

목차

■파이썬 수업

■ 1. 대화식 모드로 프로그래밍 하기	
■ 2. 텍스트 에디터로 프로그래밍 하기	
■ 3. 변수명 만들기	문제 1
- type	
■ 4. 변수에 값대입하기	문제 2
- 객체 참조 변수 개수 확인	
■ 5. 주석 처리하기 (#)	
- 주석처리 방법	
- 라인유지하기	
- 여러행 주석처리	
■ 6. 자료형 개념 배우기	문제 5 ~ 6
- 리스트변수	문제 3 ~ 4
- csv 파일 로드	
■ 8. 들여쓰기 개념 배우기	문제 7 ~ 8
■ 9. if 문 개념 배우기 ① (if ~ else)	문제 9 ~ 11
■ 10. if 문 개념 배우기 ② (if ~ elif)	문제 12
■ 11. for 문 개념 배우기 ① (for)	문제 13 ~ 22
- range(연속숫자)	
- just	
- 누적루프	
■ 12. for 문 개념 배우기 ② (for~continue~break)	
- continue	문제 29 ~ 35
*문자열 포매팅 이해하기	
- break	문제 36
■ 13. for 문 개념 배우기 ③ (for~else)	
■ 14. while 문 개념 배우기(while~continue~break)	문제 37 ~ 42, 50
■ 15. None 개념 배우기	문제 43 ~ 44
- 파이썬 연산자 정리	
■ 16. 정수형 자료 이해하기	문제 45
■ 17. 실수형 자료 이해하기	문제 46
■ 19. 대입연산자 이해하기 (=)	문제 47 ~ 49
■ 20. 사칙 연산자 이해하기 (+, -, *, /, **)	문제 51 ~ 57
■ 21. 연산자 축약 이해하기 (+=, -=, *=, /=)	문제 58
■ 22. True 와 False 이해하기	문제 59 ~ 60
■ 23. 관계 연산자 이해하기 (==, !=, <, <=, >, >=)	문제 61 ~ 63
- pandas	
■ 24. 논리 연산자 이해하기 (and, or, not)	문제 64 ~ 65
- 기타 비교 연산자 비교	
- lambda	문제 66 ~ 72
■ 26. 시퀀스 자료형 이해하기	문제 73
■ 27. 시퀀스 자료 인덱싱 이해하기	문제 74 ~ 76
■ 28. 시퀀스 자료 슬라이싱 이해하기	문제 77 ~ 80
■ 29. 시퀀스 자료 연결하기 (+)	문제 81 ~ 82
■ 30. 시퀀스 자료 반복 이해하기 (*)	
■ 31. 시퀀스 자료 크기 이해하기 (len)	문제 83
■ 32. 문자열의 멤버체크 이해하기 (in)	문제 84 ~ 89
■ 33. 문자열 이해하기	문제 90 ~ 91
■ 34. 문자열 포매팅 이해하기	문제 93 ~ 98

-sleep

■ 35. 이스케이프 문자 이해하기

■ 36. 리스트 이해하기 ([])

문제 99 ~ 107

- 리스트변수의 유용한 메소드 함수

- set(정렬)

■ 오라클과 파이썬의 그룹함수 비교

문제 108 ~ 116

■ 37. 튜플 이해하기 (())

■ 38. 딕셔너리 자료형 이해하기 ({ })

문제 117 ~ 131

■ 39. (파이썬) 함수 이해하기 (def)

- 문자함수

문제 132 ~ 137

- 날짜함수

문제 139 ~ 152

- 파이썬 모듈

문제 146

- 일반함수

문제 153 ~ 154

- 몬테카를로 알고리즘

문제 155 ~ 159

■ 40. 함수 인자(매개변수) 이해하기

문제 160 ~ 161

■ 41. 지역변수와 전역변수 이해하기 (global)

문제 162

■ 42. 함수 리턴값 이해하기(return)

문제 163

■ 43. 파이썬 모듈 이해하기

문제 164

■ 44. 파이썬 패키지 이해하기

■ 45. 파이썬 모듈 임포트 하기 ① (import)

■ 46. 파이썬 모듈 임포트 이해하기 ② (from ~ import)

■ 47. 파이썬 모듈 임포트 이해하기 ③ (import ~ as)

■ 48. 파일 열고 닫기 (open, close)

-Image

■ 49. 클래스 이해하기 (class)

문제 168 ~ 172

■ 50. 클래스 멤버와 인스턴스 멤버 이해하기

문제 176

■ 51. 클래스 메소드 애해하기

■ 기초 통계에 필요한 함수들 생성

- 평균값, 중앙값, 최빈값, 분산, 표준편차, 범위, 사분위수 범위

문제 177 ~ 184

- 막대 그래프, 원형 그래프, plot 그래프, 라인 그래프, 사분위수 그래프

문제 185 ~ 189

■ 51. 클래스 메소드 이해하기 ②

문제 190 ~198

■ 52. 클래스 생성자

■ 53. 클래스 소멸자 이해하기

■ 54. 클래스 상속 이해하기

문제 199 ~ 202

- 다중상속

■ 55. 예외처리 이해하기 ① (try ~ except)

문제 203 ~ 207

■ 56. 예외처리 이해하기 ② (try ~ except ~ else)

■ 복수개의 except 절 사용하기

문제 208

■ 57. 예외처리 이해하기 ③ (try ~ except ~ finally)

문제 209

■ 58. 예외처리 이해하기 ④ (try ~ except Exception as e)

문제 210

■ 59. 사용자 정의 예외 처리

문제 211

■ 60. 사용자 입력받기 (input)

■ 61. 자료형 확인하기 (type)	
■ 62. 나눗셈에서 나머지만 구하기 (%)	문제 212
■ 63. 몫과 나머지 구하기 (divmod)	
■ 68. 반올림 수 구하기 (round)	문제 228 ~ 230
■ 69. 실수형 자료를 정수형 자료로 변환하기 (int)	문제 231
■ 70. 정수형 자료를 실수형으로 변환하기 (float)	
■ 71. 정수 리스트에서 소수만 걸러내기(filter)	문제 232 ~ 235
■ 75. 문자열에서 특정 위치의 문자 얻기	
■ 76. 문자열에서 지정한 구간의 문자열 얻기	
■ 77. 문자열에서 홀수 번째 문자만 추출하기	
■ 78. 문자열을 거꾸로 출력하기	
■ 파이썬의 리스트가 스택 구조임을 확인하시오 !	문제 246 ~ 247
■ 데이터 정제를 파이썬으로 하는 방법 (텍스트 마이닝 클래스)	
■ 79. 두 개의 문자열 합치기(+)	문제 248
■ 80. 문자열을 반복해서 새로운 문자열로 만들기	
■ 81. 문자열에서 특정 문자가 있는지 확인하기 (in)	문제 249 ~ 250
■ 82. 문자열에서 특정 문자열이 있는지 확인하기 (in)	문제 251 ~ 263
■ 83. 파이썬 Comprehension	
- list Comprehension	문제 264 ~ 265
- set Comprehension	문제 266
- Dictionary Comprehension	문제 267
■ 84. 문자열이 알파벳인지 검사하기(isalpha)	문제 269 ~ 270
■ 85. 문자열이 숫자인지 검사하기(isdigit)	
■ 86. 문자열에 특수문자가 있는지 검사하기	문제 272 ~ 273
■ 87. 문자열에서 대소문자 변환하기 (upper, lower)	
■ 88. 문자열에서 좌우 공백 제거하기 (lstrip, rstrip, strip) outline	
■ 89. 재귀 알고리즘 기본 (구구단, 별)	문제 274 ~ 280
■ 91. 문자열에 있는 문자 개수 구하기 (count)	문제 281
■ 92. 문자열에서 특정 문자(열) 위치 찾기(find)	
■ 93. 문자열을 특정 문자(열)로 분리하기 (split)	문제 283
■ 94. 문자열을 특정 문자(열)로 결합하기(join)	
■ 95. 문자열에서 특정 문자(열)을 다른 문자(열)로 바꾸기 (replace)	문제 284
■ 96. 문자열을 바이트 객체로 바꾸기 (encode)	
■ 97. 바이트 객체를 문자열로 바꾸기 (decode)	문제 285
■ 98. 버블정렬	
■ 99. 순차적인 정수 리스트 만들기 (range)	문제 288 ~ 289
■ 100. 리스트 요소를 무작위로 섞기 (shuffle)	문제 290
■ 100. 리스트에서 특정 위치의 요소 얻기	
■ copy 와 deepcopy 모듈의 차이	
■ 101. 리스트에서 특정 요소의 위치 구하기(index)	
■ 102. 리스트에서 특정위치의 요소 변경하기	

■ 103. 리스트에서 특정 구간에 있는 요소 추출하기	
■ 104. 리스트에서 짝수번째 요소만 추출하기	
■ 105. 리스트 요소 순서를 역순으로 만들기 ① (reverse)	
■ 106. 리스트 요소 순서를 역순으로 만들기 ②	
■ 107. 리스트 합치기 (+)	
■ 108. 리스트 반복하기 (*)	
■ 109. 리스트에 요소 추가하기 (append)	
■ 110. 리스트의 특정 위치에 요소 삽입하기 (insert)	
■ 111. 리스트의 특정위치의 요소 제거하기 (del)	문제 302
■ 112. 리스트에서 특정 요소 제거하기 (remove)	문제 303
■ 113. 리스트에서 특정 구간에 있는 모든 요소 제거하기 (remove)	
■ 114. 리스트에 있는 요소 개수 구하기 (len)	
■ 115. 리스트에서 특정 요소의 개수 구하기 (count)	
■ 116. 리스트 제거하기 (del)	
■ 117. 리스트 요소 정렬하기 ① (sort)	문제 306 ~ 307
■ 120. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기 (enumerate)	문제 308
■ 121. 리스트의 모든 요소의 합 구하기 (sum)	
■ 122. 리스트 요소가 모두 참인지 확인하기 (all, any)	
■ 123. 파이썬에서 (언더바)를 사용하는 경우	
■ 123. 사전에 요소 추가하기	문제 311 ~ 312
■ Defaultdict 이 무엇인지 ?	문제 313 ~ 316
■ 124. 사전의 특정 요소 값 변경하기	
■ 125. 사전의 특정 요소 제거하기 (del)	
■ 126. 사전의 모든 요소 제거하기 (clear)	
■ 127. 사전에서 키만 추출하기 (keys)	
■ 128. 사전에서 값만 추출하기 (values)	문제 321 ~ 325
■ 129. 사전 요소를 모두 추출하기 (items)	문제 326
■ 130. 사전 정렬하기(sorted)	문제 327 ~ 328
■ 131. 문자 코드 값 구하기(ord)	문제 329
■ 132. 코드값에 대응하는 문자 얻기(chr)	
■ 133. 문자열로 된 식을 실행하기 (eval)	문제 330
■ 134. 이름없는 한줄짜리 함수 만들기(lambda)	문제 331 ~ 332
■ 135. 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기 (map)	문제 333 ~ 334
■ 136. 텍스트 파일을 읽고 출력하기 (read)	
■ 137. 텍스트 파일을 한줄 씩 읽고 출력하기 (readlines)	문제 335 ~ 336
■ 139. 화면에서 사용자 입력을 받고 파일로 쓰기 (write)	문제 337
■ 140. 텍스트 파일에 한줄씩 쓰기 (writelines)	
■ 141. 텍스트 파일 복사하기 (read, write)	
■ 142. 바이너리 파일 복사하기 (read, write)	문제 338
■ 143. 파일을 열고 자동으로 닫기 (with ~ as)	문제 339
■ 144. 파일의 특정 부분만 복사하기 (seek, read, write)	
■ 145. 파일 크기 구하기 (ospathgetsize)	
■ 146. 파일 삭제하기 (osremove)	문제 340
■ 147. 파일 이름 바꾸기 (osrename)	

■ 148. 파일 위치 바꾸기	
■ 149. 디렉토리에 있는 파일목록 얻기 (os.listdir)	문제 341 ~ 346
■ 150. 워킹 디렉토리 확인하고 바꾸기	
■ 151. 디렉토리 생성하기 (os.mkdir)	
■ 152. 디렉토리 제거하기 (os.rmdir)	문제 346
■ 153. 하위 디렉토리 및 파일 전체 삭제하기 (shutil.rmtree)	
■ 154. 파일이 존재하는지 체크하기 (os.path.exists)	문제 349
■ 155. 파일인지 디렉토리인지 확인하기 (os.path.isfile, os.path.isdir)	문제 350
■ 156. 현재 시간을 년-월-일 시:분:초 로 출력하기 (location, strftime)	
■ 157. 올해 경과된 날짜수 계산하기 (localtime)	
■ 158. 오늘의 요일 계산하기 (localtime)	
■ 159. 프로그램 실행 시간 계산하기 (카카오 4 번 문제)	
■ 160. 주어진 숫자를 천단위 구분하기	
■ 161. 문자열의 각 문자를 그 다음 문자로 변경하기	
■ 162 ~ 172 예제 : 웹 스크롤링	
1. 웹 스크롤링 기본 문법	문제 351 ~ 368
2. ebs 레이디 버그 게시판 댓글 스크롤링	문제 369 ~ 382
3. 한겨레 신문사 웹 스크롤링 코드 구현	문제 383 ~ 387
4. 중앙일보 신문사	문제 388 ~ 390
5. 동아일보 신문사	문제 391
6. 이미지(사진) 데이터 스크롤링	
- 네이버	문제 393
- 구글	문제 394
- 다음	문제 395
- Bing	문제 396
- Instagram	문제 397
- 서울시 응답소 게시판 (Javascript 포함)	문제 398 ~ 412
■ 173. ~ 177 파이썬으로 게임 프로그램 개발	문제 407 ~ 412

■ 문제 모음

※ 알고리즘

- [이진탐색 알고리즘](#)
- [버블정렬](#)
- [삽입정렬](#)
- [탐욕 알고리즘](#)

※ pandas

- [65. Pandas 를 이용해서 조건절 사용 데이터 검색 방법](#)
- [66. Pandas 를 이용한 조인](#)
- [67. Pandas 를 이용한 서브쿼리](#)
- [73. 판다스와 오라클 그룹함수 비교](#)
- [74. pandas 와 오라클 분석함수 비교](#)

파이썬 설치방법

1. <https://www.anaconda.com/download/#windows> 접속 !
2. Python 3.7 version 다운로드

■파이썬 수업

파이썬을 통해서 구현해야 하는 목표

1. 데이터 분석을 잘하기 위해서 -----> R 과 파이썬

SQL, R, 파이썬을 다 잘할줄 아는 인재



분석 패키지 상당수 있어서 분석하기 편하다.



잘 분석한 결과 파워포인트를 넘어서 분석 솔루션을 만들기를 원한다.

2. 딥러닝 기술을 이용해서 이미지 분류 코드 완성

건강 폐사진 vs 질병 폐사진을 컴퓨터가 구분할 수 있도록 파이썬으로 신경망 구현

■ 1. 대화식 모드로 프로그래밍 하기

1. 대화식 모드

* 파이썬을 실행하는 방법

시작 --- 모든프로그램 --- Anaconda3 ----- Anaconda Prompt 실행

```
(base) C:\Users\WAdministrator>python
>>> a=1
>>> b=1
>>> a+b
2
>>> exit()
(base) C:\Users\WAdministrator>
```

2. 배치모드

(base) C:\Users\WAdministrator>notepad a.py

메모장에서

a=1

b=1

print(a+b)

입력하고 저장

(base) C:\Users\WAdministrator>python a.py

2

■ 2. 텍스트 에디터로 프로그래밍 하기

1. spyder 프로그램

- 한라인씩 드래그 해서 ctrl + enter

- F9 번을 누른다

a=1

b=1

a+b

2. 파이참(pycharm) 프로그램

시작 --- 모든프로그램 --- JetBrains ----- 파이참 실행

파이썬 디렉토리에 new file 만들어서 실행

a=1

b=1

a+b

ctrl + shift + F10 (출력)

■ 3. 변수명 만들기

[문제 1](#)

* 변수 이름 규칙

1. 변수 이름에는 다음 문자만 사용할 수 있다.

- 소문자(a~z)
- 대문자(A~Z)
- 숫자(0~9)
- 언더 스코어(_)

2. 변수 이름은 숫자로 시작할 수 없다.

3. 예약어를 사용할 수 없다

예 : False, Class, Is, Return

* 일반 변수 사용 예 :

'변수 안에는 데이터가 한개'

```
a = 100
```

```
print(a)
```

```
print(type(a))
```

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

a 변수의 데이터 유형을 확인하겠다.

■ 4. 변수에 값대입하기

문제 2

설명 : 파이썬에서는 변수에 값을 대입할 때 = (assignment) 기호를 사용한다.

예제 : a = 7788

```
v_empno number(10) = 7788 ; <----PL/SQL 코드
```

파이썬의 코드는 심플함을 철학으로 한다.

```
import this
```

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

파이썬에서 사용하는 변수는 객체를 가리키는 것 !

a = 3 # 3이라는 값을 가지는 정수형 자료형이 자동으로 메모리에 생성이 되고 a는 변수 이름
 이고 3이라는 객체가 저장된 메모리의 위치를 가리키는 것이다.

```
a -----> 3
           ↑
         정수형 객체
```

변수 a는 객체가 저장된 메모리의 위치를 가리키는 레퍼런스라고 할 수 있다.

```
a = 3
b = 3
a is b
```

```
True
```

a가 3을 가리키고 있고 b도 3을 가리키고 있다.
즉, 3을 가리키는 변수는 2개가 있는 것이

■ 5. 주석 처리하기 (#)

프로그램에서 코멘트는 인터프리터에 의해 무시되는 텍스트의 한 부분이다.

코드를 설명하거나 나중에 어떤 문제를 고치기 위해 표시하는 등 다양한 목적으로 코멘트를 사용할 수 있다.

1. 주석 처리하는 방법 (#)

더하기 하는 프로그램입니다.

a=1

b=1

print(a+b)

2. 라인을 유지하기 (\)

1 + 2 \

+3

3. 여러행 주석처리하는 방법 (""")

""" This program was made W

by Mr song """

■ 6. 자료형 개념 배우기

문제 5 ~ 6

* 자료형이란 ?

"프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료 형태로 사용되는 모든 것을 뜻한다"

* 파이썬의 자료형

1. 숫자
2. 문자
3. 리스트 : 데이터의 목록을 다루는 자료형

```
예 : d = [1, 2, 3]
      d[0] = 5          # 리스트 변수는 변경이 가능하다.
      print(d)
```

4. 튜플 : 데이터의 목록을 다루는 자료형인데 리스트와는 다르게 변경이 불가능한 자료형

```
예 : f = (1, 2, 3)
      print( type( f ) )

<class 'tuple'>

f[0] = 5
TypeError: 'tuple' object does not support item assignment
```

데이터가 변경이 안되기 때문에 믿을 수 있는 데이터가 된다

5. 딕셔너리 : key 와 value 를 조합해서 사용하는 자료형

```
예 : m={}
      m['python'] = 'www.python.org'
           ↑           ↑
      인덱스, key      값, value

      print(m)
      {'python': 'www.python.org'}

      print(m.keys())
      dict_keys(['python'])

      print (m.values())
      dict_values(['www.python.org'])
```

6. 집합 자료형 : 리스트 데이터 집합의 교집합, 합집합, 차집합을 출력할 때 유용하게 사용되는 자료형

```
예 : s1 = set( [1, 2, 3, 4, 5, 6])
      s2 = set( [4, 5, 6, 7, 8, 9])

      print ( s1 & s2 )          #교집합
      {4, 5, 6}

      print ( s1 | s2 )          #합집합
      {1, 2, 3, 4, 5, 6, 7, 8, 9}

      print ( s1 - s2 )          #차집합
      {1, 2, 3}
```

★ 리스트 변수

문제 3 ~ 4

"변수 안에 여러개의 데이터(요소)가 들어있는 변수"

```
d = [1000, 2000, 3000, 4000]
print(d)
print(type(d))
```

```
[1000, 2000, 3000, 4000]
<class 'list'>
```

■ 8. 들여쓰기 개념 배우기

[문제 7 ~ 8](#)

파이썬이 다른 언어보다 보기 쉽고 코드가 간결한 이유는 바로 콜론(:)을 사용해서 들여쓰기를 하도록 만들었기 때문이다.

파이썬은 블록 구분을 들여쓰기를 통해서 한다.

콜론(:) 뒤에 나오는 명령어를 다음줄에 작성하려면 ★4칸★ 들여쓰기를 해야한다.

예제 : 월급이 3000 인 직원들의 이름과 월급을 출력하시오 !

```
import csv

file = open("c:WWemp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if int( emp_list[5] ) ==3000:                # 오라클로치면 int --> to_number
        print( emp_list[1], emp_list[5] )
```

■ 9. if 문 개념 배우기 ① (if ~ else)

[문제 9 ~ 11](#)

```
if 문 예제 1 :  
    x = 1  
    y = 2  
    if x >= y:  
        print( 'x가 y보다 크거나 같습니다.' )  
    else:  
        print ( 'x가 y보다 작습니다.' )  
  
x가 y보다 작습니다.
```

※ 파이썬에서 콜론(:)을 쓰는 경우 4가지

1. if 문
2. for loop 문
3. while loop 문
4. def 함수문

```
if 문 예제 2 :  
    a = int( input( '숫자를 입력하세요 ~ ' ) )      # input 입력하게 유도하는 명령어  
  
    if a%2 == 0:  
        print( '짝수입니다.' )  
    else:  
        print( '홀수입니다.' )  
  
숫자를 입력하세요 ~ 8  
짝수입니다.
```

■ 10. if 문 개념 배우기 ② (if ~ elif)

[문제 12](#)

예 :

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

if a < b:
    print( a, '는 ', b, '보다 작습니다.')
elif a == b:
    print( a, '는 ', b, '랑 같습니다.')
else:
    print( a, '는 ', b, '보다 큼니다.')
```

■ 11. for 문 개념 배우기 ① (for)

[문제 13 ~ 22](#)

파이썬의 for 문의 기본적인 구조는 다음과 같다.

```
for 변수 in 리스트(또는 튜플, 문자열):  
    수행할 문장 1  
    수행할 문장 2
```

리스트나 튜플, 문자열의 첫번째 요소부터 마지막 요소까지 차례로 변수에 대입되어 "수행할 문장 1", "수행할 문장 2" 등이 수행된다.

문법 : for 반복변수 in 순서열:

↑
리스트 , 튜플, 딕셔너리
[] () {}
실행문

예제 : for i in (1, 2, 3):
 print(i)

예제 : for i in 'I am a boy':
 print(i)

■ 12. for 문 개념 배우기 ② (for~continue~break)

1. continue 문

[문제 29 ~ 35](#)

"반복문이 실행되는 동안 특정 코드 블록을 실행하지 않고 다른 코드 블록만 실행되게 할 때 사용하는 문법"

예 : "0 부터 9 까지 출력을 하는데 홀수는 출력하지 마라 !"

