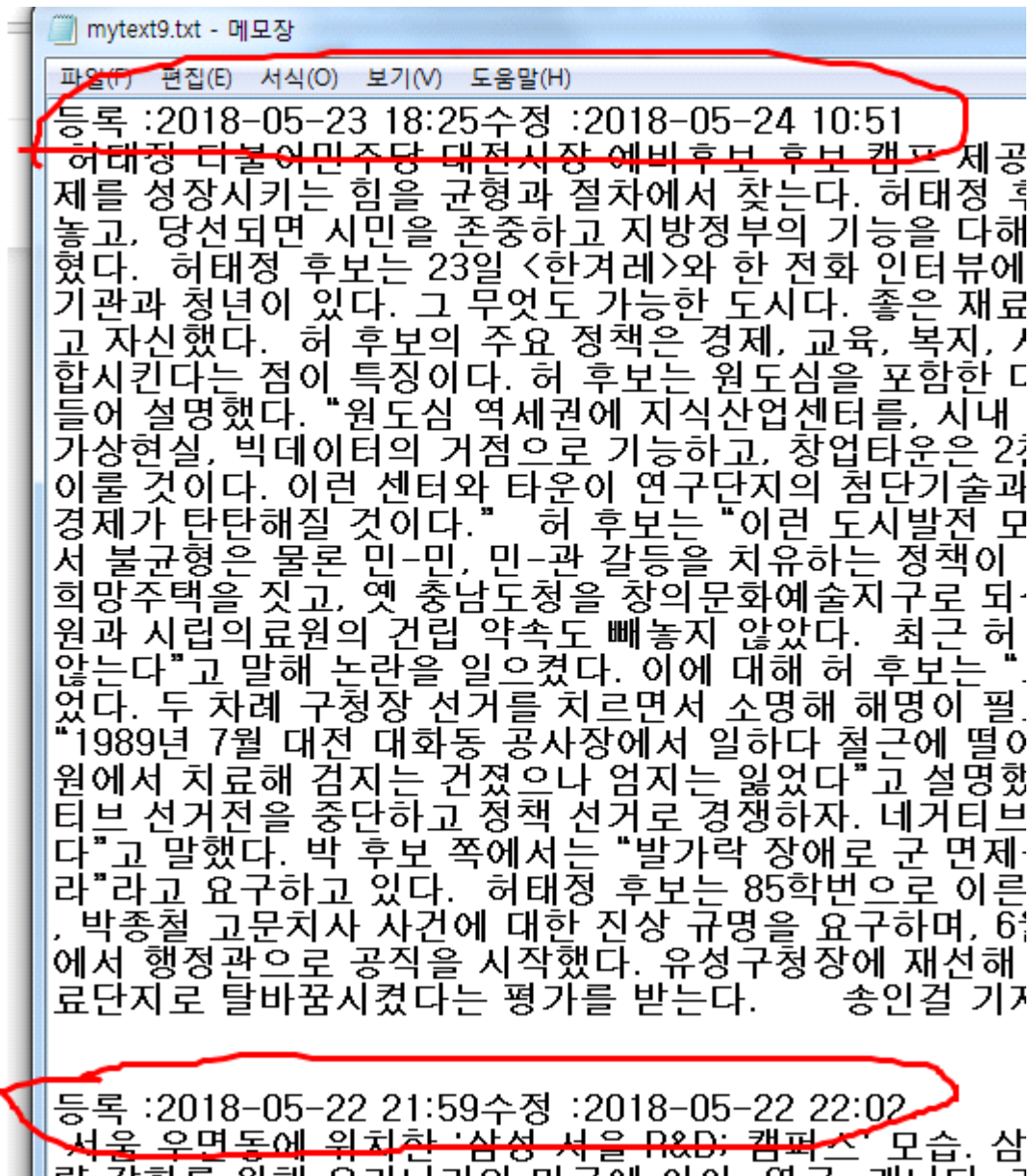
    for link1, link2 in zip(soup.find_all('p', class_="date-time"),
                           soup.find_all('div', class_="text")):
        params1.append(link1.get_text())
        params2.append(link2.get_text().strip())
    for i1, i2 in zip(params1,params2):
        print(i1,end=' ')
        print(i2)

f=open('C:\python_data\mytext9.txt','w',encoding='UTF-8')

for i1, i2 in zip(params1, params2):
    f.write('%s\n %s\n\n\n' %(i1,re.sub('[\r\n]',",",i2) ))

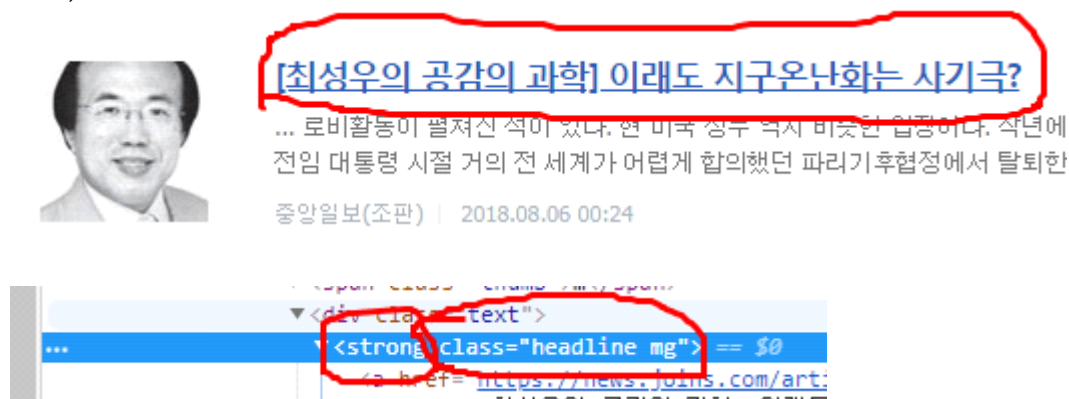
f.close()
hni2_crawling()

```



문제330. 한겨레 신문사 웹스크롤링 코드를 가져와서 중앙일보에 맞춰 웹스크롤링되게 구현.

(검색어 : 영화)
(보기)



답)

```

import urllib.request
from bs4 import BeautifulSoup
import re

def joongang_crawling():
    params1 = []
    for i in range(10):
        list_url = "https://search.joins.com/JoongangNews?  

page=" + str(i) + "&Keyword=%EC%98%81%ED%99%94  

&SortType=New&SearchCategoryType=JoongangNews"
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("UTF-8")
        soup = BeautifulSoup(result, 'html.parser')

        for j in soup.find_all('strong', class_="headline mg"):
            for k in j:
                params1.append(k.get('href'))
    return params1

joongang_crawling()

```

```

['https://news.joins.com/article/22841252',
 'https://news.joins.com/article/22841251',
 'https://news.joins.com/article/22840494',
 'https://news.joins.com/article/22840394',
 'https://news.joins.com/article/22840143',
 'https://news.joins.com/article/22838941',
 'https://news.joins.com/article/22838304',
 'https://news.joins.com/article/22838282',
 'https://news.joins.com/article/22838270',
 'https://news.joins.com/article/22838252']

```

```

import urllib.request
from bs4 import BeautifulSoup
import re

def joongang_crawling():
    params1 = []
    for i in range(10):
        list_url = "https://search.joins.com/JoongangNews?  

page=" + str(i) + "&Keyword=%EC%98%81%ED%99%94  

&SortType=New&SearchCategoryType=JoongangNews"

```

```

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("UTF-8")
soup = BeautifulSoup(result,'html.parser')

for j in soup.find_all('strong',class_="headline mg"):
    for k in j:
        params1.append(k.get('href'))
return params1

```

#상세검색

```

def joongang2_crawling():
    params1 = []
    params2 = []
    list_url=joongang_crawling()

    for i in range(len(list_url)):    #하나씩 파싱해줘야 한다
        url = urllib.request.Request(list_url[i])
        result = urllib.request.urlopen(url).read().decode("UTF-8")
        soup = BeautifulSoup(result,'html.parser')

        for link1, link2 in zip(soup.find_all('div', class_="byline"),
                                soup.find_all('div', class_="article_body")):
            params1.append(link1.get_text())
            params2.append(link2.get_text())
        for i1, i2 in zip(params1,params2):
            print(i1.strip(),end=' ')
            print(i2.strip() )

joongang2_crawling()

```

[중앙일보]

입력 2018.08.06 00:24

| 종합 27면 지면1
고, 일본을 비롯한 지구촌 곳곳이 비슷하게 재난적인 상황에 처해있다
아니다. 이로 인한 기후 교란으로 여름에는 더욱 더워지고 겨울에는 (·
· 가뭄 · 한파 · 폭설 등이 워낙 자주 일어나다 보니 이제는 기상이변(·
일치단결해서 지구온난화의 주범인 온실가스 감축을 위해 적극적인 할
도 각국의 이해관계가 충돌하다 보니 실효성 있는 대책에 합의하기가
나라들이 선진국이어서 그런 것만은 아니다. 지구온난화가 가속화될
이다. 반면에 온실가스를 가장 많이 배출하는 중국과 미국, 그리고 곧
떠넘기려는 행태마저 보여 왔다. 공감의 과학 8/6 심
한다. 지구온난화는 사실이 아니거나 설령 그렇다 해도 다른 원인에
이 여전히 적지 않다. 이들은 한술 더 떠서 지구온난화는 일부 과학자
주장마저 서슴지 않는다. 자국의 산업 보호를 위해 교토의
지구온난화에 대비한 조치는 필요 없다는 캠페인과 로비활동이 펼쳐진
임 대통령 시절 거의 전 세계가 어렵게 합의했던 파리기후협정에서 틀
다수가 기후 재난으로 이웃 나라로 피난 가야만 하는 극한적 상황이
론가

[중앙일보]

입력 2018.08.06 00:24

문제331. 위의 결과를 text파일로 생성되게하시오.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import re
```

```
def joongang_crawling():
```

```
    params1 = []
```

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

```
        list_url = "https://search Joongs.com/JoongangNews?
```

```
page= \" + str\(i\) + \"&Keyword=%EC%98%81%ED%99%94
```

```
&SortType=New&SearchCategoryType=JoongangNews"
```

```
        url = urllib.request.Request(list_url)
```

```
        result = urllib.request.urlopen(url).read().decode("UTF-8")
```

```
        soup = BeautifulSoup(result,'html.parser')
```

```
        for j in soup.find_all('strong',class_="headline mg"):
```

```
            for k in j:
```

```
                params1.append(k.get('href'))
```

```
    return params1
```

```
#상세검색
```

```
def joongang2_crawling():
```

```
    params1 = []
```

```
    params2 = []
```

```

list_url=joongang_crawling()

for i in range(len(list_url)):    #하나씩 파싱해줘야 한다
    url = urllib.request.Request(list_url[i])
    result = urllib.request.urlopen(url).read().decode("UTF-8")
    soup = BeautifulSoup(result,'html.parser')

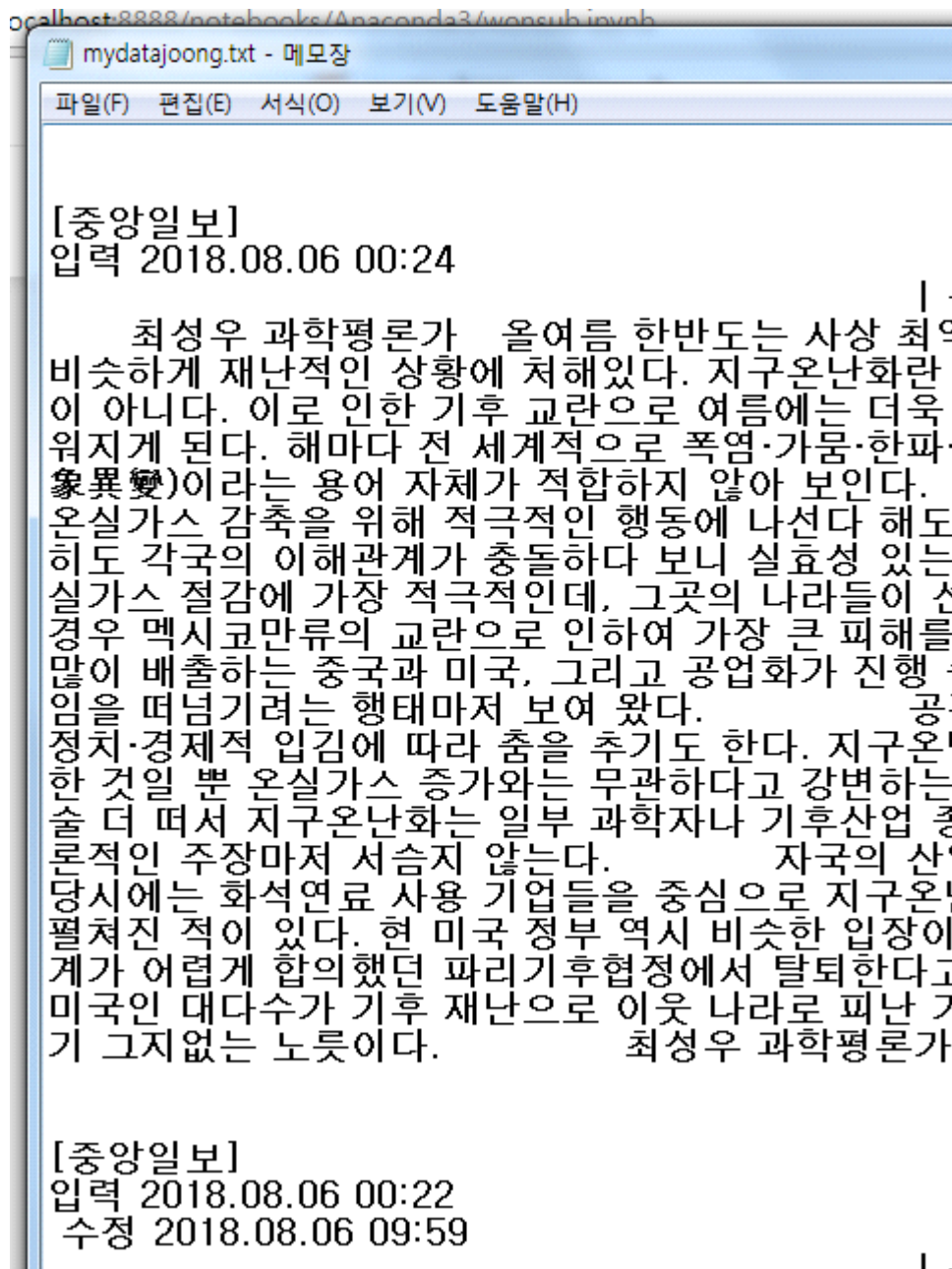
    for link1, link2 in zip(soup.find_all('div', class_="byline"),
                           soup.find_all('div', class_="article_body")):
        params1.append(link1.get_text())
        params2.append(link2.get_text())
        for i1, i2 in zip(params1,params2):
            print(i1.strip(),end=' ')
            print(i2.strip() )

f=open('C:\python_data\mydatajoong.txt','w',encoding='UTF-8')

for i1, i2 in zip(params1, params2):
    f.write('%s %s\n' %(i1,re.sub('[\r\n]','',i2) ))

f.close()
joongang2_crawling()

```



문제332. 지금 스크롤링한 텍스트 파일로 워드 클라우드를 그리시오.

답)

```
from wordcloud import WordCloud, STOPWORDS    # 워드 클라우딩 모듈
import matplotlib.pyplot as plt                # 시각화 모듈
from os import path                             # 텍스트 파일을 불러오기 위한 open,
path 하기 위해 os 임포트

d = path.dirname("c://python_data//")          # 텍스트 파일이 있는 상위 디렉토리를
path로 지정
text = open(path.join(d, "mydatajoong.txt"), mode="r", encoding="UTF-8").read()
# 텍스트파일을 open 하는데 reading만 되게 (mode="r"), UTF-8 방식으로 불러옴
(UTF-8)
```



```

wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs_ttf',
# 폰트 위치(거의 기본적으로 C://Windows//Fonts 안에 들어있습니다)
                      stopwords=STOPWORDS, background_color='white',
# STOPWORDS 옵션은 공백/줄바꾸기 기준으로 단어를 추출해 냅니다
                      width=1000,
# background_color는 워드클라우드 배경색을 나타냅니다. 'black'으로하면 검은색이
# 됩니다.
                      height=800,
# width와 height는 워드클라우드의 크기를 지정해 줍니다.
                      colormap='PuRd').generate(text)
# colormap은 워드 색깔을 지정해주는데 첨부한 색감표를 사용하시면 됩니다.
generate() 메소드는
# 워드 클라우드를 생성합니다
plt.figure(figsize=(13,13))    # matplotlib의 pyplot을 figsize로 생성합니다
plt.imshow(wordcloud)         # 워드 클라우드 이미지를 pyplot에 띄웁니다
plt.axis("off")               # pyplot에 x, y축 표시를 없앱니다.
plt.show()

```

문제333. 동아일보에서 검색어를 영화로 검색해서 웹스크롤링을 하는 파이썬 코드 구현.

```

답)
import urllib.request
from bs4 import BeautifulSoup
import re

def donga_crawling():
    params1 = []
    for i in range(10):
        list_url = "http://news.donga.com/search?p=" + str(i*15+1) + "&query=%EC%
98%81%ED%99%94&check_news=1&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')

        #p태그 밑에있는 a태그 부분만 가져와라
        for k in range(15):
            for j in soup.find_all('p', class_='tit')[k]('a'):
                params1.append(j.get('href'))

```

```
return params1
```

```
#상세검색
```

```
def donga2_crawling():
```

```
    params1 = []
```

```
    params2 = []
```

```
    list_url=donga_crawling()
```

```
    for i in range(len(list_url)):    #하나씩 파싱해줘야 한다
```

```
        url = urllib.request.Request(list_url[i])
```

```
        result = urllib.request.urlopen(url).read().decode("UTF-8")
```

```
        soup = BeautifulSoup(result,'html.parser')
```

```
        for link1, link2 in zip(soup.find_all('span', class_="date01"),
```

```
                                soup.find_all('div', class_="article_txt")):
```

```
            params1.append(link1.get_text())
```

```
            params2.append(link2.get_text())
```

```
        for i1, i2 in zip(params1,params2):
```

```
            print(i1.strip(),end=' ')
```

```
            print(i2.strip() )
```

```
f=open('C:\python_data\wwwmydatadonga.txt','w',encoding='UTF-8')
```

```
for i1, i2 in zip(params1, params2):
```

```
    f.write('%s %s\n\n\n' %(i1,re.sub('[\r\n]','',i2) ))
```

```
f.close()
```

```
donga2_crawling()
```

워드클라우드

```
from wordcloud import WordCloud, STOPWORDS    # 워드 클라우딩 모듈
```

```
import matplotlib.pyplot as plt                # 시각화 모듈
```

```
from os import path                            # 텍스트 파일을 불러오기 위한 open,
```

```
path 하기 위해 os 임포트
```

```
d = path.dirname("c://python_data//")        # 텍스트 파일이 있는 상위 디렉토리를  
path로 지정
```

```
text = open(path.join(d, "mydatadonga.txt"), mode="r", encoding="UTF-8").read()
```

텍스트파일을 open 하는데 reading만 되게 (mode="r"), UTF-8 방식으로 불러옴 (UTF-8)

```
wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs_ttf',
# 폰트 위치(거의 기본적으로 C://Windows//Fonts 안에 들어있습니다)
                        stopwords=STOPWORDS, background_color='white',
# STOPWORDS 옵션은 공백/줄바꾸기 기준으로 단어를 추출해 냅니다
                        width=1000,
# background_color는 워드클라우드 배경색을 나타냅니다. 'black'으로하면 검은색이
# 됩니다.
                        height=800,
# width와 height는 워드클라우드의 크기를 지정해 줍니다.
                        colormap='PuRd').generate(text)
# colormap은 워드 색깔을 지정해 주는데 첨부한 색감표를 사용하시면 됩니다.
generate() 메소드는
# 워드 클라우드를 생성합니다
plt.figure(figsize=(13,13)) # matplotlib의 pyplot을 figsize로 생성합니다
plt.imshow(wordcloud) # 워드 클라우드 이미지를 pyplot에 띄웁니다
plt.axis("off") # pyplot에 x, y축 표시를 없앱니다.
plt.show()
```

n사 이미지 스크롤링

문제334. (이미지 스크롤링) 네이버에서 아이언맨으로 이미지 검색을 했을때 나오는 이미지들

을모두 스크롤링 하시오.

답)

```
import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
binary = 'D:\chromedriver_win32'
browser = webdriver.Chrome(binary)
```

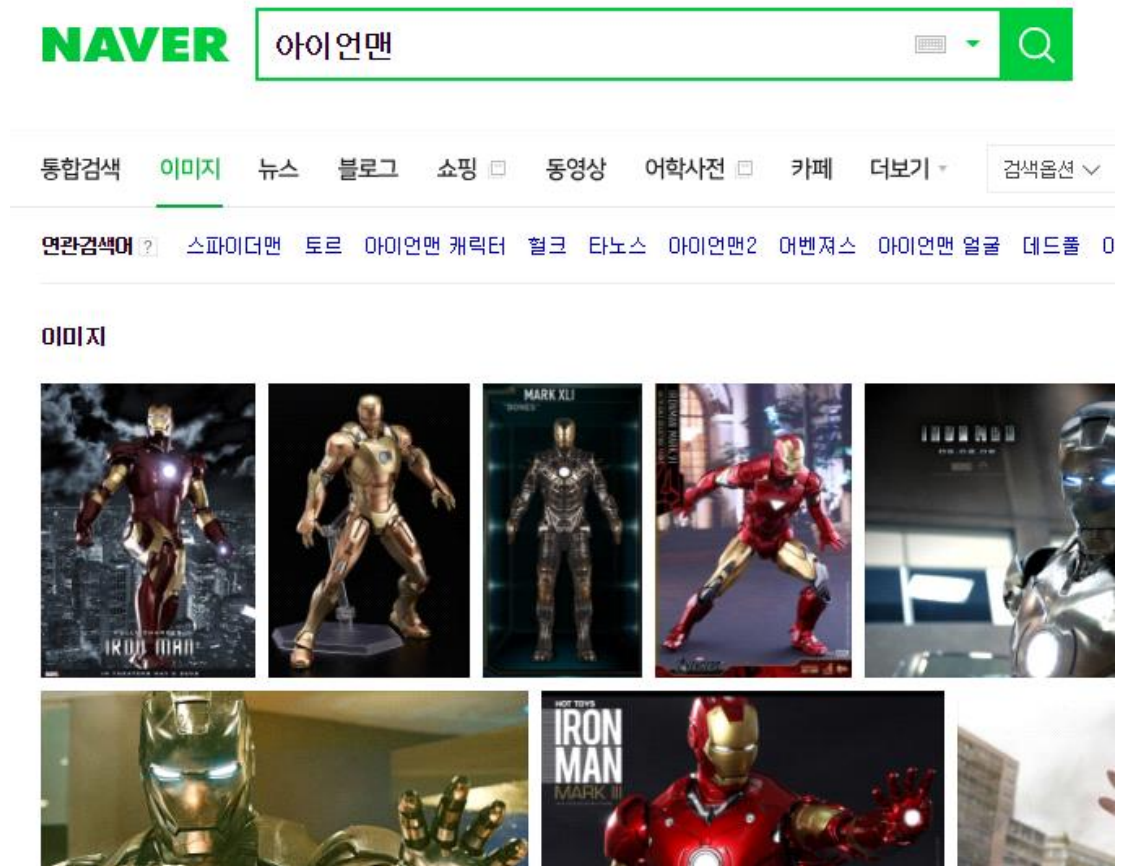
#네이버 이미지 URL

```
browser.get("https://search.naver.com/search.naver?
where=image&sm=stb_nmr&")
```

```
elem = browser.find_element_by_id("nx_query") #클래스말고 id를 찾는게 더 좋다
#find_elements_by_class_name("") #클래스로 찾기
```

검색어 입력

```
elem.send_keys("아이언맨")
elem.submit()
```



설명

: selenium으로 스크롤링해야지 해당 사이트에 민폐를 끼치지 않고 가져올 수 있다.

#스크롤을 내려야지 더보기 생기는데 이 작업하는 방법

반복할 횟수

```
for i in range(1,6):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5) #5초마다 한번씩 내린다

time.sleep(5)
html = browser.page_source
soup = BeautifulSoup(html,"html.parser")
print(soup)
```



```
# 다운받을 폴더경로 입력
urllib.request.urlretrieve(p, "d:/naverimages/" + str(a) + ".jpg" )
a = a + 1
```

```
fetch_detail_url()
```

종합)

```
import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
binary = 'D:\chromedriver_win32\chromedriver.exe'
browser = webdriver.Chrome(binary)
```

#네이버 이미지 URL

```
browser.get("https://search.naver.com/search.naver?
where=image&sm=stb_nmr&")
elem = browser.find_element_by_id("nx_query") #클래스말고 id를 찾는게 더 좋다
#find_elements_by_class_name("")
```

검색어 입력

```
elem.send_keys("아이언맨")
elem.submit()
```

반복할 횟수

```
for i in range(1,6):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5) #5초마다 한번씩 내린다
```

```
time.sleep(5)
html = browser.page_source
soup = BeautifulSoup(html,"html.parser")
print(soup)
print(len(soup))
```

이미지에 해당하는 URL가져오기

```
def fetch_list_url():
    params = []
```

```

imgList = soup.find_all("img", class="_img")
for im in imgList:
    params.append(im["src"])
return params
print(fetch_list_url())

```

#이미지를 폴더에 저장하는 방법

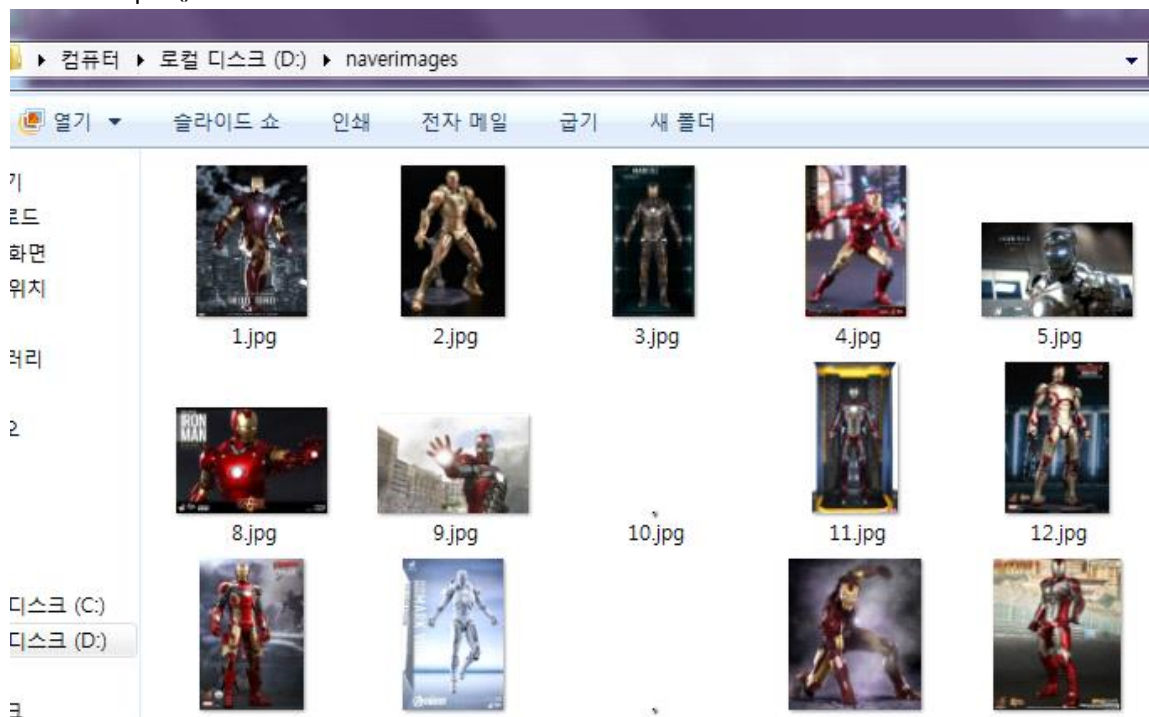
```

def fetch_detail_url():
    params = fetch_list_url()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "d:/naverimages/" + str(a) + ".jpg" )
        a = a + 1

```

fetch_detail_url()

browser.quit()



가끔 이미지가 잘 안나오는게 있는데 인터넷이 느려서 잘 못가져와서 그렇다.
인터넷이 빠르거나 sleep을 더 오래주면 가져올 수 있음.

g사 이미지 스크롤링

문제335. 구글에서 햄버거로 검색했을때의 사진을 d드라이브 밑에 googleimages라는 폴더에

저장되게 하시오.

답)

```
import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver

# 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크

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 위의 위치에 둔다
# 팬텀 js로 하면 백그라운드로 실행할 수 있음
binary = 'D:\Wchromedriver_win32\chromedriver.exe'

# 브라우저를 인스턴스화
browser = webdriver.Chrome(binary)

# 네이버의 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")

# 네이버의 이미지 검색에 해당하는 input 창의 id 가 'nx_query' 임(검색창에 해당하는 html코드를 찾아서 elem 사용하도록 설정)
# input창 찾는 방법은 원노트에 있음
# find_elements_by_class_name("") --> 클래스 이름으로 찾을때는 이렇게
elem = browser.find_element_by_id("lst-ib")

##### 검색어 입력
#####

# elem 이 input 창과 연결되어 스스로 햄버거 검색
elem.send_keys("햄버거")
# 웹에서의 submit 은 엔터의 역할을 함
elem.submit()

##### 반복할 횟수
```



```
#####
```

```
# 스크롤을 내리려면 브라우저 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번  
하고 End키를 눌러야함
```

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

```
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
```

```
    time.sleep(10)          # END 키 누르고 내려가는데 시간이 걸려서 sleep 해  
    줌
```

```
time.sleep(10)          # 네트워크 느릴까봐 안정성 위해 sleep 해줌
```

```
html = browser.page_source      # 크롬브라우저에서 현재 불러온 소스 가져옴
```

```
soup = BeautifulSoup(html, "html.parser") # html 코드를 검색할 수 있도록 설정
```

```
##### 그림파일 저장
```

```
#####
```

```
def fetch_list_url():
```

```
    params = []
```

```
    imgList = soup.find_all("img", class_="rg_ic rg_i") # 네이버 이미지 url 이 있는  
img 태그의 _img 클래스에 가서
```

```
    for im in imgList:
```

```
        try :
```

```
            params.append(im["src"])          # params 리스트에 image url 을 담  
음
```

```
        except KeyError:
```

```
            params.append(im["data-src"])
```

```
    return params
```

```
def fetch_detail_url():
```

```
    params = fetch_list_url()
```

```
    for idx,p in enumerate(params,1):
```

```
        # 다운받을 폴더경로 입력
```

```
        urllib.request.urlretrieve(p, "d:/googleImages/" + str(idx) + ".jpg")
```

```
if __name__ == '__main__':
```

```
    # 메인 실행 함수
```

```
fetch_detail_url()
```

```
# 끝나면 브라우저 닫기
```

```
browser.quit()
```



d사 이미지 웹스크롤링

문제336. 다음 이미지 검색에서 벤 으로 검색해서 나오는 이미지들을 d드라이브 밑에 **daumimages**라는 폴더에 저장되게 하시오.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
from selenium import webdriver
```

```
from selenium.webdriver.common.keys import Keys
```

```
import time
```

```
binary = 'D:\chromedriver_win32\chromedriver.exe'
```

```
browser = webdriver.Chrome(binary)
```

```
browser.get("http://search.daum.net/search?
```

```
nil_suggest=btn&w=img&DA=SBC&q=")
```

```
elem = browser.find_element_by_id("q")
```

```
# find_elements_by_class_name("")
```

```
# 검색어 입력
```

```

elem.send_keys("벤")
elem.submit()

# 반복할 횟수
for i in range(1,3):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(20)

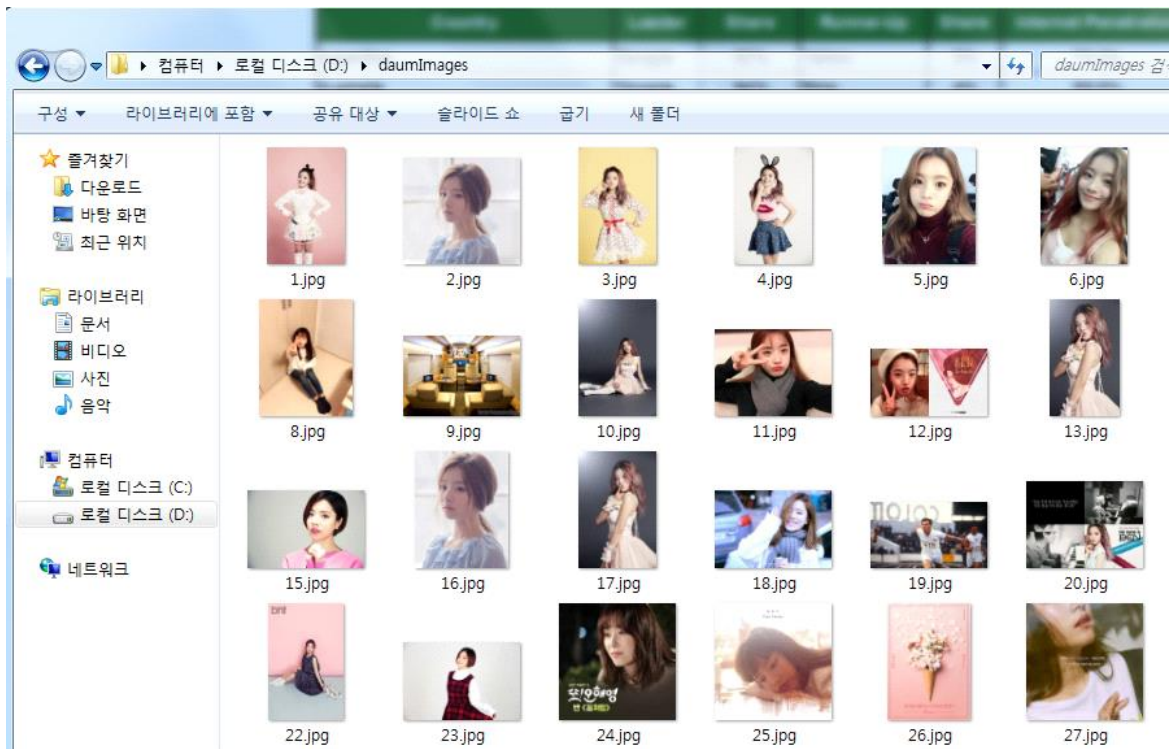
time.sleep(20)
html = browser.page_source
soup = BeautifulSoup(html, "lxml")
# print(soup)
# print(len(soup))

def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="thumb_img")
    for im in imgList:
        params.append(im["src"])
    return params

def fetch_detail_url():
    params = fetch_list_url()
    print(params)
    a = 1
    for p in params:
        print(p)
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "d:/daumImages/" + str(a) + ".jpg")
        a = a + 1

fetch_detail_url()
browser.quit()

```



b사 이미지 웹스크롤링

문제337. B사 에서 선풍기 사진으로 조회했을때 나오는 이미지를 d드라이브 밑에 bing에 저장

하시오.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
from selenium import webdriver
```

```
from selenium.webdriver.common.keys import Keys
```

```
import time
```

```
binary = 'D:\chromedriver_win32\chromedriver.exe'
```

```
browser = webdriver.Chrome(binary)
```

```
browser.get("https://www.bing.com/?scope=images&FORM=Z9LH1") #빙 이미지 검색 url
```

```
elem = browser.find_element_by_id("sb_form_q")
```

```
# 검색어 입력
```

```
elem.send_keys("선풍기")
```

```
elem.submit()
```

```
# 반복할 횟수
```

```
for i in range(1, 3):
```

```
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
```

```

        time.sleep(5)
time.sleep(5)
html = browser.page_source
soup = BeautifulSoup(html, "lxml")

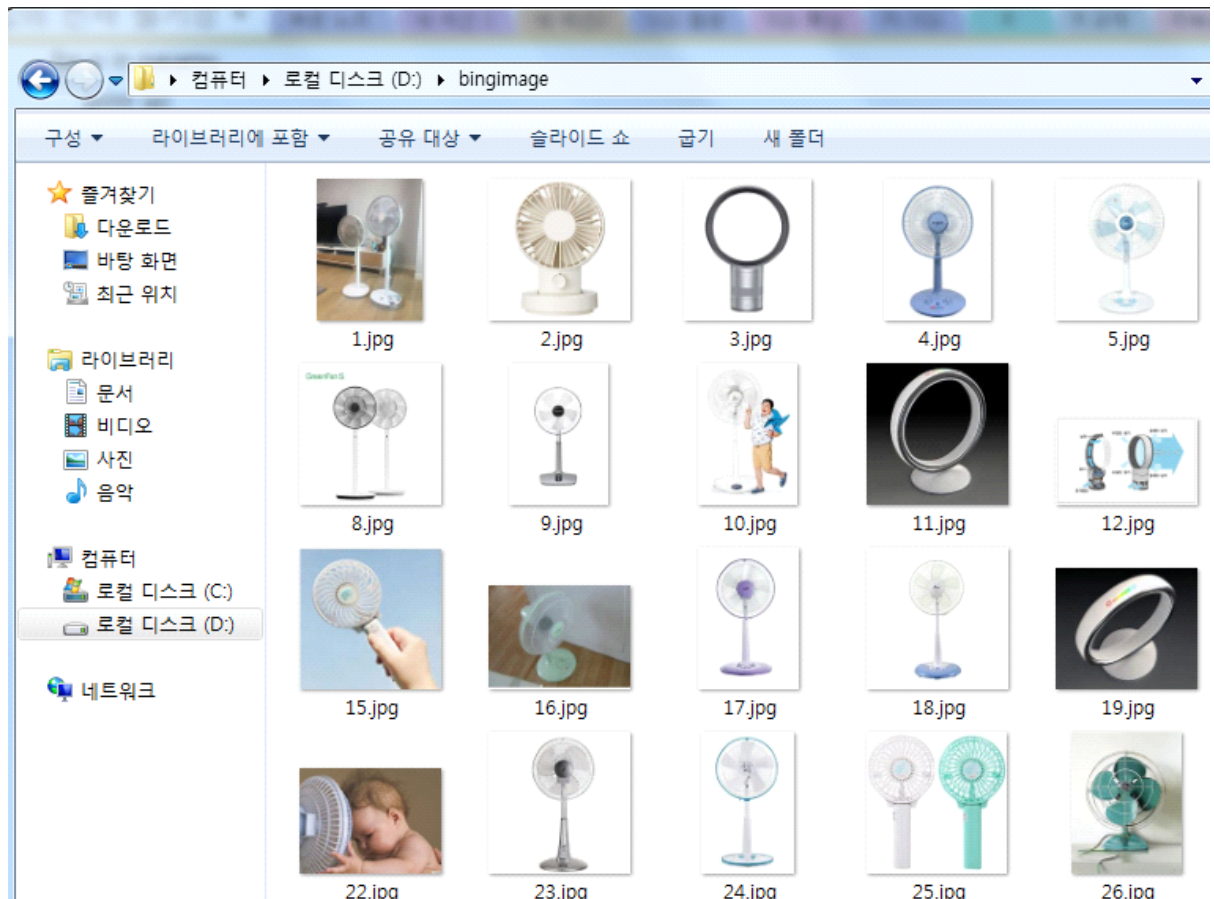
def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="mimg")
    for im in imgList:
        params.append(im["src"])
    return params

def fetch_detail_url():
    params = fetch_list_url()
    a = 1
    for p in params:
        urllib.request.urlretrieve(p, "d:/bingimage/" + str(a) + ".jpg")
        a = a + 1

fetch_detail_url()

browser.quit()

```



서울시 응답소 게시판

문제338. 서울시 응답소 게시판의 글을 웹스크롤링을 하기 위해 페이지마다 html코드를 가져올

수 있도록 하시오.

답)

1. 페이지 번호 가져오기

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import os
```

```
import re
```

```
def fetch_list_url():
```

```
    params=[]
```

```
    for i in range(1,30):
```

```
        list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
```

```
        request_header=urllib.parse.urlencode( {"page":i} )
```

```
        print(request_header)
```

```
fetch_list_url()
```

```
page=1
page=2
page=3
page=4
page=5
page=6
page=7
page=8
page=9
page=10
page=11
page=12
page=13
page=14
page=15
page=16
page=17
page=18
page=19
page=20
page=21
page=22
page=23
page=24
page=25
page=26
page=27
page=28
page=29
```

문제339. 아래의 서울시 응답소 게시판 url 과 페이지 번호를 같이 제공해야 해당 페이지의 html코드를 내려받을 수 있다. 해당 페이지의 html코드를 출력하시오.

답)

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    params=[]
    for i in range(1,30):
        list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header=urllib.parse.urlencode( {"page":i} )
        #print(request_header) page = 1, page2 ...가 출력

        request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게
        #변환
        print(request_header)
```

fetch_list_url()

```
b'page=1'  
b'page=2'  
b'page=3'  
b'page=4'  
b'page=5'  
b'page=6'  
b'page=7'  
b'page=8'  
b'page=9'  
b'page=10'  
b'page=11'  
b'page=12'  
b'page=13'
```

```
import urllib.request  
from bs4 import BeautifulSoup  
import os  
import re
```

```
def fetch_list_url():
```

```
    params=[]
```

```
    for i in range(1,30):
```

```
        list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
```

```
        request_header=urllib.parse.urlencode( {"page":i} )
```

```
        #print(request_header) page = 1, page2 ...가 출력
```

```
        request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게  
        변환
```

```
        print(request_header)
```

```
        #페이지마다 html 코드 출력
```

```
        url=urllib.request.Request(list_url, request_header)
```

```
        res=urllib.request.urlopen(url).read().decode("UTF-8")
```

```
        print(res)
```

```
fetch_list_url()
```



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="k
    <head>

        <title>서울특별시 응답소</title>

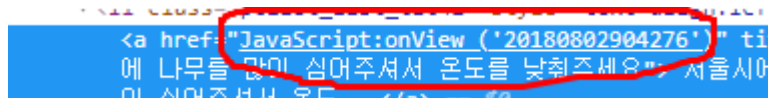
        <meta name="viewport" content="width=device-widt
=no, target-densityDpi=medium-dpi" />
        <meta http-equiv="X-UA-Compatible" content="IE=
        <meta property="og:url" content="http:
        <meta property="og:type" content="websi
        <meta property="og:title" content="서울
        <meta property="og:description" content="천만
불편, 건의, 질의 민원과 더불어 인권침해, 민생침해 등 다양한 시민

```

문제340. 게시글의 상세 url 부분을 스크롤링 하시오.

답)

1. 게시글의 자바스크립트 번호를 복사 한다.



JavaScript: onView ('20180802904276')

2. 게시글의 제목을 따오는 작업

```
soup = BeautifulSoup(res, "html.parser")
```

```
soup2 = soup.find_all("li", class_="pclist_list_tit42")
```

```
print(soup2)
```

```

[<li class="pcclist_list_tit42" style="text-align:left;">
<a href="JavaScript: onView ('20180802904276')" title="서울시에 나무를 많이 심어주셔서 온도를 낮춰주세요":
온도...</a>
</li>, <li class="pcclist_list_tit42" style="text-align:left;">
<a href="JavaScript: onView ('20180802901294')" title="종량제봉투에 비닐봉투 모양으로 손잡이 만들어주세요
손잡이...</a>
</li>, <li class="pcclist_list_tit42" style="text-align:left;">
<a href="JavaScript: onView ('20180801905041')" title="보라매 독서실을 서울시에서 직접 운영하라!!"> 보라매
영...</a>
...

```

문제341. 20180802904276 부분만 추출하시오.

답)

```
import urllib.request
```

```
from bs4 import BeautifulSoup
```

```
import os
```

```
import re
```

```
def fetch_list_url():
```

```
    params=[]
```

```

for i in range(1,30):
    list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
    request_header=urllib.parse.urlencode( {"page":i} )
    #print(request_header) page = 1, page2 ...가 출력

    request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게
    변환

    #페이지마다 html 코드 출력
    url=urllib.request.Request(list_url, request_header)
    res=urllib.request.urlopen(url).read().decode("UTF-8")

    soup = BeautifulSoup(res, "html.parser")
    soup2 = soup.find_all("li",class_="plist_list_tit2")

    for soup3 in soup2:
        soup4 = soup3.find('a')['href']
        print(re.search("[0-9]{14}", soup4).group()) #0~9까지 숫자를 14자리 가져온
    다

```

```

fetch_list_url()

```

```

20180802904276
20180802901294
20180801905041
20180801904003
20180801900113
20180731902058
20180731901736
20180731901243
20180730905546
20180730905546

```

문제342. 위의 결과의 숫자를 **params**라는 리스트에 담고 **return** 하도록 위의 함수를 수정 하시

오.

답)

```

import urllib.request
from bs4 import BeautifulSoup
import os
import re

```

```

def fetch_list_url():
    params=[]

```

```

for i in range(1,30):
    list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
    request_header=urllib.parse.urlencode( {"page":i} )
    #print(request_header) page = 1, page2 ...가 출력

    request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게
    변환

    #페이지마다 html 코드 출력
    url=urllib.request.Request(list_url, request_header)
    res=urllib.request.urlopen(url).read().decode("UTF-8")

    soup = BeautifulSoup(res, "html.parser")
    soup2 = soup.find_all("li",class_="pclist_list_tit42")

    for soup3 in soup2:
        soup4 = soup3.find('a')['href']
        params.append(re.search("[0-9]{14}", soup4).group())
    return params
fetch_list_url()
['20180802904276',
'20180802901294',
'20180801905041',
'20180801904003',
'20180801900113',
'20180731902058',
'20180731901736',
'20180731901243',
'20180730905546',
'20180730905058',
'20180730904362',
'20180730903363',
'20180730901324',
'20180730900779',
'20180729902355',
'20180729901148',
'20180729900348',
'20180728904568',
'20180728900000'

```

게시글 자바 스크립트

```

function onView (Seq) {
    var Frm = document.mainForm;
    Frm.RCEPT_NO.value = Seq;
    Frm.action = "/Shr/Shr01/Shr01_vie.jsp";
    Frm.submit ();
}

```

```
}
```

위의 결과 숫자 리스트를 자바 스크립트 RCEPT_NO에 넣으시오.

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    params=[]
    for i in range(1,30):
        list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header=urllib.parse.urlencode( {"page":i} )
        #print(request_header) page = 1, page2 ...가 출력

        request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게
        변환

        #페이지마다 html 코드 출력
        url=urllib.request.Request(list_url, request_header)
        res=urllib.request.urlopen(url).read().decode("UTF-8")

        soup = BeautifulSoup(res, "html.parser")
        soup2 = soup.find_all("li",class_="plist_list_tit42")

        for soup3 in soup2:
            soup4 = soup3.find('a')['href']
            params.append(re.search("[0-9]{14}", soup4).group())
    return params

#상세 게시글 페이지 html 가져오는 부분
def fetch_list_url2():
    params2 = fetch_list_url()

    for i in params2:
        detail_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01\_vie.jsp"
        #게시글 클릭 후 URL
        request_header = urllib.parse.urlencode({'RECP_NO':str(i)})
        request_header = request_header.encode('UTF-8')
```

```
url = urllib.request.Request(detail_url, request_header)
res = urllib.request.urlopen(url).read().decode('UTF-8')
soup = BeautifulSoup(res, "html.parser")
print(soup)
```

fetch_list_url2()

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http:

<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>서울특별시 응답소</title>
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1
i" name="viewport"/>
<meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"/>
<meta content="http://eungdapso.seoul.go.kr" property="og:url"/>
<meta content="website" property="og:type"/>
<meta content="서울특별시 응답소" property="og:title"/>
<meta content="천만 서울시민의 의견을 접수받는 온라인 민원 창구입니
생침해 등 다양한 시민의 소리를 접수하고 처리합니다." property="og:des
<meta content="http://eungdapso.seoul.go.kr/images/ico/logo.jpg" prop
<meta content="천만 서울시민의 의견을 접수받는 온라인 민원 창구입니
생침해 등 다양한 시민의 소리를 접수하고 처리합니다." name="descripti
<meta content="bb9dfdae0be8dbdfdbcfadd2c7e3c4c907a2170a" name="naver-
<link href="/images/ico/logo_icon_114.png" rel="shortcut icon"/>
```

문제343. 위의 전체 html 코드에서 div 태그에 form_table 클래스에 해당하는 html코드만 가져

오게 하시오.

보기)

```
... <div class="form_table"> == $0
<table cellpadding="0" cellspacing="0">
```

제목	서울시에 나무를 많이 심어주셔서 온도를 낮춰주세요
공개일	2018-08-02
<p>Q. 상담내용</p> <p>안녕하세요</p> <p>저는 동작구 대방동에 살고 있는 시민이구요</p> <p>서울로 올라와서 산지는 17년정도 되었습니다.</p> <p>건의드리고자 하는 것은 이제는 기후가 변화되어 여름철은 과거와는 달리 여름철이 길고 더워지고 있는데요.</p> <p>특히 올해는 더 기온이 높습니다. 그리고 새벽에도 기온이 떨어지지 않고 높은데요.</p> <p>얼마전에 폭전이 40도가 넘었다고 기사가 났었는데, 폭전은 새벽기온이 26도, 서울은 30도가 넘더라구요. 즉 서울은 밤에도 기온이 내려가지 않는 열대야 현상을 보이는데요.</p> <p>최근에 도심 나무 심기에 관심이 많아서 자료를 찾아보던중에 대구시 나무심기 기사를 보게 되었습니다. 대구가 전국에서 더운지역</p>	

답)

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re
```

```

def fetch_list_url():
    params=[]
    for i in range(3):
        list_url="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header=urllib.parse.urlencode( {"page":i} )
        #print(request_header) page = 1, page2 ...가 출력

        request_header = request_header.encode('UTF-8') #컴퓨터가 알아볼 수 있게
        변환

        #페이지마다 html 코드 출력
        url=urllib.request.Request(list_url, request_header)
        res=urllib.request.urlopen(url).read().decode("UTF-8")

        soup = BeautifulSoup(res, "html.parser")
        soup2 = soup.find_all("li",class_="plist_list_tit42")

        for soup3 in soup2:
            soup4 = soup3.find('a')['href']
            params.append(re.search("[0-9]{14}", soup4).group())
    return params

#상세 게시글 페이지 html 가져오는 부분
def fetch_list_url2():
    params2 = fetch_list_url()

    for i in params2:
        detail_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01\_vie.jsp" #게시글 클릭 후 URL
        request_header = urllib.parse.urlencode({'RECP_NO':str(i)})
        request_header = request_header.encode('UTF-8')

        url = urllib.request.Request(detail_url, request_header)
        res = urllib.request.urlopen(url).read().decode('UTF-8')

        soup = BeautifulSoup(res, "html.parser")
        soup2 = soup.find("div",class_="form_table")
        print(soup2)

```

```
fetch_list_url2()
```

문제344. div 태그에 form_table 클래스 안쪽에 table 태그에 해당하는 html코드에 접근하시

오.(table 태그가 여러 개 이므로 find_all을 사용해야 한다.)

답)

웹사이트 막혀서 종합소스로 해보시오.

```
import urllib.request # 웹브라우저에서 html 문서를 얻어오기 위해 통신하는 모듈
from bs4 import BeautifulSoup # html 문서 검색 모듈
import os
import re
```

```
def get_save_path():
```

```
    save_path = input("Enter the file name and file location : " ) #텍스트 파일 만들 위치와 이름
```

```
    save_path = save_path.replace(" ", "/")
```

```
    if not os.path.isdir(os.path.split(save_path)[0]):
```

```
        os.mkdir(os.path.split(save_path)[0])
```

```
    return save_path
```

```
def fetch_list_url():
```

```
    params = []
```

```
    for j in range(1,5):
```

```
        list_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01\_lis.jsp"
```

```
        request_header = urllib.parse.urlencode({"page": j})
```

```
        # print (request_header) # 결과 page=1, page=2 ..
```

```
        request_header = request_header.encode("utf-8")
```

```
        # print (request_header) # b'page=29'
```

```
        url = urllib.request.Request(list_url, request_header)
```

```
        # print (url) # <urllib.request.Request object at 0x00000000021FA2E8>
```

```
        res = urllib.request.urlopen(url).read().decode("utf-8")
```

```
        soup = BeautifulSoup(res, "html.parser")
```

```

        soup2 = soup.find_all("li", class_="plist_list_tit42")
        for soup3 in soup2:
            soup4 = soup3.find("a")["href"]
            params.append(re.search("[0-9]{14}", soup4).group())

    return params

def fetch_list_url2():

    params2 = fetch_list_url()

    f = open(get_save_path(), 'w', encoding = "utf-8")

    for i in params2:

        detail_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01\_vie.jsp"

        request_header = urllib.parse.urlencode({"RCEPT_NO": str(i) })
        request_header = request_header.encode("utf-8")

        url = urllib.request.Request(detail_url, request_header)
        res = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup(res, "html.parser")
        soup2 = soup.find("div", class_="form_table")

        tables = soup2.find_all("table")
        table0 = tables[0].find_all("td")
        table1 = tables[1].find("div", class_="table_inner_desc")
        table2 = tables[2].find("div", class_="table_inner_desc")

        date = table0[1].get_text()
        title = table0[0].get_text()
        question = table1.get_text(strip=True)
        answer = table2.get_text(strip=True)

        f.write("==" * 30 + "\n")
        f.write(title + "\n")
        f.write(date + "\n")

```



```

f.write(question + "\n")
f.write(answer + "\n")
f.write("==" * 30 + "\n")

f.close()

```

fetch_list_url2()

Enter the file name and file location :c://python_data//ttt.txt

문제345. 서울시 응답소 게시판의 글들을 워드 클라우드로 시각화 하시오.

```

답)
def word_cloud():
    from wordcloud import WordCloud, STOPWORDS    # 워드 클라우딩 모듈
    import matplotlib.pyplot as plt              # 시각화 모듈
    from os import path                           # 텍스트 파일을 불러오기 위한 open,
    path 하기 위해 os 임포트
    import re

    d = path.dirname("C:/python_data/")           # 텍스트 파일이 있는 상위
    디렉토리를 path로 지정
    file_name = input('워드 클라우드를 그릴 텍스트 파일명 입력: ex) abc.txt \n')
    text = open(path.join(d, file_name), mode="r", encoding="UTF-8").read()    # 텍
    스트파일을 open 하는데 reading만 되게 (mode="r"), UTF-8 방식으로 불러옴(UTF-8)

    text = re.sub("있다","",text)
    text = re.sub("있는","",text)
    text = re.sub("대한","",text)
    text = re.sub("한다","",text)
    text = re.sub("하는","",text)
    text = re.sub("했다","",text)
    text = re.sub("통해","",text)
    text = re.sub("이런", "", text)
    text = re.sub("위해", "", text)
    text = re.sub("위한", "", text)
    text = re.sub("것이다", "", text)
    text = re.sub("것은", "", text)
    text = re.sub("다른", "", text)
    text = re.sub("같은", "", text)
    text = re.sub("등을", "", text)
    text = re.sub("아니라", "", text)

```

```
text = re.sub("이라고", "", text)
text = re.sub("것으로", "", text)
text = re.sub("밝혔다", "", text)
text = re.sub("많이", "", text)
text = re.sub("함께", "", text)
text = re.sub("가장", "", text)
text = re.sub("사진", "", text)
text = re.sub("없다", "", text)
text = re.sub("것이", "", text)
text = re.sub("된다", "", text)
text = re.sub("없는", "", text)
text = re.sub("라는", "", text)
text = re.sub("그런", "", text)
text = re.sub("하고", "", text)
text = re.sub("어떻게", "", text)
text = re.sub("어떤", "", text)
text = re.sub("많은", "", text)
text = re.sub("하지만", "", text)
text = re.sub("그는", "", text)
text = re.sub("지난", "", text)
text = re.sub("이후", "", text)
text = re.sub("어떤", "", text)
text = re.sub("것도", "", text)
text = re.sub("아니다", "", text)
text = re.sub("때문에", "", text)
text = re.sub("따르면", "", text)
text = re.sub("에서", "", text)
text = re.sub("않고", "", text)
text = re.sub("있었다", "", text)
text = re.sub("이어", "", text)
text = re.sub("좋은", "", text)
text = re.sub("지음", "", text)
text = re.sub("너무", "", text)
text = re.sub("올해", "", text)
text = re.sub("이번", "", text)
text = re.sub("제가", "", text)
text = re.sub("보면", "", text)
text = re.sub("당시", "", text)
text = re.sub("다시", "", text)
text = re.sub("모든", "", text)
```

```
text = re.sub("이렇게", "", text)
text = re.sub("다양한", "", text)
text = re.sub("않는다", "", text)
text = re.sub("그러나", "", text)
text = re.sub("그의", "", text)
text = re.sub("라고", "", text)
text = re.sub("같다", "", text)
text = re.sub("최근", "", text)
text = re.sub("있을", "", text)
text = re.sub("않은", "", text)
text = re.sub("대해", "", text)
text = re.sub("됐다", "", text)
text = re.sub("특히", "", text)
text = re.sub("설명", "", text)
text = re.sub("없었다", "", text)
text = re.sub("보다", "", text)
text = re.sub("있고", "", text)
text = re.sub("때문이다", "", text)
text = re.sub("오후", "", text)
text = re.sub("그리고", "", text)
text = re.sub("것을", "", text)
text = re.sub("연합뉴스", "", text)
text = re.sub("않는", "", text)
text = re.sub("있습니다", "", text)
text = re.sub("합니다", "", text)
text = re.sub("것입니다", "", text)
text = re.sub("서울특별시", "", text)
text = re.sub("현재", "", text)
text = re.sub("서울시", "", text)
text = re.sub("감사", "", text)
text = re.sub("검토부서", "", text)
text = re.sub("또한", "", text)
text = re.sub("주시기 바랍니다", "", text)
text = re.sub("있도록", "", text)
text = re.sub("있어", "", text)
text = re.sub("입니다", "", text)
text = re.sub("생각", "", text)
text = re.sub("서울", "", text)
text = re.sub("있으며", "", text)
text = re.sub("운영", "", text)
```

```

text = re.sub("관련", "", text)
text = re.sub("저는", "", text)
text = re.sub("가정에 건강과", "", text)
text = re.sub("담당자", "", text)
text = re.sub("드립니다", "", text)
text = re.sub("대하여", "", text)
text = re.sub("하여", "", text)
text = re.sub("바랍니다", "", text)
text = re.sub("알려", "", text)

```

```

wordcloud = WordCloud(font_path='C://Windows//Fonts//HMKMRHD.TTF',
# 폰트 위치(거의 기본적으로 C://Windows//Fonts 안에 들어있습니다)
                        stopwords=STOPWORDS, background_color='white',
# STOPWORDS 옵션은 공백/줄바꾸기 기준으로 단어를 추출해 냅니다
                        width=1000, # background_color는 워드클라우드 배경색을 나
타냅니다. 'black'으로하면 검은색이 됩니다.
                        height=800, # width와 height는 워드클라우드의 크기를 지정
해 줍니다.
                        colormap='winter').generate(text)
# colormap은 워드 색깔을 지정해주는데 첨부한 색감표를 사용하시면 됩니다.
generate() 메소드는
plt.figure(figsize=(13,13)) # matplotlib의 pyplot을 figsize로 생성합니다
plt.imshow(wordcloud) # 워드 클라우드 이미지를 pyplot에 띄웁니다
plt.axis("off") # pyplot에 x, y축 표시를 없앱니다.
plt.show()

```

```
word_cloud()
```

파이썬2 페이지 157

필수 알고리즘

2018년 7월 25일 수요일 오후 3:18

문제1. 아래의 리스트를 정렬해서 출력하시오.