```
for i in range(10):  
    if i%2 == 1:                # 2로 나눈 나머지 값이 1 이 된다면  
        continue              # i 가 홀수 이므로 실행하지 말고  
    print(i)                   # 다른 코드블록만 실행되게 해라!
```

0
2
4
6
8

※ 문자열 포매팅 이해하기 (34 번)

"변하는 값을 포함하는 문자열을 표현하기 위해 하나의 양식으로 문자열을 만드는것"

예 : %s - 문자열
 %d - 정수
 %f - 실수
 %% - %기호

2. for 문에서 사용하는 break 문

[문제 36](#)

"루프를 중단시키는 역할을 하는 문법"

예제 :

```
scope = [1, 2, 3]  
  
for i in scope:  
    print(i)  
    break  
else:  
    print('perfect')
```

1

■ 13. for 문 개념 배우기 ③ (for~else)

```
a = int( input( '숫자 불러' ) )  
for i in range(1, 101):  
    if i <= a:  
        print(i)  
else:  
    print('perfect')
```

설명 : for ~ else 문에서 else 뒤의 실행 코드는 for 반복문을 모두 성공적으로 수행해야만 실행합니다

■ 14. while 문 개념 배우기(while~continue~break)

[문제 37 ~ 42, 50](#)

문법 :

```
while 조건문:
    수행할 문장 1
    수행할 문장 2
```

설명 : 위의 조건문에 만족할 동안에 계속해서 수행할 문장을 반복하겠다.

예제 :

```
x=0
while x < 10:
    x = x + 1          # 1 2 3 4 5 6 7 8 9 10
    if x < 3:
        continue
    print(x)
    if x > 7:
        break
```

※ for loop 문으로 할 수 있는 것은 다 while loop 문으로 할 수 있다.

※ loop 문이 필요한 이유?

실행문을 반복해서 실행하고자 할 때 필요하다.
단순작업을 자동화 시킬 수 있다.

■ 15. None 개념 배우기

[문제 43 ~ 44](#)

"None 은 Types.NonTypes 의 유일한 값으로, 값이 존재하지 않는 변수에 대입하여 이 변수에 아무런 값이 없다는 것을 나타내기 위해 주로 활용된다. "

N은 항상 대문자

예 : val = None

None 이 대입된 변수 val 은 아무런 값도 없는 빈 깡통변수라고 생각하면 된다.

예제 :

```
val = None
condition = 1

if condition == 1:
    val = [1, 2, 3]
else:
    val = 'I love Python'
print(val)

[1, 2, 3]
```

설명 : 위의 예제를 보면 val 에 리스트가 할당되기도 하고 문자열이 할당되기도 하기 때문에 처음 만들 때 None 을 할당해서 빈 깡통 변수를 만드는 것이다.

* 파이썬 연산자 정리

	오라클	vs	파이썬
1. 산술연산자	+ - * / mod power		+ - * / % ** 2**3=8
2. 비교연산자	> >= < <= = !=		> >= < <= == !=
3. 논리연산자	and or not		& !
4. 기타 비교연산자	between .. and in is null like		<= & >= in == '' ^, \$, 정규식 함수

■ 16. 정수형 자료 이해하기

[문제 45](#)

"자연수 (1, 2, 3,)와 음수 (-1, -2, -3,)와 0으로 이루어진 수 체계를 정수라 한다.

예제 :

a = 123
b = -178
c = 0

일반 프로그래밍에서 지원하는 정수형 상수의 범위는

-2,147,473,647 ~ 2,147,473,647 인데

파이썬에서는 메모리가 허용하는 범위에서 지원 가능한 수를 사용할 수 있다.

■ 17. 실수형 자료 이해하기

[문제 46](#)

"실수는 소수로 나타낼 수 있는 유리수와 소수로 나타낼 수 없는 무리수로 구성된 집합이다."

파이썬은 실수를 지원하기 위해서 부동 소수형을 제공한다.

★ 부동 소수형의 특징

1. 8바이트만 이용해서 수를 표현한다.
즉, 한정된 범위의 수만 표현할 수 있다.
2. 디지털 방식으로 소수를 표현해야 하므로 정밀도의 한계가 있다.

예제 : `a = 3.14`

`type(a)`

`Out[1]: float`

■ 19. 대입연산자 이해하기 (=)

[문제 47 ~ 49](#)

a = 1

왼쪽의 변수에 오른쪽의 값을 대입해라 ~

▷ 대입 연산자와 일반 연산자 비교

1. = : 대입 연산자

2. == : 같다

3. in : 여러개의 값 비교

■ 20. 사칙 연산자 이해하기 (+, -, *, /, **)

[문제 51 ~ 57](#)

* 오라클의 기타 비교 연산자와 파이썬 연산자 비교

오라클	vs	파이썬
between .. and		<= & >=
in		in
is null		= ' '
like		^, \$, 정규식 함수

■ 21. 연산자 축약 이해하기 (+=, -=, *=, /=)

[문제 58](#)

예제 :

a = 0

a += 1

a의 값은 ? a = a + 1, a = 1

a -= 5

a의 값은 ? a = a - 5, a = -4

a *= 2

a의 값은 ? a = a * 2, a = -8

a /= 4

a의 값은 ? a = a / 2, a = -2

■ 22. True 와 False 이해하기

[문제 59 ~ 60](#)

예제 :

```
a = True
b = False
print(a==1) # True 가 출력됨
print(b!=0) # False 가 출력됨
```

■ 23. 관계 연산자 이해하기 (==, !=, <, <=, >, >=)

[문제 61 ~ 63](#)

▷ pandas 모듈 설치

아나콘다 프롬프트 들어가서

```
pip install pandas
```

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

※ 판다스 기본 문법

```
emp [ 열 선택 ] [ 행 선택 ]
```

※대용량 환경에서 검색속도가 빠르다 !

■ 24. 논리 연산자 이해하기 (and, or, not)

[문제 64 ~ 65](#)

★ 기타 비교 연산자를 오라클과 파이썬 그리고 판다스와 비교

오라클	vs	파이썬	vs	판다스
between .. and		<= and >=		<= & >=
in		in		isin
is null		== ''		apply 함수
like		emp_list[1][0]		isnull()

※ lambda 표현식 ?

한번 쓰고 말 함수를 람다로 만들면 편하다!

[문제 66 ~ 72](#)

여러줄의 코드를 딱 한줄로 만들어 주는 문법
오라클로 설명하면 "이름 없는 함수"

예 : 파이썬에서 함수를 생성하는 방법

```
def hap(x,y):  
    return x + y  
print ( hap(10, 20))
```

■ 26. 시퀀스 자료형 이해하기

[문제 73](#)

문자열 'abcde'는 a, b, c, d, e 가 순서대로 나열되어 있는 시퀀스 자료형이다.

```
예제 : strdata = 'abcde'      # a b c d e
      print (strdata[0])      0 1 2 3 4
                                -2-1
      a

      listdata = [ 'a', 'b', 'c', 'd', 'e' ]
                  0   1   2   3   4
                  -5  -4  -3  -2  -1
```

■ 27. 시퀀스 자료 인덱싱 이해하기

[문제 74 ~ 76](#)

예제 :

```
strdata = 'Time is money!!'    # 문자열 변수
listdata = [1, 2, [4, 5, 6] ]  # 리스트 변수
                                # 리스트 안에 리스트가 있는 경우
```

■ 28. 시퀀스 자료 슬라이싱 이해하기

[문제 77 ~ 80](#)

예제 :

```
strdata = 'Time is money'
```

```
print ( strdata[1:5])          # 형식식 [시작문자열번호 : 끝 문자열 번호 : 배수의숫자
```

만]

```
ime
```

■ 29. 시퀀스 자료 연결하기 (+)

[문제 81 ~ 82](#)

예제 :

```
strdata1 = 'i love '  
strdata2 = 'Python'
```

```
print( strdata1 + strdata2)
```

```
i love Python
```


■ 30. 시퀀스 자료 반복 이해하기 (*)

예제 :

```
a = int ( input( '숫자를 입력하세요 !' ) )  
for i in range( 1, a + 1 ):  
    print( '★' * i )
```

숫자를 입력하세요 !5

```
★  
★★  
★★★  
★★★★  
★★★★★
```

■ 31. 시퀀스 자료 크기 이해하기 (len)

[문제 83](#)

```
strdata1 = 'I love Python'  
print( len(strdata1) )  
13
```

■ 32. 문자열의 멤버체크 이해하기 (in)

[문제 84 ~ 89](#)

"자료에 어떤 값이 있는지 없는지 확인하는 방법"

예제 :

```
listdata = [1, 2, 3, 4]
result1 = 5 in listdata
result2 = 4 in listdata

print( result1 )
print( result2 )
False
True
```

■ 33. 문자열 이해하기

[문제 90 ~ 91](#)

▷ 문자열을 나타내는 방법 3가지

1. ' 문자열 ' ----> 싱글
2. " 문자열 " ----> 더블
3. """ 문자열 """ ----> 더블 3개

예제 :

```
data1 = 'i am python programer'
data2 = 'you are a programer'
data3 = """ i love
        i love
        i love """
```

```
data4 = "my son's names is john"      # 문자열에 싱글을 나타내고 싶으면
data5 = 'name of my son is "john"'    # 문자열에 더블을 나타내고 싶으면
```

■ 34. 문자열 포매팅 이해하기

[문제 93 ~ 98](#)

"변하는 값을 포함하는 문자열을 표현하기 위해 하나의 양식으로 문자열을 만드는 것"

예 :

%s : 문자열
%d : 정수
%f : 실수
%% : % 기호

예제 :

```
txt1 = '자바'; txt2 = '파이썬'  
print ('나는 %s 보다 %s 에 더 익숙합니다.' %(txt1, txt2) )
```

나는 자바 보다 파이썬 에 더 익숙합니다.

■ 35. 이스케이프 문자 이해하기

★ 이스케이프 문자

1. Wn : 줄바꾸기
2. Wt : 탭
3. Wenter : 줄계속
4. WW : W 기호 자체

예제 : `print('나는 파이썬을 사랑합니다. Wn 파이썬은 자바보다 W
훨씬 쉽습니다.')`

■ 36. 리스트 이해하기 ([])

[문제 99 ~ 107](#)

▷ 리스트 변수란 ?

"데이터의 목록을 다루는 자료형"

대괄호 [] 안에 데이터를 입력해서 관리하는 변수

★★★★ 리스트 변수의 유용한 메소드 함수 ★★★★★

1. `append()` : 리스트의 끝에 새로운 요소를 추가
2. `extend()` : 기존 리스트에 다른 리스트를 이어 붙임
3. `insert()` : 리스트에 특정위치에 새로운 요소를 입력
4. `remove()` : 리스트에 요소를 삭제
5. `pop()` : 리스트에 마지막 요소를 제거
6. `index()` : 리스트에 특정위치의 요소를 출력할 때 사용
7. `count()` : 리스트에 요소의 건수를 출력
8. `sort()` : 리스트의 요소를 정렬
9. `reverse()` : 리스트에 요소의 순서를 반대로 뒤집을 때 사용

예제 :

```
emp_list = []  
emp_list.append('a')  
print(emp_list)  
  
emp_list.append('b')  
print(emp_list)
```

■ 오라클과 파이썬의 그룹함수 비교

[문제 108 ~ 116](#)

오라클	vs	파이썬
max		max
min		min
count		len
sum		sum
avg		사용자 정의 함수

예 :

```
sal_list = [800, 1250, 2000, 3000, 5000]
print(max(sal_list))
```

5000

■ 37. 튜플 이해하기 (())

* 튜플(tuple) 변수란 ?

리스트 변수와는 다르게 요소 데이터의 변경이 불가능한 자료형

리스트는 대괄호 [] 로 요소들을 들었는데

튜플은 소괄호 () 로 요소를 감싼다.

"변경이 안되므로 튜플로 만든 데이터에 대한 신뢰도가 높아진다."

예제 :

```
tuple1 = (1, 2, 3, 4, 5)
```

```
tuple2 = ('a', 'b', 'c', 'd')
```

```
tuple3 = (1, 'a', 'abc', [1, 2, 3])
```

```
tuple1[3][1] = 6
```

```
TypeError: 'int' object does not support item assignment
```

■ 38. 딕셔너리 자료형 이해하기 ({ })

[문제 117 ~ 131](#)

* 딕셔너리 자료형이란?

딕셔너리 자료형은 key 와 values 를 조합해서 사용하는 자료형

예 :

```
dic = {}
```

```
dic['파이썬'] = 'www.python.org'
```

↑ ↑
key values

```
print(dic)  
{'파이썬': 'www.python.org'}
```

```
print(dic.keys())  
dict_keys(['파이썬'])
```

```
print(dic.values())  
dict_values(['www.python.org'])
```

네이버 번역기 (파파고), 구글 번역기는 신경망을 통해서 수많은 영문과 번역된 한글을 입력시켜서 학습을 시킨다.

부산대 대학원에서 만든 번역기는 규칙기반으로 만들었다.

■ 39. (파이썬) 함수 이해하기 (def)

▷ 파이썬 함수를 생성해야 하는 이유?

내가 만든 스크립트를 다른사람도 유용하게 사용하게끔 하고 싶다면 코드를 함수로 만들면 된다.

* 오라클의 내장 함수와 파이썬의 내장함수 비교

	오라클	vs	파이썬
1. 문자함수 문제 132 ~ 137	upper lower initcap substr replace length rtrim ltrim rpad lpad instr		upper() lower() 사용자 정의 함수 사용자 정의 함수 replace() len() rstrip() lstrip() 사용자 정의 함수 사용자 정의 함수 사용자 정의 함수
2. 숫자 함수	round trunc mod power		round() trunc() % pow(), **
3. 날짜함수 문제 139 ~ 152	month_between add_month next_day last_day		사용자 정의 함수 relativedelta 사용자 정의 함수 monthrange
4. 변환함수	to_char to_number to_date		str() int(), float() datetime.strptime()
5. 일반 함수 문제 153 ~ 154	nvl decode case		사용자 정의 함수 사용자 정의 함수 if 문

▷ 함수 생성 예제 :

```
def add_number(n1, n2):  
    ret = n1 + n2  
    return ret  
  
print ( add_number(10, 25))  
  
35
```

★ import 에 대한 설명 (파이썬 200 제 45 번)

functions.py 를 가져오겠다

```

      ↑
import common.functions
      ↑
    common 이라는 패키지(폴더)

```

실습 : common.zip 을 압축을 해제하는데 압축을 해제해서 common 이라는 폴더가 생성되었는지 확인을하고 common 폴더안에 여러개의 .py 파일중에 functions.py 가 있는지 확인하시오 !

```

C:\Users\Administrator\spyder-py3\untitled0.py
:
:
common 폴더 갖다놓기

```

▷ import 문을 만나면 파이썬 모듈(py 파일)을 찾는 순서

[문제 146](#)

import date time 명령어를 수행하면 ?

1. 파이썬 내장 모듈

예제 :

```

import sys
print (sys.builtin_module_names)

```

2. sys.path 에 정의 되어있는 디렉토리

```

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

```

예제 : sys.path 의 내용을 프린트하시오 !

```

C:\ProgramData\Anaconda3\lib\site-packages

```

※ site-packages 란?

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

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

예 :

```

C:\ProgramData\Anaconda3\lib\site-packages

```

★ 몬테카를로 알고리즘

[문제 155 ~ 159](#)

" 수많은 노가다를 통해서 답을 알아내는 것"

1. 강화학습 : 컴퓨터가 하나의 게임을 수차례 스스로 하면서 게임을 배워나가는 방법
2. 데이터 분석 : 6기 조은성 학생 지금 강남에 삼성생명에서 데이터 분석 프로젝트를 하고있는데 몬테카를로 알고리즘으로 수행

예 : 삼성생명에서 금융사고가 났을 때를 대비해서 평상시에 가지고 있어야 할 자금이 어떻게 되는지를 예측
컴퓨터로 수많은 금융사고를 랜덤으로 일으켜서 대비해야할 금액을 알아냄

원의 넓이 구하는 공식 ?

반지름 x 반지름 x pi(3.14)

■ 40. 함수 인자(매개변수) 이해하기

[문제 160 ~ 161](#)

예제 :

```
def add_txt(t1, t2 = '파이썬'):
    print(t1+' : ' + t2)

add_txt('베스트')

add_txt(t2 = '대한민국', t1 = '1등')

베스트 : 파이썬
1등 : 대한민국

def add_txt(t1 = '쉬운', t2 = '파이썬'):
    print(t1+' : ' + t2)
```

▷ 가변형 매개변수 ?

매개변수의 수가 유동적인 함수를 만들고 싶을 때 사용하는 변수.

함수를 실행할 때 매개변수를 10 개, 20 개를 입력해도 제대로 동작한다.

예제 :

```
def merge_string(*text_list):
    result = ''
    for s in text_list:
        result += s + ' '
    return result

print(merge_string('아버지가', '방에', '들어가시려다가', '부엌으로', '가셔서', '짜장면이', 'W',
                    '싫다고', '하셨어'))
```

■ 41. 지역변수와 전역변수 이해하기 (global)

[문제 162](#)

"변수는 자신이 생성된 범위 (코드블록) 안에서만 유효하다. 함수 안에서 만든 변수는 함수 안에서만 살아있다가 함수 코드의 실행이 종료되면 그 생명을 다한다."

이것을 지역변수라고 한다.

이와는 반대로 함수 외부에서 만든 변수는 프로그램이 살아있는 동안에 함께 살아있다가 프로그램이 종료되면 같이 소멸 된다.

이렇게 프로그램 전체를 유효범위로 가지는 변수를 전역변수 라고 한다.

예 :

```
param = 10
strdata = '전역변수'

def func1():
    strdata = '지역변수'
    print(strdata)

print(func1())

지역변수
```

* 지역변수를 설명하는 코드

```
param = 10

def func2(param):
    param = 1

func2(param)

print(param)          # 무엇이 출력될까?

10

"함수 안에서 만든 변수는 함수 안에서만 살아있다가 함수 코드의 실행이 종료되면 그
생명을 다한다."

이 변수를 지역변수라고 한다.
```

* 전역변수를 설명하는 코드

```
param = 10

def func3():
    global param
    param = 50

func3()

print(param)

"전역변수는 함수내부 뿐만 아니라 함수 외부에서도 사용되는 프로그램 변수이다."
```

* 프로그램이란 ? 여러가지 기능(함수)를 모아놓은 코드

※ 어떤 변수를 전역변수로 사용해야 하는가 ?

프로그램 전체에서 공통적으로 사용되고 잘 변하지 않는 데이터는 전역변수로 사용한다.

예 :

```
pi = 3.141592653589793
```

함수 1(기능 : 원의 넓이를 구하는 함수):

```
global pi  
print(pi * 반지름 * 반지름)
```

함수 1(기능: 두개의 원의 넓이를 구하는 함수):

```
global pi  
print(pi.....)
```


■ 42. 함수 리턴값 이해하기(return)

[문제 163](#)

"모든 함수는 이름을 갖고 있다. 이 이름을 불러주면 파이썬은 그 이름 아래 정의되어 있는 코드를 실행한다.

이때 함수를 부르는 행위를 호출이라고 하고 함수를 부르는 코드를 호출자라고 한다.

함수가 호출자에게 결과를 돌려주는 것을 반환(return)이라고 한다."

예제 :

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

예제 :

```
param = 10  
  
def func3():  
    global param  
    param = 50  
  
func3()  
print(param)          # 무엇이 출력될까?
```

■ 43. 파이썬 모듈 이해하기

[문제 164](#)

" 파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 한다."

이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게 하나의 파일로 묶어놓은 것을 모듈이라고 한다.

외부에 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로 가져와서 자유롭게 사용할 수 있도록 해야 하는데 이런 일을 파이썬에서는 모듈을 import 한다고 한다.

예제 :

```
import time
print('5초간 프로그램을 정지합니다.')
time.sleep(5)
print('5초가 지났습니다.')
```

■ 44. 파이썬 패키지 이해하기

"우리가 음악파일을 저장할때도 장르별로 폴더를 만들어서 별도로 저장하듯이 파이썬 모듈도 음악처럼 갯수가 많아지면 폴더(모듈 꾸러미) 별로 관리를 해야 관리 편해지는데 이 폴더(디렉토리)가 패키지다."

C:\Users\Administrator\W.spyder-py3\my_loc

__init__.py 는 대개 비워두는 것이 보통인데 이 파일을 손대는 경우는 언제인가?

답 : __all__ 변수를 조정할 때 손을 댈다.

↓

패키지로 부터 반입할 목록을 정의할 때 사용

__init__.py 를 열어서 아래의 내용을 적으시오

```
__all__ = [ 'cal_test3', 'oracle_func' ]
```

위와 같이 __init__ 에 __all__를 설정해놓으면 패키지의 모듈을 불러올 때 아래의 명령어가 가능해진다.

```
from my_loc import *
```

원래는 어떻게 해야 하나면 ?

```
from my_loc import cal_test3
from my_loc import oracle_func
```

■ 45. 파이썬 모듈 임포트 하기 ① (import)

설명 : import 문을 만나면 파이썬 모듈을 찾는 순서

1. 파이썬 내장 모듈
2. sys.path 에 정의되어있는 디렉토리

■ 46. 파이썬 모듈 임포트 이해하기 ② (from ~ import)

C:\Users\Administrator\W.spyder-py3\my_loc(패키지)

↓
cal_test3.py (모듈)
↓
plus (함수)
minus (함수)
multiply (함수)
divide (함수)

C:\Users\Administrator\W.spyder-py3\temp.py

temp.py 에서 cal_test3.py 모듈 안의 plus 함수를 사용하고 싶다면 ?

```
import my_loc.cal_test3  
print(my_loc.cal_test3.plus(10,20))    # 너무 길어서 불편해!
```

↓ (좀더 편하게 코딩)

```
import my_loc.cal_test3 as mc  
print( mc.plus(10,20))                  # 좀 나아졌네, 좀 더 줄여볼까??
```

↓ (좀더 편하게 코딩)

```
from my_loc.cal_test3 import plus  
print(plus(10,20))                      # 엔 아무것도 안붙여도 되네?  
                                         근데 plus 만 쓸수가 있네
```

↓ (좀더 편하게 코딩)

```
from my_loc.cal_test3 import *  
print(plus(10,20))                      # 이제 cal_test3 함수들 다 쓸 수 있다 !!  
print(minus(20,10))
```

■ 47. 파이썬 모듈 임폴트 이해하기 ③ (import ~ as)

예제 :

```
import my_loc.cal_test3 as mc  
print( mc.plus(10,20))
```

■ 48. 파일 열고 닫기 (open, close)

예제 : 이미지 파일을 파이썬에서 여는 방법

```
import PIL.Image as pilimg
import numpy as np
import matplotlib.pyplot as plt

im = pilimg.open('c:\\\\lena.png')
pix = np.array(im)

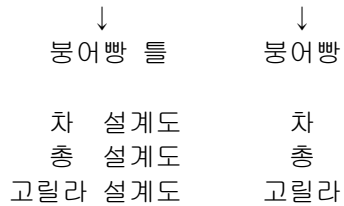
plt.imshow(pix)
```

■ 49. 클래스 이해하기 (class)

[문제 168 ~ 172](#)

클래스 (자료형) -----> 객체

클래스는 자료형이고 아직 객체가 되지 않았다.



* 클래스가 필요한 이유에 대한 실습

▷ 계산기 함수 생성

```
result = 0
```

```
def adder(num):
    global result
    result = result + num
    return result
```

```
print( adder(3) )
print( adder(4) )
```

```
print( adder(2) )
print( adder(5) )
```

14 가 나와버린다.

결국 계산기가 하나 더 필요하다.

adder 함수 하나만으로는 결과값을 따로 유지할 수 없다.

```
result1 = 0
result2 = 0
```

```
def adder1(num):
    global result1
    result1 = result1 + num
    return result1
```

```
def adder2(num):
    global result2
    result2 = result2 + num
    return result2
```

```
print( adder1(3) )
print( adder1(4) )
```

```
print( adder2(2) )
print( adder2(5) )
```

계산기 1 과 계산기 2 를 가지고 계산을 했는데 하지만 계산기가 3 개, 4 개, 5 개, 10 개로 점점 더 많이 필요해진다면 어떻게 해야 할 것인가?

그 때마다 전역변수와 함수를 추가할 것인가?

이 문제를 클래스가 간단하게 해결해 준다.