보기)

```
data = [1, 7, 12, 11, 44, 23, 14, 47, 51, 64, 67, 77, 140, 672, 871]
```

답)

```
data.sort()
```

```
print(data)
```

```
[1, 7, 11, 12, 14, 23, 44, 47, 51, 64, 67, 77, 140, 672, 871]
```

이진탐색

문제2. (알고리즘 문제1) 이진 탐색 알고리즘을 구현하려면?

보기)

```
data = [1, 7, 11, 12, 44, 23, 14, 47, 51, 64, 67, 77, 140, 672, 871]
```

```
binary_search(data,7)
```

숫자 7이 있습니다.

```
binary_search(data, 8)
```

숫자 8이 없습니다.

답)

```
def binary_search(data,x):
```

```
    data.sort()
```

```
    while True:
```

```
        mid=len(data)//2
```

```
        if x<data[mid]:
```

```
            data=data[:mid]
```

```
        elif x>data[mid]:
```

```
            data=data[mid+1:]
```

```
        elif x==data[mid]:
```

```
            print('숫자' ,x,'이 있습니다.')
```

```
            break
```

```
    if len(data)==1:
```

```
        print('숫자',x, '이 없습니다.')
```

```
        break
```

문제3. (알고리즘 문제2) 순차 탐색 알고리즘을 파이썬으로 구현하시오.

보기)

```
data = [1, 7, 11, 12, 44, 23, 14, 47, 51, 64, 67, 77, 140, 672, 871]
```

답)

```
data = [1, 7, 11, 12, 44, 23, 14, 47, 51, 64, 67, 77, 140, 672, 871]
```

```
def sequential_search(data,x):  
    for i in range(len(data)):  
        if data[i]==x:  
            return '숫자',x,'이 있습니다'  
        elif i==(len(data)-1):  
            return '숫자',x,'이 없습니다'
```

```
sequential_search(data,872)
```

재귀 알고리즘

문제4. (알고리즘 문제3) 재귀 알고리즘 기본(구구단, 별)

재귀 함수란?

1. 재귀 함수는 함수내에서 다시 자신을 호출한 후 그 함수가 끝날때까지 함수 호출 이후의 명령문이 수행되지 않는다.

2. 반복문 + 스택구조가 결합된 함수

먼저 들어간 데이터가 가장 마지막에 나오는 구조

나중에 들어간 데이터가 먼저 나오는 구조 (후입선출)

ex)

```
def hap(a,b):  
    print(a+b)
```

```
def gop(a,b):  
    print(a*b)
```

```
def hap_gop(a,b):  
    hap(a,b)  
    gop(a,b)
```

```
hap_gop(2,3)
```

설명 : hap_gop 함수는 그냥 a와 b값을 받아와서 hap 함수와 gob함수에 던져주는 역할만 수행한다. hap_gop함수가 hap 함수와 gop함수를 호출했다.

그런데 재귀함수는 자기 자신의 함수를 호출한다.

ex2)

```
def countdown(n):
```

```
    if n==0:
```

```
        print('발사')
```

```
    else:
```

```
        print(n)
```

```
        countdown(n-1) #재귀함수는 함수내에서 다시 자신을 호출한 후 함수가 끝날때까  
                        #지 함수 호출 이후의 명령문은 실행하지 않는다.
```

```
countdown(3)
```

```
3
```

```
2
```

```
1
```

```
발사
```

위의 countdown 함수를 변형해서 구구단 2단을 출력하시오.

답)

```
def countdown(n):
```

```
    if n!=0:
```

```
        print('2*%d=%d' %(n,2*n))
```

```
        countdown(n-1)
```

```
countdown(9)
```

```
2*9=18
```

```
2*8=16
```

```
2*7=14
```

```
2*6=12
```

```
2*5=10
```

```
2*4=8
```

```
2*3=6
```

```
2*2=4
```

```
2*1=2
```

위의 결과를 반대로 출력하시오.

답)

```
def countdown(n):
```

```
    if n!=10:
```

```
        print('2*%d=%d' %(n,2*n))
```

```
        countdown(n+1)
```



```
countdown(1)
```

```
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
```

구구단 전체를 재귀로

```
def gugu(n,k):
    if n==10:
        return 0
    elif k!=10:
        print(n,'X', k, '=', n*k)
        gugu(n,k+1)
    elif k==10:
        gugu(n+1,1)
```

```
gugu(2,1)
```

문제4. (알고리즘 문제4) 재귀 알고리즘 팩토리얼

보기)

```
factorial(5)
120
```

답)

```
def factorial(n):
    if n!=1:
        return n*factorial(n-1)
    else:
        return 1
```

```
factorial(5)
```

문제5. ★을 아래와 같이 재귀 함수로 구현하시오

보기)

```
star_print(7)
```

```
★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★
★★★★★★★★
```

답)

```
def star_print(n):
    if n!=0:
        star_print(n-1)
        print ('★'*n)

star_print(7)
```

문제6. ★을 재귀 함수로 구현하시오

보기)

```
star_print2(7)
★★★★★★★
★★★★★★★
★★★★★★
★★★★★
★★★★
★★★
★★
★
```

답)

```
def star_print(n):
    if n!=0:
        print('★'*n)
        return star_print(n-1)
star_print(7)
```

• 최대공약수

문제7. loop문을 이용해서 최대 공약수를 구하시오.

보기)

```
print( find_gcd(20,16) )
4
```

답)

```
def find_gcd(a,b):
    while b > 0:
        remain = a % b
        a = b
        b = remain
    print(a)
```

```
find_gcd(20,16)
```

```
4
```

문제8. 최대 공약수를 loop문을 이용하지 않고 재귀로 구현하시오.

답)

```
def find_gcd2(a,b):  
    if b>0:  
        remain=a%b  
        a=b  
        b = remain  
        find_gcd2(a,b)  
    return  
print(a)
```

```
find_gcd2(108,72)
```

```
36
```

답2)

```
def max_yaksu(a,b):  
    if a%b != 0:  
        return max_yaksu(b, a%b)  
    return b  
print(max_yaksu(20,16))
```

문제9. 최대 공약수 loop문을 이용하지 말고 재귀로 구현하시오.(매개변수3개)

답)

```
def max_yaksu(a,b,c):  
    if a%b != 0:  
        return max_yaksu(b, a%b, c)  
  
    elif b%c != 0:  
        return max_yaksu(a, c, b%c)  
    return c
```

```
print(max_yaksu(96,72,45))
```

버블정렬

문제10. 아래의 리스트를 버블정렬하는 함수를 생성하시오.(loop문)

보기)

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
print(bubble_sort(a) )
```

[1,2,3,4,5,7,8,10]

답)

```
def bubble_sort(a_list):
    for i in range(len(a_list)):
        for j in range(len(a_list)-1):
            if a_list[j]>a_list[j+1]:
                (a_list[j],a_list[j+1])=(a_list[j+1],a_list[j])
    return a_list
a = [5, 4, 3, 2, 1, 8, 7, 10]
print(bubble_sort(a))
```

문제11. 버블정렬을 재귀로 구현하시오.

답)

```
def bubble_sort(a_list,i=0,j=0):
    if j == len(a_list):
        return a_list
    elif i<len(a_list)-1:
        if a_list[i]>a_list[i+1]:
            (a_list[i],a_list[i+1])=(a_list[i+1],a_list[i])
            return bubble_sort(a_list,i+1,j)
        else:
            return bubble_sort(a_list,0,j+1)
```

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
print(bubble_sort(a))
[1, 2, 3, 4, 5, 8, 7, 10]
```

문제12. 삽입정렬 알고리즘 파이썬으로 구현

답)

```
def in_sort(a_list):
    for i in range(len(a_list)):
        val=a_list[i]
        j=i-1
        # print('i=',i,'j=',j,'val=',val)
        # print(a_list)
        while j >= 0 and a_list[j] > val:
            a_list[j+1]=a_list[j]
            j-=1
        # print('i=',i,'j=',j,'val=',val)
```

```
#         print(a_list)
    a_list[j+1]=val
    return a_list
a = [5, 4, 3, 2, 1, 8, 7, 10]
print(in_sort(a))
```

탐욕 알고리즘

"매 순간마다 최선의 선택하는 것 즉, 선택할때마다 가장 좋다고 생각되는 것을 선택해 나가며 최종적인 해답을 구하는 알고리즘 입니다.

이 알고리즘을 설계할 때 유의할 점은 전체를 고려하는게 아니라 문제를 해답을 구하므로 전체적인 최적의 해가 될 수 있는 경우가 존재 한다."

최단 거리 알고리즘 구현하기 위해서 필요한 알고리즘

```
ex)
print(greedy())
액수를 입력 : 14
거스름돈의 종류 입력 : 10 7 1
```

결과 : 10원 1개, 7원 0개, 1원 4개

```
답)
def greedy():
    price=int(input('액수를 입력 : '))#14
    coin=list(map(int, input('거스름돈 종류 입력 : ').split()))
    coin.sort()
    coin.reverse()

    for i in range(len(coin)):
        if price>=coin[i]:
            remain=price//coin[i]
            price=price-coin[i]*remain
            print(coin[i],'원',remain,'개 사용')
        else:
            print(coin[i],'원','0개 사용')

greedy()

액수를 입력 : 14
거스름돈 종류 입력 : 10 7 1
10 원 1 개 사용
7 원 0개 사용
1 원 4 개 사용
```

비밀 지도

문제13.

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.

1. 지도는 한 변의 길이가 n 인 정사각형 배열 형태로, 각 칸은 "공백"(" ") 또는 "벽"("#") 두 종류로 이루어져 있다.
2. 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.
3. "지도 1"과 "지도 2"는 각각 정수 배열로 암호화되어 있다.
4. 암호화된 배열은 지도의 각 가로줄에서 벽 부분을 1, 공백 부분을 0으로 부호화했을 때 얻어지는 이진수에 해당하는 값의 배열이다.

5.

	#			#	$01001_{(2)} = 9$
#		#			$10100_{(2)} = 20$
#	#	#			$11100_{(2)} = 28$
#			#		$10010_{(2)} = 18$
	#		#	#	$01011_{(2)} = 11$

#	#	#	#		$11110_{(2)} = 30$
				#	$00001_{(2)} = 1$
#		#		#	$10101_{(2)} = 21$
#				#	$10001_{(2)} = 17$
#	#	#			$11100_{(2)} = 28$

▼

#	#	#	#	#
#		#		#
#	#	#		#
#			#	#
#	#	#	#	#

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.

매개변수	값
n	5
arr1	[9, 20, 28, 18, 11]
arr2	[30, 1, 21, 17, 28]
출력	["#####", "# # #", "### #", "# ##", "#####"]

매개변수	값
n	6
arr1	[46, 33, 33, 22, 31, 50]
arr2	[27, 56, 19, 14, 14, 10]
출력	["#####", "### #", "## ##", "#### ", "#####", "### #"]

번호를 넣으세요 5

첫번째 배열? 9 20 28 18 11

두번째 배열? 30 1 21 17 28

['#####', '# # #', '### #', '# ##', '#####']

풀기위해 알아야 할 힌트1)

아래의 결과를 분석하시오.

```
print(bin(9))
```

```
: print(bin(9))
```

```
0b1001
```

설명 : 바이너리 형태로 변환

풀기위해 알아야 할 힌트2)

위의 결과를 1001만 나오게 출력.

답)

```
aa=bin(9)
```

```
res=aa.replace('0b','')
```

```
print(res)
```

```
1001
```

풀기위해 알아야 할 힌트3)

위의 결과의 1001을 # # 으로 출력. 1은 #으로 0은 블랭크로 출력.


```

답)
aa=bin(9)
res=aa.replace('0b','')
res2=res.replace('1','#').replace('0',' ')
print(res2)
# #

```

풀기위해 알아야 할 힌트4)

아래의 내용을 구현하시오.

보기)

번호를 넣으세요 5

첫번째 배열?9 20 28 18 11

두번째 배열?30 1 21 17 28

```
['0b11111', '0b10101', '0b11101', '0b10011', '0b11111']
```

```

답)
num=int(input('번호 입력 : '))

arr1=list(map(int, input('첫번째 배열 : ').split()))
arr2=list(map(int, input('두번째 배열 : ').split()))

res=[]
for i in range(num):
    res.append(bin(arr1[i] | arr2[i]))

print(res)

```

```

번호 입력 : 5
첫번째 배열 : 9 20 28 18 11
두번째 배열 : 30 1 21 17 28
['0b11111', '0b10101', '0b11101', '0b10011', '0b11111']

```

문제 답)

```

num=int(input('번호 입력 : '))

arr1=list(map(int, input('첫번째 배열 : ').split()))
arr2=list(map(int, input('두번째 배열 : ').split()))

res=[]

```

```

for i in range(num):
    a=bin(arr1[i] | arr2[i]).replace('0b','').replace('1','#').replace('0',' ')

    res.append(a)

print(res)

```

```

번호 입력 : 5
첫번째 배열 : 9 20 28 18 11
두번째 배열 : 30 1 21 17 28
['#####', '# # #', '### #', '# ##', '#####']

```

다트게임

문제14. 카카오톡 게임별의 하반기 신규 서비스로 다트 게임을 출시하기로 했다. 다트 게임은 다트판에 다트를 세 차례 던져 그 점수의 합계로 실력을 겨루는 게임으로, 모두가 간단히 즐길 수 있다.

갓 입사한 무지는 코딩 실력을 인정받아 게임의 핵심 부분인 점수 계산 로직을 맡게 되었다. 다트 게임의 점수 계산 로직은 아래와 같다.

1. 다트 게임은 총 3번의 기회로 구성된다.
2. 각 기회마다 얻을 수 있는 점수는 0점에서 10점까지이다.
3. 점수와 함께 Single(S), Double(D), Triple(T) 영역이 존재하고 각 영역 당첨 시 점수에서 1제곱, 2제곱, 3제곱 (점수¹, 점수², 점수³)으로 계산된다.
4. 옵션으로 스타상(*), 아차상(#)이 존재하며 스타상(*) 당첨 시 해당 점수와 바로 전에 얻은 점수를 각 2배로 만든다. 아차상(#) 당첨 시 해당 점수는 마이너스된다.
5. 스타상(*)은 첫 번째 기회에서도 나올 수 있다. 이 경우 첫 번째 스타상(*)의 점수만 2배가 된다. (예제 4번 참고)
6. 스타상(*)의 효과는 다른 스타상(*)의 효과와 중첩될 수 있다. 이 경우 중첩된 스타상(*) 점수는 4배가 된다. (예제 4번 참고)
7. 스타상(*)의 효과는 아차상(#)의 효과와 중첩될 수 있다. 이 경우 중첩된 아차상(#)의 점수는 -2배가 된다. (예제 5번 참고)
8. Single(S), Double(D), Triple(T)은 점수마다 하나씩 존재한다.
9. 스타상(*), 아차상(#)은 점수마다 둘 중 하나만 존재할 수 있으며, 존재하지 않을 수도 있다.

0~10의 정수와 문자 S, D, T, *, #로 구성된 문자열이 입력될 시 총점수를 반환하는 함수를 작성하라.

출력 형식

3번의 기회에서 얻은 점수 합계에 해당하는 정수값을 출력한다.

예) 37

입출력 예제

예제	dartResult	answer	설명
1	1S2D*3T	37	$1^1 * 2 + 2^2 * 2 + 3^3$
2	1D2S#10S	9	$1^2 + 2^1 * (-1) + 10^1$
3	1D2S0T	3	$1^2 + 2^1 + 0^3$
4	1S*2T*3S	23	$1^1 * 2 * 2 + 2^3 * 2 + 3^1$
5	1D#2S*3S	5	$1^2 * (-1) * 2 + 2^1 * 2 + 3^1$
6	1T2D0D#	-4	$1^3 + 2^2 + 3^2 * (-1)$
7	1D2S3T*	59	$1^2 + 2^1 * 2 + 3^3 * 2$

수행결과 :

문자열을 입력하세요 : 1S2D*3T

37

문자열을 입력하세요 : 1D2S#10S

9

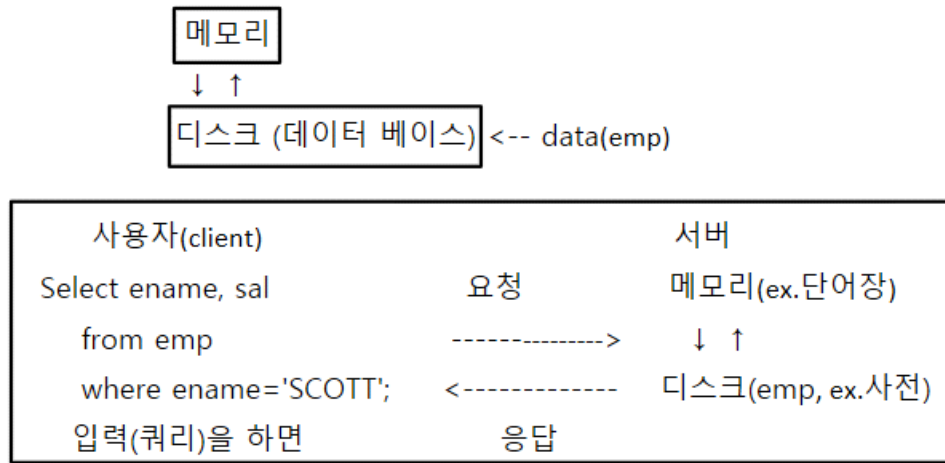
LRU 알고리즘

"데이터 베이스의 메모리 사용을 효율적으로 하기 위해서 사용하는 알고리즘"

디스크에서 **한번 읽은 데이터는 메모리에 올려놓고** 똑같은 데이터를 다음번에 검색할 때,
메모리에서 찾아서 빠르게 검색

- MRU (Most Recently Used) : 가장 최근에 사용한 블록
- LRU (Last Recently Used) : 거의 최근에 사용한 블록

오라클 아키텍처



문제15. 지도개발팀에서 근무하는 제이지는 지도에서 도시 이름을 검색하면 해당 도시와 관련된 맛집 게시물들을 데이터베이스에서 읽어 보여주는 서비스를 개발하고 있다.

이 프로그램의 테스트 업무를 담당하고 있는 어피치는 서비스를 오픈하기 전 각 로직에 대한 성능 측정을 수행하였는데, 제이지가 작성한 부분 중 데이터베이스에서 게시물을 가져오는 부분의 실행시간이 너무 오래 걸린다는 것을 알게 되었다.

어피치는 제이지에게 해당 로직을 개선하라고 다투하기 시작하였고, 제이지는 DB 캐시를 적용하여 성능 개선을 시도하고 있지만 캐시 크기를 얼마로 해야 효율적인지 몰라 난감한 상황이다.

어피치에게 시달리는 제이지를 도와, DB 캐시를 적용할 때 캐시 크기에 따른 실행시간 측정 프로그램을 작성하시오.

입력 형식

- 캐시 크기(cacheSize)와 도시이름 배열(cities)을 입력받는다.
- cacheSize는 정수이며, 범위는 $0 \leq \text{cacheSize} \leq 30$ 이다.
- cities는 도시 이름으로 이뤄진 문자열 배열로, 최대 도시 수는 100,000개이다.
- 각 도시 이름은 공백, 숫자, 특수문자 등이 없는 영문자로 구성되며, 대소문자 구분을 하지 않는다. 도시 이름은 최대 20자로 이루어져 있다.

출력 형식

- 입력된 도시이름 배열을 순서대로 처리할 때, "총 실행시간"을 출력한다.

조건

- 캐시 교체 알고리즘은 LRU(Least Recently Used)를 사용한다.

- cache hit일 경우 실행시간은 1이다. (메모리에서 읽은 경우)
- cache miss일 경우 실행시간은 5이다. (디스크에서 읽은 경우)

입출력 예제

캐시크기(cache Size)	도시이름(cities)	실행시간
3	["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "Jeju", "Pangyo", "Seoul", "NewYork", "LA"]	50
3	["Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul"]	21
2	["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "SanFrancisco", "Seoul", "Rome", "Paris", "Jeju", "NewYork", "Rome"]	60
5	["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "SanFrancisco", "Seoul", "Rome", "Paris", "Jeju", "NewYork", "Rome"]	52
2	["Jeju", "Pangyo", "NewYork", "newyork"]	16
0	["Jeju", "Pangyo", "Seoul", "NewYork", "LA"]	25

풀기위해 알아야 하는 방법1)

아래와 같이 결과를 출력되게 하시오.

보기)

도시이름 입력 : Jeju, Pangyo, Seoul, NewYork, LA, Jeju, Pangyo, Seoul, NewYork, LA

결과)

```
cities=list(map(str,input('도시이름들 입력 : ').split(',')))
```

```
print(cities)
```

```
도시이름들 입력 : Jeju, Pangyo, Seoul, NewYork, LA, Jeju, Pangyo, Seoul, NewYork, LA
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA']
```

풀기위해 알아야 하는 방법2)

Cities에서 받아온 데이터를 cities_l라는 리스트에 하나씩 추가되게 하시오.

답)

```
cities=list(map(str,input('도시이름들 입력 : ').split(',')))
```

```
cities_l=[]
```

```
for i in range(len(cities)):
```

```
    cities_l.append(cities[i])
```

```
print(cities_l)
```

```

도시이름들 입력 : Jeju, Pangyo, Seoul, NewYork, LA, Jeju, Pangyo, Seoul, NewYork, LA
['Jeju']
['Jeju', ' Pangyo']
['Jeju', ' Pangyo', ' Seoul']
['Jeju', ' Pangyo', ' Seoul', ' NewYork']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul', ' NewYo
rk']
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul', ' NewYo
rk', ' LA']

```

풀기위해 알아야 하는방법3)

위의 결과와 같이 cities_1에 데이터가 들어갈때마다 cnt라는 변수에 5가 더해지게 하시오.
답)

```
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]
```

#컴프리헨션!!

```

cities_1=[]
cnt=0

```

```

for i in cities:
    cities_1.append(i)
    cnt+=5
    print(cities_1)
    print(cnt)

```

```

도시이름들 입력 : Jeju, Pangyo, Seoul, NewYork, LA, Jeju, Pangyo, Seoul, NewYork, LA
['Jeju']
5
['Jeju', ' Pangyo']
10
['Jeju', ' Pangyo', ' Seoul']
15
['Jeju', ' Pangyo', ' Seoul', ' NewYork']
20
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA']
25
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju']
30
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo']
35
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul']
40
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul', ' NewYo
rk']
45
['Jeju', ' Pangyo', ' Seoul', ' NewYork', ' LA', ' Jeju', ' Pangyo', ' Seoul', ' NewYo
rk', ' LA']
50

```

풀기위해 알아야 하는 방법4)

For loop문 if문을 추가해서 입력하려는 데이터가 cities_1에 있는지 확인해서 있으면 cnt+1에 더하고 없으면 cnt+5를 더하게 하시오.

그리고 cities_1에는 cities의 데이터는 계속 입력되게 하시오.

답)

```
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]
```

```
cities_l=[]  
cnt=0
```

```
for i in cities:
```

```
    if i in cities_l:
```

```
        cities_l.append(i)
```

```
        cnt+=1
```

```
    else:
```

```
        cities_l.append(i)
```

```
        cnt+=5
```

```
print(cities_l)
```

```
print(cnt)
```

```
도시이름들 입력 : Jeju, Pangyo, Seoul, NewYork, LA, Jeju, Pangyo, Seoul, NewYork, LA
```

```
['Jeju']
```

```
5
```

```
['Jeju', 'Pangyo']
```

```
10
```

```
['Jeju', 'Pangyo', 'Seoul']
```

```
15
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork']
```

```
20
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA']
```

```
25
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju']
```

```
26
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo']
```

```
27
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo', 'Seoul']
```

```
28
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo', 'Seoul', 'NewYork']
```

```
29
```

```
['Jeju', 'Pangyo', 'Seoul', 'NewYork', 'LA', 'Jeju', 'Pangyo', 'Seoul', 'NewYork', 'L
```

```
A']
```

```
30
```

풀기위해 알아야 하는 방법5)

아래의 a라는 리스트의 요소의 개수를 구하고 a라는 리스트의 내용을 다 지우시오.

보기)

```
a=['jeju','pangyo','seoul']
```

답1)

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

```
del a
```

```
print(a)
```

```
-----  
NameError
```

```
<ipython-input-30-edf3f0a84500> in
```

```
4 # for i in a:
```

```
5 #     i.pop()
```

```
----> 6 print(a)
```

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

답2) 컴프리헨션!!

```
a=['jeju','pangyo','seoul']  
print(len(a))
```

```
[a.pop() for i in range(len(a))]
```

```
print(a)
```

```
3  
[]
```

풀기위해 알아야 하는 방법6)

도시이름을 물어보기 전에 캐시 사이즈를 물어보게 하시오.

보기)

캐시사이즈 입력 : 3

도시 이름 입력 : Jeju, ...

답)

```
cacheSize=int(input('캐시크기 입력 : '))
```

```
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]
```

```
cities_l=[]
```

```
cnt=0
```

```
for i in cities:
```

```
    if i in cities_l:
```

```
        cities_l.append(i)
```

```
        cnt+=1
```

```
    else:
```

```
        cities_l.append(i)
```

```
        cnt+=5
```

```
    print(cities_l)
```

```
    print(cnt)
```

풀기위해 알아야 하는 방법7)

캐시 사이즈 만큼 도시의 요소들이 들어오도록 코드를 작성하시오.

답)

```
cacheSize=int(input('캐시크기 입력 : '))
```

```
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]
```

```
cities_l=[]
```

```
cnt=0
```

```
cacheSize=int(input('캐시크기 입력 : '))
```

```
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]
```

```
cities_l=[]
```

```
cnt=0
```



```

for i in cities:
    if i in cities_l:
        cnt+=1
    elif i not in cities_l:
        cnt+=5
    if len(cities_l) >= cacheSize:
        cities_l.pop(0)

    cities_l.append(i)

print(cities_l)
print(cnt)

5
['Jeju', 'Pangyo']
10
['Jeju', 'Pangyo', 'Seoul']
15
['Pangyo', 'Seoul', 'NewYork']
20
['Seoul', 'NewYork', 'LA']
25
['NewYork', 'LA', 'Jeju']
30
['LA', 'Jeju', 'Pangyo']
35
['Jeju', 'Pangyo', 'Seoul']
40
['Pangyo', 'Seoul', 'NewYork']
45
['Seoul', 'NewYork', 'LA']
50

```

최종답)

```

cacheSize=int(input('캐시크기 입력 : '))
cities=[i.strip() for i in input('도시이름들 입력 : ').split(',')]

cities_l=[]
cnt=0

if cacheSize == 0:
    cnt=len(cities)*5
else:
    for i in cities:
        if i in cities_l:
            cnt+=1
        elif i not in cities_l:
            cnt+=5
        if len(cities_l) >= cacheSize:
            cities_l.pop(0)

        cities_l.append(i)
        #print(cities_l)
    print(cnt)

```

문제.16

4. 셔틀버스(난이도: 중)

카카오에서는 무료 셔틀버스를 운행하기 때문에 판교역에서 편하게 사무실로 올 수 있다. 카카오의 직원은 서로를 '크루'라고 부르는데, 아침마다 많은 크루들이 이 셔틀을 이용하여 출근한다.

이 문제에서는 편의를 위해 셔틀은 다음과 같은 규칙으로 운행한다고 가정하자.