```
class Calculator:          # 계산기 설계도(클래스)를 만든다.
    def __init__(self):     # 계산기 초기화하는 함수 반드시 함수 이름 __init__
        self.result = 0

    def adder(self, num):    # 더하기 하는 함수
        self.result += + num
        return self.result1
```

설계도(클래스)를 가지고 계산기 2개 생성

```
cal1 = Calculator()
cal2 = Calculator()

print(cal1.adder(3))
print(cal1.adder(4))
print(cal2.adder(2))
print(cal2.adder(5))
```

■ 50. 클래스 멤버와 인스턴스 멤버 이해하기

문제 176

* 용어설명

```
gun1 = Gun()  
  ↑   ↑  
객체 클래스(설계도)
```

```
gun2 = Gun()
```

gun1 과 gun2 는 클래스는 동일하지만 객체는 서로 다르다.

인스턴스 = 클래스() # 클래스를 가지고 객체(인스턴스)를 생성 했다!

* 클래스에서 사용하는 변수 2 가지

1. 클래스 변수 ? 객체에 의해서 변경이 되면 안되는 변수, 반드시 고정값으로 사용되어야만 하는 데이터는 클래스로 선언해서 사용한다.
단체의 단체명과 같이 클래스로 만들어진 모든 인스턴스가 공유하는 데이터를 담는 변수
2. 인스턴스 변수 ? 객체에 의해서 변경이 될 수 있는 데이터의 변수
사람의 이름과 같이 각각의 인스턴스마다 가지고 있는 고유한 데이터를 담는 변수

예제 : 클래스 변수의 예제

"회사의 사원을 관리하는 클래스(설계도)를 생성"

```
class Employees:  
    raise_amount = 1.1                      # 클래스 변수  
  
    def __init__(self, first, last, pay):                      # 생성자(인자) 초기화  
        self.first = first  
        self.last = last  
        self.pay = pay  
        self.email = first.lower() + '.' + last.lower() + '@gmail.com'  
  
    def full_name(self):                      # 사원 전체이름을 출력하는 함수  
        return '{}{}'.format(self.first, self.last)  
  
    def apply_raise(self):                      # 월급을 인상하는 함수  
        self.pay = int( self.pay*self.raise_amount)                      # 인스턴스 변수 사용  
        #self.pay = int( self.pay*Employees.raise_amount)# 클래스 변수(무조건 1.1)  
emp_yongsik = Employees('yongsik', 'kim', 5000000)
```

예제 1. 위의 Employee 설계도를 가지고 emp_yongsik 객체를 실체화 시켜서 기존 월급과 인상된 월급 2 가지를 같이 print 하시오

```
print(emp_yongsik.pay)                      # 5000000 출력  
emp_yongsik.apply_raise()  
print(emp_yongsik.pay)                      # 5500000 출력
```

예제 2. 용식과 후림이 사원 객체를 아래와 같이 생성하고 월급도 동일하게 해서 각각 프린트 해보시오 !

```
emp_yongsik = Employees('yongsik','kim',5000000)
emp_huryim = Employees('huryim','lee',5000000)
```

```
emp_huryim.raise_amount = 1.2
```

이 작업을 못하게 막을때 = 클래스변수로 선언!

```
print(emp_yongsik.pay)
emp_yongsik.apply_raise()
print(emp_yongsik.pay)
```

```
print(emp_huryim.pay)
emp_huryim.apply_raise()
print(emp_huryim.pay)
```

■ 51. 클래스 메소드 이해하기

```
class MyClass:          # 낙타등 표기법 (CamelCase)
    def sayHello(self):
        print('안녕하세요')

    def sayBye(self, name):
        print('%s! 다음에 보자!' %name)

obj = MyClass()
obj.sayHello()          # 객체.메소드()
obj.sayBye('철수')
```

예제 : gun = Gun()
 ↓ ↓
 객체 클래스

gun.shoot(3)
 ↓ ↓
 객체 메소드

.(점) 붙이고 쓰면 메소드 ?

[1, 2, 3, 4, 5]
 ↓
 객체

[1, 2, 3, 4, 5].count(2)
 ↓ ↓
 객체 메소드

* 리스트 변수(객체)의 유용한 메소드

1. count()
2. append()
3. insert()
4. remove()
5. pop()
6. index()
7. sort()
8. reverse()

```
a = [1, 2, 3, 4, 5]
a.count(2)
```

변수 a는 리스트 객체의 메모리 주소를 가지고 있어서 a.count(2)라고 하면 a를 통하여 [1, 2, 3, 4, 5] 객체의 count 메소드를 실행한 것이다.

■ 기초 통계에 필요한 함수들 생성

1. 중심 성향을 분석하는 통계방법
평균값, 중앙값, 최빈값, 분산, 표준편차, 범위, 사분위수 범위

[문제 177 ~ 184](#)

2. 그래프
막대 그래프, 원형 그래프, plot 그래프, 라인 그래프, 사분위수 그래프

[문제 185 ~ 189](#)

★ 평균값과 중앙값이 필요한 이유?

석우형님 나이를 290으로 바꾸고 두값을 비교하면

석우형님 나이 290

37.0	# 평균
27.0	# 중앙값

석우형님 나이 40

27.74074074074074	# 평균
27.0	# 중앙값

평균값은 이상치 데이터(잡음 데이터)에 민감해서 평균값만 가지고 데이터를 대표하기에는 정보가 부족하기 때문에 중앙값이 필요하다.

☆ 최빈값 ? 가장 빈번하게 출현하는 데이터

☆ 분산과 표준편차

분산 ? 데이터의 퍼짐정도 (평균에서 거리가 얼마나 떨어졌는지)

표준편차 ? 평균에 대한 오차

☆ 범위, 사분위수 범위

★ 범위 : 최대값 - 최소값

★ 사분위수 범위

전체 데이터를 정렬했을 때 갯수를 4등분 한 분위수를 의미

제 1 사분위수(Q1) : 데이터의 25%가 이 값보다 작거나 같음.

제 2 사분위수(Q2) : 데이터의 50%가 이 값보다 작거나 같음.(중위수)

제 3 사분위수(Q3) : 데이터의 75%가 이 값보다 작거나 같음.

사분위간 범위 : 제 1 사분위수와 제 3 사분위수 간의 거리(Q3-Q1)이므로, 데이터의 중간 50%에 대한 범위입니다.

■ 51. 클래스 메소드 이해하기 ②

[문제 190 ~198](#)

"기초 통계에 필요한 함수들을 가지고 class 를 생성"

예 :

```
stats = stats()
```

```
a = [24, 23, 21, 20, 30, 44, 41]
```

```
stats.mean(a)
```

■ 52. 클래스 생성자

"객체가 만들어질 때 호출되는 함수를 생성자(__init__)라고 한다."

생성자는 초기화 할때 자주 사용한다.

예제 :

```
class My_Stats:
    def __init__(self):
        print('통계 객체가 생성되었습니다.')
```

■ 53. 클래스 소멸자 이해하기

"객체가 사라질 때 호출되는 함수를 소멸자 (__del__)라고 한다.

예제 :

```
class Myclass:
    def __init__(self):
        self.var = '안녕하세요 !'
        print('Myclass 인스턴스 객체가 생성되었습니다.')

    def __del__(self):
        print('Myclass 인스턴스가 메모리에서 제거됩니다.')

object1 = Myclass()
del object1
```


■ 54. 클래스 상속 이해하기

[문제 199 ~ 202](#)

상속이란 ? 클래스들 끼리 유산(기능) 을 물려주는 것

부모 클래스 -----> 자식 클래스
기능

상속을 받게 되면 부모의 기능을 굳이 자식 클래스에 코딩을 하지 않아도 된다.

예제 :

```
graph TD
    Card[Card 클래스] --> Movie[영화할인 card]
    Card --> Mart[마트 할인 card]
```

예제 1 :

```
class father:
    def base_method(self):
        print('hello~~~~~')

class child(father):
    pass

father = father()
father.base_method()

child = child()
child.base_method()
```

☆ 오버라이드 : 자식 클래스에서 부모클래스의 함수를 가져와서 수정하면 부모클래스의 함수를 무시하고 자식 클래스의 함수로 적용되게 하는 것을 말한다.

★ 다중 상속

다중 상속이란 두개 이상의 클래스를 상속 받는것을 말한다. 이 경우에는 두 클래스의 모든 속성을 물려받게 된다. 이는 하나의 자식 클래스가 두개 이상의 부모 클래스를 가지는것이라고 할 수 있다.

```
graph TD
    father1 --> child
    father2 --> child
```

예제:

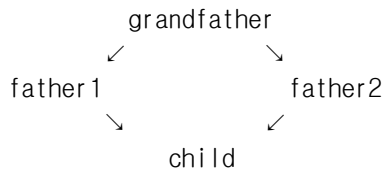
```
class father1:
    def func(self):
        print("지식")

class father2:
    def func(self):
        print('지혜')

class child(father1, father2):
    def childfunc(self):
        father1.func(self)
        father2.func(self)
```

```
child1 = child()
child1.childfunc()
```

☆ 다중 상속시 주의할 점 "죽음의 다이아몬드 상속"



예제 :

```
class grandfather:
    def __init__(self):
        print('튼튼한 두팔')

class father1(grandfather):
    def __init__(self):
        grandfather.__init__(self)
        print("지식")

class father2(grandfather):
    def __init__(self):
        grandfather.__init__(self)
        print("지혜")

class grandchild(father1,father2):
    def __init__(self):
        father1.__init__(self)
        father2.__init__(self)
        print('자기 만족도가 높은 삶')

grandchild = grandchild()
```

결과 :

```
튼튼한 두팔
지식
튼튼한 두팔
지혜
자기 만족도가 높은 삶
```

설명 : 팔이 4 개가 되어버렸다. 다중상속시 주의할 점 이다. 그래서 팔이 한번만 출력되게 하려면 super 를 사용해야 한다.

```
class grandfather:
    def __init__(self):
        print('튼튼한 두팔')

class father1(grandfather):
    def __init__(self):
        print("지식")
        super().__init__()

class father2(grandfather):
    def __init__(self):
        print("지혜")
        super().__init__()
```

```

class grandchild(father1,father2):
    def __init__(self):
        super().__init__()
        print('자기 만족도가 높은 삶')

grandchild = grandchild()

```

--- super() 와 관련하여 문제 생성 스크립트

```

class Card:
    def __init__(self):
        self.card = 0
        print('카드가 발급 되었습니다.')
    def charge(self, num):
        self.card += num
        print('{}원 충전되었습니다.'.format(num))
    def consume(self, num, loc):
        if self.card <= num:
            print('잔액이 부족합니다.')
        else:
            print('{}에서 {}원 사용했습니다.'.format(loc, num))
            self.card -= num
    def print(self):
        print('잔액이 {}원 입니다.'.format(self.card))

```

```

class m_card(Card):
    def consume(self, num, loc):
        if loc == '영화관':
            num = num*0.8
        super().consume(num, loc)

```

```

class t_card(Card):
    def consume(self, num, loc):
        if loc == '마트':
            num = num*0.9
        super().consume(num, loc)

```

```

class movie_mart_trans_card(m_card,t_card):
    def consume(self, num, loc):
        if loc == '교통':
            num = num*0.5
        super().consume(num, loc)

```

■ 55. 예외처리 이해하기 ① (try ~ except)

[문제 203 ~ 207](#)

예외 ?

"프로그램에서 에러가 발생했을 때, 에러를 처리하는 기능"

예제 :

```
def my_divide():  
    x = input(' 분자의 숫자를 입력하세요 ~')  
    y = input(' 분모의 숫자를 입력하세요 ~')  
    return int(x)/int(y)
```

```
print( my_divide() )
```

분자의 숫자를 입력하세요 ~10

분모의 숫자를 입력하세요 ~2

5.0

분모에 만약 0을 넣은다면 ZeroDivisionError: division by zero 에러가 난다
근데 이것은 코드의 에러가 아니라서 보기 편한 메시지로 에러 화면을 바꾸자!

예 : 홍플러스 계산원분이 고객님의 홍플러스 카드가 단말기에서 오류메세지가 나온다면?

■ 56. 예외처리 이해하기 ② (try ~ except ~ else)

예제 :

```
try:
    실행할 코드블럭
except:
    예외 처리 코드 블럭
else:
    except 절을 만나지 않았을 경우 실행할 코드 블럭
```

예제(문제 205 수정) :

```
def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')
    name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
    try:
        sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
        # 반드시 수행이 되어야 하는 중요한 코드를 처리해야 else 이후에 다른 코드들이
        # 돌고 싶게 하고 싶을 때 사용

    except:
        return '해당사원은 없습니다.'
    else:
        print('월급 추출에 성공했습니다.')          # 추가
        return sal                                   # 리턴 자리 변경

print( find_sal())

월급을 알고 싶은 사원명을 입력하세요 ~ scott
월급 추출에 성공했습니다.
[3000]
```

try와 except 사이의 코드가 문제가 없다는 것을 확실히 눈으로 확인하고 넘어갈 수 있다
(else 문으로 넘어간다) 는 장점이 있다.

■ 복수개의 except 절 사용하기

[문제 208](#)

" 예외처리를 여러개를 나열 할 수 있다."

예제 : 아래의 두 수를 나누는 코드는 작성하는데 아래의 두가지가 다 실행되게 하시오 !

분자 숫자를 입력하세요 ~ 10

분자 숫자를 입력하세요 ~ 0

0으로 나눌 수 없습니다.

분자 숫자를 입력하세요 ~ 10

분자 숫자를 입력하세요 ~ aaa

나눌 수 없습니다.

코드 :

```
def my_divide():
    try:
        x = input ('분자 숫자를 입력하세요~ ')
        y = input ('분모 숫자를 입력하세요~ ')
        z = int(x)/int(y)
        return z
    except ZeroDivisionError:                # 0으로 나눌때의 예외처리
        return ('0으로 나눌 수 없습니다.')
    except KeyboardInterrupt:
        return( '사용자에 의해서 프로그램이 종료됩니다.')
        #ctrl + c <----- 를 누르면
    except:
        return ('나눌 수 없습니다.')

print(my_divide())
```

다른 예외처리 절 : 구글링~

■ 57. 예외처리 이해하기 ③ (try ~ except ~ finally)

[문제 209](#)

예제 :

```
try:
    실행할 코드 블록
except:
    예외처리 코드 블록
finally:
    실행할 코드가 성공했든 실패했든 무조건 실행되는 블록
```

코드 : 이름을 물어보게 하고 이름을 입력하면 월급이 출력되는 코드

```
def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')
    name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
    try:
        sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
        return sal
    except:
        return '해당사원은 없습니다.'
    finally:
        print('저는 무조건 실행됩니다.')

print( find_sal())
```

```
월급을 알고 싶은 사원명을 입력하세요 ~ scott
저는 무조건 실행됩니다.
[3000]
```

```
월급을 알고 싶은 사원명을 입력하세요 ~ wjdtjdgh
저는 무조건 실행됩니다.
해당사원은 없습니다.
```

■ 58. 예외처리 이해하기 ④ (try ~ except Exception as e)

[문제 210](#)

"코드상의 오류를 출력할 때 사용"

try 문에서는 오류가 나면 except 문을 사용하기 때문에 어떤 오류인지 확인할 수가 없기 때문에 오류를 확인하기 위해서 활용한다.

예제 :

```
try:
    print(param)
except Exception as e:
    print(e)
```

name 'param' is not defined <----- 위의 코드가 실패한 이유를 보여주고 있다.

■ 59. 사용자 정의 예외 처리

[문제 211](#)

"파이썬 입장에서 봤을때는 오류가 아닌데 프로그래머가 이건 오류이다 라고 일부러 프로그램이 안돌게 만들게 오류 메시지를 출력하게 되는 경우에 사용한다."

금액이 안맞는 프로그램이 있다면 <----- 사고(코드 오류)

예제 :

```
def 함수명:
    실행코드

    if 어떤 조건:
        raise Exception('예외가 발생했습니다.')
    else:
        return 변수명
```

■ 60. 사용자 입력받기 (input)

예제 :

```
k = input('<값>을 입력하세요 : ')
```

```
print( '당신이 입력한 값은 <'+ k +'> 입니다. ')
```

■ 61. 자료형 확인하기 (type)

예제 :

```
numdata = 57
strdata = '파이썬'
listdata = [1,2,3]
dicdata = {'a':1, 'b':2}
```

```
def func():
    print('안녕하세요')
```

```
print(type(numdata))
print(type(strdata))
print(type(listdata))
print(type(dicdata))
print(type(func))
```

■ 62. 나눗셈에서 나머지만 구하기 (%)

[문제 212](#)

예제 :

```
a = 11113
b = 23
ret = a%b
print(ret)
```

4

■ 63. 몫과 나머지 구하기 (divmod)

예제 :

```
a = 11113
b = 23
ret1, ret2 = divmod(a, b)  # 몫, 나머지
    ↑    ↑
    몫   나머지

print (ret1, ret2)
```

■ 68. 반올림 수 구하기 (round)

[문제 228 ~ 230](#)

예제 :

```
print( round( 16.554, 0 ) )
```

17.0

1 6 . 5 5 4

-2 -1 0 1 2 3

```
print( round( 16.554, 1 ) )
```

17.6

■ 69. 실수형 자료를 정수형 자료로 변환하기 (int)

[문제 231](#)

- 파이썬의 변환 함수 정리

	Data type	변환함수
1. 정수형	int	int()
2. 문자형	str	str()
3. 실수형	float	float()
4. 리스트형	list	list()
5. 튜플형	tuple	tuple()

■ 70. 정수형 자료를 실수형으로 변환하기 (float)

예제 :

```
a = float(10)
print(a)           # 10.0 으로 출력됨
```


■ 71. 정수 리스트에서 소수만 걸러내기(filter)

[문제 232 ~ 235](#)

합성수

vs

소수

6

7

2 x 3

두개의 수로 못나눈다.

(1 과 자기 자신으로만 나눌 수 있는 수)

■ 75. 문자열에서 특정 위치의 문자 얻기

예제 241. 아래의 txt1 문자열에서 '가'를 출력하시오!

```
txt1 = '이 또한 지나가리라.'  
  
print( txt1[-4] )
```

■ 76. 문자열에서 지정한 구간의 문자열 얻기

예제 242. 아래의 txt1 문자열에서 '가리라'를 출력하시오 !

```
txt1 = '이 또한 지나가리라.'  
  
print( txt1[-4:])
```

■ 77. 문자열에서 홀수 번째 문자만 추출하기

예제 243. 아래의 txt2 문자열에서 홀수번째 문자만 추출하시오 !

```
txt2 = 'aAbBcCdDeEfGhHiIjJkK'  
  
print( txt1[0::2])
```

예제 244. 아래의 txt2 문자열에서 짝수번째 문자만 추출하시오 !

```
txt2 = 'aAbBcCdDeEfGhHiIjJkK'  
  
print( txt1[1::2])
```

■ 78. 문자열을 거꾸로 출력하기

```
txt3 = 'abcdefg'  
print(txt3[::-1])
```

■ 파이썬의 리스트가 스택 구조임을 확인하시오 !

[문제 246 ~ 247](#)

1. 스택 --> 후입선출

2. 큐 --> 선입선출

예 : 오라클에서 데이터의 무결성을 보호하기 위한 lock 이 바로 큐 구조이다.

A	B	C

scott(A)	scott(B)	scott(c)
1. update emp		
set sal = 9000		
where ename = 'ALLEN';		
	2. update emp	
	set sal = 0	
lock	where ename = 'ALLEN';	
		3. update emp
		set sal = 80
		where ename = 'ALLEN';
4. commit	lock	

■ 데이터 정제를 파이썬으로 하는 방법 (텍스트 마이닝 클래스)

" 텍스트 마이닝 클래스를 생성 "

■ 79. 두 개의 문자열 합치기(+)

[문제 248](#)

예제 :

```
filename = input('저장할 파일 이름을 입력하세요 : ')\na = '당신이 저장한 파일은 ' + filename + ' 입니다.'\nprint(a)
```

저장할 파일 이름을 입력하세요 : 카카교교
당신이 저장한 파일은 카카교교 입니다.

■ 80. 문자열을 반복해서 새로운 문자열로 만들기

예제 : msg1 = '여러분'
print(msg1*3)

■ 81. 문자열에서 특정 문자가 있는지 확인하기 (in)

[문제 249 ~ 250](#)

예제 :

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

```
임의 문장을 입력하세요 ~ i am boy
당신이 입력한 문장에는 a가 있습니다.
```

■ 82. 문자열에서 특정 문자열이 있는지 확인하기 (in)

[문제 251 ~ 263](#)

예제 :

```
msg = input('임의 문장을 입력하세요 ~ ')
if 'is' in msg:
    print('당신이 입력한 문장에는 is 가 있습니다.')
else:
    print('당신이 입력한 문장에는 is 가 없습니다.')
```

임의 문장을 입력하세요 ~ i am boy
당신이 입력한 문장에는 is 가 없습니다.

■ 83. 파이썬 Comprehension

※ Python Comprehension

1. list Comprehension
2. set Comprehension
3. Dictionary Comprehension

★ 1. list Comprehension

[문제 264 ~ 265](#)

문법 : 출력 표현식 for 요소 in 입력시퀀스 if 조건식

예제 :

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

print(newlist)

[1, 4, 9]
```

★ 2. set Comprehension

[문제 266](#)

{ 출력 표현식 for 요소 in 입력시퀀스 if 조건식 }

예제 :

```
oldset = {1, 2, 3, 3, 4}
newset = {i*i for i in oldlist}
print(newlset)

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

★ 3. Dictionary Comprehension

[문제 267](#)

{Key:Value for 요소 in 입력시퀀스 if 조건식 }

예제 :

```
id_name = {1 : '김준구', 2: '장보겸', 3: '정지엽'}
a = {val:key for key,val in id_name.items()}
print(a)
```


■ 84. 문자열이 알파벳인지 검사하기(isalpha)

[문제 269 ~ 270](#)

- * 설명 :
1. 알파벳이 맞는지 확인 : `isalpha`
 2. 숫자가 맞는지 확인 : `isdigit`
 3. 공백이 맞는지 확인 : `isspace`

■ 85. 문자열이 숫자인지 검사하기(isdigit)

예제 271. 셜록홈즈 대본에는 숫자가 몇개 있는지 확인하시오 !

```
text_file = open('c:\wwwsherlock.txt')
lines = text_file.readlines()
total = 0

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

print(total)
```

132

■ 86. 문자열에 특수문자가 있는지 검사하기

[문제 272 ~ 273](#)

겨울왕국에는 특수문자가 몇개나 있는지 확인해보기

특수문자 = 전체문자 - (문자+숫자+공백)

■ 87. 문자열에서 대소문자 변환하기 (upper, lower)

예제 :

```
txt = 'A lot of Thing occur each day.'  
print( txt.upper() )  
print( txt.lower() )
```

■ 88. 문자열에서 좌우 공백 제거하기 (lstrip, rstrip, strip)

예제 : txt = ' 양쪽에 공백이 있는 문자열 입니다. '

```
print( txt.lstrip() )
print( txt.rstrip() )
print( txt.strip() )
```

■ 89. 재귀 알고리즘 기본 (구구단, 별)

[문제 274 ~ 280](#)

★ 재귀함수란?

1. 재귀함수는 함수 내에서 다시 자신을 호출한 후 그 함수가 끝날때 까지 함수 호출 이후의 명령문을 수행하지 않는다.
2. "반복문 + 스택구조가 결합된 함수"
먼저 들어간 데이터가 가장 마지막에 나오는 구조
나중에 들어간 데이터가 가장 먼저 나오는 구조 (후입선출)

예제 :

```
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 a 와 b 값을 받아와서 hap 함수와 gop 함수에 던져주는 역할만 수행한다.
hap_gop 함수가 hap 함수와 gop 함수를 호출했다.

" 그런데 재귀함수는 자기 자신의 함수를 호출한다."

```
예제 : def countdown(n):  
        # | 3 | 2 | 1 |  
        if n == 0:  
            # | 3 | 2 | 1 |  
            print('발사')  
        else:  
            # |  
            print(n)  
            # | 3 | 2 | 1 |  
            countdown(n-1) # | countdown(2) | countdown(1) | countdown(0) |
```

```
countdown(3)
```

```
3
```

```
2
```

```
1
```

```
발사
```

※ 재귀함수 를 사용했을 때의 장점

1. 코드가 간결해진다.
2. 루프문을 사용하지 않아도 된다.

■ 91. 문자열에 있는 문자 개수 구하기 (count)

[문제 281](#)

```
예제 : txt = ' A lot of thing occur each day, every day.'  
word_count1 = txt.count('o')  
word_count2 = txt.count('day')  
word_count3 = txt.count(' ')  
  
print(word_count1)  
print(word_count2)  
print(word_count3)
```

■ 92. 문자열에서 특정 문자(열) 위치 찾기(find)

예제 :

```
txt = 'A lot of things occur each day, every day.'  
  
result1 = txt.find('e')  
result2 = txt.find('day')  
  
print( result1 )  
print( result2 )
```


■ 93. 문자열을 특정 문자(열)로 분리하기 (split)

[문제 283](#)

예제:

```
url = 'http://www.naver.com/news/today=20181205'  
log = 'name:홍길동 age:17 sex:남자 nation:조선'  
  
result = url.split('/')  
print(result)
```

■ 94. 문자열을 특정 문자(열)로 결합하기(join)

예제 :

```
loglist = ['2018/12/05', '200', 'ok', '이 또한 지나가리라']  
bond = ';'

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

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

[문제 284](#)

```
예제 : txt = 'My password is 1234'
      result1 = txt.replace('1', '0')
      print(result)
```

■ 96. 문자열을 바이트 객체로 바꾸기 (encode)

* 인코딩이란 ?

"인코딩이란 사람이 인지할 수 있는 형태의 데이터를 약속된 규칙에 의해 컴퓨터가 사용하는 0과 1로 변환하는 과정을 말한다."

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

미국에서 제정된 ASCII(미국 정보 교화 표준 부호)는 1960년대 제정된 문자집합으로 이후 개발된 문자집합의 토대를 이루고 있다.

ASCII는 7비트 만을 이용해서 음이 아닌 수 (0 ~ 127)의 문자집합내의 문자를 할당한 약속이다.

예 : 숫자 65는 'A', 97은 'a'를 할당했다.

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

```
A
65
```

이 아스키 코드표로 미국 엔지니어들은 행복하게 살아가고 있었지만 다른나라 사람들은 그러지 못했다.

1. 서유럽

독일어의 움라우트, 스페인어의 물결표, 터키어의 시딜러, 프랑스어의 악센트는 7비트로 표현하지 못했다.
그래서 미국 표준을 보완한 새로운 국제 표준이 등장했다.

미국 표준인 0 ~ 127 영역은 그대로 유지하고 128 ~ 255 영역에 서유럽어에서 필요한 문자를 추가했다.

128(2의 7승)에 256(2의 8승)인 8비트로 표현하는 문자집합을 추가해서 서유럽어는 컴퓨터로 구현 가능하게 했다.

2. 아시아

아시아에서는 한자, 일본어, 한글이 8비트에 들어맞지 않았다.
아시아를 위해서 더블 캐릭터 셋 16비트로 된 문자집합을 제공해서 해결했다.
65,536(2의 16승) 개의 문자를 할당할 수 있다.

EUC-KR, CP949가 여기에 해당된다.

그런데 갑자기 크게 두가지 문제가 발생했다.

1. 포스코 더 샵 아파트

```
insert into emp(empno, ename, address)
values(1111, '김인호', '포스코더샵')
```

```
select * from emp;
```

```
1111, 김인호, 포스코더?
```

2. EUC-KR를 지원하는 일본어용 소프트웨어를 한글윈도우에서 실행하면 텍스트 깨진다.

이 문제를 해결하기 위해서 나온게 바로 유니코드이다.
UTF-7, UTF-8, UTF-6, UTF-32
이중에 UTF-8을 가장 많이 사용한다.

```
f = open('c:₩₩ladybug3.txt','r',encoding = 'UTF-8')
```

■ 97. 바이트 객체를 문자열로 바꾸기 (decode)

[문제 285](#)

encoding : 사람이 알아볼 수 있는 언어 --> 컴퓨터가 알아볼 수 있는 숫자로 변경
decoding : 컴퓨터가 알아볼 수 있는 언어 --> 사람이 알아볼 수 있는 언어로 변경

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

       c_txt = b_txt.decode()
       print(c_txt)
```

■ 98. 버블정렬

예제 286. 아래의 리스트를 버블정렬하는 함수를 생성하시오 !
(loop 문으로 수행)

```
def bubble_sort(data):
    dummy = 0                #자리를 바꾸기 위한 더미변수
    for j in range(len(data)):
        for i in range(len(data)-1):
            if data[i] > data[i+1]:
                dummy = data[i]
                data[i] = data[i+1]
                data[i+1] = dummy
        return data

a = [5, 4, 3, 2, 1, 8, 7, 10]
print(bubble_sort(a))
```

예제 287. 버블정렬을 재귀로 구현하시오 !(오늘의 마지막 문제)

```
def bubble_sort(data, j=0, i=0):
    dummy = 0
    if data[i] > data[i+1]:
        dummy = data[i]
        data[i] = data[i+1]
        data[i+1] = dummy
    if i == len(data)-2:
        j = j+1
        i = -1
    if j <= len(data):
        bubble_sort(data, j, i+1)
    return data

a = [5, 4, 3, 2, 1, 8, 7, 10]
print(bubble_sort(a))
```

■ 99. 순차적인 정수 리스트 만들기 (range)

[문제 288 ~ 289](#)

예제 :

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


■ 100. 리스트 요소를 무작위로 섞기 (shuffle)

[문제 290](#)

" 순서용 자료를 뒤죽박죽 섞어 놓는 함수 "

예제 :

```
from random import shuffle
listdata = list(range(1,11))
for i in range(3):
    shuffle(listdata)
    print(listdata)
```

■ 100. 리스트에서 특정 위치의 요소 얻기

예제 291. 아래의 리스트에서 숫자 2 와 알파벳 b 와 숫자 5를 출력하시오 !

```
listdata = [ 1, 2, 'a', 'b', 'c', [4,5,6] ]  
  
print( listdata[1], listdata[3], listdata[5][1])
```

■ copy 와 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]
```

이번에는 가리키지 말고 복사 하는 방법은 없는가 ?

a 가 가지고 있는 값으로 b 변수를 생성할때 같은 값으로 복사하는 방법은 ?

1. [:] 을 이용한다.

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

3. deepcopy 모듈을 이용한다.

```
a = [1,2,[3,4]]
b = copy(a)
```

```
print(b) # [1,2,[3,4]]
```

```
a[2][0] = 5
print(b) #[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(b) # [1,2,[3,4]]
```

※ 설명 : 리스트 안의 리스트가 있으면 deepcopy 를 써서 Copy 해야 한다.

■ 101. 리스트에서 특정 요소의 위치 구하기(index)

예제 292. 아래의 리스트를 가지고 아래의 결과를 출력하시오 !

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

결과 : 지구는 태양계에서 3 번째에 위치하고 있습니다.
지구는 태양계에서 9 번째에 위치하고 있습니다.

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

```
for i in range(len(a)):
    if a[i] == '지구':
        print('지구는 태양계에서 %d 번째에 위치하고 있습니다.' %i)
```

↓

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성', '지구', '목성', '지구']
print('지구는 태양계에서 ' + str(a.index('지구')) + ' 번째에 위치하고 있습니다.')
print('지구는 태양계에서 ' + str(a.index('지구', -3)) + ' 번째에 위치하고 있습니다.')
print('지구는 태양계에서 ' + str(a.index('지구', -1)) + ' 번째에 위치하고 있습니다.')
```

■ 102. 리스트에서 특정위치의 요소 변경하기

예제 293. 아래의 리스트에서 화성을 영어로 Mars 라고 변경 하시오 ! (단 숫자를 이용하지 말고 리스트의 index 메소드를 활용해서 수행하시오)

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

결과 :

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

■ 103. 리스트에서 특정 구간에 있는 요소 추출하기

예제 294. 아래의 리스트에서 아래의 결과를 출력하시오 !

```
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']

```
print('태양계의 암석형 행성 : ' + a[1:4])
```

```
print('태양계의 가스형 행성 : ' + a[4:])
```

■ 104. 리스트에서 짝수번째 요소만 추출하기

문제 295. 아래의 리스트에서 짝수번째 요소와 홀수번째 요소를
각각 출력하시오 ! (점심시간 문제)

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']  
print(str(a[1::2]))  
print(str(a[::2]))
```


■ 105. 리스트 요소 순서를 역순으로 만들기 ① (reverse)

예제 296. 아래의 리스트를 역순으로 만드시오 !

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

# 결과 :
# ['해왕성', '천왕성', '토성', '목성', '화성', '지구', '금성', '수성', '태양']

a.reverse()
print(a)
```

■ 106. 리스트 요소 순서를 역순으로 만들기 ②

예제 297. 이번에는 아래의 리스트를 역순으로 만들어서 b라는 변수에 저장해서 출력되게 하시오 !

```
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']
```

※ 설명 : reverse 와 reversed 의 차이는 ?

reverse 는 a 의 요소를 완전히 변경해버리지만 reversed 는 변경하지 않는다.

■ 107. 리스트 합치기 (+)

예제 298. 아래의 두개의 리스트를 하나로 합쳐서 출력하시오 !

```
listdata1 = ['a', 'b', 'c', 'd', 'e']  
listdata2 = ['f', 'g', 'h', 'i', 'j']  
  
print(listdata1 + listdata2)  
#['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

■ 108. 리스트 반복하기 (*)

문제 299. 아래의 리스트로 아래의 결과를 출력하시오 !

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

```
# 결과 :  
# [0, 1, 2, 0, 1, 2]
```

■ 109. 리스트에 요소 추가하기 (append)

예제 300. 아래와 같이 리스트에 추가할 값을 물어보게 하고 입력하면 리스트에 내용이 출력되게 하시오 !

```
def add_list():
    a=[]
    while True:
        a.append(input('리스트에 추가할 값을 입력하세요 ~'))
        print(a)

print( add_list() )

#리스트에 추가할 값을 입력하세요 ~a
#['a']
#리스트에 추가할 값을 입력하세요 ~b
#['a', 'b']
#리스트에 추가할 값을 입력하세요 ~c
#['a', 'b', 'c']
```

■ 110. 리스트의 특정 위치에 요소 삽입하기 (insert)

문제 301. 아래의 a 리스트에 '소행성'이라는 요소를 입력하시오 !

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

print(a)
# 결과 :
# ['태양', '수성', '금성', '지구', '화성', '소행성', '목성', '토성', '천왕성', '해왕성']
```

■ 111. 리스트의 특정위치의 요소 제거하기 (del)

[문제 302](#)

※ del 과 remove 의 차이?

remove 는 요소명으로 삭제하는 것 : `a.remove('토성')`
del 은 리스트의 인덱스번호로 삭제하는 것 : `del a[-3]`

■ 112. 리스트에서 특정 요소 제거하기 (remove)

[문제 303](#)

※ del 과 remove 의 차이?

remove 는 요소명으로 삭제하는 것 : `a.remove('토성')`
del 은 리스트의 인덱스 번호로 삭제하는 것 : `del a[-3]`

■ 113. 리스트에서 특정 구간에 있는 모든 요소 제거하기 (remove)

문제 304. 아래의 리스트에서 토성을 포함해서 토성 이후에 나오는 요소를 지우시오 !

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

■ 114. 리스트에 있는 요소 개수 구하기 (len)

■ 115. 리스트에서 특정 요소의 개수 구하기 (count)

문제 305. len 과 count 의 차이가 무엇인가 ?
아래의 리스트로 테스트 해보시오

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

print(len(a))
print(a.count('지구'))
# 9
# 1
```

※ 설명 : len 은 리스트 전체의 요소의 갯수를 세는 함수이고 count 는 리스트 안에 특정 요소가 몇건 있는지 세는 함수

■ 116. 리스트 제거하기 (del)

```
namelist = ['Mary', 'Same', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']  
del namelist  
  
print(namelist)  
# NameError: name 'namelist' is not defined
```

■ 117. 리스트 요소 정렬하기 ① (sort)

[문제 306 ~ 307](#)

■ 120. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기 (enumerate)

[문제 308](#)

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

예제 :

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

0 body

1 foo

2 bar

↑

index

■ 121. 리스트의 모든 요소의 합 구하기 (sum)

예제 309. 아래의 리스트의 요소를 다 더한 값을 출력하시오 !

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

print(sum(listdata))
# 65
```

■ 122. 리스트 요소가 모두 참인지 확인하기 (all, any)

문제 310. 아래의 리스트 요소중에서 하나라도 True가 있으면 True를 출력하시오 !

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

print( any(a))      #True
print( all(a))      #False
```

■ 123. 파이썬에서 _(언더바)를 사용하는 경우

1. 인터프리터에서 마지막 값을 저장할 때
2. 값을 무시하고 싶을 때
3. 여러개의 값을 무시하고 싶을 때
3. 변수나 함수명에 특별한 의미를 부여하고 싶을 때
4. 숫자 또는 문자값의 자릿수 구분을 위한 구분자로서 사용할 때

예제 :

1. 인터프리터에서 마지막 값을 저장할 때

```
>>> 10
>>> _
>>> _*3
30
>>> _*20
600
```

2. 값을 무시하고 싶을 때

```
>>> x, y = (1, 2)
>>> x
>>> y
>>> x, _, y = (1, 2, 3)
>>> x
1
>>> y
3
```

3. 여러개의 값을 무시하고 싶을 때

```
>>> x, *__, y = (1, 2, 3, 4, 5)
>>> x
1
>>> y
5
```

4. for loop 문 사용시 index 를 무시하고 싶을 때

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

5. 변수나 함수명에 특별한 의미를 부여하고 싶을 때

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

6. 숫자 또는 문자 값의 자릿수 구분을 위한 구분자로서 사용할 때

```
dec_base = 1_000_000_000_000_000
print(dec_base)
```


■ 123. 사전에 요소 추가하기

[문제 311 ~ 312](#)

예제 :

```
sol = {}
```

```
sol['태양'] = 'sum'
```

↑

키

↑

값

```
print(sol)
```

소녀시대 : 소원을 말해봐, Gee, Oh!

비틀즈 : Let it be, Yesterday, Hey jude, Imagine

↑

키

↑

값들

■ Defaultdict 이 무엇인지 ?

[문제 313 ~ 316](#)

"defaultdict()은 말 그대로 dictionary의 기본값을 정의하고 값이 없더라도 에러를 출력하지 않고 기본값을 출력한다."

예제 :

```
d = {}  
d['one'] = 'a'  
  
print(d)  
#결과 : {'one': 'a'}
```

※ 설명 : 딕셔너리를 구성하려면 위와 같이 키와 값 두개가 동시에 필요하다.

위의 코드를 다시 수행하는데 아래와 같이 해보시오 !

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

그런데 defaultdict()를 사용하면 키만 생성하고 값이 없어도 에러가 나지 않는다.

```
from collections import defaultdict  
  
d3 = defaultdict(list)  
d3['one']  
d3['two']  
  
print(d3)  
# defaultdict(<class 'list'>, {'one': [], 'two': []})
```

■ 124. 사전의 특정 요소 값 변경하기

예제 317. 아래의 딕셔너리의 값중에서 Let it be를 Woman으로 변경하시오 !

```
dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']

dic['Beatles'][1] = 'Woman'
print(dic)
# 결과 : {'Beatles': ['Yesterday', 'Woman', 'Hey Jude'], '소녀시대': ['소원을 말해봐',
#                               'Gee', '다만세']}
```

■ 125. 사전의 특정 요소 제거하기 (del)

예제 318. 아래의 딕셔너리에서 소녀시대의 키와 값을 전부 지우시오 !

```
dic = {}  
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']  
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']  
del dic['소녀시대']  
  
print(dic)  
# 결과 : {'Beatles': ['Yesterday', 'let it be', 'Hey Jude']}
```

■ 126. 사전의 모든 요소 제거하기 (clear)

예제 319. 아래의 리스트에 모든 요소를 다 삭제하시오 !

```
dic = {}  
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']  
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']  
dic.clear()  
  
print(dic)  
# 결과 : {}
```

■ 127. 사전에서 키만 추출하기 (keys)

예제 320. 아래의 딕셔너리에서 Key에 해당하는 부분만 추출하시오 !

```
dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']

pr_dic = ''
for i in dic.keys():
    pr_dic += i + ', '
print(pr_dic[:-2])
# 결과 : Beatles, 소녀시대
```

■ 128. 사전에서 값만 추출하기 (values)

[문제 321 ~ 325](#)

■ 129. 사전 요소를 모두 추출하기 (items)

[문제 326](#)

- * 딕셔너리 자료형에서 데이터를 추출하는 방법
 1. key 만 추출 : `box.keys()`
 2. 값만 추출 : `box.values()`
 3. 둘다 추출 : `box.items()`

■ 130. 사전 정렬하기(sorted)

[문제 327 ~ 328](#)

■ 131. 문자 코드 값 구하기(ord)

[문제 329](#)

ASCII 코드 값을 출력하는 함수

A -----> 65
 인코딩

예제 : `print(ord('A'))`

■ 132. 코드값에 대응하는 문자 얻기(chr)

예제 : `print(chr(65))`

■ 133. 문자열로 된 식을 실행하기 (eval)

[문제 330](#)

"문자열을 명령문으로 인식하게 하는 함수"

예제 :

```
a = '2+3'
print(a, '=', eval(a))
# 결과 : 2+3 = 5
```

■ 134. 이름없는 한줄짜리 함수 만들기(lambda)

[문제 331 ~ 332](#)

```
예제 : add = lambda x,y : x+y
           입력값   실행문
       print(add(1,3))
```

■135. 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기 (map)

[문제 333 ~ 334](#)

map(A 함수명, A 함수에 제공할 매개변수값들)
매개변수 값들을 바꿔가면서 함수를 반복 실행 하고자 할 때 사용한다.

예제 :

```
multi = lambda x : x*x
a = [1, 2, 3, 4, 5]
b = map(multi, a)      # map (숫자 or 문자함수, 함수인자) 숫자 or 문자함수를 리스트원소
                        # 마다 적용해 b 메모리 주소에 저장(출력할때는 list 화해서 출력)

print(list(b))
[1, 4, 9, 16, 25]
```

■ 136. 텍스트 파일을 읽고 출력하기 (read)

예제 :

```
f = open('c:\\\\stockcode_part.txt','r')
data = f.read()
print(data)
f.close()          # 파일을 닫는다.
                  # 파일모습 그대로 한번에 출력
```

■ 137. 텍스트 파일을 한줄 씩 읽고 출력하기 (readlines)

[문제 335 ~ 336](#)

예제 :

```
f = open('c:WWWstockcode_part.txt','r')
data = f.readlines()
print(data)
f.close()
#리스트형식으로 라인마다 나뉘서 출력
```


■ 139. 화면에서 사용자 입력을 받고 파일로 쓰기 (write)

[문제 337](#)

예제 :

```
text = input('파일에 저장할 내용을 입력하세요 ~ ')
```

```
f = open('c:\\mydata.txt', 'w')
```

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

```
f.close()
```

■ 140. 텍스트 파일에 한줄씩 쓰기 (writelines)

```
text_save = []
f = open('c:\\mydata.txt', 'w')
while True:
    text = input('파일에 저장할 내용을 입력하세요 ~ ')
    text_save.append(text + '\n')
    if text == '':
        break
f.writelines(text_save)
f.close()
```

※ writelines 는 리스트 자료형도 파일에 저장할 수 있다.

■ 141. 텍스트 파일 복사하기 (read, write)

예제 :

```
f = open('c:Wstockcode_part.txt', 'r')
h = open('c:Wstockcode_copy.txt', 'w')

data = f.read()
h.write(data)

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

※ 특정 폴더 밑에 있는 파일들을 파이썬에서 읽어오는 방법

- 폐결절 사진과 정상 폐사진
- 개와 고양이 사진

```
import os
path_dir = 'c:WW'
file_list = os.listdir(path_dir)
file_list.sort()
for items in file_list:
    print(items)
```

- 파일명만 가지고 오는 듯?

■ 142. 바이너리 파일 복사하기 (read, write)

[문제 338](#)

"이미지 파일을 파이썬으로 복사하는 방법"

예제 :

```
bufsize = 1024                                # 이미지를 저장할 최소 기본 메모리 사이즈
f = open('c:WW1.png', 'rb')                   # 이미지가 바이너리 파일이기때문에
h = open('c:WW1_copy.png', 'wb')              # (이진파일, 숫자로된 파일)

data = f.read(bufsize)

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

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

■ 143. 파일을 열고 자동으로 닫기 (with ~ as)

[문제 339](#)

예제 :

```
f = open('c:\\\\stockcode.txt','r')
data = f.readlines()
print(data)
f.close()                # 파일을 닫는 코드를 명시해줬다.
```

■ 144. 파일의 특정 부분만 복사하기 (seek, read, write)

"파일을 전부 copy 하는게 아니라 일부분만 copy 하는 방법"

예제 :

```
spos = 105      # 파일을 읽는 위치 지정 105 바이트 위치부터
size = 500      # 읽을 크기를 지정 500 바이트 까지
f = open("c:\\\\stockcode_part.txt", 'r')
h = open('c:\\\\stockcode.txt', 'w')

f.seek(spos)
data = f.read(size)
h.write(data)

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

■ 145. 파일 크기 구하기 (os.path.getsize)

예제 :

```
from os.path import getsize

file1 = "c:WWWstockcode_part.txt"
file2 = "c:WWW1.png"

file_size1 = getsize(file1)
file_size2 = getsize(file2)

print( 'file name : %s / file size : %d' %(file1, file_size1))
print( 'file name : %s / file size : %d' %(file2, file_size2))
```

■ 146. 파일 삭제하기 (osremove)

[문제 340](#)

예제 :

```
from os import remove
target_file = 'c:\WWWstockcode_copy.txt'
k = input('[%s] 파일을 삭제하시겠습니까? ( Y / N)' %target_file)
if k == "Y":
    remove(target_file)
    print('[%s]를 삭제했습니다.' %target_file)
```

설명 : shift + delete 와 같은 작동이 일어난다.

■ 147. 파일 이름 바꾸기 (osrename)

예제 :

```
from os import rename
target_file = 'c:Wwstockcode.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:Wstockcode.txt]에 대한 새로운 파일 이름을 입력하세요 : c:Wwffff.txt
# [c:Wstockcode.txt] --> [c:Wwffff.txt]로 파일이름이 변경되었습니다.
```

■ 148. 파일 위치 바꾸기

```
from os import rename

target_file = '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)
```

■ 149. 디렉토리에 있는 파일목록 얻기 (os.listdir)

[문제 341 ~ 346](#)

"딥러닝때 사용할 데이터 전처리 코드 작성"

※ 딥러닝 수업때 사용할 코드가 크게 2가지

1. 이미지 데이터를 파이썬으로 로드하는 함수
2. 신경망 구현함수

■ 150. 워킹 디렉토리 확인하고 바꾸기

- 워킹 디렉토리 확인하기

```
import os
pdir = os.getcwd()
print(pdir)
# C:\Users\WAdministrator\WPycharmProjects
```

↑

프로젝트 워킹 디렉토리 (project, .py 가 저장된다.)

- 워킹 디렉토리 변경하기

```
import os
pdir = os.getcwd()
os.chdir("c:WW")
print(os.getdir())
```

워킹 디렉토리가 c:WW 로 변경됨

■ 151. 디렉토리 생성하기 (osmkdir)

```
import os
newfolder = input('새로 생성할 디렉토리 이름을 입력하세요 : ')
os.mkdir(newfolder)

# 새로 생성할 디렉토리 이름을 입력하세요 : nerwe
# C:\Users\Administrator\PycharmProjects 에 nerwe 폴더가 생김 !
```

■ 152. 디렉토리 제거하기 (osrmdir)

[문제 346](#)

예제 :

c 드라이브 밑에 data7 이라는 폴더를 만들고 테스트 하시오 !