- 셔틀은 09:00부터 총 n 회 t 분 간격으로 역에 도착하며, 하나의 셔틀에는 최대 m 명의 승객이 탈 수 있다.
- 셔틀은 도착했을 때 도착한 순간에 대기열에 선 크루까지 포함해서 대기 순서대로 태우고 바로 출발한다. 예를 들어 09:00에 도착한 셔틀은 자리가 있다면 09:00에 줄을 선 크루도 탈 수 있다.

일찍 나와서 셔틀을 기다리는 것이 귀찮았던 콘은, 일주일간의 집요한 관찰 끝에 어떤 크루가 몇 시에 셔틀 대기열에 도착하는지 알아냈다. 콘이 셔틀을 타고 사무실로 갈 수 있는 도착 시각 중 제일 늦은 시각을 구하여라.

단, 콘은 게으르기 때문에 같은 시각에 도착한 크루 중 대기열에서 제일 뒤에 선다. 또한, 모든 크루는 잠을 자야 하므로 23:59에 집에 돌아간다. 따라서 어떤 크루도 다음날 셔틀을 타는 일은 없다.

입력 형식

셔틀 운행 횟수 n , 셔틀 운행 간격 t , 한 셔틀에 탈 수 있는 최대 크루 수 m , 크루가 대기열에 도착하는 시각을 모은 배열 `timetable`이 입력으로 주어진다.

- $0 < n \leq 10$
- $0 < t \leq 60$
- $0 < m \leq 45$
- `timetable`은 최소 길이 1이고 최대 길이 2000인 배열로, 하루 동안 크루가 대기열에 도착하는 시각이 HH:MM 형식으로 이루어져 있다.


```
{'09:00': ['08:00'], '09:10': ['09:09', '09:10', '09:10']}
```

풀기위해 알아야 할 힌트2)

아래와 같이 물어보게 하시오.

보기)

```
실행 횟수 : 2
배차 간격 : 10
최대 탑승 인원수 : 2
08:00, 08:03, 08:04, 08:02
```

답)

```
n=input("실행 횟수 : ")
t=input("배차 간격 : ")
m=input("최대 탑승 인원수 : ")
timetable=[i.strip() for i in input().split(', ')]
```

풀기위해 알아야 할 힌트3)

divmod(540,60)의 결과가 무엇?

답)

```
print(divmod(540,60))
```

(9, 0)

divmod(550,60)의 결과 무엇?

답)

```
print(divmod(550,60))
```

(9, 10)

풀기위해 알아야 할 힌트4)

9를 09로 출력하려면 어떻게 해야하는가?

답)

```
print(str(9).zfill(2))
```

```
print(str(9).zfill(2))
```

09

그러면, 0 을 00으로 출력하려면?

답)

```
print(str(0).zfill(2))
```

풀기위해 알아야 할 힌트5)

9와 0을 09:00으로 출력하려면 어떻게 해야하는가?

답)

```
print(str(9).zfill(2)+'-'+str(0).zfill(2))
```

09:00

풀기위해 알아야 할 힌트6)

540과 550을 받아서 아래와 같이 출력하려면 어떻게 해야하는가?

보기)

```
['09:00', '09:10']
```

답)

```
numstr=[540,550]
```

```
def numstr1(x):
    e_list=[]
    for i in numstr:
        a=divmod(i,60)
        b=str(a[0]).zfill(2)+'-'+str(a[1]).zfill(2)
        e_list.append(b)
    return e_list
```

```
print(numstr1(numstr))
```

풀기위해 알아야 할 힌트7)

아래의 결과를 받아서 결과를 출력하시오.

보기)

```
n=input("운행 횟수 : ")
```

```
t=input("배차 간격(분) : ")
```

```
print(timetable(2,10))
```

결과)

```
['09:00', '09:10']
```

답)

```
n=int(input("운행 횟수 : "))
```

```
t=int(input("배차 간격(분) : "))
```

```
m=int(input("최대 탑승 인원수 : "))
```

```
numstr=[540,550]
```

```
def numstr1(x):
    e_list=[]
    for i in numstr:
        a=divmod(i,60)
        b=str(a[0]).zfill(2)+'-'+str(a[1]).zfill(2)
        e_list.append(b)
    return e_list
```

```
def timetable(n,t):
    bus = 540
    e_list=[]
    for i in range(n):
```

```

        bus+=(i*t)
        return numstr1(bus)

print(timetable(n,t))

```

풀기위해 알아야 할 힌트8)

만든 2개의 함수로 아래의 결과를 출력하시오.

```

결과)
['09:10', '09:09', '08:00', '09:10']

['09:00', '09:10']

```

```

답)
n=int(input("운행 횟수 : "))
t=int(input("배차 간격(분) : "))
m=int(input("최대 탑승 인원수 : "))
timetable=[i.strip() for i in input().split(',')]

from collections import defaultdict

numstr=[540,550]

def numstr1(x):
    e_list=[]
    for i in numstr:
        a=divmod(i,60)
        b=str(a[0]).zfill(2)+'-'+str(a[1]).zfill(2)
        e_list.append(b)
    return e_list

def timetable_f(n,t):
    bus = 540
    e_list=[]
    for i in range(n):
        bus+=(i*t)
        return numstr1(bus)

print(timetable_f(n,t))

dic=defaultdict(list)

```

풀기위해 알아야 할 힌트9)

아래 두개의 리스트를 가지고 아래의 딕셔너리를 생성하시오.

```

보기)
[09:10,09:09,08:00,09:10]

['09:00', '09:10']

```

```

답)
crew = ['09:10', '09:09', '08:00', '09:10']
bus = ['09:00', '09:10']

```

```

dic= {}

for i in bus:
    aaa = []
    for k in crew: # a = ['09:10', '09:09', '08:00', '09:10']
        if k <= i: # b = ['09:00', '09:10']
            aaa.append(k)
            dic[i] = aaa
            crew.pop(crew.index(k))
    print(dic)
{'09:00': ['08:00']}
{'09:00': ['08:00'], '09:10': ['09:10', '09:10']}

```

-----적용

```

n=int(input("운행 횟수 : "))
t=int(input("배차 간격(분) : "))
m=int(input("최대 탑승 인원수 : "))
timetable=[i.strip() for i in input().split(',')]

```

```

numstr=[540,550]

```

```

def numstr1(x):
    e_list=[]
    for i in numstr:
        a=divmod(i,60)
        b=str(a[0]).zfill(2)+'-'+str(a[1]).zfill(2)
        e_list.append(b)
    return e_list

```

```

def timetable_f(n,t):
    bus = 540
    e_list=[]
    for i in range(n):
        bus+=(i*t)
    return numstr1(bus)

```

```

print(timetable)
print(timetable_f(n,t))

```

```

dic={}
for tf in timetable_f(n,t):
    aaa=[]
    for t in timetable:
        if t <= tf:
            aaa.append(t)
            dic[tf]=aaa
            timetable.pop(timetable.index(t))
    print(dic)

```

```

운행 횟수 : 2
배차 간격(분) : 10
최대 탑승 인원수 : 2
09:10, 09:09, 08:00, 09:10
['09:10', '09:09', '08:00', '09:10']
['09:00', '09:10']
{'09:00': ['08:00']}
{'09:00': ['08:00'], '09:10': ['09:10', '09:09', '09:10']}

```

풀기위해 알아야 할 힌트10)

위의 결과에서 딕셔너리의 value의 개수가 2개 이상인 것의 key와 value를 출력하시오.

답)

```

n=int(input("운행 횟수 : "))
t=int(input("배차 간격(분) : "))
m=int(input("최대 탑승 인원수 : "))
timetable=[i.strip() for i in input().split(',')]

numstr=[540,550]

def numstr1(x):
    e_list=[]
    for i in numstr:
        a=divmod(i,60)
        b=str(a[0]).zfill(2)+'-'+str(a[1]).zfill(2)
        e_list.append(b)
    return e_list

def timetable_f(n,t):
    bus = 540
    e_list=[]
    for i in range(n):
        bus+=(i*t)
        return numstr1(bus)

print(timetable)
print(timetable_f(n,t))

dic={}
for tf in timetable_f(n,t):
    aaa=[]
    for t in timetable.copy():
        if t <= tf:
            aaa.append(t)
            dic[tf]=aaa
            timetable.pop(timetable.index(t))
print(dic)
for i in dic:
    if len(dic.values())>=2:
        print(i,dic[i])

```



```

운행 횟수 : 2
배차 간격(분) : 10
최대 탑승 인원수 : 2
09:10, 09:09, 08:00, 09:10
['09:10', '09:09', '08:00', '09:10']
['09:00', '09:10']
{'09:00': ['08:00'], '09:10': ['09:10', '09:09', '09:10']}
09:00 ['08:00']
09:10 ['09:10', '09:09', '09:10']

```

풀기위해 알아야 할 힌트11)

9시 30분 버스에 2명만 승차할 수 있고 아래의 버스에는 3명이 있다. 이 세명중에 최소시간을 출력하시오.

보기)

09:10 ['09:10', '09:09', '09:10']

시간관계상 광흠이 코드보고 이해하도록

```
import re
```

```
n = int(input()) # 버스 횟수
```

```
t = int(input()) # 버스 간격
```

```
m = int(input()) # 탈 수 있는 최대 수
```

```
timetable = input().split(',') # 각 크루의 대기열 도착 시간
```

```
for i in range(len(timetable)):
```

```
    timetable[i] = int(re.sub('[-.#/?:$“” }]', "", timetable[i])) #특수문자 제거
```

```
timetable.sort()
```

```
time=900 #첫차 시간
```

```
lst = 0 #마지막 탑승인원의 탑승 시간
```

```
ls = n*m # 최대 탈수있는 인원
```

```
cnt=0
```

```
for i in range(n): #0,1
```

```
    for j in range(m): #0,1
```

```
        if len(timetable) > j: #인덱스 범위를 초과하지 않기 위함
```

```
            if (timetable[j] <= time): #time변수보다 같거나 작을때만 뺌
```

```
                cnt+=1
```

```
                lst = timetable[j]
```

```
        if cnt > 0:
```

```
            timetable=timetable[cnt:] # timetable에서 탑승 한 사원 삭제
```

```
            time+=t
```

```
rst = " # 최종 결과 답을 변수
```

```
if m-cnt > 0: ## 탑승인원보다 자리에 여유가 있으므로 막차시간에 타준다.
```

```

h = ((n-1)*t)//60 # 시간으로 나눔
m = ((n-1)*t)%60 # 분으로 나눔
rst = str(int(900+(h*100)+m)) # 시간, 분을 더함

else :
    lst = str(lst)
    if lst[-2:] == '00': # 00분 일경우 -1 을 빼면 99분이 되므로 100 -41 (59) 해줌
        rst = str(int(lst)-41)
    else :
        rst=str(int(lst)-1)

if (len(rst)<4): # 4자리 수가 아니면 앞에 부족한 만큼 0을 더함
    rst='0'*(4-len(rst))+rst
rst = rst[0:-2]+'.'+rst[-2:]
print(rst)

```

문제17. 단층 퍼셉트론 and게이트의 가중치를 계산하는 프로그램 구현

보기)

처음에 줄 가중치

```
w = np.array([[0.3], [0.4], [0.1]])
```

```
import numpy as np
```

```
inputs = np.array([[0,0],[0,1],[1,0],[1,1]])
```

```
targets = np.array([[0],[0],[0],[1]])
```

```
import pcn_kws
```

```
p = pcn_kws.pcn(inputs,targets)
```

```
p.pcntrain(inputs,targets,0.05,18)
```

결과)

```
w0 = 0.4
```

```
w1 = 0.3
```

```
w2 = 0.1
```

답)

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from matplotlib import font_manager, rc
```

```
x = np.array([[-1,0,0],[-1,0,1],[-1,1,0],[-1,1,1]])
```

```
y = np.array([[0],[0],[0],[1]])
```

```
def step_function(a):
```

```
    return np.array( a>= 0, dtype=np.int)
```

```
def pcntrain(x,y,r,repeat):
```

```
    w = np.array([[0.3], [0.4], [0.1]])
```

```
    cnt=0
```

```
    while True:
```

```

for j in range(4): #입력
    k = w[0]*x[j][0] + w[1]*x[j][1] + w[2]*x[j][2]
    if y[j]-step_function(k) != 0:
        for h in range(3):
            w[h]=w[h]+r*x[j][h]*(y[j]-step_function(k))
        cnt+=1
    if cnt==repeat:
        break
return w

print(pcctrain(x,y,0.05,18))

[[0.4]
 [0.3]
 [0.1]]

```

자카드 유사도

문제18. 뉴스 클러스터링

여러 언론사에서 쏟아지는 뉴스, 특히 속보성 뉴스를 보면 비슷비슷한 제목의 기사가 많아 정작 필요한 기사를 찾기가 어렵다. Daum 뉴스의 개발 업무를 맡게 된 신입사원 튜브는 사용자들이 편리하게 다양한 뉴스를 찾아볼 수 있도록 문제점을 개선하는 업무를 맡게 되었다.

개발의 방향을 잡기 위해 튜브는 우선 최근 화제가 되고 있는 “카카오 신입 개발자 공채” 관련 기사를 검색해보았다.

- 카카오 첫 공채.. ‘블라인드’ 방식 채용
- 카카오, 합병 후 첫 공채.. 블라인드 전형으로 개발자 채용
- 카카오, 블라인드 전형으로 신입 개발자 공채
- 카카오 공채, 신입 개발자 코딩 능력만 본다
- 카카오, 신입 공채.. “코딩 실력만 본다”
- 카카오 “코딩 능력만으로 2018 신입 개발자 뽑는다”

기사의 제목을 기준으로 “블라인드 전형”에 주목하는 기사와 “코딩 테스트”에 주목하는 기사로 나누는 걸 발견했다. 튜브는 이들을 각각 묶어서 보여주면 카카오 공채 관련 기사를 찾아보는 사용자에게 유용할 듯싶었다.

유사한 기사를 묶는 기준을 정하기 위해서 논문과 자료를 조사하던 튜브는 “자카드 유사도”라는 방법을 찾아냈다.

자카드 유사도는 집합 간의 유사도를 검사하는 여러 방법 중의 하나로 알려져 있다. 두 집합 A, B 사이의 자카드 유사도 $J(A, B)$ 는 두 집합의 교집합 크기를 두 집합의 합집합 크기로 나눈

값으로 정의된다.

예를 들어 집합 $A = \{1, 2, 3\}$, 집합 $B = \{2, 3, 4\}$ 라고 할 때, 교집합 $A \cap B = \{2, 3\}$, 합집합 $A \cup B = \{1, 2, 3, 4\}$ 이 되므로, 집합 A, B 사이의 자카드 유사도 $J(A, B) = 2/4 = 0.5$ 가 된다. 집합 A와 집합 B가 모두 공집합일 경우에는 나눗셈이 정의되지 않으니 따로 $J(A, B) = 1$ 로 정의한다.

자카드 유사도는 원소의 중복을 허용하는 다중집합에 대해서 확장할 수 있다. 다중집합 A는 원소 "1"을 3개 가지고 있고, 다중집합 B는 원소 "1"을 5개 가지고 있다고 하자. 이 다중집합의 교집합 $A \cap B$ 는 원소 "1"을 $\min(3, 5)$ 인 3개, 합집합 $A \cup B$ 는 원소 "1"을 $\max(3, 5)$ 인 5개 가지게 된다. 다중집합 $A = \{1, 1, 2, 2, 3\}$, 다중집합 $B = \{1, 2, 2, 4, 5\}$ 라고 하면, 교집합 $A \cap B = \{1, 2, 2\}$, 합집합 $A \cup B = \{1, 1, 2, 2, 3, 4, 5\}$ 가 되므로, 자카드 유사도 $J(A, B) = 3/7$, 약 0.42가 된다.

이를 이용하여 문자열 사이의 유사도를 계산하는데 이용할 수 있다. 문자열 "FRANCE"와 "FRENCH"가 주어졌을 때, 이를 두 글자씩 끊어서 다중집합을 만들 수 있다. 각각 {FR, RA, AN, NC, CE}, {FR, RE, EN, NC, CH}가 되며, 교집합은 {FR, NC}, 합집합은 {FR, RA, AN, NC, CE, RE, EN, CH}가 되므로, 두 문자열 사이의 자카드 유사도 $J(\text{"FRANCE"}, \text{"FRENCH"}) = 2/8 = 0.25$ 가 된다.

입력으로 들어온 두 문자열의 자카드 유사도를 출력한다. 유사도 값은 0에서 1 사이의 실수이므로, 이를 다루기 쉽도록 65536을 곱한 후에 소수점 아래를 버리고 정수부만 출력한다.

예제 입출력

str1	str2	answer
FRANCE	french	16384
handshake	shake hands	65536
aa1+aa2	AAA12	43690
E=M*C^2	e=m*c^2	65536

풀기위해 알아야한 힌트)

아래의 str1과 str2의 철자를 가지고 아래와 같이 결과를 출력하시오.

보기)

```
str1= 'FRANCE'.lower()
str2='french'.lower()
['fr','ra','an','nc','ce']
['fr','re','en','nc','eh']
```

답)

```

str1= 'FRANCE'.lower()
str2='french'.lower()

str1_list=[]
str2_list=[]
for i in range(1,len(str1)):
    str1_list.append(str1[i-1]+str1[i])
for i in range(1,len(str2)):
    str2_list.append(str2[i-1]+str2[i])
print(str1_list)
print(str2_list)

['fr', 'ra', 'an', 'nc', 'ce']
['fr', 're', 'en', 'nc', 'ch']

```

답)

import re

```

str1=str(input().lower())
str2=str(input().lower())

```

```

str1_list=[]
str2_list=[]

```

```

for i in range(1,len(str1)):
    str1_list.append(str1[i-1]+str1[i])
for i in range(1,len(str2)):
    str2_list.append(str2[i-1]+str2[i])

```

```

print(str1_list)
print(str2_list)

```

#특수문자와 숫자와 공백 제거

```

temp1=[]
temp2=[]
for i in range(len(str1_list)):
    if str1_list[i].isalpha():
        temp1.append(str1_list[i])

```

```

for i in range(len(str2_list)):
    if str2_list[i].isalpha():
        temp2.append(str2_list[i])

```

```

print(temp1)
print(temp2)
A=[]

```

#교집합 카운트

```

inter_cnt = 0

```

```

for i in range(len(temp1)):
    if temp1[i] in temp2:
        inter_cnt += 1

```

```

print(inter_cnt)

```

```

if len(temp1)==0 and len(temp2)==0:

```

```

print(1*65536)
else:
    #합집합 개수
    B=len(temp1)-(inter_cnt) + len(temp2)
    print(int(inter_cnt/B*65536))

E=M*C^2
e=m*c^2
['e=', 'm', 'm*', '*c', 'c^', '^2']
['e=', 'm', 'm*', '*c', 'c^', '^2']
[]
[]
0
65536

```

문제19. 프렌즈 4블럭

블라인드 공채를 통과한 신입 사원 라이언은 신규 게임 개발 업무를 맡게 되었다. 이번에 출시할 게임 제목은 "프렌즈4블록".

같은 모양의 카카오프렌즈 블록이 2×2 형태로 4개가 붙어있을 경우 사라지면서 점수를 얻는 게임이다.



만약 판이 위와 같이 주어질 경우, 라이언이 2×2로 배치된 7개 블록과 콘이 2×2로 배치된 4개 블록이 지워진다. 같은 블록은 여러 2×2에 포함될 수 있으며, 지워지는 조건에 만족하는 2×2 모양이 여러 개 있다면 한꺼번에 지워진다.



블록이 지워진 후에 위에 있는 블록이 아래로 떨어져 빈 공간을 채우게 된다.



만약 빈 공간을 채운 후에 다시 2×2 형태로 같은 모양의 블록이 모이면 다시 지워지고 떨어지고를 반복하게 된다.



위 초기 배치를 문자로 표시하면 아래와 같다.

```
TTTANT
RRFACC
RRRFCC
TRRRAA
TTMMMF
TMMTTJ
```

각 문자는 라이언(R), 무지(M), 어피치(A), 프로도(F), 네오(N), 튜브(T), 제이지(J), 콘(C)을 의미한다

입력으로 블록의 첫 배치가 주어졌을 때, 지워지는 블록은 모두 몇 개인지 판단하는 프로그램을 제작

입력 형식

- 입력으로 판의 높이 m , 폭 n 과 판의 배치 정보 `board`가 들어온다.
- $2 \leq n, m \leq 30$
- `board`는 길이 n 인 문자열 m 개의 배열로 주어진다. 블록을 나타내는 문자는 대문자 A에서 Z가 사용된다.

출력 형식

입력으로 주어진 판 정보를 가지고 몇 개의 블록이 지워질지 출력하라.

입출력 예제

m	n	board	answer
4	5	["CCBDE", "AAADE", "AAABF", "CCBBF"]	14
6	6	["TTTANT", "RRFACC", "RRRFCC", "TRRRAA", "TTMMMF", "TMMTTJ"]	15

답)

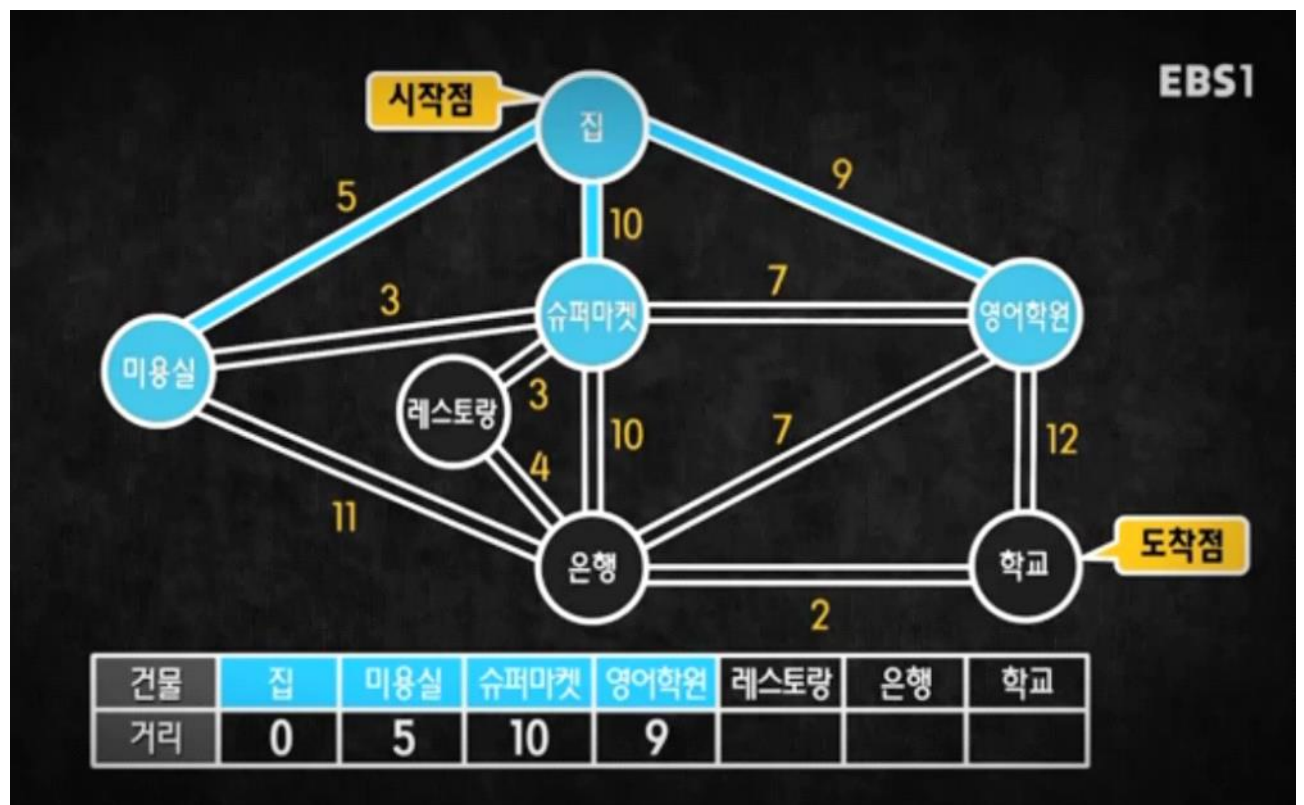
다이나믹 프로그래밍(최단거리)

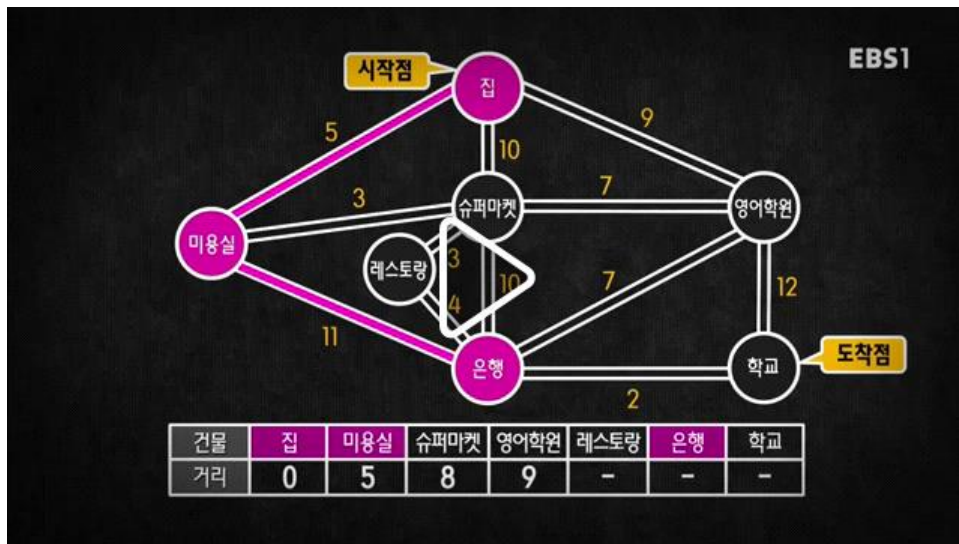
다이나믹 프로그래밍 이란?

: 큰 문제는 작은 문제들이 여러 개 모여 있는 것으로 작은 문제를 하나씩 해결하면 된다.

문제20. 최단 거리 구하기

■ 블라인드 코딩 시험 대비 : 최단거리 알고리즘을 파이썬으로 구현하시오 !





코드 구현 결과 :

집, 미용실, 슈퍼마켓, 영어학원, 레스토랑, 은행, 학교 중 출발지 선택 >? 집

집 에서 집까지의 최단거리는 0 입니다.
 집 에서 미용실까지의 최단거리는 5 입니다.
 집 에서 슈퍼마켓까지의 최단거리는 8 입니다.
 집 에서 영어학원까지의 최단거리는 9 입니다.
 집 에서 레스토랑까지의 최단거리는 11 입니다.
 집 에서 은행까지의 최단거리는 15 입니다.
 집 에서 학교까지의 최단거리는 17 입니다.

1. 집->미 : 5, 집->슈: 10, 집->영 : 9 = 가장 짧은 곳이 미용실이다. **집->미** 를 dequeue 한다
2. 집->미->슈:8, 집->미->은:16 = 1번에서 메모리에서 안뺀 집->슈: 10, 집->영 : 9을 가져온다.
집->미->슈:8, 집->미->은:16, **집->슈: 10**, 집->영
 집->미->슈:8가 집->슈:10 보다 빠르므로 **집->미->슈:8**를 dequeue 시킨다.
3. 집->미->슈->레:11, **집->미->슈->은:18**, 집->미->슈->영:15
 집->미->은:16, **집->영:9** 을보면 같은 원리로 **집->미->슈->은:18**, **집->영:9** 을 dequeue시킨다.
4. **집->영->은 :16**, 집->영->학:21, 집->미->슈->레:11, 집->미->은:16 에서
집->영->은 :16, **집->미->슈->레:11** 을 dequeue 시킨다.
5. 집->미->슈->레->은: 15, 집->영->학:21, 집->미->은:16 에서
집->미->슈->레->은: 15을 dequeue 시킨다.
6. 집->미->슈->레->은->학:17, 집->영->학:21 에서
집->미->슈->레->은->학:17 을 dequeue시킨다.

답)

```
import pandas as pd
idx = ['집','미용실','슈퍼마켓','영어학원','레스토랑','은행','학교']
start= input('집, 미용실, 슈퍼마켓, 영어학원, 레스토랑, 은행, 학교 중 출발지 선택 ')
a=idx.index(start)
short=[[0,5,10,9,999,999,999],
        [5,0,3,999,999,11,999],
        [10,3,0,7,3,10,999],
        [9,999,7,0,999,7,12],
```

```

[999,999,3,999,0,4,999],
[999,11,10,7,4,0,2],
[999,999,999,12,999,2,0]]
df=pd.DataFrame(short)
print(df)
for i in df:
    loc=df[a][i]
    for j in df:
        if df[a][j]>=loc+df[i][j]:
            df[a][j]=loc+df[i][j]
print("start point:",start)
for i in df:
    print(start,'>>',idx[i],':',df[a][i])
집, 미용실, 슈퍼마켓, 영어학원, 레스토랑, 은행, 학교 중 출발지 선택 집
    0    1    2    3    4    5    6
0    0    5   10    9  999  999  999
1    5    0    3  999  999   11  999
2   10    3    0    7    3   10  999
3    9  999    7    0  999    7   12
4  999  999    3  999    0    4  999
5  999   11   10    7    4    0    2
6  999  999  999   12  999    2    0
start point: 집
집 >> 집 :: 0
집 >> 미용실 :: 5
집 >> 슈퍼마켓 :: 8
집 >> 영어학원 :: 9
집 >> 레스토랑 :: 11
집 >> 은행 :: 15
집 >> 학교 :: 17

```


것을 tkinter 에게 알려준다.

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼

외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)

```
canvas.configure(background='black') # 캔버스 색깔 변경하는 방법
```

```
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추  
라고 캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려  
주는것이다.
```

문제347. 창이 바로 꺼지는것을 막으려면?

mainloop()라고 불리는 애니메이션 루프를 추가해야한다.

답)

```
from tkinter import *
```

```
import random
```

```
import time
```

```
tk = Tk() # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Ping Pong Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여  
한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이  
다.
```

```
tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할  
것을 tkinter 에게 알려준다.
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼

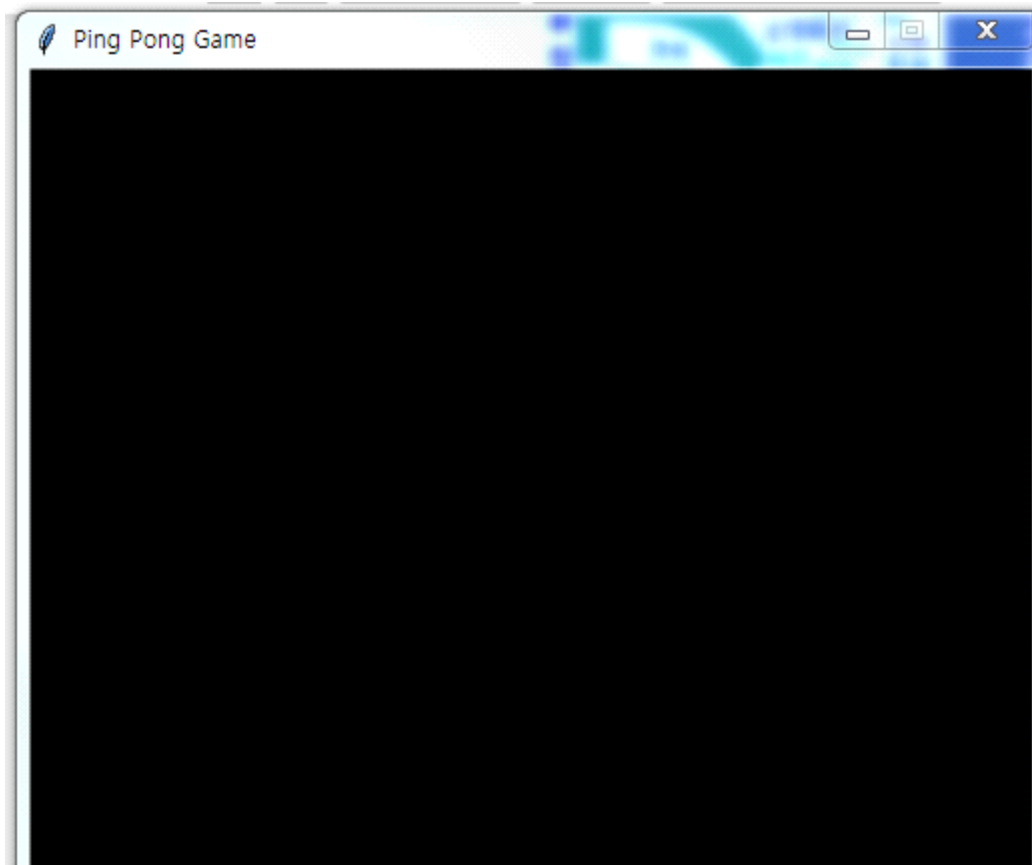
외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)

```
canvas.configure(background='black') # 캔버스 색깔 변경하는 방법
```

```
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추  
라고 캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려  
주는것이다.
```

```
tk.mainloop()
```



문제348. 캔버스의 색깔을 하얀색으로 바꾸시오.

답)

```
from tkinter import *  
import random  
import time
```

```
tk = Tk()    # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Ping Pong Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여  
한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이  
다.
```

```
tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할  
것을 tkinter 에게 알려준다.
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

```
# bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
```

```
# 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
```

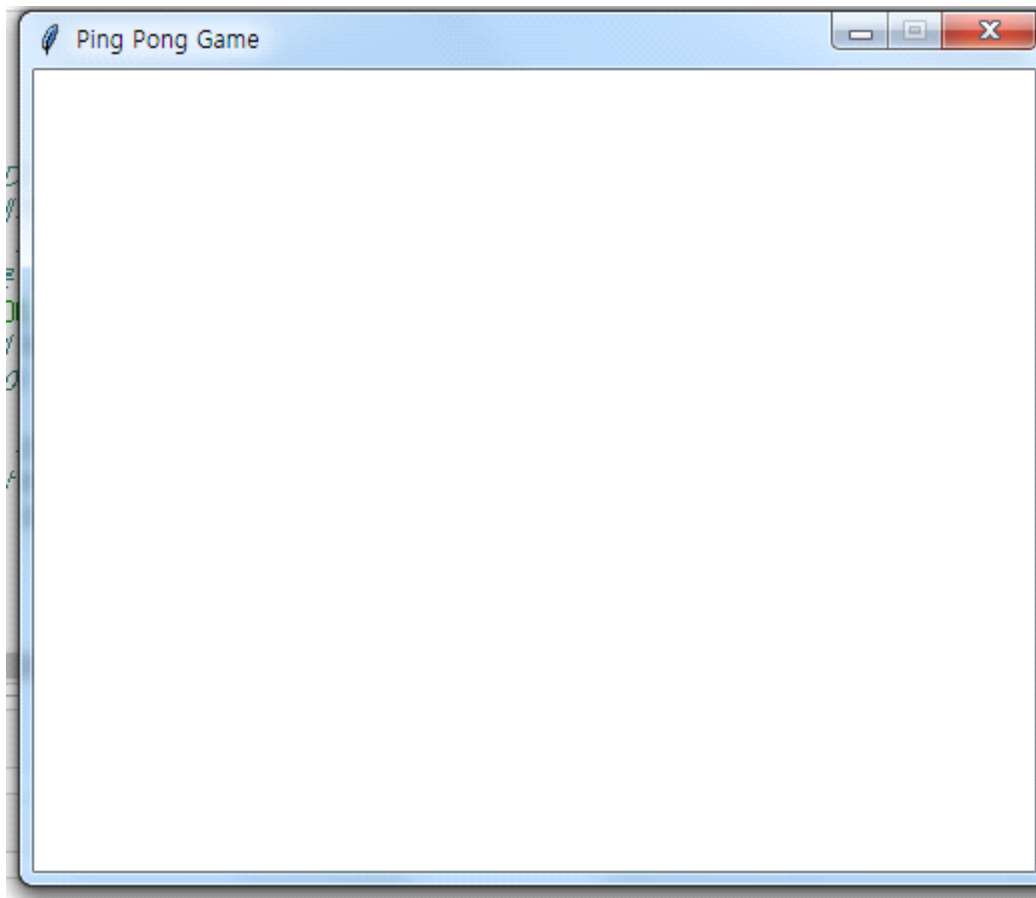
```
canvas.configure(background='white') # 캔버스 색깔 변경하는 방법
```

```
canvas.pack()    # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추  
라고 캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려
```

주는것이다.

tk.mainloop()



문제349. 볼을 캔버스안에 담으시오.

답)

```
from tkinter import *
```

```
import random
```

```
import time
```

```
class Ball:
```

```
    def __init__(self, canvas, color):
```

```
        self.canvas = canvas
```

```
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval :  
object 형태 타입) #실제로는 □인데 ○모양으로 나오는것, 10,10,25,25는 공의 크기
```

```
                #      ↑  ↑  ↑  ↑
```

```
        #좌측상단 구석에서 시작하는 x y   우측 상단 구석에서 시작하는 x y
```

```
        self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
```

```
    def draw(self):
```

```
        pass
```

```

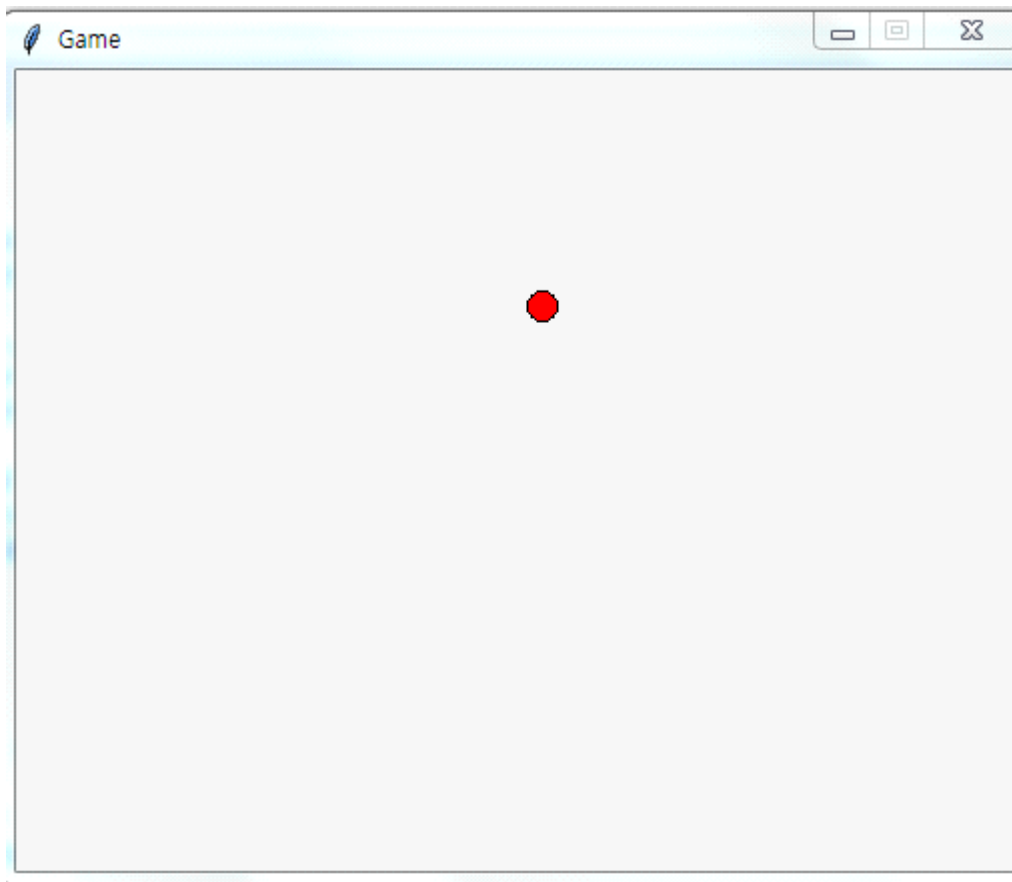
tk = Tk()    # 1. tk 를 인스턴스화 한다.
tk.title("Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할
    것을 tkinter 에게 알려준다.
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
    # bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
    # 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)

canvas.pack()    # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추
    라고 캔버스에에 말해준다.
tk.update()    # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려
    주는것이다.

ball = Ball(canvas,'red')

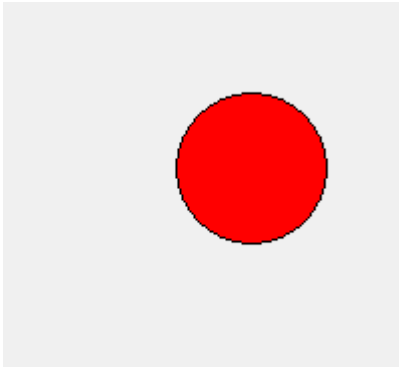
while 1:
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화
    면을 그리고
    tk.update()
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !

```

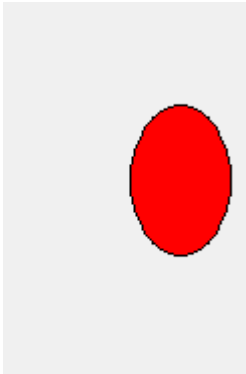


설명 :

100, 100, 25, 25



100, 100, 50, 25



문제350. 캔버스를 검정색으로하고 공의 색깔을 하얀색으로 변경하시오.

답)

```
class Ball:
```

```
    def __init__(self, canvas, color):
```

```
        self.canvas = canvas
```

```
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval :  
object 형태 타입) #실제로는 □인데 ○모양으로 나오는것, 10,10,25,25는 공의 크기  
#      ↑  ↑  ↑  ↑  ↑
```

```
#좌측상단 구석에서 시작하는 x y 우측 상단 구석에서 시작하는 x y
```

```
        self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
```

```
    def draw(self):
```

```
        pass
```

```
tk = Tk()    # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이  
다.
```

```
tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할  
것을 tkinter 에게 알려준다.
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

```
# bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼  
# 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
```

```
canvas.configure(background='black')
```

```
canvas.pack()    # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추  
라고 캔버스에에 말해준다.
```

```
tk.update()  # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려  
주는것이다.
```

```
ball = Ball(canvas,'white')
```

```
while 1:
```

```
    tk.update_idletasks()  # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화  
    면을 그리고
```

```
    tk.update()
```

```
    time.sleep(0.01)  # 100분의 1초마다 잠들어라 !
```

문제351. 캔버스의 공의 시작위치가 천정위가 되게하시오.

```
관련 코드 : self.canvas.move(self.id,245,100)
```

공, x축, y축

답)

```
class Ball:
```

```
    def __init__(self, canvas, color):
```

```
        self.canvas = canvas
```

```
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval :  
object 형태 타입) #실제로는 □인데 ○모양으로 나오는것, 10,10,25,25는 공의 크기
```

```
                #      ↑   ↑   ↑   ↑
```

```
        #좌측상단 구석에서 시작하는 x y   우측 상단 구석에서 시작하는 x y
```

```
        self.canvas.move(self.id, 245, 0)
```

```
    def draw(self):
```

```
        pass
```

```
tk = Tk()    # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이
```

다.

tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할 것을 tkinter 에게 알려준다.

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

```
# bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
```

```
# 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
```

```
canvas.configure(background='black')
```

```
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추  
라고 캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려  
주는것이다.
```

```
ball = Ball(canvas,'white')
```

```
while 1:
```

```
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화  
    면을 그리고
```

```
    tk.update()
```

```
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !
```

문제352. 공을 중앙으로 다시 원위치시키고 패들이 키보드의 방향키로 움직이도록 하시오.

관련코드 : self.canvas.move(self.id, self.x, self.y)

공, 공의x방향, 공의 y방향

힌트)

패들 클래스에 init 함수에서 아래의 코드를 작성

방향키와 공을 움직이는 함수를 매칭 시켜준다.

```
def turn_left(self, evt):
```

```
    self.x = -9
```

```
def turn_right(self, evt):
```

```
    self.y = 9
```

init함수에 추가

```
self.canvas.bind_all('<KeyPress-Left>',self.trun_left) #방향키 왼쪽을 누르면 turn_left  
작동
```

답)

일단 완성코드를 실행해보시오.

```

from tkinter import *
import random
import time

# ball = Ball(canvas, paddle, 'white')
class Ball:
    def __init__(self, canvas, paddle, color):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
        #서 남 동 북
        canvas.configure(background='black') # 캔버스의 색깔을 검정색으로 하겠다.

        self.canvas.move(self.id, 245, 100)

        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

        self.hit_bottom = False # 바닥에 닿으면 게임 끝나는 코드를
        # 구현하기 위해서 쓰는 변수

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)
        if pos[1] <= 0:
            self.y = 3
        if pos[3] >= self.canvas_height:
            self.hit_bottom = True
        if pos[0] <= 0:
            self.x = 3
        if pos[2] >= self.canvas_width:
            self.x = -3
        if self.hit_paddle(pos) == True:
            self.y = -3

```

```

def hit_paddle(self,pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]:
        if pos[3] >= paddle_pos[1] and pos[1] <= paddle_pos[3]:
            return True
    return False

class Paddle:
    def __init__(self,canvas,color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0,0,100,10,fill=color)
        self.canvas.move(self.id, 200, 400)
        self.x = 0
        self.canvas_width = self.canvas.winfo_width()
        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)

    def draw(self):
        self.canvas.move(self.id, self.x, 0)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:
            self.x = 0

    def turn_left(self,evt):
        self.x = -9

    def turn_right(self,evt):
        self.x = 9

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=600, height=500, bd=0, highlightthickness=0)
canvas.pack()
tk.update()
paddle = Paddle(canvas,'white')
ball = Ball(canvas, paddle, 'white')

```

```

while 1:
    ball.draw()
    paddle.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.02)

```

문제353. 패들이 위아래로도 움직여지게 하시오.

```

답)
from tkinter import *
import random
import time

# ball = Ball(canvas, paddle, 'white')
class Ball:
    def __init__(self, canvas, paddle, color):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
        #서 남 동 북
        canvas.configure(background='black') # 캔버스의 색깔을 검정색으로 하겠다.

        self.canvas.move(self.id, 245, 100)

        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

        self.hit_bottom = False # 바닥에 닿으면 게임 끝나는 코드를
        # 구현하기 위해서 쓰는 변수

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)
        if pos[1] <= 0:

```

```

        self.y = 3
    if pos[3] >= self.canvas_height:
        self.hit_bottom = True
    if pos[0] <= 0:
        self.x = 3
    if pos[2] >= self.canvas_width:
        self.x = -3
    if self.hit_paddle(pos) == True:
        self.y = -3

def hit_paddle(self,pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]:
        if pos[3] >= paddle_pos[1] and pos[1] <= paddle_pos[3]:
            return True
    return False

class Paddle:
    def __init__(self,canvas,color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0,0,100,10,fill=color)
        self.canvas.move(self.id, 200, 400)
        self.x = 0
        self.y = 0
        self.canvas_width = self.canvas.winfo_width()
        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)
        self.canvas.bind_all('<KeyPress-Up>',self.turn_up)
        self.canvas.bind_all('<KeyPress-Down>',self.turn_down)

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:
            self.x = 0

    def turn_left(self,evt):
        self.x = -9

```

```

def turn_right(self,evt):
    self.x = 9

def turn_up(self,evt):
    self.y = -9

def turn_down(self,evt):
    self.y = 9

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=600, height=500, bd=0, highlightthickness=0)
canvas.pack()
tk.update()
paddle = Paddle(canvas,'white')
ball = Ball(canvas, paddle, 'white')

while 1:
    ball.draw()
    paddle.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.02)

```


핑퐁 강화학습

2018년 8월 8일 수요일 오후 2:20

문제1. 캔버스를 생성하시오.

답)

```
from tkinter import *  
import random  
import time
```

tk = Tk() # 1. tk 를 인스턴스화 한다.

tk.title("Ping Pong") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.

tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.

tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것을
tkinter 에게 알려준다.

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼

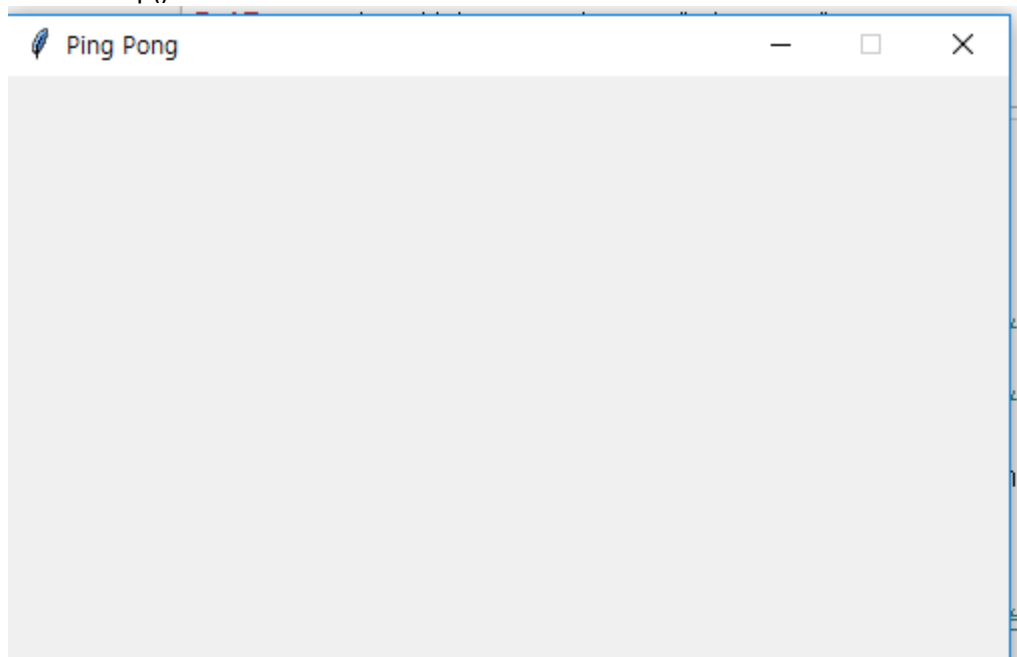
외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)

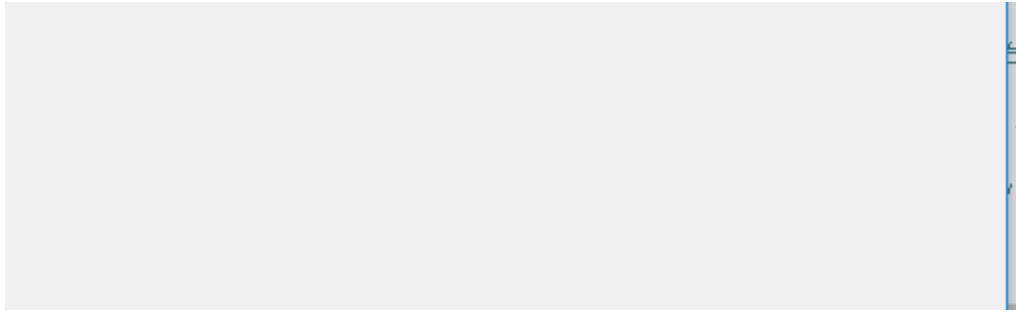
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라
고 캔버스에에 말해준다.

tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주
는것이다.

#창이 바로 꺼지는것을 막으려면 ? mainloop() 라 불리는 애니메이션 루프를 추가해야
한다.

```
tk.mainloop()
```





문제 2) 캔버스의 색을 노란색으로 변경하시오.

답)

```
from tkinter import *  
import random  
import time
```

```
tk = Tk() # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Ping Pong") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
```

```
tk.wm_attributes("-topmost", 1) # 4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것을  
tkinter 에게 알려준다.
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

```
# bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
```

```
# 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
```

```
canvas.configure(background='yellow') # 캔버스 색깔 변경하는 방법
```

```
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라  
고 캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주  
는것이다.
```

```
#창이 바로 꺼지는것을 막으려면 ? mainloop() 라 불리는 애니메이션 루프를 추가해야  
한다.
```

```
tk.mainloop()
```



문제3. 캔버스 위에 ball클래스 생성.

답)

```
from tkinter import *
import random
import time
```

```
class Ball:
```

```
    def __init__(self, canvas, color):
        self.canvas = canvas
```

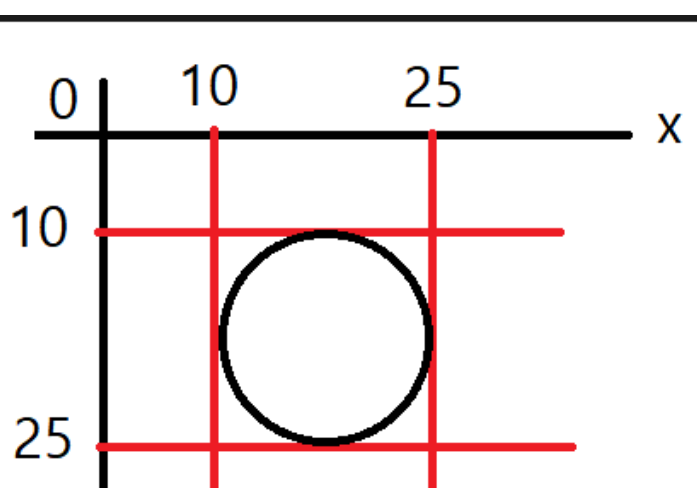
```
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 크기 및 색깔(oval : object 형태
```

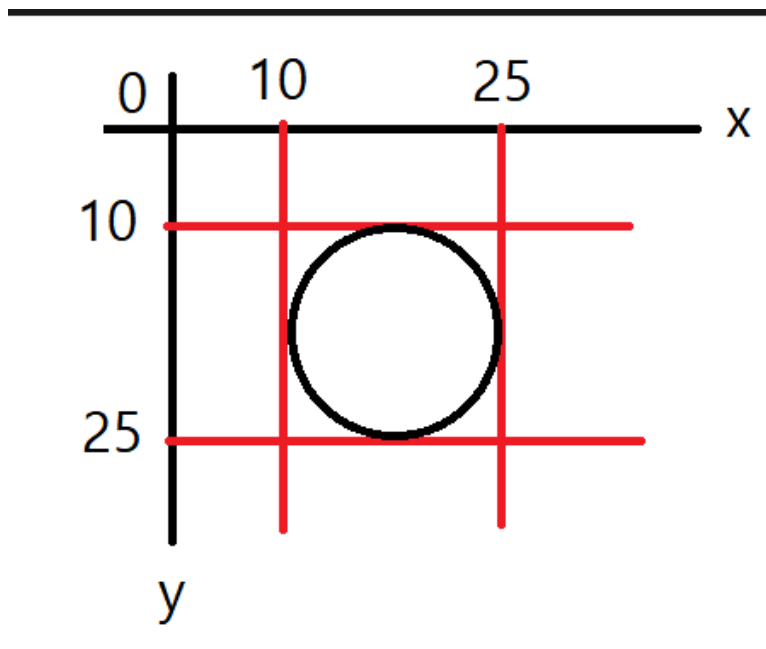
타입)

#볼 자신

↑ ↑ ↑ ↑

#좌측상단 구석에서 시작하는 x y 우측 하단 구석에서 시작하는 x y





```
self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
```

```
def draw(self):
    pass
```

```
tk = Tk() # 1. tk 를 인스턴스화 한다.
```

```
tk.title("Game") # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
```

```
tk.resizable(0, 0) # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
```

```
tk.wm_attributes("-topmost", 1) #4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것을
tkinter 에게 알려준다.
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
```

```
# bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
```

```
# 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
```

```
canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라고
캔버스에에 말해준다.
```

```
tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주는
것이다.
```

```
ball = Ball(canvas,'red')
```

```
while 1:
```

```
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그
    리고
```

```
    tk.update()
```

```
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !
```

문제4. 공이 2개가 캔버스 위에 뜨게 하시오. (빨강, 파랑)

답)

```
from tkinter import *
import random
import time
```

```
class Ball:
```

```

def __init__(self, canvas, val1, val2, color):
    self.canvas = canvas
    self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval : object 형태
타입)
    #볼자신          ↑ ↑ ↑ ↑
    #좌측상단 구석에서 시작하는 x y 우측 하단 구석에서 시작하는 x y
    self.canvas.move(self.id, val1, val2) #공의 위치

def draw(self):
    pass

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
    canvas.pack()
tk.update()
ball = Ball(canvas, 245, 100, 'red')
ball2 = Ball(canvas, 245, 200, 'blue')
while 1:
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```

문제5. 공을하나 빼고 게임이 시작할때 공이 위로 올라가게 하시오.

```

답)
from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
        #볼자신          ↑ ↑ ↑ ↑
        #좌측상단 구석에서 시작하는 x y 우측 하단 구석에서 시작하는 x y
        self.canvas.move(self.id, val1, val2) #공의 위치

    def draw(self):
        self.canvas.move(self.id, 0, -1) # canvas의 move 함수를 호출한다.
        # 0은 가로로 움직이지 마라 !
        # -1 은 화면의 위쪽으로 1픽셀 움직이라 !

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)

```

```

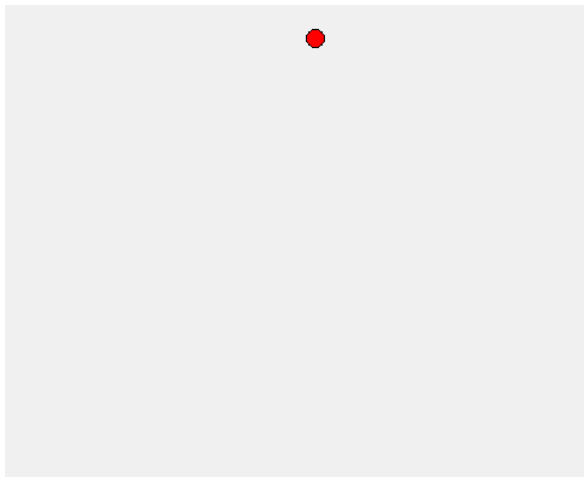
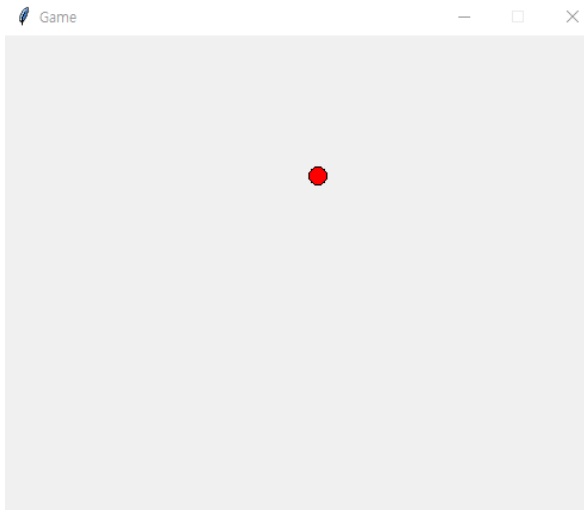
canvas.pack()
tk.update()
ball = Ball(canvas, 245, 100, 'red')

```

```

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```



문제6. 게임이 시작할때 캔버스에 공의 첫 시작위치가 천정 위가 되게하고 시작하자마자 공이 아래로

내려오게 하시오.

답)

```

from tkinter import *
import random
import time

```

```

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

```

#볼자신

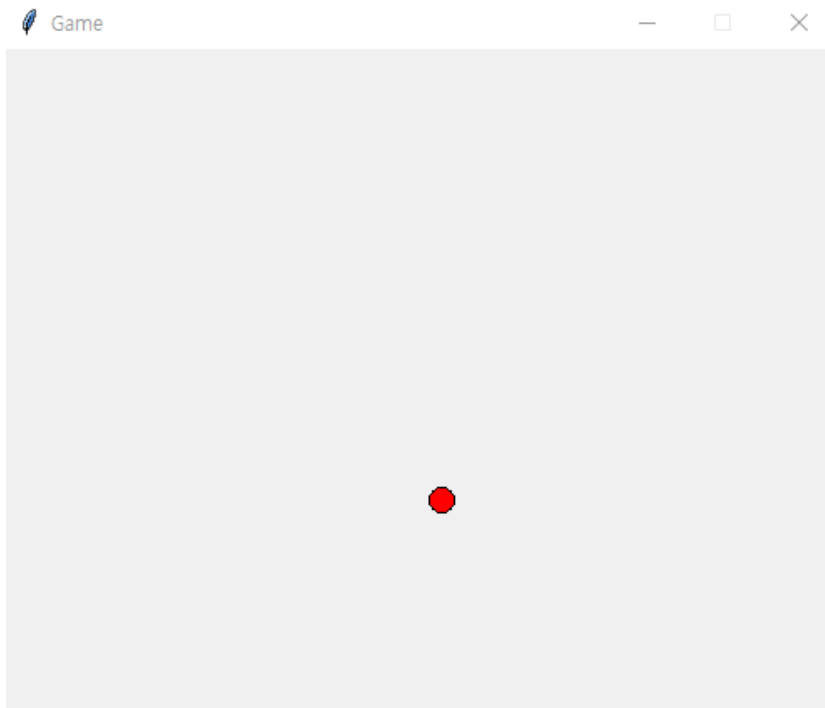
↑ ↑ ↑ ↑

```
self.canvas.move(self.id, val1, val2)
```

```
def draw(self):  
    self.canvas.move(self.id, 0, +1)
```

```
tk = Tk()  
tk.title("Game")  
tk.resizable(0, 0)  
tk.wm_attributes("-topmost", 1)  
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)  
    canvas.pack()  
tk.update()  
ball = Ball(canvas, 245, 0, 'red')
```

```
while 1:  
    ball.draw()  
    tk.update_idletasks()  
    tk.update()  
    time.sleep(0.01)
```



문제7. 게임이 시작할때 공이 왼쪽 대각선 또는 오른쪽 대각선으로 랜덤하게 튀어올라가게 하시오.

답)

```
from tkinter import *  
import random  
import time
```

```
class Ball:  
    def __init__(self, canvas, val1, val2, color):  
        self.canvas = canvas  
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 크기 및 색깔(oval : object 형태  
타입)
```

```

        #볼자신          ↑ ↑ ↑ ↑
        #좌측상단 구석에서 시작하는 x y 우측 하단 구석에서 시작하는 x y
        self.canvas.move(self.id, val1, val2) #공의 위치

        starts = [-3,-2,-1,1,2,3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)

canvas.pack()
tk.update()
ball = Ball(canvas, 245, 200, 'red')

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```

문제8. 캔버스의 전체 높이와 전체 넓이가 어떻게 되는지 확인하시오.