```
import os
target_folder = 'c:\\data7'
os.rmdir(target_folder)
```

■ 153. 하위 디렉토리 및 파일 전체 삭제하기 (shutil.rmtree)

```
import os
import shutil

target_folder= 'c:\data7'
shutil.rmtree(target_folder)
```

■ 154. 파일이 존재하는지 체크하기 (os.path.exists)

[문제 349](#)

c 드라이브 밑에 data7 폴더를 생성한 후에 테스트 하세요 ~

```
from os.path import exists
```

```
target_folder = 'c:\\data7'
exists(target_folder)
```


■ 155. 파일인지 디렉토리인지 확인하기 (os.path.isfile, os.path.isdir)

[문제 350](#)

예제 :

```
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)
```

■ 156. 현재 시간을 년-월-일 시:분:초 로 출력하기 (location, strftime)

```
from time import localtime, strftime
print( strftime('%Y-%m-%d %X', localtime()))
```

```
# 2018-12-11 12/11/18
```

■ 157. 올해 경과된 날짜수 계산하기 (localtime)

```
from time import localtime

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

print(start_day, elapsed_day)

# 2018-01-01 345
```

■ 158. 오늘의 요일 계산하기 (localtime)

```
from time import localtime

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

t = localtime()
print(t)

today = '%d-%d-%d' %(t.tm_year, t.tm_mon, t.tm_mday)

print( t.tm_wday)

week = weekdays[t.tm_wday]

print('[%s] 오늘은 [%s]입니다.' %(today, week))

# [2018-12-11] 오늘은 [화요일]입니다.
```

■ 159. 프로그램 실행 시간 계산하기 (카카오 4 번 문제)

```
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: 0:00:00.142008
# 총 계산 시간: 142ms
```

■ 160. 주어진 숫자를 천단위 구분하기

```
num = input('아무 숫자를 입력하세요: ')

if num.isdigit():
    num = num[::-1]
    ret = ''
    for i, c in enumerate(num):
        i += 1
        if i != len(num) and i%3 == 0:
            ret += (c + ',')
        else:
            ret += c
    ret = ret[::-1]
    print(ret)
else:
    print('입력한 내용 [%s]: 숫자가 아닙니다.' %num)

# 아무 숫자를 입력하세요: 999999999999
# 999,999,999,999
```

■ 161. 문자열의 각 문자를 그 다음 문자로 변경하기

예제 :

```
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)
```

```
# 문장을 입력하세요: wjdtjdgh
```

```
# jdtjdghw
```

■ 162 ~ 172 예제 : 웹 스크롤링

★ 저축해야할 웹스크롤링 함수

- | | | |
|---------------------------|----------------|----------------|
| 1. 웹 스크롤링 기본 문법 | (문제 351 ~ 368) | |
| 2. ebs 레이디 버그 게시판 댓글 스크롤링 | (문제 369 ~ 382) | ebs_scroll() |
| 3. 한겨레 신문사 웹 스크롤링 코드 구현 | (문제 383 ~ 387) | hani_scroll() |
| 4. 중앙일보 신문사 | (문제 388 ~ 390) | chuan_scroll() |
| 5. 동아일보 신문사 | (문제 391) | donga_scroll() |

주가에 영향을 주는 요소 빅데이터
기사를 읽는 인공지능 신경망(아직 없음)
(미래에셋 로보 어드바이저)

6. 이미지(사진) 데이터 스크롤링(selenium)

- 네이버 (문제 393)
- 구글 (문제 394)
- 다음 (문제 395)
- Bing (문제 396)
- 인스타그램 (문제 397)

웹스크롤링을 하기 위해서 사용할 파이썬 모듈 ?

"beautiful soup 모듈"

웹스크롤링을 하는 이유 ? "데이터 수집을 위해서"

데이터를 수집하는 방법 : 1. 웹스크롤링 기술 이용
2. 딥러닝 기술을 이용

* html 이란 ?

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

■ 173. ~ 177 파이썬으로 게임 프로그램 개발

※ 머신러닝이란 ? "컴퓨터가 스스로 학습하는 것"

※ 머신러닝의 종류 3가지 ?

1. 지도학습 : 라벨(정답)이 있는 데이터를 학습하는 것
2. 비지도 학습 : 라벨(정답)이 없는 데이터를 학습하는 것
3. 강화 학습 : 상과 벌을 통해서 데이터를 만들어 내며 학습하는 방법

알파고 --> 머신러닝이 붐을 일으켰다.



강화학습

■ 문제 모음

문제 1. 위의 예제를 따라서 문자형 변수를 만들고 type 을 확인하시오!

```
b= 'heaven'

print(type(b))
<class                                'str'>
```

문제 2. a 와 b 는 정말 같은 객체를 가리키는 것인가?

```
a = 3
b = 3
import sys
print( sys.getrefcount(3) )      # 3 이라는 정수형 객체를 참조하는 변수의 갯수를 확인할 수 있다.

6175
```

문제 3. d 로 만든 리스트 변수에서 숫자 1000 을 출력하시오 !

```
print(d[0])

1000
```

* 파이썬은 시작을 0 부터 시작한다.

문제 4. d list 변수 안에 있는 요소들을 하나씩 다 출력하시오 !

```
d = [1000, 2000, 3000, 4000]

for i in d:
    print(i)                # 앞에 4 을 띄어서 쓰는것을 파이썬 약속이다.
                           괄호를 생략하기 때문에 띄어쓰기를 기호로 쓴다.
                           괄호 대신 공백 !
```

* 설명 : 자바나 C 언어를 사용하던 사람들은 뒤에 세미콜론 붙이는 습관이 있는데 파이썬에서는 기본적으로 구문 뒤에 아무것도 붙이지 않는다. if 문, while loop, for loop, def, class 문의 뒤에는 콜론(:)을 꼭 붙여야 한다.

파이썬이 다른 언어보다 보기 쉽고 소스코드가 간결한 이유는 바로 콜론(:)을 사용해서 들여쓰기를 하도록 만들었기 때문이다.

콜론(:)뒤에 나오는 명령어를 다음 줄에 작성하려면 들여쓰기를 해야한다.

예제 :

```
for i in range(100):        # range 연속된 숫자를 생성
    print(i)
```

문제 5. emp2.csv 를 파이썬으로 로드해서 print 하시오 !

```
import csv                  # csv 파일을 가져와서 쓰겠다.
```

```

file = open("c:\\Wemp2.csv", 'r')          # file이라는 변수에 디렉토리를 오픈해라 ('r'은 read)
                                           # WW를 쓰는 습관을 들여라 (We 라는 예약어가 있어서)
emp_csv = csv.reader(file)                 # emp_csv 라는 변수에 파일을 읽어서 넣겠다.

for emp_list in emp_csv:                   # 모두 출력
    print(emp_list)

['7839', 'KING', 'PRESIDENT', '0', '1981-11-17', '5000', '0', '10']
['7698', 'BLAKE', 'MANAGER', '7839', '1981-05-01', '2850', '0', '30']
['7782', 'CLARK', 'MANAGER', '7839', '1981-05-09', '2450', '0', '10']
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']
['7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-10', '1250', '1400', '30']
['7499', 'ALLEN', 'SALESMAN', '7698', '1981-02-11', '1600', '300', '30']
['7844', 'TURNER', 'SALESMAN', '7698', '1981-08-21', '1500', '0', '30']
['7900', 'JAMES', 'CLERK', '7698', '1981-12-11', '950', '0', '30']
['7521', 'WARD', 'SALESMAN', '7698', '1981-02-23', '1250', '500', '30']
['7902', 'FORD', 'ANALYST', '7566', '1981-12-11', '3000', '0', '20']
['7369', 'SMITH', 'CLERK', '7902', '1980-12-09', '800', '0', '20']
['7788', 'SCOTT', 'ANALYST', '7566', '1982-12-22', '3000', '0', '20']
['7876', 'ADAMS', 'CLERK', '7788', '1983-01-15', '1100', '0', '20']
['7934', 'MILLER', 'CLERK', '7782', '1982-01-11', '1300', '0', '10']

```

문제 6. 이름과 월급을 출력하시오!

```

import csv

file = open("c:\\Wemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1], emp_list[5])

```

※ 파일 불러오는 과정을 꼭 써야 한다. 따로 DB에 저장시키는 개념이 아니다.

문제 7. 직업이 ANALYST, CLECK 인 사원들의 이름과 직업을 출력하시오 !

```

import csv

file = open("c:\\Wemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[2] in ('ANALYST', 'CLERK'):
        print(emp_list[1], emp_list[5])

JAMES 950
FORD 3000
SMITH 800
SCOTT 3000
ADAMS 1100
MILLER 1300

```

문제 8. 부서번호가 10번인 사원들의 이름과 월급과 부서번호를 출력하시오 !

```

import csv

file = open("c:\\Wemp2.csv", 'r')

```

```
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if int(emp_list[7]) == 10 :
        print(emp_list[1], emp_list[5], emp_list[7])

        KING 5000 10
        CLARK 2450 10
        MILLER 1300 10
```

문제 9. 숫자를 2개를 각각 아래처럼 물어보게 하고 아래처럼 출력되게 하시오 !

첫번째 숫자를 입력하세요 ~ 1

두번째 숫자를 입력하세요 ~ 2

a가 1 이고 는 2 보다 작습니다.

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

```
if a < b:
    print( a, '는 ', b, '보다 작습니다.')
else:
    print( a, '는 ', b, '보다 큼니다.')
```

문제 10. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 월급이 출력되게 하시오 !

이름을 입력하세요 ~ SCOTT
3000

```
a = str( input( '이름을 입력하세요 ~'))
```

```
import csv
```

```
file = open("c:\WWemp2.csv",'r')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    if a == emp_list[1]:
        print(emp_list[5])
```

문제 11. 위의 문제를 다시 수행하는데 scott 을 대문자로 입력하던 소문자로 입력하던 상관없이 출력되게 하시오!

```
a = str( input( '이름을 입력하세요 ~' ) )
```

```
import csv
```

```
file = open("c:\WWemp2.csv",'r')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    if a.upper() == emp_list[1]:
        print(emp_list[5])
```

이름을 입력하세요 ~scott

3000

이름을 입력하세요 ~SCOTT
3000

문제 12. 이름을 물어보게 하고 이름을 입력해서 해당 사원의 월급이 3000 이상이면 고소득자 입니다. 라는 메세지출력, 2000 ~ 3000 사이이면 적당합니다. 라는 메세지 출력, 2000 이하면 월급 인상자 입니다. 라는 메세지가 출력되게 하시오 !

답 1

```
a = str( input( '이름을 입력하세요 ~' ) )
import csv
file = open("c:WWemp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[1] == a.upper():
        if int(emp_list[5]) >= 3000:
            print( '고소득자 입니다.' )
        elif int(emp_list[5]) >= 2000:
            print( '적당합니다.' )
        else:
            print( '월급 인상자 입니다.' )

이름을 입력하세요 ~scott
고소득자 입니다.
```

답 2

```
a = str( input( '이름을 입력하세요 ~' ) )
import csv
file = open("c:WWemp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[1] == a.upper() and int(emp_list[5]) >= 3000:
        print( '고소득자 입니다.' )
    elif emp_list[1] == a.upper() and int(emp_list[5]) >= 2000:
        print( '적당합니다.' )
    elif emp_list[1] == a.upper() and int(emp_list[5]) < 2000:
        print( '월급 인상자 입니다.' )

이름을 입력하세요 ~scott
고소득자 입니다.
```

문제 13. 아래의 리스트에 있는 요소(값)들을 하나씩 뽑아서 출력하시오 !

```
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']

for i in ['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']:
    print( i )
```

문제 14. 숫자 1번부터 10000번까지 세로로 출력하시오 !

```
for i in range(10001):
    print(i+1)
```

문제 15. 아래와 같이 숫자 짝수를 출력하시오 !

2
4
6
8
10

```
for i in range(10):  
    if i%2 == 1:  
        print(i+1)
```

문제 16. 아래와 같이 결과를 출력하시오

1
3
5
7
9

```
for i in range(10):  
    if i%2 == 1:  
        print(i)
```

문제 17. for loop 문을 이용해서 숫자 1부터 10 까지 출력하시오 !

```
for i in range(1,11):          # range( 시작, 끝, 증가 ) 시퀀스 생성  
    print(i)
```

1
2
3
4
5
6
7
8
9
10

설명 : 2씩 증가하고 싶으면 ?

```
for i in range(1, 11, 2):  
    print(i)
```

```
>>>4*'k'
```

```
'kkkk'
```

```
>>>5*'★'
```

```
'★★★★★'
```

문제 18. 아래와 같이 결과를 출력하시오 !

★
★★

```

★ ★ ★
★ ★ ★ ★
★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★

```

```

for i in range(1,10):
    print( i*'★' )

```

문제 19. 숫자를 물어보게 하고 숫자를 입력하면 ★이 출력되게 하시오 !

숫자를 입력하세요 ~ 10

```

★
★ ★
★ ★ ★
★ ★ ★ ★
★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

```

```

a = int(input('숫자를 입력하세요 ~'))

```

```

for i in range(1,a+1):
    print( i*'★' )

```

문제 20. 숫자를 물어보게 하고 숫자를 입력하면 ★이 아래와 같이 출력되게 하시오 !

숫자를 입력하세요 ~10

```

★ ★ ★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★ ★
★ ★ ★ ★ ★
★ ★ ★ ★
★ ★ ★
★ ★
★

```

```

a = int(input('숫자를 입력하세요 ~'))

```

```

for i in range(a, 0, -1):
    print( i*'★' )

```

문제 21. 아래와 같이 숫자를 두 번 물어보게 하고 ★을 출력해서 사각형을 만드시오 !

가로의 숫자를 입력하세요 ~5

세로의 숫자를 입력하세요 ~6

```

★ ★ ★ ★ ★
★ ★ ★ ★ ★

```

```

★★★★★
★★★★★
★★★★★
★★★★★

```

```

a = int(input('가로의 숫자를 입력하세요 ~'))
b = int(input('세로의 숫자를 입력하세요 ~'))
for i in range(b):
    print( a*'★' )

```

문제 22. 아래와 같이 ★를 출력하시오 ! (댓글로 올리세요 ~)

```

  ★
 ★★
★★★
★★★★
★★★★★

```

```

a = int(input('가로의 숫자를 입력하세요 ~'))
for i in range(1,a+1):
    print( (a-i)*' ', i*'★' )

```

문제 23. 구구단 2 단을 출력하시오 !

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

```

```

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

```

문제 24. 구구단 2 단 부터 9 단 까지 출력하시오 !

```

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

```

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
:
:
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63

```


9 x 8 = 72
9 x 9 = 81

▷ 구구단을 세로가 아닌 가로로 출력하기 위해 알아야 하는 내용

```
[ SQL> select lpad(sal, 20, ' ')
[      from emp;
[
[      설명 : 월급을 출력하는데 전체 20 자리 잡고 출력하고 월급이 출력되고 남은 나머지 공간에
[      공백을 채워 넣어라 ~
[
[ python> 'k'.rjust(20)
[ '                k'
[
[ result = ''
[
[ for i in range(2, 10):
[     result += str(i) + ' x '      # +=는 전 루프결과에 현재루프결과를 덧붙인다아~(누적)
[     print(result)
```

문제 25. 구구단 2 단을 가로로 출력하시오 !

2x2=4 2x3=6 2x4=8 2x5=10 2x6=12 2x7=14 2x8=16 2x9=18

```
a = 2
result = ''
for i in range(2,10):
    result += (str(a) + 'x' + str(i) + '=' + str(a*i) + ' ')
print(result)
```

문제 26. 구구단 2 단부터 9 단까지 출력하시오 !

2x 2= 4	2x 3= 6	2x 4= 8	2x 5=10	2x 6=12	2x 7=14	2x 8=16	2x 9=18
3x 2= 6	3x 3= 9	3x 4=12	3x 5=15	3x 6=18	3x 7=21	3x 8=24	3x 9=27
4x 2= 8	4x 3=12	4x 4=16	4x 5=20	4x 6=24	4x 7=28	4x 8=32	4x 9=36
5x 2=10	5x 3=15	5x 4=20	5x 5=25	5x 6=30	5x 7=35	5x 8=40	5x 9=45
6x 2=12	6x 3=18	6x 4=24	6x 5=30	6x 6=36	6x 7=42	6x 8=48	6x 9=54
7x 2=14	7x 3=21	7x 4=28	7x 5=35	7x 6=42	7x 7=49	7x 8=56	7x 9=63
8x 2=16	8x 3=24	8x 4=32	8x 5=40	8x 6=48	8x 7=56	8x 8=64	8x 9=72
9x 2=18	9x 3=27	9x 4=36	9x 5=45	9x 6=54	9x 7=63	9x 8=72	9x 9=81

```
result = ''
for a in range(2,10):
    result = ''
    for i in range(2,10):
        result += (str(a).rjust(2) + 'x' + W
                    str(i).rjust(2) + '=' + W
                    str(a*i).rjust(2) + ' ')
    print(result)
```

문제 27. 숫자를 물어보게하고 숫자를 입력하면 해당 단이 가로로 출력되게 하시오 !

2x 2= 4 2x 3= 6 2x 4= 8 2x 5=10 2x 6=12 2x 7=14 2x 8=16 2x 9=18

```
result = ''
a = int( input( '숫자를 입력하세요 ~'))
for i in range(2,10):
    result += (str(a).rjust(2) + 'x' + W
               str(i).rjust(2) + '=' + W
               str(a*i).rjust(2) + ' ' )
print(result)
```

문제 28. (점심시간 문제) 아래와 같이 숫자를 물어보게하고 숫자를 입력하면 해당하는 ★마름모를 출력되게 하시오 !

예쁜거 뽑아줄게 숫자좀 불러보소 : 5

```
★
★★
★★★
★★★★
★★★★★
★★★★★
★★★★
★★★
★★
★
```

```
a = int(input('예쁜거 뽑아줄게 숫자좀 불러보소 : '))
for i in range(1,2*a+1):
    if i <= a:
        print( (a-i)*' ', i*'★' )
    else:
        print( (i-a)*' ', ((2*a) - i)*'★' )
```

문제 29. 숫자 1 부터 10 까지 출력하는데 중간에 5는 출력되지 않게 하시오 !

```
for i in range(10):
    if i == 5:
        continue
    print(i)
```

0
1
2
3
4
6
7
8
9

문제 30. 아래의 학생 점수 리스트를 이용해서 60 점 이상인 학생들에게는 축하 메시지를 보내고 나머지 학생들

에게는 아무런 메세지도 전하지 않는 프로그램을 작성하시오 !

```
jumsu = [ 90, 25, 67, 45, 80 ]          # 점수 입력
num=0

for i in jumsu:
    num = num + 1
    if i < 60:
        continue
    print ("%d번 학생 축하합니다. 합격입니다." %num)    # %num 이 %d 와 연결되어 학생번호 출력

1번 학생 축하합니다. 합격입니다.
3번 학생 축하합니다. 합격입니다.
5번 학생 축하합니다. 합격입니다.
```

문제 31. 위의 문제의 코드를 수정해서 아래와 같이 결과가 출력되게 하시오 !

```
1번 학생 90 점입니다. 축하합니다. 합격입니다.
3번 학생 67 점입니다. 축하합니다. 합격입니다.
5번 학생 80 점입니다. 축하합니다. 합격입니다.

jumsu = [ 90, 25, 67, 45, 80 ]
num=0

for i in jumsu:
    num = num + 1
    if i < 60:
        continue
    print ("%d번 학생"%num, "%d 점입니다. 축하합니다. 합격입니다." %jumsu[num-1])
```

문제 33. 우리반 테이블의 데이터를 출력하는데 나이가 30 살 이상인 학생들만 출력하시오 !

```
import csv

file = open("c:\Wemp7.csv",'r')
emp2_csv = csv.reader(file)

for emp2_list in emp2_csv:
    if int(emp2_list[1]) >= 30:
        print(emp2_list)

['김용원', '31', '1988-12-25', '컴퓨터과학과', 'dev@xerato.com', 'sk', '경기도 용인']
['오세희', '31', '1988-04-01', '치기공학과', 'sehee5542@hanmail.net', 'sk', '서울시']
['허석우', '40', '1979-05-31', '정보통신공학과', 'heosukwoo@gmail.com', 'lg', '경기']
['안혜진', '30', '1989-06-06', '통계학과', 'rpagkffo@naver.com', 'sk', '서울시 관악구']
['김진철', '33', '1986-02-26', '물리학과', '7and3@naver.com', 'sk', '서울시 관악구']
['김용식', '32', '1987-03-31', '분자생물학', 'kumbanova@gmail.com', 'lg', '경기도']
```

문제 34. 우리반 테이블의 데이터를 출력하는데 나이가 30 살 이상인 학생들만 출력하고 아래와 같이 메세지가 출력되게 하시오 !

김용원 학생 축하합니다. 늑었습니다.
오세희 학생 축하합니다. 늑었습니다.
허석우 학생 축하합니다. 늑었습니다.
안혜진 학생 축하합니다. 늑었습니다.
김진철 학생 축하합니다. 늑었습니다.
김용식 학생 축하합니다. 늑었습니다.

```
import csv

file = open("c:\\Wemp7.csv", 'r')
emp2_csv = csv.reader(file)

for emp2_list in emp2_csv:
    if int(emp2_list[1]) >= 30:
        print(emp2_list[0], "학생 축하합니다. 늑었습니다.")
```

문제 35. 우리반 테이블의 데이터 전체를 출력하는데 통계학과 학생들만 빼고 출력하시오 !
(continue 사용해서 출력하시오 !)

```
import csv

file = open("c:\\Wemp7.csv", 'r')
emp2_csv = csv.reader(file)

for emp2_list in emp2_csv:
    if emp2_list[3] in ('통계학과', '정보통계보험수리학과'):
        continue
    print(emp2_list)
```

문제 36. 위의 스크립트를 수정해서 1부터 100 까지 숫자가 출력되게 하는 loop 문을 만드는데 숫자를 물어보게
해서 해당 숫자에서 loop 문이 종료되게 하시오 !
(break 문 반드시 사용)

```
a = int( input( '숫자 불러' ) )
for i in range(1, 101):
    if i <= a:
        print(i)
    elif i == a:
        break
```

문제 37. while loop 문을 이용해서 숫자 1부터 100 까지 출력하는 스크립트를 만드는데
숫자를 물어보게 하고 해당 숫자만큼 숫자가 출력되게 하시오 !

```
숫자를 입력하세요 ~ 5
1
2
3
4
5
6

a = int( input( '숫자를 입력하세요 ~' ))
x = 0
while x <= 100:
```

```

x = x + 1
if x <= a:
    print(x)
if x > a:
    break

```

문제 38. 아래와 같이 숫자를 물어보게 하고 숫자를 입력하면 ★이 출력되게 하시오 !

```

숫자를 입력하세요 ~7
★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★

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

```

문제 39. 아래와 같이 출력되게 하시오 !

```

숫자를 입력하세요 ~7
★★★★★★★
★★★★★★★
★★★★★★
★★★★★
★★★★
★★★
★★
★

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

답 2
a = int(input('숫자를 입력하세요 ~'))
x = a+1
while x > 0:
    x -= 1
    print(x*'★')

```

문제 40. 아래와 같이 ★로 세모를 출력하시오

```

숫자를 입력하세요 ~4
★
★★
★★★

```

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

문제 41. 아래와 같이 결과를 출력하시오 ! (마지막 문제)
카페에 반드시 댓글로 달아주세요 !

숫자를 입력하세요 ~5

```
a = int(input('숫자를 입력하세요 ~'))
x = 0
while x < 4*a:
    x += 1
    if x < a:
        print((2*a-1)*' ', (a-x)*' ', x*'★')
    elif x < 2*a:
        print((x-a)*' ', (4*a-x)*'★')
    elif x < 3*a:
        print((3*a-x)*' ', x*'★')
    else:
        print((x-a)*' ', (4*a-x)*'★')
```

문제 42. 팩토리얼을 while loop 문으로 구현하시오 ! (점심시간 문제)

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

숫자를 입력하세요 ~5

문제 46. 소수점 43.2 에서 43.1 을 빼면 얼마인지 파이썬으로 구현하시오 !

```
a = 43.2
b = 43.1
a-b
```

```
Out[2]: 0.100000000000000142
```

설명 : 부동 소수형은 정밀도의 한계를 가지고 있다. 이것은 파이썬만의 문제가 아니라, 부동 소수형이 어떻게 구성되고 동작되는가를 정의하는 표준인 IEEE 754 를 따르는 모든 컴퓨터 시스템의 문제이다.

IEEE 754 는 제한된 메모리를 이용해서 (파이썬의 경우는 8 바이트) 실수를 표현하기 때문에 제한된 정밀도를 갖는다.

문제 47. 아래의 리스트 변수에서 positive 라는 단어는 몇개 나오는가 ?

```
word = ['winter', 'cold', 'positive', 'negative']
sum = 0
```

```
for i in word:
    if i == 'positive':
        sum = sum + 1
```

```
print(sum)
```

또는

```
word = ['winter', 'cold', 'positive', 'negative']
word.count('positive')
```

↑ ↑
객체 함수(메소드)

문제 48. word 리스트에서는 positive 와 negative 단어가 몇개가 들어있는가?