답)

```

def __init__(self, canvas, val1, val2, color):
    self.canvas = canvas
    self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

    self.canvas.move(self.id, val1, val2)

    starts = [-3,-2,-1,1,2,3]
    random.shuffle(starts)

    self.x = starts[0]
    self.y = -3

    self.canvas_height = self.canvas.winfo_height()
    print('높이', self.canvas_height)
    self.canvas_width = self.canvas.winfo_width()
    print('넓이', self.canvas_width)

```

```

높이 400
넓이 500

```

문제9. 공이 키보드 방향키로 움직여지게 하시오.

답)

```
class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)

        self.x = 0
        self.y = 0

        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
# 키보드 방향키 <- 을 누르면 turn_left 라는 함수가 실행되게해라!
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)
# 키보드 방향키 -> 을 누르면 turn_right 라는 함수가 실행되게해라!

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)

    def turn_left(self,evt):
        self.x = -3

    def turn_right(self,evt):
        self.x = 3
```

문제10. 공이 위아래로도 움직여지게 하시오.

답)

```
from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)

        self.x = 0
        self.y = 0

        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
# 키보드 방향키 <- 을 누르면 turn_left 라는 함수가 실행되게해라!
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)
# 키보드 방향키 -> 을 누르면 turn_right 라는 함수가 실행되게해라!
        self.canvas.bind_all('<KeyPress-Up>',self.turn_up)
        self.canvas.bind_all('<KeyPress-Down>',self.turn_down)

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)

    def turn_left(self,evt):
        self.x = -3
```

```

def turn_right(self,evt):
    self.x = 3

def turn_up(self,evt):
    self.y = -3

def turn_down(self,evt):
    self.y = 3
tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)

canvas.pack()
tk.update()
ball = Ball(canvas, 245, 200,'red')

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```

문제11. 공의 현재 좌표를 출력하시오.

```

답)
from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)

        self.x = 0
        self.y = 0

        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
        # 키보드 방향키 <- 을 누르면 turn_left 라는 함수가 실행되게해라!
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)
        # 키보드 방향키 -> 을 누르면 turn_right 라는 함수가 실행되게해라!
        self.canvas.bind_all('<KeyPress-Up>',self.turn_up)
        self.canvas.bind_all('<KeyPress-Down>',self.turn_down)

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)
        print(pos)

    def turn_left(self,evt):
        self.x = -3

```

```

def turn_right(self,evt):
    self.x = 3

def turn_up(self,evt):
    self.y = -3

def turn_down(self,evt):
    self.y = 3
tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)

canvas.pack()
tk.update()
ball = Ball(canvas, 245, 200,'red')

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)
[198.0, 210.0, 213.0, 225.0]
[201.0, 210.0, 216.0, 225.0]
[204.0, 210.0, 219.0, 225.0]
[207.0, 210.0, 222.0, 225.0]
[210.0, 210.0, 225.0, 225.0]
[213.0, 210.0, 228.0, 225.0]
[216.0, 210.0, 231.0, 225.0]
[219.0, 210.0, 234.0, 225.0]
[222.0, 210.0, 237.0, 225.0]
[225.0, 210.0, 240.0, 225.0]
[228.0, 210.0, 243.0, 225.0]
[231.0, 210.0, 246.0, 225.0]
[234.0, 210.0, 249.0, 225.0]
[237.0, 210.0, 252.0, 225.0]
-----

```

문제12. 공을 캔버스 화면 밖으로 안나가도록 하시오.

설명)

```
pos = [170.0, 434.0, 185.0, 449.0]
```

공의 좌측 상단 x,y 공의 우측 하단 x,y

1. 공이 위로 올라가서 천정에 부딪히면 스스로 아래로 튀기게 하는 코드

-----천정의 위치를 0 으로 보고

↑
○

```
if pos[1] <= 0 :
```

```
    self.y = 1
```

2. 공이 아래로 내려가서 바닥에 부딪히면 스스로 위로 튕기게 하는 코드

```
pos = [170.0, 434.0, 185.0, 449.0]
```

공의 좌측 상단 x,y 공의 우측 하단 x,y

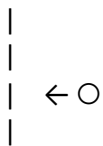


----- 높이 400

```
if pos[3] >= self.canvas_height:
```

```
    self.y = -1
```

3. 공이 서쪽(왼쪽)으로 가서 벽에 부딪히면 다시 동쪽(오른쪽)으로 가게하는 코드

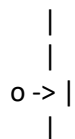


```
pos = [170.0, 434.0, 185.0, 449.0]
```

공의 좌측 상단 x,y 공의 우측 하단 x,y

```
if pos[0] <= 0:  
    self.x = 1
```

4. 공이 동쪽(오른쪽)으로 가서 벽에 부딪히면 다시 서쪽(왼쪽)으로 가게하는 코드



```
pos = [170.0, 434.0, 185.0, 449.0]
```

공의 좌측 상단 x,y 공의 우측 하단 x,y

```
if pos[2] >= self.canvas_width:  
    self.x = -1
```

답)

```
from tkinter import *  
import random  
import time
```

```
class Ball:  
    def __init__(self, canvas, val1, val2, color):  
        self.canvas = canvas  
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
```

```

self.canvas.move(self.id, val1, val2)

self.x = 0
self.y = 0

self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
# 키보드 방향키 <- 을 누르면 turn_left 라는 함수가 실행되게해라!
self.canvas.bind_all('<KeyPress-Right>',self.turn_right)
# 키보드 방향키 -> 을 누르면 turn_right 라는 함수가 실행되게해라!
self.canvas.bind_all('<KeyPress-Up>',self.turn_up)
self.canvas.bind_all('<KeyPress-Down>',self.turn_down)

self.canvas_height = self.canvas.winfo_height()
self.canvas_width = self.canvas.winfo_width()

def draw(self):
    self.canvas.move(self.id, self.x, self.y)

    pos = self.canvas.coords(self.id)
    if pos[1] <= 0 :
        self.y = 3
    if pos[3] >= self.canvas_height:
        self.y = -3
    if pos[0] <= 0:
        self.x = 3
    if pos[2] >= self.canvas_width:
        self.x = -3

def turn_left(self,evt):
    self.x = -3

def turn_right(self,evt):
    self.x = 3

def turn_up(self,evt):
    self.y = -3

def turn_down(self,evt):
    self.y = 3
tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)

canvas.pack()
tk.update()
ball = Ball(canvas, 245, 200,'red')

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```

문제13. 공이 키보드가 아니라 랜덤으로 움직여지게 하시오.

답)

```
from tkinter import *
import random
import time
```

```
class Ball:
```

```
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.canvas_height = self.canvas.winfo_height()

        self.canvas_width = self.canvas.winfo_width()

        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3
```

```
    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)

        if pos[1] <= 0:
            self.y = 3
        if pos[3] >= self.canvas_height:
            self.y = -3
        if pos[0] <= 0:
            self.x = 3
        if pos[2] >= self.canvas_width:
            self.x = -3
```

```
tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()
```

```
ball1 = Ball(canvas, 250, 100, 'red')
```

```
while 1:
    ball1.draw()
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을
    그리고
    tk.update()
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !
```

문제14. 공이 2개가 게임 시작할때 랜덤으로 움직여지게 하시오.

답)

```

from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)

        if pos[1] <= 0:
            self.y = 3
        if pos[3] >= self.canvas_height:
            self.y = -3
        if pos[0] <= 0:
            self.x = 3
        if pos[2] >= self.canvas_width:
            self.x = -3

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)

canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

ball1 = Ball(canvas, 250, 100, 'red')
ball2 = Ball(canvas, 250, 200, 'blue')

while 1:
    ball1.draw()
    ball2.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)

```

문제15. 패들을 canvas에 올리시오.

답)

```

from tkinter import *
import random
import time

```

```

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

        self.x = starts[0]
        self.y = -3

    def draw(self):
        self.canvas.move(self.id, self.x, self.y)
        pos = self.canvas.coords(self.id)

        if pos[1] <= 0:
            self.y = 3
        if pos[3] >= self.canvas_height:
            self.y = -3
        if pos[0] <= 0:
            self.x = 3
        if pos[2] >= self.canvas_width:
            self.x = -3

#-----
class Paddle:
    def __init__(self, canvas, y_loc, color):
        self.canvas = canvas

        self.id = canvas.create_rectangle(0, 0, 100, 10, fill=color) # 패들의 높이와 넓이 그리고
        색깔

        self.canvas.move(self.id, 200, y_loc) # 패들 사각형을 200,300 에 위치
        self.x = 0 # 패들이 처음 시작할때 움직이지 않게 0으로 설정
        self.canvas_width = self.canvas.winfo_width() # 캔버스의 넓이를 반환한다. 캔버스 밖
        으로 패들이 나가지 않도록

        self.canvas.bind_all('<KeyPress-Left>', self.turn_left) # 왼쪽 화살표 키를 '<KeyPress-
        Left>' 라는 이름로 바인딩

        self.canvas.bind_all('<KeyPress-Right>', self.turn_right) # 오른쪽도 마찬가지로 바인딩
        한다.

    def draw(self):
        pos = self.canvas.coords(self.id)
        # print(pos)

        self.canvas.move(self.id, self.x, 0)

```



```
# 패들이 화면의 끝에 부딪히면 공처럼 튕기는게 아니라 움직임이 멈춰야한다.
# 그래서 왼쪽 x 좌표(pos[0]) 가 0 과 같거나 작으면 self.x = 0 처럼 x 변수에 0 을
# 설정한다. 같은 방법으로 오른쪽 x 좌표(pos[2]) 가 캔버스의 폭과 같거나 크면
# self.x = 0 처럼 변수에 0 을 설정한다.
```

```
def turn_left(self, evt): # 패들의 방향을 전환하는 함수
    self.x = -3
```

```
def turn_right(self, evt):
    self.x = 3
```

```
def move(self, x):
    self.x = x
```

```
#-----
```

```
tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
```

```
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()
```

```
ball1 = Ball(canvas,250,100,'red')
ball2 = Ball(canvas,250,200,'blue')
paddle = Paddle(canvas,370,'blue')
```

```
while 1:
    ball1.draw()
    ball2.draw()
    paddle.draw()
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을
    그리고
    tk.update()
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !
```

문제16. 패들을 밖으로 못나가게 하시오.

```
답)
from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, val1, val2, color):
        self.canvas = canvas
        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)

        self.canvas.move(self.id, val1, val2)
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)
```

```

self.x = starts[0]
self.y = -3

def draw(self):
    self.canvas.move(self.id, self.x, self.y)
    pos = self.canvas.coords(self.id)

    if pos[1] <= 0:
        self.y = 3
    if pos[3] >= self.canvas_height:
        self.y = -3
    if pos[0] <= 0:
        self.x = 3
    if pos[2] >= self.canvas_width:
        self.x = -3

class Paddle:
    def __init__(self, canvas, y_loc, color):
        self.canvas = canvas

        self.id = canvas.create_rectangle(0, 0, 100, 10, fill=color) # 패들의 높이와 넓이 그리고
        색깔

        self.canvas.move(self.id, 200, y_loc) # 패들 사각형을 200,300 에 위치
        self.x = 0 # 패들이 처음 시작할때 움직이지 않게 0으로 설정
        self.canvas_width = self.canvas.winfo_width() # 캔버스의 넓이를 반환한다. 캔버스 밖
        으로 패들이 나가지 않도록

        self.canvas.bind_all('<KeyPress-Left>', self.turn_left) # 왼쪽 화살표 키를 '<KeyPress-
        Left>' 라는 이름으로 바인딩
        self.canvas.bind_all('<KeyPress-Right>', self.turn_right) # 오른쪽도 마찬가지로 바인딩
        한다.

    def draw(self):
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0 and self.x < 0: # 패들의 위치가 왼쪽 끝이고, 이동하려는 방향이 왼쪽이
        면 함수 종료(이동 안 함)
            return
        elif pos[2] >= self.canvas_width and self.x > 0: # 패들의 위치가 오른쪽 끝이고, 이동하
        려는 방향이 오른쪽이면 함수 종료
            return
        self.canvas.move(self.id, self.x, 0)

        self.canvas.move(self.id, self.x, 0)

        # 패들이 화면의 끝에 부딪히면 공처럼 튕기는게 아니라 움직임이 멈춰야한다.
        # 그래서 왼쪽 x 좌표(pos[0]) 가 0 과 같거나 작으면 self.x = 0 처럼 x 변수에 0 을
        # 설정한다. 같은 방법으로 오른쪽 x 좌표(pos[2]) 가 캔버스의 폭과 같거나 크면
        # self.x = 0 처럼 변수에 0 을 설정한다.

```

```

def turn_left(self, evt): # 패들의 방향을 전환하는 함수
    self.x = -3

def turn_right(self, evt):
    self.x = 3

def move(self, x):
    self.x = x

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)

canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

ball1 = Ball(canvas, 250, 100, 'red')
ball2 = Ball(canvas, 250, 200, 'blue')
paddle = Paddle(canvas, 370, 'blue')

while 1:
    ball1.draw()
    ball2.draw()
    paddle.draw()
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을
    그리고
    tk.update()
    time.sleep(0.01) # 100분의 1초마다 잠들어라 !

```

문제17. 공이 패들에 맞으면 튕겨 오르게 하시오.

```

답)
from tkinter import *
import random
import time

class Ball:
    def __init__(self, canvas, paddle, color):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval : object 형태
        타입)

        self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)
        #공의 속도
        self.x = starts[0]
        self.y = -3

        self.canvas_height = self.canvas.winfo_height()
        self.canvas_width = self.canvas.winfo_width()

```

```

self.hit_bottom = False

def hit_paddle(self,pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]:
        if pos[3] >= paddle_pos[1] and pos[3] <= paddle_pos[3]:
            return True
    return False

def draw(self):
    self.canvas.move(self.id, self.x, self.y) #공을 움직이게 하는 부분
    #공이 화면 밖으로 나가지 않게 해준다
    pos = self.canvas.coords(self.id)
    if pos[1] <= 0:
        self.y = 3
    if pos[3] >= self.canvas_height: #바닥에 부딪히면 게임오버
        self.hit_bottom = True
    if pos[0] <= 0:
        self.x = 3
    if pos[2] >= self.canvas_width:
        self.x = -3
    if self.hit_paddle(pos) == True: #판에 부딪히면 위로 튕겨올라가게
        self.y = -3

class Paddle:
    def __init__(self,canvas,color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0,0,100,10,fill=color)
        self.canvas.move(self.id, 200, 300)
        self.x = 0
        self.canvas_width = self.canvas.winfo_width()
        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)

    def draw(self):
        self.canvas.move(self.id, self.x, 0)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:
            self.x = 0

    def turn_left(self,evt):
        self.x = -3

    def turn_right(self,evt):
        self.x = 3

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

```

```
paddle = Paddle(canvas,'blue')
ball = Ball(canvas, paddle, 'red')
start = False
```

#공을 약간 움직이고 새로운 위치로 화면을 다시 그리며, 잠깐 잠들었다가 다시 시작해
라!

```
while 1:
    if ball.hit_bottom == False:
        ball.draw()
        paddle.draw()
```

```
#그림을 다시 그려라! 라고 쉴새없이 명령
tk.update_idletasks()
tk.update()
```

```
#시작 전 2초간 sleep
if not start:
    time.sleep(2)
    start = True
```

```
time.sleep(0.01)
```

문제19. 다시 패들은 양옆으로만 움직이게 하고 공이 패들에 닿으면 공이 멈추게하시오.

답)

```
from tkinter import *
import random
import time
```

```
class Ball:
    def __init__(self, canvas, paddle, color):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval : object 형태
```

타입)

```
self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
starts = [-3, -2, -1, 1, 2, 3]
random.shuffle(starts)
#공의 속도
self.x = starts[0]
self.y = -3
```

```
self.canvas_height = self.canvas.winfo_height()
self.canvas_width = self.canvas.winfo_width()
self.hit_bottom = False
```

```
def hit_paddle(self,pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]:
        if pos[3] >= paddle_pos[1] and pos[3] <= paddle_pos[3]:
            return True
    return False
```

```
def draw(self):
```

```

self.canvas.move(self.id, self.x, self.y) #공을 움직이게 하는 부분
#공이 화면 밖으로 나가지 않게 해준다
pos = self.canvas.coords(self.id)
if pos[1] <= 0:
    self.y = 2
if pos[3] >= self.canvas_height: #바닥에 부딪히면 게임오버
    self.hit_bottom = True
if pos[0] <= 0:
    self.x = 2
if pos[2] >= self.canvas_width:
    self.x = -2
if self.hit_paddle(pos) == True: #판에 부딪히면 위로 튕겨올라가게
    self.y = 0
    self.x = 0

class Paddle:
    def __init__(self,canvas,color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0,0,100,10,fill=color)
        self.canvas.move(self.id, 200, 300)
        self.x = 0
        self.y = 0
        self.canvas_width = self.canvas.winfo_width()
        self.canvas.bind_all('<KeyPress-Left>',self.turn_left)
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right)

    def draw(self):
        self.canvas.move(self.id, self.x, 0)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:
            self.x = 0

    def turn_left(self,evt):
        self.x = -3

    def turn_right(self,evt):
        self.x = 3

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

paddle = Paddle(canvas,'blue')
ball = Ball(canvas, paddle, 'red')
start = False

#공을 약간 움직이고 새로운 위치로 화면을 다시 그리며, 잠깐 잠들었다가 다시 시작해

```

```

라!
while 1:
    if ball.hit_bottom == False:
        ball.draw()
        paddle.draw()

    #그림을 다시 그려라! 라고 쉴새없이 명령
    tk.update_idletasks()
    tk.update()

    #시작 전 2초간 sleep
    if not start:
        time.sleep(2)
        start = True

    time.sleep(0.01)

```

문제20. 공이 패들에 닿으면 패들과 함께 붙어서 움직이게 하시오.

```

답)
If self.hit_paddle(pos) == True:
    self.y=0
    self.x=paddle.x

```

문제21. 다시 스페이바를 누르면 공이 위로 튀어올라가게 하시오.

```

힌트)
Ball calss 에

__init__
    self.canvas.bind_all('<Key>', self.turn_space)

def turn_space(self, evt):
    starts = [-9,-8,-7,-4,-3,-2,-1]
    random.shuffle(starts)
    self.y = -3
    self.x = start[0]

```

```

답)
from tkinter import *
import random
import time

class Ball:

    def __init__(self, canvas, paddle, color):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) #공 좌표 및 색깔(oval : object 형태
타입)

        self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
        starts = [-3, -2, -1, 1, 2, 3]
        random.shuffle(starts)

```

```

#공의 속도
self.x = starts[0]
self.y = -3

self.canvas_height = self.canvas.wininfo_height()
self.canvas_width = self.canvas.wininfo_width()
self.hit_bottom = False
self.canvas.bind_all('<Key>', self.turn_space)

def hit_paddle(self, pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]:
        if pos[3] >= paddle_pos[1] and pos[3] <= paddle_pos[3]:
            return True
    return False

def draw(self):
    self.canvas.move(self.id, self.x, self.y) #공을 움직이게 하는 부분
    #공이 화면 밖으로 나가지 않게 해준다
    pos = self.canvas.coords(self.id)
    if pos[1] <= 0:
        self.y = 2
    if pos[3] >= self.canvas_height: #바닥에 부딪히면 게임오버
        self.hit_bottom = True
    if pos[0] <= 0:
        self.x = 2
    if pos[2] >= self.canvas_width:
        self.x = -2
    if self.hit_paddle(pos) == True: #판에 부딪히면 위로 튕겨올라가게
        self.y = 0
        self.x = paddle.x

def turn_space(self, evt):
    starts = [-9,-8,-7,-4,-3,-2,-1]
    random.shuffle(starts)
    self.y = -3
    self.x = start[0]

class Paddle:
    def __init__(self, canvas, color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0,0,100,10,fill=color)
        self.canvas.move(self.id, 200, 300)
        self.x = 0
        self.y = 0
        self.canvas_width = self.canvas.wininfo_width()
        self.canvas.bind_all('<KeyPress-Left>', self.turn_left)
        self.canvas.bind_all('<KeyPress-Right>', self.turn_right)

    def draw(self):
        self.canvas.move(self.id, self.x, 0)
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0:
            self.x = 0
        elif pos[2] >= self.canvas_width:

```



```

        self.x = 0

    def turn_left(self, evt):
        self.x = -3

    def turn_right(self, evt):
        self.x = 3

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

paddle = Paddle(canvas, 'blue')
ball = Ball(canvas, paddle, 'red')
start = False

#공을 약간 움직이고 새로운 위치로 화면을 다시 그리며, 잠깐 잠들었다가 다시 시작해
라!
while 1:
    if ball.hit_bottom == False:
        ball.draw()
        paddle.draw()

    #그림을 다시 그려라! 라고 쉴새없이 명령
    tk.update_idletasks()
    tk.update()

    #시작 전 2초간 sleep
    if not start:
        time.sleep(2)
        start = True

    time.sleep(0.01)

```

강화학습 완성코드

```

from tkinter import *
import random
import time
import csv

EMPTY = 0
PADDLE_HEIGHT = 360.0
PADDLE_MOVE = [-8, 8] # PADDLE이 좌우로 이동하는 속도

START = 3
END = 4

class Ball:
    def __init__(self, canvas, paddle, color, announceterm, saveterm, winval = 10, loseval = -1):

```

```

self.canvas = canvas
self.paddle = paddle
self.id = canvas.create_oval(10, 10, 25, 25, fill=color) # 공 크기 및 색깔

self.canvas.move(self.id, 245, 200) # 공을 캔버스 중앙으로 이동
starts = [-3, 3] # 공의 속도를 랜덤으로 구성하기 위해 준비한 리스트
random.shuffle(starts) # starts 리스트 중에 숫자를 랜덤으로 골라서
self.x = starts[0] # 처음 공이 패들에서 움직일때 왼쪽으로 올라갈지 오른쪽으로 올라
갈지 랜덤으로 결정되는 부분
self.y = -3 # 처음 공이 패들에서 움직일때 위로 올라가는 속도
self.canvas_height = self.canvas.winfo_height() # 캔버스의 현재 높이를 반환한다.(공이
화면에서 사라지지 않기위해)
self.canvas_width = self.canvas.winfo_width() # 캔버스의 현재 넓이를 반환한다.(공이
화면에서 사라지지 않기위해)
self.hit_bottom = False
self.values = {}
self.epsilon = 0 #랜덤율
self.alpha = 0.99 #망각계수
self.learning = True
self.cycle_data = []
self.wincount = 0
self.losecount = 0
self.gamecount = 0
self.winval = winval #성공 보상치
self.loseval = loseval #실패 보상치
self.announceterm = announceterm # 게임 횟수 프린트 텀
self.saveterm = saveterm # csv 저장 텀

# csv 파일에서 gamecount / values 불러옴
self.loadcsv()

def action(self):
    r = random.random()
    if r < self.epsilon:
        direction = self.randomChoice()
    else:
        direction = self.greedyChoice()
    x = PADDLE_MOVE[direction]
    # 머신러닝을 위한 한 사이클 내 이동 데이터 저장
    self.cycle_data.append(self.keystate(direction))
    #이동/그리기
    self.paddle.move(x)
    self.paddle.draw()

#이동 방향 랜덤으로 지정
def randomChoice(self):
    rand = random.choice([0,1])
    key = self.keystate(rand)

```

```

if key not in self.values:
    self.add(key)
return rand

```

#이동 방향 Greedy로 지정

```

def greedyChoice(self):
    val_left = self.keystate(0)
    val_right = self.keystate(1)

    if self.lookup(val_left) > self.lookup(val_right):
        return 0
    elif self.lookup(val_left) < self.lookup(val_right):
        return 1
    else:
        return random.choice([0,1])

```

```

def add(self, key):
    self.values[key] = 0

```

```

def lookup(self, key):
    if key not in self.values:
        #print(key)
        self.add(key)
    return self.values[key]

```

```

def hit_paddle(self, pos): # 패들에 공이 튀기게 하는 함수
    paddle_pos = self.canvas.coords(self.paddle.id)
    # 공의 x좌표가 패들의 너비 안에 있는지 / 공 바닥이 패들 윗면에 닿아 있는지
    if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2] \
        and pos[3] == PADDLE_HEIGHT:
        return True
    return False

```

#공을 패들로 받았는지 여부 출력(True/False)

```

def is_paddle_hit(self):
    return self.hit_paddle(self.canvas.coords(self.id))

```

```

def draw(self):
    # 볼의 현재 좌표를 출력해준다. 공 좌표( 좌상단 x,y좌표 / 우하단 x,y좌표 )
    pos = self.canvas.coords(self.id)
    # [ 255,29,270,44]

    if pos[1] <= 0:
        self.y = 3
    if pos[3] >= self.canvas_height:
        self.y = -3
    if pos[0] <= 0:
        self.x = 3
    if pos[2] >= self.canvas_width:
        self.x = -3
    if self.hit_paddle(pos) == True:
        self.y = -3

```

```

self.canvas.move(self.id, self.x, self.y) # 공을 움직이게 하는 부분

```

공이 화면 밖으로 나가지 않게 해준다

```
def keystate(self, movement):
    paddle_pos = self.canvas.coords(self.paddle.id)
    ball_pos = self.canvas.coords(self.id)
    #paddle 위치(좌측 x좌표), 공의 좌상단 x/y좌표, 공의 좌우/상하 속도(방향), paddle을
    #좌/우 중 어느 쪽으로 움직이는지
    return (paddle_pos[0], (ball_pos[0], ball_pos[1]), (self.x, self.y), movement)
```

사이클 시작 : 1, 사이클 종료 : -1, 해당 없음 : 0

```
def cyclestate(self):
    pos = self.canvas.coords(self.id)
    if pos[3] == PADDLE_HEIGHT:
        if self.y == -3:
            return START
        elif self.y == 3:
            return END
    return 0
```

결과 학습

```
def backup(self, newVal, idx):
    if idx >= 0 and self.learning:
        prevVal = self.values[self.cycle_data[idx]]
        self.values[self.cycle_data[idx]] += self.alpha * (newVal - prevVal)
        #print("key : {0}, val : {1}".format(self.cycle_data[idx], self.values[self.cycle_data[idx]]))
        self.backup(newVal * self.alpha, idx - 1)
```

게임 끝났을 시 학습 및 cycle_data 초기화

```
def gameover(self):
    if self.learning:
        #paddle로 받았으면 1, 못 받았으면 -1
        if self.is_paddle_hit():
            result_value = self.winval
            self.wincount += 1
        else:
            result_value = self.loseval
            self.losecount += 1
        self.backup(result_value, len(self.cycle_data) - 1)
        self.gamecount += 1
```

#saveterm마다 csv 저장

```
if self.gamecount % self.saveterm == 0:
    self.writecsv()
if self.gamecount % self.announceterm == 0:
    print("cycle count : {0}".format(ball.gamecount))
self.cycle_data.clear()
```

```
def winnerval(self, winner):
    if winner == 'hit':
        return 1
    elif winner == 'miss':
        return -1
    else:
```

```
return 0
```

```
#게임 결과 csv로 저장
```

```
def writcsv(self):
```

```
    try:
```

```
        # Values 저장
```

```
        Fn = open("E:\pong_value.csv", 'w', newline="")
```

```
        writer = csv.writer(Fn, delimiter=',')
```

```
        writer.writerow([self.gamecount]) #첫줄에 학습 게임 횟수 저장
```

```
        keys = self.values.keys()
```

```
        for key in keys:
```

```
            writer.writerow([key[0],
```

```
                              key[1][0],
```

```
                              key[1][1],
```

```
                              key[2][0],
```

```
                              key[2][1],
```

```
                              key[3],
```

```
                              ball.values[key]
```

```
            ])
```

```
        Fn.close()
```

```
#성공/실패 횟수 저장
```

```
Fn = open("E:\pong_score.csv", 'a', newline="")
```

```
writer = csv.writer(Fn, delimiter=',')
```

```
writer.writerow([self.wincount, self.losecount, self.gamecount])
```

```
Fn.close()
```

```
#승률의 변화를 확인하기 위해 일정 판수마다 카운트 리셋
```

```
self.wincount = 0
```

```
self.losecount = 0
```

```
print("save data in cycle {0}.".format(self.gamecount))
```

```
except Exception as e:
```

```
    print('save data failed in cycle {0}.\nError Type : {1}'.format(self.gamecount,type(e).
```

```
__name__))
```

```
def loadcsv(self):
```

```
    try:
```

```
        Fn = open("E:\pong_value.csv", 'r')
```

```
        self.gamecount = int(Fn.readline().split(',')[0]) # 첫 줄의 학습 게임 횟수 불러오기
```

```
        reader = csv.reader(Fn, delimiter=',')
```

```
        for key in reader:
```

```
            self.values[(int(float(key[0])), (int(float(key[1])), int(float(key[2]))), (int(float(key[3])),
```

```
int(float(key[4])), int(float(key[5])))] = float(key[6])
```

```
            print('Load Success! Start at cycle {0}'.format(self.gamecount))
```

```
        except Exception:
```

```
            print('Load Failed!')
```

```
class Paddle:
```

```
    def __init__(self, canvas, y_loc, color):
```

```
        self.canvas = canvas
```

```
        self.id = canvas.create_rectangle(0, 0, 100, 10, fill=color) # 패들의 높이와 넓이 그리고
```

```
색깔
```

```

self.canvas.move(self.id, 200, y_loc) # 패들 사각형을 200,300 에 위치
self.x = 0 # 패들이 처음 시작할때 움직이지 않게 0으로 설정
self.canvas_width = self.canvas.winfo_width() # 캔버스의 넓이를 반환한다. 캔버스 밖
으로 패들이 나가지 않도록

self.canvas.bind_all('<KeyPress-Left>', self.turn_left) # 왼쪽 화살표 키를 '<KeyPress-
Left>' 라는 이름으로 바인딩

self.canvas.bind_all('<KeyPress-Right>', self.turn_right) # 오른쪽도 마찬가지로 바인딩
한다.

def draw(self):
    pos = self.canvas.coords(self.id)
    # print(pos)
    if pos[0] <= 0 and self.x < 0: # 패들의 위치가 왼쪽 끝이고, 이동하려는 방향이 왼쪽이
면 함수 종료(이동 안 함)
        return
    elif pos[2] >= self.canvas_width and self.x > 0: # 패들의 위치가 오른쪽 끝이고, 이동하
려는 방향이 오른쪽이면 함수 종료
        return
    self.canvas.move(self.id, self.x, 0)

    # 패들이 화면의 끝에 부딪히면 공처럼 튕기는게 아니라 움직임이 멈춰야한다.
    # 그래서 왼쪽 x 좌표(pos[0]) 가 0 과 같거나 작으면 self.x = 0 처럼 x 변수에 0 을
    # 설정한다. 같은 방법으로 오른쪽 x 좌표(pos[2]) 가 캔버스의 폭과 같거나 크면
    # self.x = 0 처럼 변수에 0 을 설정한다.

def turn_left(self, evt): # 패들의 방향을 전환하는 함수
    self.x = -3

def turn_right(self, evt):
    self.x = 3

def move(self, x):
    self.x = x

if __name__ == '__main__':
    tk = Tk() # tk 를 인스턴스화 한다.
    tk.title("Game") # tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
    tk.resizable(0, 0) # 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
    tk.wm_attributes("-topmost", 1) # 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것을
tkinter 에게 알려준다.

    canvas = Canvas(tk, width=500, height=450, bd=0, highlightthickness=0)
    # bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
    # 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)

```

canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라고 캔버스에에 말해준다.

tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주는 것이다.

paddle = Paddle(canvas, PADDLE_HEIGHT, 'blue')

#announceterm : 현재 count 출력 term, saveterm : csv에 저장하는 term

ball = Ball(canvas, paddle, 'red', announceterm=500, saveterm=500)

start = False

공을 약간 움직이고 새로운 위치로 화면을 다시 그리며, 잠깐 잠들었다가 다시 시작해 ! "

is_cycling = False

while 1:

ball.draw()

c_state = ball.cyclestate()

if c_state == END:

#print('END')

ball.gameover()

is_cycling = False

if c_state == START or ball.is_paddle_hit():

#print('START')

is_cycling = True

if is_cycling:

ball.action()

tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그려라 !

tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주는 것이다.

#1만번 학습 후에 정상 속도로 플레이 시작(학습 결과 반영됨)

if ball.gamecount > 10000:

time.sleep(0.005) # 무한 루프중에 100분의 1초마다 잠들어라 !

머신러닝 종류 3가지

1. 지도학습 : 사람이 레이블을 알려주고 (지도)데이터를 분류해라 라고 학습시키는 것.
Ex) 산타토익(문제,답) --> 머신러닝 신경망
2. 비지도 학습 : 사람이 레이블을 주지 않고 컴퓨터가 스스로 데이터의 패턴을 발견해서 데이터를
분류하는 것.
3. 강화 학습 : 시행착오를 통해서 환경에 적응하는 학습방법.
지도학습과는 다르고, 이 행동이 맞다 라고 명시적으로 주어지는 게 아니라 보

상을

통해서 행동의 바람직함을 스스로 알아가는 학습 방법.

강화 학습에서 사용하는 용어

1. 에이전트 : 강화학습을 통해 스스로 학습하는 컴퓨터 (학습 주제)
2. 환경 : 에이전트가 행동하는 곳.
3. 상태 : 에이전트의 상태로 에이전트가 처한 특정한 상황에 대한 관찰
4. 행동 : 에이전트가 어떠한 상태에서 취할수 있는 행동
5. 보상 : 행동에 대한 보상

"강화 학습을 다른 머신러닝 기법과는 다르게 만들어주는 가장 핵심적인 요소"

1. 즉각적 보상
2. 최종 보상
6. 정책 : 에이전트가 보상을 얻으려면 행동을 해야하는데 특정 상태가 아닌 모든 상태에 대해

어떤 행동을 해야하는지 알아야 하는데 이렇게 모든 상태에 대해 에이전트가 어떤 행동을 해야하는지 정해놓은 정책.

• 딥마인드는 강화 학습을 이용해서

게임 -----> 구글 서버 센터 에너지 효율을 높이는 시스템 구축

• 특정 게임이 히트 칠려면? -----> 강화 학습 활용

1. 핵 사용자를 판별해서 막아야 한다.
2. 어떠한 무기가 게임을 재미있게 만드는지 판단할때.

• 요즘 활용 되고 있는 부분? **스마트 팩토리**

• 강화 학습의 역사

피피티보고 다시 채워!!!!!!!!!!!!!!

강화학습 입힌 코드들 시연

1. 인공지능 핑퐁 코드 --> sarsa
2. Mit 공대에서 만든 인공지능 틱택토 코드 -->
3. 경사감소법을 이용한 틱택토 코드

4. 동킹 게임 -----> sarsa 구현

```
from tkinter import *
import random
import time
import numpy as np
```



```

import csv
from copy import deepcopy

#####
##### 공의 위치 파일 저장/불러오기 #####

# 공의 위치 파일 저장(1회만 파일 저장) ##### !!!! 두번째부터는 None으로 놓기!!!!
#####
save_ballloc = 'c:\python_data\pingpong_move.csv'
# save_ballloc 파일 위치 입력(반드시 입력해야 함. save_ballloc 의 위치와 동일한 위치로
설정)
load_ballloc = 'c:\python_data\pingpong_move.csv'

##### 회귀분석 가중치 파일 저장/불러오기 #####
# 가중치 파일 저장(저장하고 싶으면 위치 입력. 아니면 None 으로 놓기)
save_weightloc = 'c:\python_data\pingpong_weight.csv'
# save_weightloc 파일 위치 입력(파일 참조하지 않으려면 None 으로 놓기)
load_weightloc = 'c:\python_data\pingpong_weight.csv'

##### 경사감소법 튜닝 #####

# 경사감소법 learning_rate(변경x)
learning_rate = 0.5
# 경사감소법 시행횟수(변경x)
training_cnt= 20000
#가능조합(learning_rate = 0.00001, training_cnt = 50000)
#가능조합(learning_rate = 0.00002, training_cnt = 25000)

#####

class Ball:
    def __init__(self, canvas, paddle, color, save=False):
        self.canvas = canvas
        self.paddle = paddle
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color) # 공 크기 및 색깔
        self.canvas.move(self.id, 245, 100) # 공을 캔버스 중앙으로 이동
        self.x = random.choice([-4,-3, -2, -1, 1, 2, 3,4]) # 처음 공이 패들에서 움직일때 왼쪽으
로 올라갈지 오른쪽으로 올라갈지 랜덤으로 결정되는 부분
        self.y = -3 # 처음 공이 패들에서 움직일때 위로 올라가는 속도
        self.canvas_height = self.canvas.winfo_height() # 캔버스의 현재 높이를 반환한다.(공
이 화면에서 사라지지 않기위해)
        self.canvas_width = self.canvas.winfo_width() # 캔버스의 현재 넓이를 반환한다.(공이
화면에서 사라지지 않기위해)
        self.hit_bottom = False
        self.save = save
        self.ball_start = []
        self.ball_end = []
        self.convertloc = self.canvas.coords(self.id)[0]

```

```

self.leftorright = 0

def hit_paddle(self, pos): # 패들에 공이 튀기게 하는 함수
    paddle_pos = self.canvas.coords(self.paddle.id)
    if self.save == True:
        if pos[3] >= paddle_pos[1] and pos[3] <= paddle_pos[3]: # 공이 패들에 닿았을때 좌
표
            return True
        elif self.save == False:
            if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]: # 공이 패들에 내려오기 직
전 좌표
                if pos[3] >= paddle_pos[1] and pos[3] <= paddle_pos[3]: # 공이 패들에 닿았을때
좌표
                    return True
            return False

##### 공이 떨어지는 가상의 좌표 #####
def endloc(self, pos):
    paddle_pos = self.canvas.coords(self.paddle.id)
    if 290 > pos[1] >= 285 and pos[3] <= paddle_pos[3] and self.y > 0: # 공이 패들 통과할 때
의 좌표
        return pos[0]

def draw(self):
    self.canvas.move(self.id, self.x, self.y) # 공을 움직이게 하는 부분
    pos = self.canvas.coords(self.id) # 볼의 현재 좌표를 출력해준다. 공 좌표( 서쪽(0) , 남
쪽(1) , 동쪽(2), 북쪽(3) )
    paddle_pos = self.canvas.coords(self.paddle.id)

#####
# 가상의 좌표를 만드는 과정
# self.leftorright는 기본은 0, 최초로 벽에 부딪혔을 때 왼쪽 벽이면 -1, 오른쪽 벽이면
1 을 출력
    if self.leftorright == 0:
        self.convertloc += float(self.x)
    elif self.leftorright != 0:
        self.convertloc += self.leftorright * abs(float(self.x))
#####
    if pos[1] <= 0:
        self.y *= -1
    if pos[3] >= self.canvas_height:
        self.x = random.choice([-1,1])
        self.y *= -1

    if pos[0] <= 0:
        self.x *= -1
        ##### 최초로 왼쪽 벽에 부딪히면 self.leftorright = -1이 됨 #####
    if self.leftorright == 0:
        self.leftorright = -1

```

```

if pos[2] >= self.canvas_width:
    self.x *= -1 # 공을 왼쪽으로 돌린다.
    ##### 최초로 오른쪽 벽에 부딪히면 self.leftorright = 1이 됨 #####
    if self.leftorright == 0:
        self.leftorright = 1

if self.hit_paddle(pos) == True:
    self.x = random.choice(range(-11,12,2))
    self.y *= -1
    ##### (공의 시작 x좌표, 시작 시 x속력, y속력, 상수1) 을 저장 #####
    self.ball_start.append([pos[0], float(self.x), float(self.y), 1.0])
    ##### (공이 떨어진 x 좌표) 를 저장
    self.ball_end.append(self.convertloc)
    ##### 패들에 부딪히면, 새로운 공의 시작 정보를 저장하기 위해 가상좌표와
    leftorright 값을 초기화 #####
    self.convertloc = pos[0]
    self.leftorright = 0

class Paddle:
    def __init__(self, canvas, color):
        self.canvas = canvas
        self.id = canvas.create_rectangle(0, 0, 100, 10, fill=color)
        self.canvas.move(self.id, 200, 300)
        self.x = 0
        self.canvas_width = self.canvas.winfo_width()

    def draw(self):
        pos = self.canvas.coords(self.id)
        if pos[0] <= 0 and self.x < 0: # 패들의 위치가 왼쪽 끝이고, 이동하려는 방향이 왼쪽이
            면 함수 종료(이동 안 함)
            return
        elif pos[2] >= self.canvas_width and self.x > 0: # 패들의 위치가 오른쪽 끝이고,이동하
            려는 방향이 오른쪽이면 종료
            return
        self.canvas.move(self.id, self.x, 0)

    ##### 공이 떨어지는 가상의 좌표를 실제 게임 내 좌표로 바꿔주는 메소드
    #####

    def convertendloc(self, convertloc):
        cnt = 0
        if convertloc in range(486):
            return convertloc
        elif convertloc < 0:
            while True:
                if cnt % 2 == 0 and cnt * -485 - convertloc in range(486):
                    return cnt * -485 - convertloc

                elif cnt % 2 == 1 and (cnt + 1) * 485 + convertloc in range(486):
                    return (cnt + 1) * 485 + convertloc
                cnt += 1
        elif convertloc > 485:

```

```

while True:
    if cnt % 2 == 0 and (cnt + 2) * 485 - convertloc in range(486):
        return (cnt + 2) * 485 - convertloc
    elif cnt % 2 == 1 and (cnt + 1) * -485 + convertloc in range(486):
        return (cnt + 1) * -485 + convertloc
    cnt += 1

##### 회귀분석식을 이용해 공이 떨어질 가상의 위치 예측하는 메소드 #####
def prediction(self, input, weight):
    return weight[0] * input[0] + weight[1] * input[1] + weight[2] * input[2] + weight[3] * input[3]

##### 공이 떨어질 위치로 패들을 움직이는 메소드 #####
def predict_move(self, convertloc):
    loc = self.convertendloc(convertloc)
    pos = self.canvas.coords(self.id)
    if pos[0]+40 < loc-5 and pos[2]-40 > loc+10:
        self.x = 0
        print('stop')
    else:
        if pos[2]-40 < loc+10:
            self.x = 3
            print('+3')
        elif pos[0]+40 > loc-5:
            self.x = -3
            print('-3')
    return self.x, 'loc', loc, 'pos', (pos[0], pos[2])

def move(self, x, y):
    self.x = x

##### 경사감소법 및 회귀분석 머신러닝 #####

class machine_learning():

    ##### 비용함수 메소드 #####
    @staticmethod
    def Loss(x, y, weight):
        loss = np.sum((x.dot(weight) - y.reshape(len(y),1)) ** 2) / (2 * len(x))
        print('loss', loss)
        return loss

    ##### 경사감소법 및 회귀분석 가중치 계산 메소드 #####
    @staticmethod
    def gradient_descent(x, alpha=0.00001, descent_cnt=1):
        X = x[:, 0:4]
        Y = x[:, 4]
        M = len(x)
        minloss = 10 ** 20

        WEIGHT = np.zeros((4,1)) # 초기 weight
        loss_history = np.zeros((descent_cnt, 1))

```

```

for cnt in range(descent_cnt):
    predictions = X.dot(WEIGHT).flatten()
    errors_x1 = (predictions - Y) * X[:, 0]
    errors_x2 = (predictions - Y) * X[:, 1]
    errors_x3 = (predictions - Y) * X[:, 2]
    errors_w0 = (predictions - Y) * X[:, 3]

    WEIGHT_backup = deepcopy(WEIGHT)
    # beta = theta - alpha * (X.T.dot(X.dot(beta)-y)/m)
    WEIGHT[0][0] = WEIGHT[0][0] - alpha * (1.0 / M) * errors_x1.sum()
    WEIGHT[1][0] = WEIGHT[1][0] - alpha * (1.0 / M) * errors_x2.sum()
    WEIGHT[2][0] = WEIGHT[2][0] - alpha * (1.0 / M) * errors_x3.sum()
    WEIGHT[3][0] = WEIGHT[3][0] - alpha * (1.0 / M) * errors_w0.sum()

    loss_history[cnt, 0] = machine_learning.Loss(X, Y, WEIGHT)

##### BOLD DRIVER 방법 #####
if minloss >= loss_history[cnt,0]:
    minloss = loss_history[cnt,0]
    alpha *= 1.1
elif minloss < loss_history[cnt,0]:
    alpha *= 0.5
    WEIGHT = WEIGHT_backup
return WEIGHT, loss_history

##### 세이브 로드 관련 클래스 #####

class SaveLoad():
    @staticmethod
    def saveCSV(ballloc, weightloc):
        try:
            if weightloc != None:
                f = open((weightloc), 'wt')
                w = csv.writer(f, delimiter=',', lineterminator='\n')

                for key in machine_learning.gradient_descent(np.array(ball_loc_save), learning_rate,
training_cnt)[0]:
                    w.writerow(key)
                f.close()
                print('weight saved')
            if ballloc != None:
                f = open((ballloc), 'wt')
                w = csv.writer(f, delimiter=',', lineterminator='\n')

                for key in ball_loc_save:
                    w.writerow(key)
                f.close()
                print('ball saved')

        except FileNotFoundError and TypeError:
            print('No Save')

    @staticmethod
    def loadCSV(ballloc, weightloc = None):
        try:
            if weightloc == None :

```

```

        pingpong = [data for data in csv.reader(open(ballloc, 'r'))]
        for pp in range(len(pingpong)):
            for p in range(5):
                pingpong[pp][p] = float(pingpong[pp][p])
            pingpong = np.array(pingpong)
            return machine_learning.gradient_descent(pingpong, learning_rate, training_cnt)[0]
    else :
        weight = [data for data in csv.reader(open(weightloc, 'r'))]
        return np.array([weight[-4], weight[-3], weight[-2], weight[-1]], dtype=float)

except FileNotFoundError :
    print('파일 로드 위치를 지정해주세요')

if __name__ == '__main__':
    ##### 머신러닝 위한 시뮬레이션용 #####
    tk = Tk() # tk 를 인스턴스화 한다.
    tk.title("Game") # tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
    tk.resizable(0, 0) # 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
    tk.wm_attributes("-topmost", 1) # 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것
    을 tkinter 에게 알려준다.
    canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
    canvas.configure(background='black')

    canvas.pack() # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라고
    캔버스에에 말해준다.

    tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려
    주는것이다.
    paddle = Paddle(canvas, 'black')
    ball = Ball(canvas, paddle, 'black', save=True)

    for i in range(10000):
        if ball.hit_bottom == False:
            ball.draw()
            paddle.move(paddle.x, 0)
            paddle.draw()

    ball_loc_save = []
    for idx_start in range(0, len(ball.ball_start)-1):
        try:
            ball_loc_save.append(ball.ball_start[idx_start]+[ball.ball_end[idx_start+1]])
        except IndexError:
            continue

    ##### 파일 세이브 #####
    SaveLoad.saveCSV(save_ballloc, save_weightloc)

    ##### 파일 로드 #####
    weight = SaveLoad.loadCSV(load_ballloc, load_weightloc)

    ##### 머신러닝 배운 후 플레이 #####
    paddle = Paddle(canvas, 'white')

```

```
while True:
    if ball.hit_bottom == False:
        ball.draw()
    try:
        convertloc = int(paddle.prediction(ball.ball_start[-1], weight)[0])
        print('prediction', paddle.predict_move(convertloc))
        paddle.move(paddle.x, 0)
    except IndexError:
        #paddle.move(random.choice([-3, 3]), 0) # 맨처음에 랜덤으로 두게 하려면 활성화
        paddle.move(ball.x,0) # 맨처음에 공을 따라가게 하려면 활성화
    paddle.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)
```

```
from math import *
import random
import copy
import time
```

```
class TicTacToe:  
    def __init__(self, state=None):  
        if state is None:  
            state = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
                    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
                    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
            self.playerJustMoved = 2  
            self.state = state  
  
    def Clone(self):
```

```

state = TicTacToe()
state.state = self.state[:]
state.playerJustMoved = self.playerJustMoved
return state

def DoMove(self, move):
    assert int(move) >= 0 and int(move) <= 80 and self.state[move] == 0
    self.playerJustMoved = 3 - self.playerJustMoved
    self.state[move] = self.playerJustMoved

def GetMoves(self):
    if self.checkState() != 0:
        return []
    else:
        moves = []
        for i in range(81):
            if self.state[i] == 0:
                moves.append(i)
        return moves

def GetResult(self, playerjm):
    result = self.checkState()
    assert result != 0
    if result == -1:
        return 0.5

    elif result == playerjm:
        return 1.0
    else:
        return 0.0

def checkState(self):
    for (x, y, z, p, q) in [(0, 1, 2, 3, 4), (9, 10, 11, 12, 13), (18, 19, 20, 21, 22), (27, 28, 29, 30, 31),
67),
        (36, 37, 38, 39, 40), (45, 46, 47, 48, 49), (54, 55, 56, 57, 58), (63, 64, 65, 66,
        (1, 2, 3, 4, 5), (10, 11, 12, 13, 14), (19, 20, 21, 22, 23), (28, 29, 30, 31, 32),
68),
        (37, 38, 39, 40, 41), (46, 47, 48, 49, 50), (55, 56, 57, 58, 59), (64, 65, 66, 67,
        (2, 3, 4, 5, 6), (11, 12, 13, 14, 15), (20, 21, 22, 23, 24), (29, 30, 31, 32, 33),
69),
        (38, 39, 40, 41, 42), (47, 48, 49, 50, 51), (56, 57, 58, 59, 60), (65, 66, 67, 68,
        (3, 4, 5, 6, 7), (12, 13, 14, 15, 16), (21, 22, 23, 24, 25), (30, 31, 32, 33, 34),
70),
        (39, 40, 41, 42, 43), (48, 49, 50, 51, 52), (57, 58, 59, 60, 61), (66, 67, 68, 69,
        (4, 5, 6, 7, 8), (13, 14, 15, 16, 17), (22, 23, 24, 25, 26), (31, 32, 33, 34, 35),
71),
        (40, 41, 42, 43, 44), (49, 50, 51, 52, 53), (58, 59, 60, 61, 62), (67, 68, 69, 70,
        (0, 9, 18, 27, 36), (1, 10, 19, 28, 37), (2, 11, 20, 29, 38), (3, 12, 21, 30, 39),
        (4, 13, 22, 31, 40), (5, 14, 23, 32, 41), (6, 15, 24, 33, 42), (7, 16, 25, 34, 43),
        (8, 17, 26, 35, 44), (9, 18, 27, 36, 45), (10, 19, 28, 37, 46), (11, 20, 29, 38, 47),
        (12, 21, 30, 39, 48), (13, 22, 31, 40, 49), (14, 23, 32, 41, 50), (15, 24, 33, 42,
51),
        (16, 25, 34, 43, 52), (17, 26, 35, 44, 53), (18, 27, 36, 45, 54), (19, 28, 37, 46,
55),
        (20, 29, 38, 47, 56), (21, 30, 39, 48, 57), (22, 31, 40, 49, 58), (23, 32, 41, 50,
59),
        (24, 33, 42, 51, 60), (25, 34, 43, 52, 61), (26, 35, 44, 53, 62), (27, 36, 45, 54,

```


63),
 67),
 71),
 75),
 79),
 58),
 62),
 70),
 78),
 45),
 45),
 53),
 57),
 57),
 65),
 65),
 73),
 73),
 30),
 34),
 40),
 46),
 50),
 56),
 60),
 66),