```
word = ['winter', 'cold', 'positive', 'negative']
sum = 0
for i in word:
    if i in ('positive', 'negative'):
        sum = sum + 1
print(sum)
```

2

문제 49. 겨울왕국 대본에는 긍정적인 단어가 몇개가 들어있는가?

라이나 생명 데이터 분석 프로젝트

보험 상담원이 상담사가 규칙에 맞는 단어를 사용해서 상담을 했는지 판독하는 프로젝트
상담원의 상담 내용 녹음(음성) --> 텍스트로 변환

예 : 90 일 이라고 이야기해야 하는데 3 달이라고 이야기하면 안되는데 규칙에 맞지 않는 단어를 사용한
상당사를 찾는데 사용하는 코드

```
winter = open("C:WWWwinter.txt",'r')
winter = winter.read().split(' ')
sum = 0

positive = open("C:WWWpositivewords.txt",'r')
positive_line = positive.read().split('\n')

for i in winter:
    if i.lower() in positive_line:
        sum += 1
print(sum)

330
```

문제 50. (오늘의 마지막 문제) while loop 문을 이용해서 최대 공약수를 출력하시오 !

```
a = int( input('첫번째 수를 입력하시오 !'))
b = int( input('두번째 수를 입력하시오 !'))
dummy = 0
i = 1
c = 1
if a <= b:
    dummy = a
    a = b
    b = dummy
while i < b:
    i += 1
    if a%i == 0:
        a = a / i
    if b%i == 0:
        b = b / i
        c = c * i
    i = 1
print('결과 : 최대 공약수는', c, '입니다.')
```

문제 51. 월급이 1000 에서 3000 사이인 직원들의 이름과 월급을 출력하시오 !

```
import csv

file = open("c:WWWemp_comm.csv",'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    if 1000 <= int(i[5]) <= 3000:
        print(i[1], i[5])

BLAKE 2850
CLARK 2450
JONES 2975
MARTIN 1250
ALLEN 1600
```

TURNER 1500
WARD 1250
FORD 3000
SCOTT 3000
ADAMS 1100
MILLER 1300

문제 52. 부서번호가 10, 20 번이 아닌 직원들의 이름과 부서번호를 출력하시오 !

```
import csv

file = open("c:\\emp_comm.csv", 'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    if int(i[7]) not in(10, 20):
        print(i[1], i[5])
```

BLAKE 2850
MARTIN 1250
ALLEN 1600
TURNER 1500
JAMES 950
WARD 1250

문제 53. 아래의 SQL 을 파이썬으로 구현하시오 !

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

```
import csv

file = open("c:\\emp_comm.csv", 'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    print(i[1], i[1][0])
```

문제 54. 아래의 SQL 을 파이썬으로 구현하시오 !

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

```
import csv

file = open("c:\\emp_comm.csv", 'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    if i[1][0] == 'S':
        print(i[1], i[1][0])
```

문제 55. 아래의 SQL 을 파이썬으로 구현하시오 !

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

```

        where ename, like '_M%';

import csv

file = open("c:\\WWemp_comm.csv",'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    if i[1][1] == 'M':
        print(i[1], i[1][0])

```

문제 56. 오라클이 rdbms를 만들어서 크게 성공을 했는데 rdbms의 꽃이 바로 join(조인)이다.

- * 조인의 방법 3가지
 1. nested loop join
 2. hash join
 3. sort merge join

문제 57. 아래의 SQL을 파이썬으로 구현하시오 !

```
"nested loop join"
```

```
SQL> select e.ename, d.loc
      from emp e, dept d
      where e.deptno = d.deptno;
```

```

import csv
file_emp = open("c:\\WWemp2.csv",'r')
emp_csv = csv.reader(file_emp)
for emp_list in emp_csv:
    file_dept = open("c:\\WWdept.csv",'r')
    dept_csv = csv.reader(file_dept)
    for dept_list in dept_csv:
        if int(dept_list[0]) == int(emp_list[7]):
            print(emp_list[1], dept_list[2])

```

문제 58. 위의 코드를 이용해서 아래와 같이 a값에 들어갈 숫자를 물어보게하고 결과가 출력되게 하시오 !

```

a 값에 들어갈 숫자를 입력하세요 ~10
3.0

a = int( input( 'a 값에 들어갈 숫자를 입력하세요 ~'))

a += 1
a -= 5
a *= 2
a /= 4
print(a)

```

문제 59. True를 사용해서 while loop문의 무한 루프문을 구현해서 숫자 7이 반복해서 출력되게 하시오 !

```

while True:
    print(7)

```

문제 60. pandas 모듈을 이용해서 emp.csv 를 출력하시오 !

문제 61. 직업이 SALESMAN 이 아닌 직원들의 이름과 직업을 출력하시오 !

- pandas 이용하지 않았을 때

```
import csv

file = open("c:\Wemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[2] != 'SALESMAN':
        print( emp_list[1], emp_list[2])
```

문제 62. 위의 결과를 Pandas 모듈을 이용해서 출력하시오 !

```
import pandas as pd

emp = pd.read_csv("c:\Wemp.csv")

result = emp[ ['ename', 'job'] ] [ emp[ 'job' ] != 'SALESMAN' ]
print(result)
```

	ename	job
0	KING	PRESIDENT
1	BLAKE	MANAGER
2	CLARK	MANAGER
3	JONES	MANAGER
7	JAMES	CLERK
9	FORD	ANALYST
10	SMITH	CLERK
11	SCOTT	ANALYST
12	ADAMS	CLERK
13	MILLER	CLERK

문제 63. 판다스를 이용해서 emp.csv 에서 월급이 3000 이상인 직원들의 이름과 월급을 출력하시오 !

```
import pandas as pd

emp = pd.read_csv("c:\Wemp.csv")

result = emp[ ['ename', 'sal'] ] [ emp[ 'sal' ] >= 3000 ]
print(result)
```

	ename	sal
0	KING	5000
9	FORD	3000
11	SCOTT	3000

문제 64. 직업이 SALESMAN 이고 월급이 1200 이상인 직원들의 이름과 월급과 직업을 출력하시오 !

- pandas 이용하지 않았을 때

```
import csv

file = open("c:\\emp2.csv", 'r')
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    if emp_list[2] == 'SALESMAN' and int(emp_list[5]) >= 1200:
        print(emp_list[1], emp_list[2])
```

```
MARTIN SALESMAN
ALLEN SALESMAN
TURNER SALESMAN
WARD SALESMAN
```

- pandas 이용했을 때

```
import pandas as pd

emp = pd.read_csv("c:\\emp.csv")

result = emp[ ['sal', 'job'] ]W
            [ (emp[ 'sal' ] >= 1200) & W
              (emp[ 'job' ] == 'SALESMAN') ]
print(result)
```

```
      sal      job
4  1250  SALESMAN
5  1600  SALESMAN
6  1500  SALESMAN
8  1250  SALESMAN
```

문제 65. 직업이 SALESMAN, ANALYST 인 직원들의 이름과 직업을 출력하시오 !

- 판다스 이용하지 않았을 때

```
import csv
file = open("c:\\emp2.csv", 'r')
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    if emp_list[2] == 'SALESMAN' or W
    emp_list[2] == 'ANALYST':
        print( emp_list[1], emp_list[2] )
```

```
MARTIN SALESMAN
ALLEN SALESMAN
```

TURNER SALESMAN
 WARD SALESMAN
 FORD ANALYST
 SCOTT ANALYST

- 판다스 이용했을 때

```
import pandas as pd

emp = pd.read_csv("c:\\emp.csv")

result = emp[ ['ename', 'job'] ]W
           [ (emp[ 'job' ].isin(W
           [ 'SALESMAN', 'ANALYST' ] ) ) ]
print(result)
```

	ename	job
4	MARTIN	SALESMAN
5	ALLEN	SALESMAN
6	TURNER	SALESMAN
8	WARD	SALESMAN
9	FORD	ANALYST
11	SCOTT	ANALYST

문제 66. 이름의 첫번째 철자가 S로 시작하는 직원들의 이름을 출력하시오 !

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

- pandas 이용하지 않은 코드

```
import csv

file = open("c:\\emp_comm.csv", 'r')
emp_comm = csv.reader(file)
for i in emp_comm:
    if i[1][0] == 'S':
        print(i[1], i[1][0])
```

- pandas 를 이용한 코드

```
import pandas as pd

emp = pd.read_csv("c:\\emp.csv")
result = emp['ename'].apply(lambda x : x[0] == 'S') # .apply(함수명 또는 lambda 표현식)
                                                    결과는 논리값( True or false)

result2 = emp[['ename']] [result==True]
print( result2 )
```

문제 67. 아래와 같이 함수를 실행하면 결과가 아래와 같이 출력되는 함수를 생성하시오 !

```

        print( gop(10,20) )

def gop(x,y):
    return x * y
print ( gop(10, 20))

200

```

문제 68. 위의 코드를 lambda 표현식으로 나타내시오 !

↓
이름없는 함수 또는 여러줄의 코드를
딱 한줄로 표현하기 위한 문법

```

print ( (lambda x, y : x * y) (10,20) )
200

```

문제 69. 오라클의 nvl 함수를 파이썬으로 구현하고 이름과 커미션을 출력하는데 커미션이 null 인 사원들은 0으로 출력되게 하시오 !

```

def nvl(x,y):
    if x is '':
        return y
    return x

import csv
file = open("c:\\emp_comm.csv",'r')
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1], nvl(emp_list[6], 0) )

KING 0
BLAKE 0
CLARK 0
JONES 0
MARTIN 1400
ALLEN 300
TURNER 0
JAMES 0
WARD 500
FORD 0
SMITH 0
SCOTT 0
ADAMS 0
MILLER 0

```

문제 70. 위의 결과를 nvl 함수를 만들지 말고 lambda 표현식으로 구현하시오 !

```

import csv
file = open("c:\\emp_comm.csv",'r')
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print(emp_list[1],W
          ( lambda x,y :y if(x is '') else x )W          # 한줄로 쓰는 if 문을 알아야 한다.
          (emp_list[6],0) )

```

※ 코드 설명 : if 문을 한줄로 표현하는 파이썬의 comprehension
 y if (x is '') else x
 x가 null 일 때는 y를 출력하고 그렇지 않으면 x를 출력하라.

KING 0
 BLAKE 0
 CLARK 0
 JONES 0
 MARTIN 1400
 ALLEN 300
 TURNER 0
 JAMES 0
 WARD 500
 FORD 0
 SMITH 0
 SCOTT 0
 ADAMS 0
 MILLER 0

문제 71. 이름의 두번째 철자가 M인 직원들의 이름을 pandas로 구현하시오 !

```
import pandas as pd

emp = pd.read_csv("c:\\W\\emp.csv")
result = emp['ename'].apply(lambda x : x[1] == 'M')
result2 = emp[['ename']] [result==True]
print( result2 )

    ename
10  SMITH
```

문제 72. 두개의 숫자를 물어보게 하고 두개의 숫자 사이의 모든 숫자를 다 더한 값이 출력되게 하시오 !

```
첫번째 수를 입력하시오 !1

두번째 수를 입력하시오 !10

a = int( input('첫번째 수를 입력하시오 !'))
b = int( input('두번째 수를 입력하시오 !'))
c = 0
for i in range(a,b+1):
    c += i
print(c,'입니다.')
```

문제 73. 이름의 끝 글자가 T로 끝나는 직원들의 이름을 출력하시오 !

```
import csv

file = open("c:\\W\\emp_comm.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[1][-1] == 'T':
```



```
print ( emp_list[1] )
```

SCOTT

예제 :

```
strdata = 'Time is money!!'  
listdata = [1, 2, [4, 5, 6] ]
```

문제 74. 위의 예제에서 strdata 문자열 변수에서 철자 i 를 출력하시오 !

```
print ( strdata[5] )  
i
```

문제 75. 아래의 listdata 변수에서 숫자 2 를 출력하시오 !

```
listdata = [1, 2, [4, 5, 6] ]  
  
print ( listdata[1] )  
print ( listdata[2][1] )  
2
```

문제 76. 아래의 listdata 변수에서 숫자 4 를 출력하시오 !

```
listdata = [1, 2, [4, 5, 6] ]  
  
print ( listdata[2][0] )  
4
```

문제 77. strdata 에서 아래의 문자만 출력하시오 !

```
strdata = 'Time is money'  
  
print ( strdata[0:7] )  
print ( strdata[ :7] )
```

문제 78. strdata 문자열 변수값에서 짝수번째 철자만 출력하시오 !

```
strdata = 'Time is money'  
  
print ( strdata[::2])
```

문제 79. 이름을 출력하고 이름의 첫번째 철자부터 세번째 철자까지 아래와 같이 출력되게 하시오 !

```
import csv  
  
file = open ( "c:WWemp_comm.csv", 'r' )  
emp_csv = csv.reader(file)
```

```

for emp_list in emp_csv:
    print( emp_list[1], emp_list[1][:3] )

```

문제 80. 위의 결과를 pandas 를 이용해서 출력하시오 !

```

import pandas as pd
emp = pd.read_csv("c:\\emp.csv")
result = emp['ename']

print (result.apply(lambda x : x[:3] ) )

```

```

0    KIN
1    BLA
2    CLA
3    JON
4    MAR
5    ALL
6    TUR
7    JAM
8    WAR
9    FOR
10   SMI
11   SCO
12   ADA
13   MIL

```

문제 81. 이름을 출력하는데 전부 소문자로 출력하시오 !

```

import csv
file = open ( "c:\\emp_comm.csv", 'r' )
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print( emp_list[1].lower() )

```

```

king
blake
clark
jones
martin
allen
turner
james
ward
ford
smith
scott
adams
miller

```

문제 82. 이름을 출력하는데 이름을 첫철자는 대문자로 출력하고 나머지는 소문자로 출력하는 아래의 SQL 을 구현하시오 !

```
(lower(), upper() 함수를 이용하세요 )
SQL> select initcap(ename)
      from emp;
```

```
import csv
file = open ( "c:WWemp_comm.csv", 'r' )
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print( emp_list[1][0].upper() +W
          emp_list[1][1:].lower() )
```

```
King
Blake
Clark
Jones
Martin
Allen
Turner
James
Ward
Ford
Smith
Scott
Adams
Miller
```

문제 83. 아래의 SQL 을 파이썬으로 구현하시오 !

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

```
import csv
file = open ( "c:WWemp_comm.csv", 'r' )
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print( emp_list[1], len(emp_list[1]) )
```

```
KING 4
BLAKE 5
CLARK 5
JONES 5
MARTIN 6
ALLEN 5
TURNER 6
JAMES 5
WARD 4
FORD 4
SMITH 5
SCOTT 5
ADAMS 5
MILLER 6
```

문제 84. 겨울왕국 대본에는 elsa가 몇번 나오는지 확인하기 위해서 겨울왕국 스크립트 한라인 한라인을 각각 list 변수에 담으시오 !



답 :

```

file = open( "c:WWWwinter.txt",'r')
for winter_list in file:
    a = winter_list.split(' ')
    print(a)
  
```

문제 85. 아래의 리스트에서 단어를 하나씩 뽑아내서 출력하시오 !

```

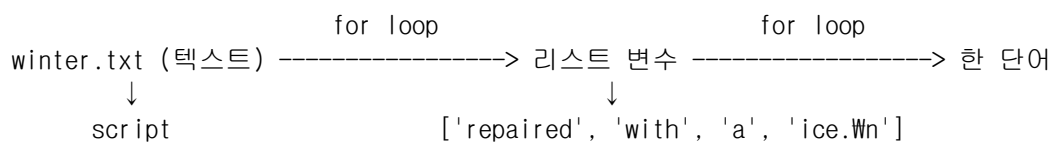
a = ['repaired', 'with', 'a', 'ice.Wn']

a = ['repaired', 'with', 'a', 'ice.Wn']
for i in a:
    print(i)
  
```

```

repaired
with
a
ice.
  
```

문제 86. 겨울왕국 스크립트를 한 단어씩 출력하시오 !



```

file = open( "c:WWWwinter.txt",'r')
for winter_list in file:
    a = winter_list.split(' ')
    for i in a:
        print(i)
  
```

문제 87. 위의 단어들이 소문자로 출력되게 하시오 !

```

file = open( "c:WWWwinter.txt",'r')
for winter_list in file:
    a = winter_list.split(' ')
    for i in a:
        print(i.lower())
  
```

문제 88. 아래의 list 에서 단어 aaa 가 몇개 있는지 count 하시오 .

```

word = ['bbb', 'ccc', 'aaa', 'ddd', 'aaa']
  
```

```
print(word.count('aaa'))
```

문제 89. 겨울왕국 대본에는 elsa 라는 단어가 몇번나오는가? anna 라는 단어는 몇번 나오는가?

```
anna = 0
elsa = 0
file = open( "c:WWWwinter.txt",'r')
for winter_list in file:
    a = winter_list.split(' ')
    for i in a:
        j = i.lower()
        anna += j.count('anna')
        elsa += j.count('elsa')
print(anna, elsa)
```

문제 90. 문자열 안에 싱글과 더블 쿼테이션 마크를 다 넣고 싶을때는 어떻게 해야 되는가?

```
data6 = """ my son's names is "john" """

print(data6)
```

문제 91. 아래의 data1 ~ data 6 변수에 있는 문자열 값을 한번에 출력하시오 !

```
data1 = 'i am python programer'
data2 = 'you are a programer'
data3 = """ i love
            i love
            i love """
data4 = "my son's names is john"
data5 = 'name of my son is "john"'
data6 = """ my son's names is "john" """

for i in range(1,6):
    dt = "data"+str( i )
    print( locals()[dt] )          #지역변수 data들에 담긴 값들을 출력하겠다.
    ※ 설명 : locals() : 지역변수의 이름과 값을 확인하는 함수
              globals() : 전역변수의 이름과 값을 확인하는 함수

i am python programer
you are a programer
i love
            i love
            i love
my son's names is john
name of my son is "john"
```

문제 93. 아래의 변수에 있는 값을 이용해서 아래의 문자를 출력하시오 !

```
txt1 = '자바'; txt2 = '파이썬'
num1 = 5
```

```
print ('%s 이 %s 보다 %d 배 더 쉽습니다.' %(txt1, txt2, num1) )
파이썬이 자바보다 5 배 더 쉽습니다.
```

문제 94. 변하는 값을 포함하는 문자열을 나타내는 아래의 코드를 실행하시오 !

```
from time import sleep # time 패키지에서 sleep 모듈을 가져와라 ~

for i in range(100):
    msg = 'Wr 진행률 %d%%' %(100-i-1)
    print(msg, end = '')
    sleep(0.1)
```

※ 설명 : Wr 리턴문자를 추가하여 msg 변수를 print 할때 줄바꿈 없이 화면에 출력한다.

print(' '*len(msg), end = '') 로 msg 길이만큼 공백문자를 줄바꿈없이 화면에 출력한다.

문제 95. 겨울왕국에서 anna를 검색해서 건수를 출력하는 코드를 실행하시오 !

```
file = open("c:\\winter.txt", 'r')

sum=0

for winter_list in file:
    a = winter_list.split(' ')
    for b in a:
        sum = sum + b.lower().count('anna')
print(sum)
685
```

문제 96. 문제 94 번과 문제 95 번 코드를 잘 조합해서 단어를 물어보게 하고 단어를 입력하면 해당 단어의 갯수가 변하는 숫자로 출력되어서 나오게 하시오 !

```
from time import sleep

file = open("c:\\winter.txt", 'r')
x = input('찾을 단어를 입력 ')
sum=0

for winter_list in file:
    a = winter_list.split(' ')
    for b in a:
        sum += b.lower().count(x)
        msg = 'Wr %d' %sum
        print(msg, end = '')
        sleep(0.00001)
```

문제 97. 위의 코드를 수정해서 아래와 같이 스크립트도 물어보게 하시오 !

분석할 스크립트를 입력하세요 ~winter.txt

찾을 단어를 입력하세요~ anna
685

```
x = input( '분석할 스크립트를 입력하세요 ~')
y = input( '찾을 단어를 입력하세요~ ')
from time import sleep

file = open("c:\\\\s" %x, 'r')
sum=0

for winter_list in file:
    a = winter_list.split(' ')
    for b in a:
        sum += b.lower().count(y)
        msg = 'Wr %d' %sum
        print(msg, end = '')
        sleep(0.000001)
```

문제 98. (점심시간 문제) 로그함수를 파이썬으로 구현하시오 !

로그의 밑수를 입력하세요 ~ 2
로그의 진수를 입력하세요 ~ 8

3 입니다.

```
x = int( input('로그의 밑수를 입력하세요 ~ ') )
y = int( input('로그의 진수를 입력하세요 ~ ') )

for i in range(y):
    if x**i == y:
        print(i, '입니다.')
```

문제 99. emp_list 에서 b요소를 제거하시오 !

```
emp_list = []
emp_list.append('a')
emp_list.append('b')

emp_list.remove('b')
print(emp_list)
```

문제 100. emp2.csv 의 sal 의 값을 sal_list 라는 비어있는 리스트 변수에 담으시오 ! (for loop 문 이용)

[5000, 2850, 2450, 2975, 1250, 1600, 1500, 950, 1250, 3000, 800, 3000, 1100, 1300]

```
import csv
```

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

```
sal_list = []
```

```
for emp_list in emp_csv:
```

```

        sal_list.append( int(emp_list[5]) )
        print(sal_list)
print(sal_list)
# append 되는 과정을 볼 수 있다.
# append 다 된 리스트를 볼 수 있다.

```

문제 101. 위의 결과를 다시 출력하는데 월급이 낮은 것부터 높은 순으로 출력하시오 !

```
[800, 950, 1100, 1250, 1250, 1300, 1500, 1600, 2450, 2850, 2975, 3000, 3000, 5000]
```

```

sal_list.sort()
print(sal_list)

```

문제 102. 위의 결과를 다시 출력하는데 월급이 높은 것부터 출력되게 하시오 !

```
[5000, 3000, 3000, 2975, 2850, 2450, 1600, 1500, 1300, 1250, 1250, 1100, 950, 800]
```

```

sal_list.reverse()
print(sal_list)

```

문제 103. 위의 sal_list 의 요소를 아래와 같이 뽑아내서 출력하시오 !

```

5000
3000
3000
2975
2850
2450
1600
1500
1300
1250
1250
1100
950
800

```

```

for i in sal_list:
    print(i)

```

문제 104. 이름과 월급을 출력하는데 월급이 높은 것부터 출력하시오 !

```

SQL> select ename, sal
      from emp
      order by sal desc:

```

석우형님 답 :

```

import csv
file = open('c:WWemp2.csv','r')
emp_csv = csv.reader(file)

sal_list = [ ]

```



```

for emp_list in emp_csv:
    sal_list.append( [ int(emp_list[5]),emp_list[1] ] )
print(sal_list)
sal_list.sort()
sal_list.reverse()

for i in sal_list:
    print(i[1],i[0])

선생님 답 :

def colcheck(data):
    return int( data[5] )

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

emp_list = []
for i in emp_csv:
    emp_list.append(i)

emp_sort_list = sorted(emp_list, reverse = True, key=colcheck)

for i in emp_sort_list:
    print(i[1], i[5])

```

문제 105. 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력하시오 !

```

SQL> select ename, hiredate
      from emp
      order by hiredate desc;

```

```

ADAMS 1983-01-15
SCOTT 1982-12-22
MILLER 1982-01-11
JAMES 1981-12-11
FORD 1981-12-11
KING 1981-11-17
MARTIN 1981-09-10
TURNER 1981-08-21
CLARK 1981-05-09
BLAKE 1981-05-01
JONES 1981-04-01
WARD 1981-02-23
ALLEN 1981-02-11
SMITH 1980-12-09

```

```

def colcheck(data):
    return data[4]

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

emp_list = []

```

```

for i in emp_csv:
    emp_list.append(i)

emp_sort_list = sorted(emp_list, reverse = True, key=colcheck)

for i in emp_sort_list:
    print(i[1], i[4])

```

문제 106. 이번에는 위의 문제를 pandas 로 구현하시오 !

	1	4
12	ADAMS	1983-01-15
11	SCOTT	1982-12-22
13	MILLER	1982-01-11
7	JAMES	1981-12-11
9	FORD	1981-12-11
0	KING	1981-11-17
4	MARTIN	1981-09-10
6	TURNER	1981-08-21
2	CLARK	1981-05-09
1	BLAKE	1981-05-01
3	JONES	1981-04-01
8	WARD	1981-02-23
5	ALLEN	1981-02-11
10	SMITH	1980-12-09

```

import pandas as pd

file = open('c:\\emp2.csv', 'r')
file2 = pd.read_csv(file, header=None) # 컬럼명 없이 읽어오겠다!
emp = file2.sort_values(4, ascending = False) # 판다스 안의 sort_values 모듈이용
print(emp[[1,4]])

```

문제 107. 이름과 월급을 출력하는데 월급이 낮은 사원부터 높은 사원 순으로 출력하시오 !(pandas 로 수행)

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

```

import pandas as pd

file = open('c:\\emp2.csv', 'r')
file2 = pd.read_csv(file, header=None)

```

```
emp = file2.sort_values(5, ascending = True)
print(emp[[1,5]])
```

문제 108. 사원 테이블에서 최대월급을 출력하시오 !

```
SQL> select max(sal)
      from emp;
```

5000

1. 판다스를 이용하지 않았을 때

```
sal_list = []

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

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

2. 판다스를 이용했을 때

```
import pandas as pd

emp_csv = pd.read_csv('c:\\emp2.csv', header = None)

print (max(emp_csv[5]))
```

문제 109. 사원 테이블에서 토탈월급을 출력하시오 !

```
SQL> select sum(sal)
      from emp;
```

29025

1. pandas 를 이용하지 않았을 때

```
sal_list = []
import csv
file = open("c:\\emp2.csv", 'r')
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    sal_list.append(int(emp_list[5]))
print(sum(sal_list))
```

2. pandas 를 이용했을 때

```
import pandas as pd

emp_csv = pd.read_csv('c:\\emp2.csv', header = None)

print (sum(emp_csv[5]))
```

문제 110. 평균월급을 출력하시오 !

2073.214285714286

1. pandas 를 이용하지 않았을 때

```
def avg(data):  
    return sum(data)/len(data)  
  
sal_list = []  
import csv  
file = open("c:WWemp2.csv", 'r')  
emp_csv = csv.reader(file)  
for emp_list in emp_csv:  
    sal_list.append(int(emp_list[5]))  
print(avg(sal_list))
```

2. pandas 를 이용했을 때

```
import pandas as pd  
  
emp_csv = pd.read_csv('c:WWemp2.csv', header = None)  
  
result = emp_csv[5].mean()  
  
print(result)
```

문제 111. 직업이 SALESMAN 인 사원들의 최대월급을 출력하시오 !

1. pandas 를 이용하지 않았을 때

```
sal_list = []  
  
import csv  
file = open("c:WWemp2.csv", 'r')  
emp_csv = csv.reader(file)  
  
for emp_list in emp_csv:  
    if emp_list[2] == 'SALESMAN':  
        sal_list.append(int(emp_list[5]))  
print(sum(sal_list))
```

2. pandas 를 이용했을 때

```
import pandas as pd  
  
emp_csv = pd.read_csv('c:WWemp.csv')  
  
result = emp_csv['sal'] [ emp_csv['job'] == 'SALESMAN'].max()  
  
print(result)
```

문제 112. 30 번 부서번호인 사원들의 토탈월급을 출력하시오 !

1. pandas 를 이용하지 않았을 때

```
sal_list = []

import csv
file = open("c:WWemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if int(emp_list[7]) == 30:
        sal_list.append(int(emp_list[5]))
print(sum(sal_list))
```

2. pandas 를 이용했을 때

```
import pandas as pd

emp_csv = pd.read_csv('c:WWemp.csv')

result = emp_csv['sal'] [ emp_csv['deptno'] == 30].sum()

print(result)
```

문제 113. 직업, 직업별 최대월급을 출력하시오 !

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

1. pandas 를 이용 했을 때

```
import pandas as pd

emp_csv = pd.read_csv('c:WWemp.csv')

result = emp_csv.groupby('job')['sal'].max()

print(result)
```

job	
ANALYST	3000
CLERK	1300
MANAGER	2975
PRESIDENT	5000
SALESMAN	1600

2. pandas 를 이용하지 않았을 때

```
# job 중복제거 리스트 만들기
job_list = []

import csv
file = open("c:WWemp2.csv", 'r')
emp_csv = csv.reader(file)
```

```

for emp_list in emp_csv:
    job_list.append(emp_list[2])

job = list(set(job_list))
job.sort()

# 직업과, 직업별 최대월급 출력
max_sal_by_job = []

for i in job:
    import csv
    file = open("c:WWemp2.csv", 'r')
    emp_csv = csv.reader(file)
    for emp_list in emp_csv:
        if i == emp_list[2]:
            max_sal_by_job.append(int(emp_list[5]))
    print(i, max(max_sal_by_job))
    max_sal_by_job = []

```

문제 114. 직업을 중복제거해서 출력하시오 !

```

job_list = []

import csv
file = open("c:WWemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    job_list.append(emp_list[2])

job = list(set(job_list))          # set 으로 중복을 제거
print(job)

```

문제 115. 직업, 직업별 토달월급을 출력하는데 직업별 토달월급이 높은것 부터 출력하시오 !

```

job_list = []

import csv
file = open("c:WWemp2.csv", 'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    job_list.append(emp_list[2])

job = list(set(job_list))
job.sort()

# 직업과, 직업별 최대월급을 리스트(sal_desc)에 입력
sum_sal_by_job = 0
sal_desc = []
for i in job:
    import csv
    file = open("c:WWemp2.csv", 'r')

```

```

emp_csv = csv.reader(file)
for emp_list in emp_csv:
    if i == emp_list[2]:
        sum_sal_by_job += int(emp_list[5])
sal_desc.append([sum_sal_by_job, i])
sum_sal_by_job = 0

# sal_desc를 sal 기준 내림차순으로 정렬하고 출력
sal_desc.sort()
sal_desc.reverse()
for j in range(0, len(sal_desc)):
    print(sal_desc[j][1], sal_desc[j][0])

```

문제 116. 위의 결과를 pandas를 이용한 것으로 수행하시오 !
(오늘의 마지막 문제)

```

import pandas as pd

emp_csv = pd.read_csv('c:\wwemp.csv')

result = emp_csv.groupby('job')['sal'].sum().sort_values(ascending=False)

print(result)

```

문제 117. 딕셔너리 자료형을 이용해서 주어가 0, 명사가 2, 동사가 1로 해서 한글과 영문을 저장하시오 !

```

dic = {}

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)

print(dic)

한글 어순 : 나는 소년 입니다.   (주어 + 보어 + 동사 )
영문 어순 : I am boy           (주어 + 동서 + 보어 )

```

문제 118. 아래와 같이 번역할 한글을 물어보게 하고 한글을 입력하면 영어로 번역하는 프로그램을 파이썬으로 작성하시오 !

번역할 한글을 입력하세요 ! 나는 소년입니다.

I am boy

```

dic = {}

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)

```

```
dic[ '입니다' ] = ('am', 1)

a = []
result = ''
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    result += i[1] + ' '
print(result)
```

문제 119. 아래와 같이 번역할 한글을 입력하세요 ~ 라는 말을 물어보게 하고 입력하면 아래와 같이 결과가 출력 되게 하시오 !

번역할 한글을 입력하세요 ~ 나는 소년 입니다.

결과 :

```
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
print (input_list)
```

문제 120. 위의 문제에서 input_list 에 있는 요소를 하나씩 뽑아내서 출력하시오 !

```
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
print (input_list)

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

문제 121. 아래의 딕셔너리 변수에서 l 를 출력하시오 !

```
dic = {}

dic[ '나는' ] = ('l', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '입이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)

dic['나는'][0]
```

문제 122. 아래의 dic 딕셔너리의 요소중에 값의 두번째인 숫자만 a 라는 비어있는 리스트를 만들고 a 리스트에 아래와 같이 입력하시오 !

번역할 한글을 입력하세요 !

결과 [0, 2, 1, 2, 1, 1]

```
dic = {}
```



```

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)
dic[ '입니다' ] = ('am', 1)

a = []
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
print (input_list)
for i in input_list:
    a.append(dic[i][1])

```

문제 123. 위의 코드를 수정해서 아래와 같이 결과가 출력되게 하시오 !

```

[[0, 'I'], [2, 'boy'], [1, 'am']]

dic = {}

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)
dic[ '입니다' ] = ('am', 1)

a = []
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
print (input_list)
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
print(a)

```

문제 124. 위의 코드를 수정해서 아래와 같이 출력되게 하시오 !

```

번역할 한글을 입력하세요 ~나는 소년 입니다
|
boy
am

result = ''
a = []
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
for i in a:
    print(i[1])

```

문제 125. 위의 코드를 수정해서 아래와 같이 출력되게 하시오 !

```

dic = {}

```

```

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)
dic[ '입니다' ] = ('am', 1)

result = ''
a = []
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    print(i[1])

```

문제 126. 위의 번역된 결과가 가로로 출력되게 하시오 !

번역할 한글을 입력하세요 ! 나는 소년입니다.
I am boy

```

dic = {}

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)
dic[ '입니다' ] = ('am', 1)

a = []
result = ''
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    result += i[1] + ' '
print(result)

```

문제 127. 나는 옷을 입습니다 라는 말이 영어로 번역되게 dic 딕셔너리를 구성하고 번역되게 하시오 !

번역할 한글을 입력하세요 ~나는 옷을 입습니다
I wearing cloth

```

dic = {}

dic[ '나는' ] = ('I', 0)
dic[ '소년' ] = ('boy', 2)
dic[ '이다' ] = ('am', 1)
dic[ '피자' ] = ('pizza', 2)
dic[ '먹는다' ] = ('eat', 1)
dic[ '입니다' ] = ('am', 1)
dic[ '입습니다' ] = ('wearing', 1)

```

```
dic[ '옷을' ] = ('cloth', 2)

a = []
result = ''
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    result += i[1] + ' '
print(result)
```

문제 128. smt_dic.csv 파일을 파이썬으로 로드해서 아래와 같이 리스트화 하시오 !

```
import csv
file = open("c:\\\\smt_dic.csv","r")
dic_csv = csv.reader(file)

for dic_list in dic_csv:
    print(dic_list)
```

문제 129. smt_dic.csv를 리스트화한 위의 코드에 for loop 문을 사용해서 4239 개의 영어와 한글 단어들을 dic 딕셔너리에 다 입력하시오 !

```
['-1', '힘없', '힘 없다', 'Be weak', '1', '박상범']
0      1      2      3      4      5
dic['희롱'] = ('Mook', 1)

dic = {}

import csv
file = open("c:\\\\smt_dic.csv","r")
dic_csv = csv.reader(file)

for dic_list in dic_csv:
    dic[dic_list[1]] = (dic_list[3], dic_list[4] )
```

문제 130. 문제 129 번 코드와 문제 127 번 코드를 합쳐서 번역 프로그램을 완성하시오 !

```
번역할 한글을 입력하세요 ~나는 소년 이다
I am boy

dic = {}

import csv
file = open("c:\\\\smt_dic.csv","r")
dic_csv = csv.reader(file)

for dic_list in dic_csv:
    dic[dic_list[1]] = (dic_list[3], int(dic_list[4]) )
```

```

a = []
result = ''
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    result += i[1] + ' '
print(result)

```

문제 131. 위의 코드에 if 문을 추가해서 번역이 안되면 번역할 수 없습니다. 라는 메시지가 출력되게 하시오 !
(점심시간) -----> 파이썬 200 제 예제 55 번 예외처리

문제 132. 오라클의 initcap 함수를 만들어서 아래와 같이 실행되게 하시오 !

```

SQL> select initcap('smith')
      from dual;

```

결과 : Smith

```

def initcap(data):
    data1 = data[0].upper() + data[1:].lower()
    return data1

print(initcap('smith'))

```

문제 133. substr 함수를 파이썬으로 생성하시오 !

```

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

```

smi

```

def substr(data, x, y):
    data1 = data[x-1:x-1+y]
    return data1

```

```

print(substr('smith', 3,2 ))

```

문제 134. 오라클의 lpad 함수를 파이썬으로 만드시오 !

```

SQL> select lpad('smith', 10 , '*' )
      from dual;

```

★★★★★smith

※ 설명 : 전체 10 자리 잡고 smith를 출력하는데 나머지 남은 왼쪽에 *을 채워넣는다.

내답 :

```
def lpad(data, x, data1):
    for i in range(x):
        result = data1*i + data
        if len(result) == x:
            return result
```

```
print(lpad('smith', 10, '★' ))
```

더 간단한 답 :

```
def lpad(data, x, data1):
    return ( (x-len(data))*data1 + data)
```

```
print(lpad('smith', 10, '★' ))
```

rjust 를 쓴 답 :

```
def lpad(data,a,b):
    return data.rjust(a,b)
```

```
print(lpad('smith',10,'*'))
```

문제 135. 오라클의 rpad 함수를 파이썬으로 구현하시오 !

```
SQL> select rpad('smith', 10 , '*' )
        from dual;
```

```
smith★★★★★
```

```
def lpad(data, x, data1):
    return ( data + (x-len(data))*data1)
```

```
print(lpad('smith', 10, '★' ))
```

문제 136. 오라클의 instr 함수를 파이썬으로 구현하시오 !

```
SQL> select instr('smith','m') from dual;
        2
```

```
def instr (data1, data2):
    for i in range(len(data1)):
        if data1[i:i+len(data2)] == data2:
            return i+1
```

```
print( instr('smith','t'))
print( instr('smith','mi') )
```

문제 137. 아래와 같이 없는 철자를 입력하면 0 이 출력되게 하시오 !

```
SQL> select instr('smith','m') from dual;
        2
```

```

def instr (data1, data2):
    for i in range(len(data1)+1): #범위를 데이터길이보다 한칸 더 늘림
        if data1[i:i+len(data2)] == data2:
            return i+1
        break
    if i == len(data1):          # i 가 data1 범위보다 큰수가 올 때까지 break 가 걸리지 않는다면
        return 0                # 0을 리턴한다.

print( instr('smith','h'))
print( instr('smith','mi') )

```

문제 139. 오늘 날짜를 출력하시오 !

```

SQL> select sysdate from dual;

from datetime import date          # datetime 패키지의 today 를 이용
print(date.today())

```

문제 140. 파이썬으로 오라클의 add_months 함수를 생성하시오 !
(오늘 부터 3 달 뒤에 돌아오는 날짜를 출력하시오)

```

SQL> select add_months(sysdate, 3)
       from dual;

```

힌트 :

```

from datetime import date
from dateutil.relativedelta import relativedelta

result = date.today() + relativedelta(months =+ 3)

print(result)

```

답 :

```

def sysdate():
    from datetime import date
    return str(date.today())

def add_months(a, b):
    import datetime
    from dateutil.relativedelta import relativedelta
    a1 = datetime.datetime.strptime(a, '%Y-%m-%d')
    result = a1 + relativedelta(months =+ int(b) )
    return result

print (add_months( sysdate(), 3))

```

```
select last_day(sysdate)
from dual;
```

2018-11-30

문제 141. 올해 2월달의 마지막 날짜를 출력하시오 !

```
from calendar import monthrange
print( monthrange(2018,2))
```

(3, 28)

↓

그 달의 총 일수

※ 3의 의미는 ? 그 달의 1일의 요일

월 : 0, 화 : 1, 수 : 2, 목 : 3, 금 : 4, 토 : 5, 일 : 6

문제 142. 위의 결과에서 28 만 출력하시오 !

```
from calendar import monthrange
print( monthrange(2018,2)[1])
```

문제 143. 아래의 날짜에서 년도만 추출하시오 !

'2018-07-18'

```
a = '2018-07-18'
from datetime import datetime
b = datetime.strptime(a, '%Y-%m-%d').year
print(b)
```

문제 144. 아래의 날짜에서 달만 추출하시오 !

'2018-07-18'

```
a = '2018-07-18'
from datetime import datetime
b = datetime.strptime(a, '%Y-%m-%d').month
print(b)
```

문제 145. (오늘의 마지막 문제) 위의 함수들을 이용해서 오라클 last_day 함수를 아래와 같이 생성하시오 !

```
print( last_day('2018-07-18') )
2018-07-31
```

```
def last_day(a):
    from datetime import datetime
    year = datetime.strptime(a, '%Y-%m-%d').year
    from datetime import datetime
    month = datetime.strptime(a, '%Y-%m-%d').month
    from calendar import monthrange
    result = a[:8] + str(monthrange(year, month)[1])
    return result

print( last_day('2018-07-18') )
```

문제 146. oracle_func.py 라는 이름으로 메모장을 열고 그 안에 어제 만들었던 오라클 함수들을 저장하시오 !

```
1. initcap
2. substr
3. rtrim
4. ltrim
5. rpad
6. pad
7. instr

import oracle_func as o
print(o.initcap('smith'))

Smith
```

문제 147. 오늘 날짜에서 하루 더한 날짜가 어떻게 되는가?

```
import datetime as d
print( d.datetime.today() + d.timedelta(days=1))
      ↓      ↓
      클래스 메소드(함수)
```

문제 148. 아래와 같은 함수를 생성하시오 !

```
print(next_day('2018-07-18',2))
2018-07-20

def next_day(date, num):
    import datetime as d
    from dateutil.relativedelta import relativedelta
    date1 = d.datetime.strptime(date, '%Y-%m-%d')
    result = date1 + d.timedelta(days = num)
    return result

print(next_day('2018-07-18',2))
```

점심시간 문제 :

```
SQL> select next_day( to_date('2018-11-23', 'RRRR-MM-DD'), '월요일')
      from dual;
```


2018-11-26

```
def next_day(date, week):
    week_dic = {}
    week_dic['월요일'] = 0
    week_dic['화요일'] = 1
    week_dic['수요일'] = 2
    week_dic['목요일'] = 3
    week_dic['금요일'] = 4
    week_dic['토요일'] = 5
    week_dic['일요일'] = 6
    import datetime as d
    from datetime import datetime as da
    date1 = d.datetime.strptime(date, '%Y-%m-%d')
    date_week = da.weekday(date1)
    if date_week != week_dic[week]:
        result = date1 + d.timedelta(days = date_week + week_dic[week])
    else:
        result = date1 + d.timedelta(days = 7)
    return result

print(next_day('2018-11-23', '월요일'))
```

문제 149. 오늘이 무슨 요일인지 출력하시오 !

```
from datetime import datetime as d
print(d.today().weekday())
```

※ 월요일 : 0
화요일 : 1
수요일 : 2
목요일 : 3
금요일 : 4
토요일 : 5
일요일 : 6

문제 150. 문제 149 번을 다시수행하는데 숫자 4 가 나오는게 아니라 금요일이라고 출력되게 하시오 !
(아래의 리스트 변수를 이용해서)

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

from datetime import datetime as d
d = d.today().weekday()
dy[d]
```

문제 151. (점심시간 문제) 아래의 데이터를 딕셔너리로 만드시오 ! (딕셔너리 이름 : week_dic)

key	value
↓	↓
※ 월요일	: 0
화요일	: 1
수요일	: 2
목요일	: 3

금요일 : 4
토요일 : 5
일요일 : 6

```
week_dic = {}  
week_dic['월요일'] = 0  
week_dic['화요일'] = 1  
week_dic['수요일'] = 2  
week_dic['목요일'] = 3  
week_dic['금요일'] = 4  
week_dic['토요일'] = 5  
week_dic['일요일'] = 6
```

■ months_between 함수를 파이썬으로 만들기

"날짜와 날짜 사이의 개월수를 출력"

```
SQL> select months_between(sysdate, hiredate)  
       from emp;
```

예제 :

```
import datetime  
d1 = '2018-07-18'  
print( type(d1))          # 문자형  
  
import datetime as d  
b = d.datetime.strptime(d1, '%Y-%m-%d')  
  
print(a)  
print( type(a))           # 날짜형
```

* 아래 두 날짜 사이의 개월 수

2017-01-01 ~ 2018-07-01

2018-2017 = 12 개월
07-01 = 6 개월

문제 152. months_between 함수를 파이썬으로 구현하시오 !

```
print( months_between('2017-01-01', '2018-07-01'))
```

18

```
def months_between(date1, date2):  
    import datetime as d  
    a = d.datetime.strptime(date1, '%Y-%m-%d')  
    b = d.datetime.strptime(date2, '%Y-%m-%d')  
    result = (a.year - b.year)*12 + (a.month - b.month)  
    return result
```

```
print( months_between('2017-12-01', '2018-01-01'))
```

문제 153. 오라클의 nvl 함수를 파이썬으로 구현하시오 !

```
SQL> select ename, nvl(comm,0)
       from emp;

def nvl(data, val):
    if data == '':
        return val
    else:
        return data

import csv
file = open("c:\\\\emp_comm.csv", "r")
emp_csv = csv.reader(file)
for emp_list in emp_csv:
    print( emp_list[1], nvl( emp_list[6], 0) )
```

문제 154. 오라클의 decode 함수를 파이썬으로 구현하시오 !

```
SQL> select ename, sal, deptno,
       decode(deptno, 10, sal*0.1, 0) as bonus
       from emp;
```

문제 155. 0 에서 1 사이의 숫자를 random 으로 생성해 보시오 !

```
import random
x = random.uniform(0,1)
print(x)
```

문제 156. 위의 코드를 for loop 문을 이용해서 10000 번 반복시키시오 !

```
import random
for i in range(10000):
    x = random.uniform(0, 1)
    y = random.uniform(0, 1)
    print(x)
    print(y)
```

문제 157. $x^2 + y^2 \leq 1$ 이면 부채꼴 안에 들어가는 점이므로 이 공식을 이용해서 pi 를 구하시오 !

```
import random
total = int( input('찍을 점의 개수를 입력'))
cnt = 0

for i in range(total):
    x = random.uniform(0, 1)
    y = random.uniform(0, 1)
    if x*x + y*y <= 1:
        cnt += 1
```

```
print((cnt/total)*4)
```

문제 158. 무한루프로 파이값을 찾아가게 만들어보자

```
from time import sleep
import random
total = 0
cnt = 0
while True:
    total += 1
    x = random.uniform(0, 1)
    y = random.uniform(0, 1)
    if x*x + y*y <= 1:
        cnt += 1
    pi = (cnt/total)*4
    msg = 'Wr 파이(pi)는 = %f' %pi
    print(msg, end = '')
    sleep(0.000000000000001)
```

문제 160. 위의 코드를 가지고 최대공약수 구하는 함수를 아래와 같이 수행되게 생성하시오 !

```
maxyaksu(16, 24)
8
```

```
def maxyaksu(a, b):
    dummy = 0
    i = 1
    c = 1
    if a <= b:
        dummy = a
        a = b
        b = dummy
    while i < b:
        i += 1
        if a%i == 0:
            a = a / i
            if b%i == 0:
                b = b / i
                c = c * i
            i = 1
    return c
```

문제 161. 가변 매개변수를 사용해서 아래와 같이 실행해도 최대공약수가 나오게 하시오 !

```
maxyaksu(16, 24, 8, 4)
4
```

```
def maxyaksu(*n):
    num = list(n)
    for j in range(len(n)-1):
        a = num[j]
        b = num[j+1]
        dummy = 0
        i = 1
```

```

c = 1
if a <= b:
    dummy = a
    a = b
    b = dummy
while i < b:
    i += 1
    if a%i == 0:
        a = a / i
    if b%i == 0:
        b = b / i
    c = c * i
    i = 1
num[j+1] = c
return c

```

문제 162. 위의 예제를 코드로 구현하시오 !

```

print( cycle_func1(5) )      # 한개의 원의 넓이 출력
78.53981633974483

```

```

print( cycle_func2(5))      # 두개의 원의 넓이 출력
157.07963267948966

```

```

pi = 3.141592653589793

```

```

def cycle_func1(radius):
    global pi
    return pi*radius*radius

```

```

print( cycle_func1(5) )

```

```

def cycle_func2(radius):
    global pi
    return pi*radius*radius*2

```

```

print( cycle_func2(5))

```

```

print(abs(-9))              # 절대값 출력하는 함수
                             (무조건 양수로 출력하는 함수)

```

문제 163. abs 쓰지 말고 if 문으로 my_abs 라는 함수를 생성하시오 !(무조건 양수로 출력하는 함수)

```

print(my_abs(9))
9
print(my_abs(-9))
9

```

```

def my_abs(num):
    if num < 0:
        num = -num
    return num

```

문제 164. 아래의 4개의 함수를 가지고 cal_test3.py 라는 모듈을 생성하시오 !

```
def plus(a, b):  
    return a+b
```

```
def minus(a, b):  
    return a - b
```

```
def multiply(a, b):  
    return a*b
```

```
def divide(a, b):  
    return a/b
```

cal_test3.py 가 있는 위치에 다른 스크립트를 열어서 cal_test3.py 를 호출해서 아래와 같이 코드를 실행해 보시오 ~

```
import cal_test3 as c
```

```
c.plus(10,20)
```

문제 166. (점심시간 문제) mean 이라는 함수를 생성 하시오 !
(가변 매개변수를 이용해서 숫자를 몇개 넣고 실행하던 상관없이 평균값이 구해지게 하시오)

```
print( mean( 7,9,2,5,4,9,8))  
6.285714285714286  
print( mean(7, 9, 2, 5, 4, 9, 8, 10, 12))  
7.333333333333333
```

```
def mean(*num):  
    num_list = list(num)  
    result = 0  
    for i in num_list:  
        result += i  
    return result/len(num_list)
```

```
print( mean( 7,9,2,5,4,9,8))
```

문제 168. Calculator 클래스(계산기 설계도)에 빼기(minus), 곱하기(multiply), 나누기(divide) 함수 3개를 추가하고 실행해보시오 !

```
class Calculator:  
    def __init__(self):  
        self.result = 0  
  
    def adder(self, num):  
        self.result += num  
        return self.result  
  
    def minus(self, num):  
        self.result -= num  
        return self.result  
  
    def multiply(self, num):  
        self.result *= num  
        return self.result
```

```

    def divide(self, num):
        self.result /= num
        return self.result

cal3 = Calculator()

print(cal3.adder(3))
print(cal3.minus(4))
print(cal3.multiply(2))
print(cal3.divide(5))

```

문제 169. 총 클래스를 생성하고 객체화 시키시오 !

```

class Gun():
    def __init__(self):                # 클래스가 실체화 될때 바로 작동
        self.bullet = 0              # 하는 메소드

    def charge(self, num):            # 충전하는 기능
        self.bullet = num

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

    def print(self):                  # 출력하는 기능
        print(' {} 발 남았습니다.'.format(self.bullet))

# 총을 실체화 시킨다.
gun1 = Gun()

# 총알을 충전한다.
gun1.charge(10)

# 총알을 쏜다.
gun1.shoot(3)

# 몇발 남았는지 확인한다.
gun1.print()

탕!
탕!
탕!
7 발 남았습니다.