(28, 37, 46, 55, 64), (29, 38, 47, 56, 65), (30, 39, 48, 57, 66), (31, 40, 49, 58,
 (32, 41, 50, 59, 68), (33, 42, 51, 60, 69), (34, 43, 52, 61, 70), (35, 44, 53, 62,
 (36, 45, 54, 63, 72), (37, 46, 55, 64, 73), (38, 47, 56, 65, 74), (39, 48, 57, 66,
 (40, 49, 58, 67, 76), (41, 50, 59, 68, 77), (42, 51, 60, 69, 78), (43, 52, 61, 70,
 (44, 53, 62, 71, 80), (0, 10, 20, 30, 40), (1, 11, 21, 31, 41), (2, 12, 22, 32, 42),
 (3, 13, 23, 33, 43), (4, 14, 24, 34, 44), (9, 19, 29, 39, 49), (10, 20, 30, 40, 50),
 (11, 21, 31, 41, 51), (12, 22, 32, 42, 52), (13, 23, 33, 43, 53), (18, 28, 38, 48,
 (19, 29, 39, 49, 59), (20, 30, 40, 50, 60), (21, 31, 41, 51, 61), (22, 32, 42, 52,
 (27, 37, 47, 57, 67), (28, 38, 48, 58, 68), (29, 39, 49, 59, 69), (30, 40, 50, 60,
 (31, 41, 51, 61, 71), (36, 46, 56, 66, 76), (37, 47, 57, 67, 77), (38, 48, 58, 68,
 (39, 49, 59, 69, 79), (40, 50, 60, 70, 80), (4, 12, 20, 28, 36), (5, 13, 21, 29, 37),
 (6, 14, 22, 30, 38), (7, 15, 23, 31, 39), (8, 16, 24, 32, 40), (5, 13, 21, 29, 37),
 (10, 18, 26, 34, 42), (11, 19, 27, 35, 43), (12, 20, 28, 36, 44), (13, 21, 29, 37,
 (14, 22, 30, 38, 46), (11, 19, 27, 35, 43), (12, 20, 28, 36, 44), (13, 21, 29, 37,
 (18, 26, 34, 42, 50), (19, 27, 35, 43, 51), (20, 28, 36, 44, 52), (21, 29, 37, 45,
 (22, 30, 38, 46, 54), (23, 31, 39, 47, 55), (24, 32, 40, 48, 56), (25, 33, 41, 49,
 (26, 34, 42, 50, 58), (23, 31, 39, 47, 55), (24, 32, 40, 48, 56), (25, 33, 41, 49,
 (30, 38, 46, 54, 62), (31, 39, 47, 55, 63), (32, 40, 48, 56, 64), (33, 41, 49, 57,
 (34, 42, 50, 58, 66), (31, 39, 47, 55, 63), (32, 40, 48, 56, 64), (33, 41, 49, 57,
 (38, 46, 54, 62, 70), (39, 47, 55, 63, 71), (40, 48, 56, 64, 72), (41, 49, 57, 65,
 (42, 50, 58, 66, 74), (39, 47, 55, 63, 71), (40, 48, 56, 64, 72), (41, 49, 57, 65,
 (43, 51, 59, 67, 75), (44, 52, 60, 68, 76), (1, 9, 10, 11, 19), (2, 10, 11, 12, 20),
 (3, 11, 12, 13, 21), (4, 12, 13, 14, 22), (5, 13, 14, 15, 23), (6, 14, 15, 16, 24),
 (7, 15, 16, 17, 25), (10, 18, 19, 20, 28), (11, 19, 20, 21, 29), (12, 20, 21, 22,
 (13, 21, 22, 23, 31), (14, 22, 23, 24, 32), (15, 23, 24, 25, 33), (16, 24, 25, 26,
 (19, 27, 28, 29, 37), (20, 28, 29, 30, 38), (21, 29, 30, 31, 39), (22, 30, 31, 32,
 (23, 31, 32, 33, 41), (24, 32, 33, 34, 42), (25, 33, 34, 35, 43), (28, 36, 37, 38,
 (29, 37, 38, 39, 47), (30, 38, 39, 40, 48), (31, 39, 40, 41, 49), (32, 40, 41, 42,
 (33, 41, 42, 43, 51), (34, 42, 43, 44, 52), (37, 45, 46, 47, 55), (38, 46, 47, 48,
 (39, 47, 48, 49, 57), (40, 48, 49, 50, 58), (41, 49, 50, 51, 59), (42, 50, 51, 52,
 (43, 51, 52, 53, 61), (46, 54, 55, 56, 64), (47, 55, 56, 57, 65), (48, 56, 57, 58,

```

(49, 57, 58, 59, 67), (50, 58, 59, 60, 68), (51, 59, 60, 61, 69), (52, 60, 61, 62,
70),
(55, 63, 64, 65, 73), (56, 64, 65, 66, 74), (57, 65, 66, 67, 75), (58, 66, 67, 68,
76),
(59, 67, 68, 69, 77), (60, 68, 69, 70, 78), (61, 69, 70, 71, 79)]:
    if self.state[x] == self.state[y] == self.state[z] == self.state[p] == self.state[q]:
        if self.state[x] == 1:
            return 1
        elif self.state[x] == 2:
            return 2

    if [i for i in range(81) if self.state[i] == 0] == []:
        return -1
    return 0

def __repr__(self):
    cells = []
    for i in range(81):
        cells.append(NAMES[self.state[i]].center(6))
    return BOARD_FORMAT.format(*cells)

class Node:
    def __init__(self, move=None, parent=None, state=None):
        self.move = move
        self.parentNode = parent
        self.childNodes = []
        self.wins = 0
        self.visits = 0
        self.untriedMoves = state.GetMoves()
        self.playerJustMoved = state.playerJustMoved

    def UCTSelectChild(self):
        s = sorted(self.childNodes, key=lambda c: c.wins / c.visits + sqrt(2 * log(self.visits) /
c.visits))
        return s[-1]

    def AddChild(self, m, s):
        n = Node(move=m, parent=self, state=copy.deepcopy(s))
        self.untriedMoves.remove(m)
        self.childNodes.append(n)
        return n

    def Update(self, result):
        self.visits += 1
        self.wins += result

    def __repr__(self):
        return "[M" + str(self.move) + " W/V " + str(self.wins) + "/" + str(self.visits) + " U" + str(
            self.untriedMoves) + "]"

    def ChildrenToString(self):
        s = ""
        for c in self.childNodes:
            s += str(c) + "\n"
        return s

```

```

def UCT(rootstate, itermx):
    rootnode = Node(state=rootstate)
    sTime = time.time()
    for i in range(itermx):
        node = rootnode
        state = copy.deepcopy(rootstate)

        # selection
        while node.untriedMoves == [] and node.childNodes != []:
            node = node.UCTSelectChild()
            state.DoMove(node.move)

        # Expansion
        if node.untriedMoves != []:
            m = random.choice(node.untriedMoves)
            state.DoMove(m)
            node = node.AddChild(m, state)

        # simulation
        while state.GetMoves() != []:
            state.DoMove(random.choice(state.GetMoves()))

        # BackPropagation
        while node != None:
            node.Update(state.GetResult(node.playerJustMoved))
            node = node.parentNode

    print(rootnode.ChildrenToString())
    eTime = time.time()
    print('AI가 수를 계산하는데 걸린 시간 : ', eTime - sTime)
    s = sorted(rootnode.childNodes, key=lambda c: c.wins / c.visits)
    return sorted(s, key=lambda c: c.visits)[-1].move

```

```

def UCTPlayGame():
    state = TicTacToe()
    while state.GetMoves() != []:
        print(str(state))
        if state.playerJustMoved == 1:
            rootstate = copy.deepcopy(state)
            m = UCT(rootstate, itermx=10000)

        else:
            m1 = input("수를 둘 행(1-9)을 입력하세요 : ")
            m2 = input("수를 둘 열(1-9)을 입력하세요 : ")
            m = m1+m2
            m = int(m)
            m = m-10-int(m1)
            state.DoMove(m)

    if state.GetResult(state.playerJustMoved) == 1.0:
        print("Player " + str(state.playerJustMoved) + " Wins!!")

    elif state.GetResult(state.playerJustMoved) == 0.0:
        print("Player " + str(3 - state.playerJustMoved) + " Wins!!")

```

```
else:  
    print("Draw!!")
```

```
if __name__ == "__main__":  
    UCTPlayGame()
```

6. 볼게짜리 AI핑퐁 코드 -----> sarsa 구현

numpy 함수 명령어

2018년 8월 17일 금요일 오후 3:33

#numpy 함수 모음

함수명	설명	예시
np.mean(x)	#평균 값 n차원 배열&매트릭스 요소들 값의 평균 값을 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.mean(a) >> 8.6
np.median(x)	#중앙 값 n차원 배열&매트릭스 요소들 값의 중앙 값을 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.median(a) >> 6.0
np.max(x)	#최대 값 n차원 배열&매트릭스 요소들 값의 최대 값을 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.max(a) >> 21
np.min(x)	#최소 값 n차원 배열&매트릭스 요소들 값의 최소 값을 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.min(a) >> 1
np.std(x)	#표준편차 n차원 배열&매트릭스 요소들 값의 표준편차를 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.std(a) >>6.437390775772433
np.var(x)	#분산 n차원 배열&매트릭스 요소들 값의 표준편차를 리턴	a = np.array([1,2,4,5,5,7,10,13,18,21]) np.var(a) >>41.44
x.flatten()	#1차원 배열로 변환 n차원 배열 -> 1차원 배열로 변환	a=np.array([[1,2,3],[4,5,6]]) b=a.flatten() print(b) >>[1 2 3 4 5 6]
a.astype(타입)	#배열의 자료형을 변환 **타입 - np.int : 정수형 - np.float : 실수형 - np.str : 문자형 등...	a=np.array([1,2,3]) b=a.astype(np.str) c=a.astype(np.bool) print(b) print(c) >>['1' '2' '3'] >>[True True True]
np.exp(x)	#자연상수 e의 x승 반환	print(np.exp(1)) >>2.718281828459045
np.arange(x)	#범위 반환 np.arange([start],end,[증가값])	a=np.arange(1,10,2) print(a) >>[1 3 5 7 9]
np.dim(x)	#배열의 차원수 반환	a=np.array([[1,2,3],[4,5,6]]) print(np.ndim(a)) >>2
x.shape	#배열의 행,열 반환 배열의 행, 열을 튜플로 반환한다. - 2차원 배열일 때, (3,2) 를 반환 했다면 3x2 배열이란 뜻.	a=np.array([[1,2,3],[4,5,6]]) print(a.shape) >>(2,3) a=np.array([[[1,2,3,4],[5,6,7,8]],[[11,22,33,44],[55,66,77,88]],[[5,3,2,9],[4,5,3,0]]]) [[[1 2 3 4] [5 6 7 8]] [[11 22 33 44] [55 66 77 88]] [[5 3 2 9]]

		<pre>[4 5 3 0]] print(a.shape) (3, 2, 4)</pre>
np.dot(x,y)	#두 행렬간의 곱(내적) **행렬의 형상이 맞지 않다면 오류발생 ! ex. (3x2)행렬 x (4x2)행렬 => 에러발생	<pre>a=np.array([[1,2],[3,4]]) b=np.array([[5,6],[7,8]]) c=np.dot(a,b) print(c) >> [[19 22] [43 50]]</pre>
np.maximum(x,y)	# 두 값 중 큰 값을 반환	<pre>np.maximum(0,4) >>4</pre>
np.argmax(x)	#배열의 요소 중 가장 큰 값의 인덱스 반환	<pre>a=np.array([3,4,1,2]) print(np.argmax(a))</pre>
np.argmin(x)	#배열의 요소 중 가장 작은 값의 인덱스 반환	<pre>a=np.array([3,4,1,2]) print(np.argmin(a))</pre>
np.zeros_like(x)	#x와 형상이 같고 그 원소가 0으로 채워진 배열을 만들어 반환	
조건주기	# 배열에서 특정 값 이상인 값만 출력하기	<pre>a=np.array([[51,55],[14,19],[0,4]]) print(a[a>15]) >> 51, 55, 19</pre>



딕셔너리 명령어

2018년 8월 17일 금요일 오후 3:41

딕셔너리를 자유자재로 사용하기 위해 딕셔너리가 자체적으로 가지고 있는 관련 함수들을 사용해 보도록 하자.

함수명	설명	예시
keys	#Key 리스트 만들기 a.keys()는 딕셔너리 a의 Key만을 모아서 dict_keys라는 객체를 리턴한다.	<pre>>>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} >>> a.keys() dict_keys(['name', 'phone', 'birth']) >>> for k in a.keys(): ... print(k) ... phone birth Name dict_keys 객체를 리스트로 변환하려면 다음과 같이 하면 된다. >>> list(a.keys()) ['phone', 'birth', 'name']</pre>
values	#Value 리스트 만들기 sort 함수는 리스트의 요소를 순서대로 정렬	<pre>>>> a.values() dict_values(['pey', '0119993323', '1118'])</pre>
items	#Key, Value 쌍 얻기 items 함수는 key와 value의 쌍을 튜플로 묶은 값을 dict_items 객체로 돌려준다.	<pre>>>> a.items() dict_items([('name', 'pey'), ('phone', '0119993323'), ('birth', '1118')])</pre>
clear	#Key: Value 쌍 모두 지우기 clear() 함수는 딕셔너리 안의 모든 요소를 삭제한다.	<pre>>>> a.clear() >>> a {}</pre>
get	#Key로 Value얻기 get(x) 함수는 x라는 key에 대응되는 value를 돌려준다. a['nokey']처럼 a 딕셔너리에 없는 키로 값을 가져오려고 할 경우 a['nokey']는 Key 오류를 발생시키고 a.get('nokey')는 None을 리턴한다 는 차이가 있다. *딕셔너리 안에 찾으려는 key 값이 없을 경우 미리 정해 둔 디폴트 값을 대신 가져오게 하고 싶을 때에는 get(x, '디폴트 값')을 사용하면 편리하다.	<pre>>>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} >>> a.get('name') 'pey' >>> a.get('phone') '0119993323'</pre>
in	#해당 Key가 딕셔너리 안에 있는지 조사하기 'name'이라는 문자열은 a 딕셔너리의 key 중 하나이다. 따라서 'name' in a를 호출하면 참(True)을 리턴한다. 반대로 'email' 은 a 딕셔너리 안에 존재하지 않는 key이므로 거짓(False)을 리턴	<pre>>>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'} >>> 'name' in a True >>> 'email' in a False</pre>

		False
defaultdict()	# 값을 주지 않아도 딕셔너리 키값을 만들어 놓을 수 있음.	<pre> from collections import defaultdict artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대', '소녀시대'] music = ['Yesterday', 'let it be', '소원을 말해봐', 'Gee', '다만세'] box=defaultdict(list) print(box) lst=[] for i,mu in enumerate(artist): box[mu].append(music[i]) # print(box) print(box) **결과 defaultdict(<class 'list'>, {}) defaultdict(<class 'list'>, {'Beatles': ['Yesterday', 'let it be'], '소녀시대': ['소원을 말해봐', 'Gee', '다만세']}) </pre>

딕셔너리에 요소 추가하기

```

s = {}
s['kk']='vv'
print(s)

```


리스트 함수 명령어

2018년 8월 17일 금요일 오후 3:47

■ 리스트 함수

함수명	설명	예시
append	#리스트에 요소 추가 append(x)는 리스트의 맨 마지막에 x를 추가시키는 함수	<pre>>>> a = [1, 2, 3] >>> a.append(4) >>> a [1, 2, 3, 4]</pre>
sort	#리스트 정렬 sort 함수는 리스트의 요소를 순서대로 정렬	<pre>>>> a = [1, 4, 3, 2] >>> a.sort() >>> a [1, 2, 3, 4]</pre>
sorted	#리스트 정렬 (특정 요소값으로) sorted는 sort와 다르게 정렬된 값을 반환 받고, 중첩리스트에서 특정 요소의 값을 기준으로 정렬 가능. sorted(iterable) 함수는 입력값을 정렬한 후 그 결과를 리스트로 리턴하는 함수	<pre>>>> aa=list(['b',2],['c',10],['a',7]) >>> sorted(aa, key=lambda x:x[1]) [['b', 2], ['a', 7], ['c', 10]] >>> sorted(aa, key=lambda x:x[0]) [['a', 7], ['b', 2], ['c', 10]] >>> sorted([3, 1, 2]) [1, 2, 3] >>> sorted(['a', 'c', 'b']) ['a', 'b', 'c'] >>> sorted("zero") ['e', 'o', 'r', 'z'] >>> sorted((3, 2, 1)) [1, 2, 3]</pre>
reverse	#리스트 뒤집기 reverse 함수는 리스트를 역순으로 뒤집어 준다. 이때 리스트 요소들을 순서대로 정렬한 다음 다시 역순으로 정렬하는 것이 아니라 그저 현재의 리스트를 그대로 거꾸로 뒤집을 뿐이다.	<pre>>>> a = ['a', 'c', 'b'] >>> a.reverse() >>> a ['b', 'c', 'a']</pre>
index	#위치 반환 index(x) 함수는 리스트에 x라는 값이 있으면 x의 위치값을 리턴. 리스트에 존재하지 않으면 값 오류(ValueError)가 발생한다.	<pre>>>> a = [1,2,3] >>> a.index(3) 2 >>> a.index(1) 0</pre>
insert	#리스트에 요소 삽입 insert(a, b)는 리스트의 a번째 위치에 b를 삽입하는 함수	<pre>>>> a = [1, 2, 3] >>> a.insert(0, 4) [4, 1, 2, 3]</pre>
remove	#리스트 요소 제거 remove(x)는 리스트에서 첫 번째로 나오는 x를 삭제하는 함수	<pre>>>> a = [1, 2, 3, 1, 2, 3] >>> a.remove(3) [1, 2, 1, 2, 3]</pre>
pop	#리스트 요소 끄집어내기 pop()은 리스트의 맨 마지막 요소를 돌려 주고 그 요소는 삭제하는 함수	<pre>>>> a = [1,2,3] >>> a.pop() 1,2 >>> a = [1,2,3]</pre>

		<pre>>>> idx = 1 >>> a.pop(idx) 1,3</pre>
count	#리스트에 포함된 요소 x의 개수 세기 count(x)는 리스트 내에 x가 몇 개 있는지 조사하여 그 개수를 돌려주는 함수	<pre>>>> a = [1,2,3,1] >>> a.count(1) 2</pre>
extend	#리스트 확장 extend(x)에서 x에는 리스트만 올 수 있으며 원래의 a 리스트에 x 리스트를 더하게 된다.	<pre>>>> a = [1,2,3] >>> a.extend([4,5]) >>> a [1, 2, 3, 4, 5] >>> b = [6, 7] >>> a.extend(b) >>> a [1, 2, 3, 4, 5, 6, 7]</pre>
enumerate	#인덱스, 값을 반환 enumerate는 "열거하다"라는 뜻이다. 이 함수는 순서가 있는 자료형(리스트, 튜플, 문자열)을 입력으로 받아 인덱스 값을 포함하는 enumerate 객체를 리턴 ※ 보통 enumerate 함수는 예제처럼 for문과 함께 자주 사용된다.	<pre>>>> for i, name in enumerate(['body', 'foo', 'bar']): ... print(i, name) ... 0 body 1 foo 2 bar</pre>
len	#요소 전체의 개수 반환 len(s)은 입력값 s의 길이(요소의 전체 개수)를 리턴하는 함수	<pre>>>> len("python") 6 >>> len([1,2,3]) 3 >>> len((1, 'a')) 2</pre>
list	#반복 가능한 자료형을 리스트형으로 반환 list(s)는 반복 가능한 자료형 s를 입력받아 리스트로 만들어 리턴하는 함수	<pre>>>> list("python") ['p', 'y', 't', 'h', 'o', 'n'] >>> list((1,2,3)) [1, 2, 3]</pre>
map	#반복 가능한 자료형의 각 요소를 특정 함수로 수행한 결과를 반환 map(f, iterable)은 함수(f)와 반복 가능한(iterable) 자료형을 입력으로 받는다. map은 입력받은 자료형의 각 요소가 함수 f에 의해 수행된 결과를 묶어서 리턴하는 함수	<pre>>>> def two_times(x): ... return x*2 ... >>> list(map(two_times, [1, 2, 3, 4])) [2, 4, 6, 8]</pre>
zip	#동일한 인덱스의 값끼리 묶어줌 zip(iterable*)은 동일한 개수로 이루어진 자료형을 묶어 주는 역할을 하는 함수	<pre>>>> list(zip([1, 2, 3], [4, 5, 6])) [(1, 4), (2, 5), (3, 6)] >>> list(zip([1, 2, 3], [4, 5, 6], [7, 8, 9])) [(1, 4, 7), (2, 5, 8), (3, 6, 9)] >>> list(zip("abc", "def")) [('a', 'd'), ('b', 'e'), ('c', 'f')]</pre>
::	#인덱싱할때 특정 간격만 뽑아올 수 있음 1::2 --> 1번째 부터 2칸씩 건너뛰 0::3 ----> 0번째 부터 3칸씩 ..	<pre>a=['태양','수성','금성','지구','화성','목성','토성','천왕성','해왕성'] print(a[1::2]) # 짝수번째 인덱스만.. ['수성', '지구', '목성', '천왕성'] print(a[2::-1]) ['금성', '수성', '태양']</pre>

■ 리스트의 요소를 제거하는 3가지 방법

함수명	설명	예시
remove	리스트의 특정 요소값을 삭제	a.remove(x) - x는 a리스트의 요소값
pop	리스트의 인덱스 위치 값을 삭제	a.pop(x) - x는 a리스트의 인덱스
del	리스트의 인덱스 위치 값을 삭제	del a[x] - x는 a리스트의 인덱스

예제1.

```
emp_list=[]  
emp_list.append('a')  
print(emp_list)
```

```
emp_list.append('b')  
print(emp_list)
```

```
['a']  
['a', 'b']
```

문자형 함수 명령어

2018년 8월 17일 금요일 오후 3:47

함수명	설명	예시
str	#문자형으로 변환 다른 타입의 변수를 문자형으로 바꿔준다.	<pre>>>> a = 3 >>> str(a) '3'</pre>
count	#문자 개수 세기 문자열 중 특정 문자의 개수를 반환	<pre>>>> a = "hobby" >>> a.count('b') 2</pre>
find	#위치 찾기 문자열 중 특정 문자가 처음 나온 위치를 반환한다. 만약 찾는 문자나 문자열이 존재하지 않는다면 -1을 반환한다. ※ 파이썬은 숫자를 0부터 세기 때문에 b의 위치는 15가 아닌 14가 된다.	<pre>>>> a = "Python is the best choice" >>> a.find('b') 14 >>> a.find('k') -1</pre>
index	#위치 찾기 문자열 중 특정 문자가 맨 처음으로 나온 위치를 반환한다. 만약 찾는 문자나 문자열이 존재하지 않는다면 오류를 발생시킨다. 앞의 find 함수와 다른 점은 문자열 안에 존재하지 않는 문자를 찾으면 오류가 발생한다는 점이다.	<pre>>>> a = "Life is too short" >>> a.index('t') 8 >>> a.index('k') Traceback (most recent call last): File "<stdin>", line 1, in <module> ValueError: substring not found</pre>
join	#문자열 삽입 문자열의 각각 문자 사이에 특정 값을 삽입한다.	<pre>>>> a = "," >>> a.join('abcd') 'a,b,c,d'</pre>
upper	# 대문자로 바꾸기 소문자를 대문자로 바꾸어 준다.	<pre>>>> a = "hi" >>> a.upper() 'HI'</pre>
lower	#소문자로 바꾸기 대문자를 소문자로 바꾸어 준다.	<pre>>>> a = "HI" >>> a.lower() 'hi'</pre>
lstrip	#왼쪽 공백 지우기 문자열 중 가장 왼쪽에 있는 한 칸 이상의 연속된 공백들을 모두 지운다. lstrip에서 l은 left를 의미한다.	<pre>>>> a = " hi " >>> a.lstrip() 'hi '</pre>
rstrip	#오른쪽 공백 지우기 문자열 중 가장 오른쪽에 있는 한 칸 이상의 연속된 공백들을 모두 지운다. rstrip에서 r은 right를 의미한다.	<pre>>>> a = " hi " >>> a.rstrip() ' hi'</pre>
strip	#양쪽 공백 지우기 문자열 양쪽에 있는 한 칸 이상의 연속된 공백들을 모두 지운다.	<pre>>>> a = " hi " >>> a.strip() 'hi'</pre>
replace	#문자열 바꾸기 replace(바뀌게 될 문자열, 바꿀 문자열)처럼 사용해서 문자열 내의 특정한 값을 다른 값으로 치환해 준다.	<pre>>>> a = "Life is too short" >>> a.replace("Life", "Your leg") 'Your leg is too short'</pre>

split	#문자열 나누기 <code>a.split()</code> 처럼 괄호 안에 아무런 값도 넣어 주지 않으면 공백(스페이스, 탭, 엔터등)을 기준으로 문자열을 나누어 준다. 만약 <code>a.split(':')</code> 처럼 괄호 안에 특정한 값이 있을 경우에는 괄호 안의 값을 구분자로 해서 문자열을 나누어 준다. 이렇게 나눈 값은 리스트에 하나씩 들어가게 된다.	<pre>>>> a = "Life is too short" >>> a.split() ['Life', 'is', 'too', 'short'] >>> a = "a:b:c:d" >>> a.split(':') ['a', 'b', 'c', 'd']</pre>
capitalize	# 문자의 첫 글자만 대문자로 첫번째 글자만 대문자로 바꾸고 나머지는 소문자로 바꾼다.	<pre>>>> print("ddDDddDD".capitalize()) Dddddddd</pre>
chr	#아스키 코드에 해당하는 문자 반환 <code>chr(i)</code> 는 아스키(ASCII) 코드값을 입력 받아 그 코드에 해당하는 문자를 출력하는 함수 ※ 아스키 코드란 0에서 127 사이의 숫자들을 각각 하나의 문자 또는 기호에 대응시켜 놓은 것이다.	<pre>>>> chr(97) 'a' >>> chr(48) '0'</pre>
ord	#문자에 해당하는 아스키코드 반환 <code>ord(c)</code> 는 문자의 아스키 코드값을 리턴하는 함수	<pre>>>> ord('a') 97 >>> ord('0') 48</pre>
eval	#문자열을 명령어로 보고 그 결과를 반환 <code>eval(expression)</code> 은 실행 가능한 문자열(<code>1+2</code> , <code>'hi' + 'a'</code> 같은 것)을 입력으로 받아 문자열을 실행한 결과값을 리턴하는 함수 *보통 <code>eval</code> 은 입력받은 문자열로 파이썬 함수나 클래스를 동적으로 실행하고 싶은 경우에 사용된다.	<pre>>>> eval('1+2') 3 >>> eval("'hi' + 'a'") 'hia' >>> eval('divmod(4, 3)') (1, 1)</pre>
input	#문자열을 입력받음 <code>input([prompt])</code> 은 사용자 입력을 받는 함수 (※ 파이썬 2.7 버전의 경우 <code>input</code> 대신 <code>raw_input</code> 을 사용해야 한다.)	<pre>>>> a = input() hi >>> a 'hi' >>> b = input("Enter: ") Enter: hi</pre>
int	#숫자형 자료형으로 변환 <code>int(x)</code> 는 문자열 형태의 숫자나 소수점이 있는 숫자 등을 정수 형태로 리턴하는 함수 * <code>int(x, radix)</code> 는 <code>radix</code> 진수로 표현된 문자열 <code>x</code> 를 10진수로 변환하여 리턴한다.	<pre>>>> int('3') 3 >>> int(3.4) 3 >>> int('1A', 16) 26 16진수로 표현된 '1A'의 10진수 값은 위와 같이 구한다.</pre>
str	#문자형 자료형으로 반환 <code>str(object)</code> 은 문자열 형태로 객체를 변환하여 리턴하는 함수	<pre>>>> str(3) '3' >>> str('hi') 'hi' >>> str('hi'.upper()) 'HI'</pre>
encode	#문자열을 바이트 객체로 바꾸기	

##문자열 짝&홀수 출력

```
txt='aAbBcCdDeEfFgG'
ret=txt[1::2] #짝수만 출력
print(ret)
ret=txt[::2]  #홀수만 출력
print(ret)
```

**** 문자에서 특정 글자를 찾는 방법**

```
if 'a' in 'abcdf'    ---> true
```

예제 (1) : 문자열의 각 문자를 그 다음 문자로 변경하기

```
text = input('문장을 입력하세요: ')

ret = ''
for i in range(len(text)):
    if i != len(text)-1:
        ret += text[i+1]
    else:
        ret += text[0]

print(ret)
```

****결과**

```
문장을 입력하세요: 안녕하세요
녕하세요안
```

숫자형 함수 명령어

2018년 8월 17일 금요일 오후 3:48

함수명	설명	예시
abs	#절대값 반환 abs(x)는 어떤 숫자를 입력으로 받았을 때, 그 숫자의 절대값을 돌려주는 함수	<pre>>>> abs(3) 3 >>> abs(-3) 3</pre>
count	#문자 개수 세기 문자열 중 특정 문자의 개수를 반환	<pre>>>> a = "hobby" >>> a.count('b') 2</pre>
divmod	#나눈 몫, 나머지 값 반환 divmod(a, b)는 2개의 숫자를 입력으로 받는다. 그리고 a를 b로 나눈 몫과 나머지를 튜플 형태로 리턴하는 함수	<pre>>>> divmod(7, 3) (2, 1) >>> divmod(1.3, 0.2) (6.0, 0.099999999999999978)</pre>
hex	#16진수 반환 hex(x)는 정수값을 입력받아 16진수(hexadecimal)로 변환하여 리턴하는 함수	<pre>>>> hex(234) '0xea' >>> hex(3) '0x3'</pre>
max	#최대값 반환 max(iterable)은 인수로 반복 가능한 자료형을 입력받아 그 최대값을 리턴하는 함수	<pre>>>> max([1, 2, 3]) 3 >>> max("python") 'y'</pre>
min	#최소값 반환 min(iterable)은 max 함수와 반대로, 인수로 반복 가능한 자료형을 입력받아 그 최소값을 리턴하는 함수	<pre>>>> min([1, 2, 3]) 1 >>> min("python") 'h'</pre>
oct	#8진수 문자열로 반환 oct(x)는 정수 형태의 숫자를 8진수 문자열로 바꾸어 리턴하는 함수	<pre>>>> oct(34) '0o42' >>> oct(12345) '0o30071'</pre>
pow	#제공한 결과 반환 pow(x, y)는 x의 y 제곱한 결과값을 리턴하는 함수	<pre>>>> pow(2, 4) 16 >>> pow(3, 3) 27</pre>
range	#범위의 값을 반복가능한 객체로 반환 range([start,] stop [,step])는 for문과 함께 자주 사용되는 함수이다. 이 함수는 입력받은 숫자에 해당되는 범위의 값을 반복 가능한 객체로 만들어 리턴	<pre>*인자 1개일 경우 >>> list(range(5)) [0, 1, 2, 3, 4] *인자 2개일 경우 >>> list(range(5, 10)) [5, 6, 7, 8, 9] *인자 3개일 경우 >>> list(range(1, 10, 2)) [1, 3, 5, 7, 9] >>> list(range(0, -10, -1))</pre>

		[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]
round	#반올림 round(number[, ndigits]) 함수는 숫자를 입력받아 반올림 해 주는 함수	<pre>>>> round(4.6) 5 >>> round(4.2) 4</pre>