```

문제 170. 총알 충전할 때 몇발 충전 되었습니다. 라는 메세지가 출력되게 하시오 !

```

class Gun:
    def __init__(self):

```

```

self.bullet = 0

def charge(self, num):
    self.bullet = num
    print('{}발 충전 되었습니다.'.format(self.bullet))

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

def print(self):
    print('{}발 남았습니다.'.format(self.bullet))

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

10 발 충전 되었습니다.
탕!
탕!
탕!
7 발 남았습니다.

```

문제 170. 고릴라 클래스를 구현해서 고릴라 객체를 만드시오 ~

기능 : 1. 소리지르는 기능 : shout

우와 ~~~~~~
소리를 지를 수 없습니다. <----- 바나나가 없으면

2. 바나나 먹는 기능 : eat

3. 바나나가 뱃속에 몇개 남았는지 프린트 하는 기능

바나나를 10개 먹었으면 소리를 10번 지를 수 있다.

```

class Monky:
    def __init__(self):
        self.banana = 0

    def eat(self, num):
        self.banana = num
        print('고릴라가 바나나 {}개를 먹었습니다.'.format(self.banana))

    def shout(self, num):
        for i in range(num):
            if self.banana > 0:
                print('우와~~~~~')
                self.banana -= 1
            if self.banana == 0:
                print('소리를 지를 수 없습니다.')
                break

```



```

        def print(self):
            print('고릴라 뱃속에 바나나가 {}개 남았습니다.'.format(self.banana))

        monkey = Monkey()
        monkey.eat(10)
        monkey.shout(3)
    monkey.print()

```

문제 171. 위의 고릴라의 기능중에 걷는 기능을 추가하시오 !
메소드의 이름은 walk 라고 한다.

```

print(go.walk(3))
뚜벅 !
뚜벅 !
뚜벅 !

```

바나나가 없으면 배가 고파 걸을 수 없습니다. 메세지 출력
바나나가 없으면 배가고파 소리를 지를 수 없습니다.

바나나를 10 개를 먹었으면 10 번 소리를 지를 수 있다.
바나나를 10 개를 먹었으면 10 번 걸을 수 있다.
바나나를 10 개를 먹었으면 5 번 소리지를 수 있고 5 번 걸을 수 있다.

```

class Monkey:
    def __init__(self):
        self.banana = 0

    def eat(self, num):
        self.banana = num
        print('고릴라가 바나나 {}개를 먹었습니다.'.format(self.banana))

    def shout(self, num):
        for i in range(num):
            if self.banana > 0:
                print('우와~~~~~')
                self.banana -= 1
            if self.banana == 0:
                print('바나나가 없으면 배가 고파 걸을 수 없습니다.')
                print('바나나가 없으면 배가고파 소리를 지를 수 없습니다.')
                break

    def walk(self, num):
        for i in range(num):
            if self.banana > 0:
                print('뚜벅 !')
                self.banana -= 1
            if self.banana == 0:
                print('바나나가 없으면 배가 고파 걸을 수 없습니다.')
                print('바나나가 없으면 배가고파 소리를 지를 수 없습니다.')
                break

    def print(self):
        print('고릴라 뱃속에 바나나가 {}개 남았습니다.'.format(self.banana))

monkey = Monkey()
monkey.eat(10)

```

```
monky.shout(3)
monky.walk(7)
monky.print()
```

문제 172. Card 클래스를 생성해서 카드를 발급하시오 !

Movie_Card 클래스

- 충전 기능 (Charge)
- 소비 기능 (Consume)
 - Consume(3000, '마트')
 - Consume(10000, '영화') # 20% 할인해서
- print 기능 (print)

```
class Movie_Card:
    def __init__(self):
        self.card = 0

    def charge(self, num):
        self.card += num
        print('{}원 충전되었습니다.'.format(num))

    def consume(self, num, loc):
        if loc == '영화관':
            num = int(num*0.8)
        self.card -= num
        if self.card <= 0:
            print('잔액이 부족합니다.')
        else:
            print('{}에서 {}원 사용했습니다.'.format(loc, num))

    def print(self):
        print('잔액이 {}원 입니다.'.format(self.card))
```

```
card1 = Movie_Card()
```

```
card1.charge(20000)
card1.print()
```

```
card1.consume(3000, '마트')
card1.consume(10000, '영화관')
card1.print()
card1.consume(13000, '마트')
```

```
20000 원 충전되었습니다.
잔액이 20000 원 입니다.
마트에서 3000 원 사용했습니다.
영화관에서 8000 원 사용했습니다.
잔액이 9000 원 입니다.
잔액이 부족합니다.
```

문제 176. (점심시간 문제) 아래와 같이 고릴라 두 객체를 생성하는데 고릴라가 태어날때 부터 바나나를 뱃속에 넣고 태어날 수 있게 클래스를 생성하시오 !

```
anwooyong_gorilla = Gorilla(10)          # 바나나 10개 충전하면서 생성
```

```

gehyuk_gorilla = Gorilla(10)                # 바나나 10 개 충전하면서 생성
anwooyong_gorilla.shout(3)
우와~~~~~
우와~~~~~
우와~~~~~

class Gorilla:
    def __init__(self, num):
        self.banana = num

    def eat(self, num):
        self.banana = num
        print('고릴라가 바나나 {}개를 먹었습니다.'.format(num))

    def shout(self, num):
        for i in range(num):
            if self.banana > 0:
                print('우와~~~~~')
                self.banana -= 1
            if self.banana == 0:
                print('바나나가 없으면 배가 고파 걸을 수 없습니다.')
                print('바나나가 없으면 배가고파 소리를 지를 수 없습니다.')
                break

    def walk(self, num):
        for i in range(num):
            if self.banana > 0:
                print('뚜벅 !')
                self.banana -= 1
            if self.banana == 0:
                print('바나나가 없으면 배가 고파 걸을 수 없습니다.')
                print('바나나가 없으면 배가고파 소리를 지를 수 없습니다.')
                break

    def print(self):
        print('고릴라 뱃속에 바나나가 {}개 남았습니다.'.format(self.banana))

```

문제 177. 7기 학생들의 나이 평균과 8기 학생들의 나이 평균을 구하시오 !

```

import numpy as np

age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,32,28,28,24]

a = np.array(age_7) # 일반 리스트를 numpy 형 리스트로 변환
b = np.array(age_8) # 일반 리스트를 numpy 형 리스트로 변환

print(a.mean())
26.7

print(b.mean())
27.74074074074074

```

문제 178. 7기와 8기학생들의 나이의 중앙값(가운데 값)을 출력하시오 !

```

import numpy as np

age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,32,W
28,28,24]

a = np.array(age_7) # 일반 리스트를 numpy 형 리스트로 변환
b = np.array(age_8) # 일반 리스트를 numpy 형 리스트로 변환

print( np.median(a))
27.0
print( np.median(b))
27.0

```

문제 179. 위의 9기 학생들의 최빈값을 구하시오 !

```

age_9 = [2,2,2,2,2,2,2,2,2,2,52,52,52,52,52,52,52,52,52,52]

import numpy as np
from collections import Counter
cnt = Counter(age_9)
print(cnt.most_common()[0:3])

```

문제 180. 어느 학습 성적 표준편차가 작다는 것은 다음 중 무엇을 의미하는가?

1. 성적이 나쁘다.
2. 성적이 비슷하다. #답
3. 성적이 우수하다.
4. 성적에 차이가 많다.

문제 181. 7기와 8기의 분산을 각각 구해보시오 !

```

age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,32,W
28,28,24]

print(np.var(age_7))
print(np.var(age_8))

2.8766666666666665
11.080932784636488

```

문제 182. 7기와 8기의 나이의 표준편차를 각각 구해보시오 !

```

age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,32,W
28,28,24]

print(np.std(age_7))

```

```
print(np.std(age_8))
```

```
1.69607389776114  
3.328803506462418
```

문제 183. 7기와 8기 학생들의 나이의 범위 데이터값을 각각 구하시오 !

```
def rang(data):  
    return max(data) - min(data)  
  
age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]  
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,27,32,28,28,24]  
age_9 = [2,2,2,2,2,2,2,2,2,2,2,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52,52]  
  
print(rang(age_7))  
print(rang(age_8))
```

문제 184. 사분위수 범위를 구하는 함수를 생성하고 7기와 8기의 사분위수 범위를 구하시오! (Q3 - Q1)
75% 25%

```
def data_range(args):  
    import numpy as np  
    result = np.quantile(args, 0.75) - np.quantile(args, 0.25)  
    return result  
print(data_range(age_7))
```

★ 1. 라인 그래프

<<<<< 파이썬 그래프 총정리 >>>>>

☆ 1. 라인 그래프 그리기

```
import numpy as np  
import matplotlib.pyplot as plt          # 그래프를 그릴 수 있는 함수들이 있음  
  
x = np.array([0,1,2,3])                  # x 축 데이터를 생성  
y = np.array([-1,0.2,0.9,2.1])           # y 축 데이터를 생성  
  
plt.plot(x,y)                            # 점을 찍음  
plt.grid(True)                           # 모눈종이  
plt.show()                               # 그림을 출력
```

문제 185. 아래의 직선의 방정식을 그래프로 시각화 하시오 !

$$y = 4x + 10$$

```
import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.arange(-10, 10)  
y = np.array(4*x + 10)
```

```
plt.plot(x,y)
plt.grid(True)
plt.show()
```

★ 2. 막대 그래프

예제:

```
import numpy as np
import matplotlib.pyplot as plt

y1_value = (21.6, 23.6, 45.8, 77.0, 102.2, 133.3, 327.9, 348.0, 137.6, 49.3, 53.0, 24.9)
x_name=('1', '2', '3', '4', '5','6', '7', '8', '9', '10', '11', '12')
n_groups = len(x_name)
index = np.arange(n_groups)

plt.bar(index, y1_value, tick_label=x_name, align='center')    # align : 막대 위치 원,오, 중앙

plt.xlabel('month')
plt.ylabel('average rainfall (mm)')
plt.title('Weather Bar Chart')
plt.xlim( -1, n_groups)
plt.ylim( 0, 400)
plt.show()
```

문제 186. 아래의 데이터를 막대그래프로 시각화 하시오 !

```
y1_value = (5600, 4150, 5000, 8275, 6000)
x_name=('SALESMAN', 'CLERK', 'PRESIDENT', 'MANAGER', 'ANALYST')

import numpy as np
import matplotlib.pyplot as plt

y1_value = (5600, 4150, 5000, 8275, 6000)
x_name=('SALESMAN', 'CLERK', 'PRESIDENT', 'MANAGER', 'ANALYST')
n_groups = len(x_name)
index = np.arange(n_groups)

plt.bar(index, y1_value, tick_label=x_name, align='center', color=['red','blue','green','W
    'yellow','purple'], width = 0.5)

plt.xlabel('job')
plt.ylabel('sum(sal)')
plt.title('EMP')
plt.xlim( -1, n_groups)
plt.ylim( 0, 10000)
plt.show()
```

문제 187. 아래의 데이터를 막대그래프로 시각화 하시오 !

```
y1_value = (800, 22825, 4300, 1100)
x_name=('1980', '1981', '1982', '1983')

import numpy as np
```

```

import matplotlib.pyplot as plt

y1_value = (800, 22825, 4300, 1100)
x_name=('1980', '1981', '1982', '1983')
n_groups = len(x_name)
index = np.arange(n_groups)

plt.bar(index, y1_value, tick_label=x_name, align='center', color =
['red','blue','green','purple'],W
width = 0.5)

plt.xlabel('year')
plt.ylabel('sum(sal)')
plt.title('EMP')
plt.xlim( -1, n_groups)
plt.ylim( 0, 25000)
plt.show()

```

★ 3. 원형 그래프

막대 그래프가 단순히 크기비교라면 원형그래프는 백분율 표시를 시각화 할때 유용하다.

예제 :

```

import matplotlib.pyplot as plt
import numpy as np
from matplotlib import font_manager, rc
from matplotlib import style

font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)
style.use('ggplot')

labels = ['삼성전자', 'SK 하이닉스', 'LG 전자', '네이버', '카카오']
ratio = [50, 20, 10, 10, 10]
plt.pie(ratio, labels=labels, shadow=True, startangle=90)
plt.show()

```

문제 188. 아래의 데이터를 원형그래프로 시각화 하시오 !
(백분율로 표시하시오 !)

```

import matplotlib.pyplot as plt
import numpy as np

labels = ['SALESMAN', 'CLERK', 'PRESIDENT', 'MANAGER', 'ANALYST']
ratio = [5600, 4150, 5000, 8275, 6000]
explode = (0.0, 0.5, 0.0, 0.0, 0.0)
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', 'red']

plt.pie(ratio, explode=explode, labels=labels, shadow=False, startangle=90, colors=colors, W
autopct='%1.1f%%')
plt.show()

```

★ 4. plot 그래프

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(5)                                # seed : 랜덤값이 일률적으로 뱉힌다.
x = np.arange(1, 101)
y = 20 + 3 * x + np.random.normal(0, 60, 100)      # (평균, 표준편차, 갯수)
plt.plot(x, y, "o")                                # "o" 점모양
plt.show()
```

문제 189. (오늘의 마지막 문제) 7기와 8기의 나이데이터를 plot 그래프로 각각 시각화 하시오 !

```
import numpy as np
import matplotlib.pyplot as plt

age_8 = [26,27,27,27,26,27,28,31,28,27,25,26,23,26,26,25,24,26,31,40,28,30,33,27,32,27,26]
age_7 = [28,27,24,27,28,28,26,27,28,26,26,27,25,27,26,25,27,29,25,28,26,25,28,25,24,27,32,28,W
28,24]

x = np.arange(1, len(age_7)+1)
y = np.array(age_7)

plt.plot(x, y, "o")
plt.title('age_7')
plt.ylim( 22, 45)
plt.show()

x = np.arange(1, len(age_8)+1)
y = np.array(age_8)

plt.plot(x, y, "o")
plt.title('age_8')
plt.ylim( 22, 45)
plt.show()
```

문제 190. mean 함수를 생성해서 아래와 같이 수행하면 실행되게 하시오 !

```
a = [24, 23, 21, 20, 30, 44, 41]

print(mean(a))
```

내 답 :

```
def mean(data):
    num = 0
    for i in data:
        num += i
    return num/len(data)
```

선생님 답 :

```
def mean(data):
    return sum(data)/len(data)
```


문제 191. Stats 라는 클래스를 만들고 Stats 클래스 안에 mean 메소드를 추가하여 아래와 같이 실행되게 하시오 !

```
class Stats:

    def mean(self, data):
        return sum(data)/len(data)

stats1 = Stats()

a = [24, 23, 21, 20, 30, 44, 41]

print( stats1.mean(a) )
```

문제 192. 분산 함수를 생성하시오 !

* 분산 : 데이터의 퍼짐 정도

1 공식

```
def variance(data):
    avg = sum(data)/len(data)
    num = 0
    for i in data:
        num += (i - avg)**2
    return num/len(data)
```

2 공식

```
def variance(self, data):
    avg = mean(data)
    num = 0
    for i in data:
        num += i**2
    avg_2 = num/len(data)
    return avg_2 - avg**2
```

문제 193. 표준편차를 구하는 함수를 생성하시오 !

* 표준편차 ? 평균과의 오차(분산에 루트를 씌운것)

```
import math
print( math.sqrt(4))
```

실행 결과 :

```
def mean(data):
    return sum(data)/len(data)

def variance(data):
    avg = mean(data)
    num = 0
    for i in data:
        num += i**2
    avg_2 = num/len(data)
    return avg_2 - avg**2

def stddev(data):
    var = variance(data)
    std = var**0.5
    return std
```

```
print( stats1.stddev(a) )
```

문제 194. 중앙값을 구하는 함수를 생성하시오
(어제 배웠던 numpy 이용해서 구현)

```
a = [24,23,21,20,30,44,41]
print( median(a) )
```

답:

```
def median(args) :
    import numpy as np
    a = np.array(args)
    b = np.median(a)
    return b
```

```
a = [24,23,21,20,30,44,41]
print( median(a) )
```

-- 넘파이 이용하지 않은 코드 :

```
def median(lst):
    lst.sort()
    if len(lst)%2==0 :
        return (lst[int(len(lst)/2)-1]+lst[int((len(lst)/2)+1)-1])/2
    else:
        return lst[int((len(lst)/2)-0.5)]
```

문제 195. 최빈값을 구하는 함수를 생성하시오 !

```
age_9 = [ 2,2,2,2,2,2,2,2,2, 52,52,52,52,52,52,52,52,52]
```

```
def mode(data):
    dis_data = list(set(data))
    cnt = 0
    mod = []
    mod_data = []
    for i in dis_data:
        for j in data:
            if i == j:
                cnt += 1
        mod.append(cnt)
        cnt = 0
    mod_index = 0
    data_mod_index = []
    for i in mod:
        if max(mod) == i:
            data_mod_index.append(mod_index)
        mod_index += 1
    result = []
    for i in data_mod_index:
        result.append(dis_data[i])
    return result
```

문제 196. (점심시간 문제) 위에서 만든 기초통계 함수들을 가지고 통계 클래스(Stats)를 생성하시오!

```
class Stats:
    def __init__(self):
        print('통계 객체가 생성되었습니다.')
    def mean(self, data):
        return sum(data)/len(data)

    def varience(self, data):
        avg = self.mean(data)
        num = 0
        for i in data:
            num += i**2
        avg_2 = num/len(data)
        return avg_2 - avg**2

    def stddev(self, data):
        var = self.varience(data)
        std = var**0.5
        return std

    def median(self, args) :
        import numpy as np
        a = np.array(args)
        b = np.median(a)
        return b

    def mode(self, data):
        dis_data = list(set(data))
        cnt = 0
        mod = []
        mod_data = []
        for i in dis_data:
            for j in data:
                if i == j:
                    cnt += 1
            mod.append(cnt)
            cnt = 0
        mod_index = 0
        data_mod_index = []
        for i in mod:
            if max(mod) == i:
                data_mod_index.append(mod_index)
            mod_index += 1
        result =[]
        for i in data_mod_index:
            result.append(dis_data[i])
        return result

a = [24, 23, 21, 20, 30, 44, 41]
stats1 = Stats()
print (stats1.mean(a))
print (stats1.median(a))
print (stats1.mode(a))
print (stats1.varience(a))
print (stats1.stddev(a))
```

문제 197. 위에서 만든 기초 통계 클래스를 모듈화 해서 아래와 같이 다른창에 실행되게 하시오 !

```
import My_Stats as m
a = [24, 23, 21, 20, 30, 44, 41]
stats1 = m.My_Stats()
print(stats1.stddev(a))
```

통계 객체가 생성되었습니다.
9.071147352221455

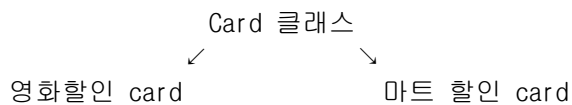
★★★ 꿀팁 ! Ctrl + 함수 클릭하면 원래의 모듈파일이 열린다. ★★★

문제 198. range 함수와 percent_range(사분위 범위) 함수를 각각 My_Stats 클래스에 추가하시오!

```
def range(self, data):
    return max(data) - min(data)

def percent_range(self, args):
    result = np.quantile(args, 0.75) - np.quantile(args, 0.25)
    return result
```

문제 199. 아래의 카드사의 상속코드를 구현하기 위해서 먼저 부모 클래스인 Card 클래스를 구현하시오 !



구현 내용 : card의 기본 기능

1. __init__함수 : 카드 발급시 카드가 발급되었습니다. 메세지 출력
2. 충전 기능 (charge)
3. 소비 기능 (consume)
"잔액부족 메세지 출력"
4. 잔액 프린트 기능(print)

```
class Card:
    def __init__(self):
        self.card = 0

    def charge(self, num):
        self.card += num
        print('{}원 충전되었습니다.'.format(num))

    def consume(self, num, loc):
        self.card -= num
        if self.card <= 0:
            self.card += num
            print('잔액이 부족합니다.')
        else:
            print('{}에서 {}원 사용했습니다.'.format(loc, num))

    def print(self):
        print('잔액이 {}원 입니다.'.format(self.card))
```

문제 200. 부모 클래스인 Card 클래스의 기능을 상속받아서 영화할인 카드 클래스를 생성하시오 !
(영화 할인 20%로 해서 생성하시오)

1. 그냥 그대로 상속 받은 버전의 코드

```
class Movie_card(Card):  
    pass
```

2. 오버라이드한 버전의 코드 (부모 consume 안쓰고 내꺼 쓰겠다는 뜻)

```
class Movie_card(Card):  
    pass  
  
    def consume(self, num, loc):  
        if loc == '영화관':  
            num = num*0.8  
            self.card -= num  
            if self.card <= 0:  
                self.card += num  
                print('잔액이 부족합니다.')  
            else:  
                print('{}에서 {}원 사용했습니다.'.format(loc, num))
```

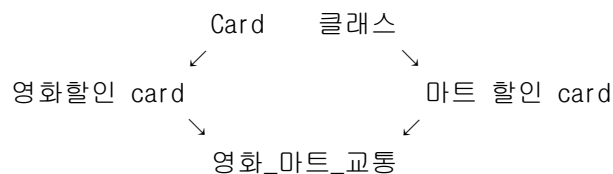
문제 201. 마트 할인 카드 클래스를 생성하시오 ! (할인율 10%)

```
class Mart_card(Card):  
    pass  
  
    def consume(self, num, loc):  
        if loc == '마트':  
            num = num*0.9  
            self.card -= num  
            if self.card <= 0:  
                self.card += num  
                print('잔액이 부족합니다.')  
            else:  
                print('{}에서 {}원 사용했습니다.'.format(loc, num))
```

문제 202. 손자 Card를 생성하시오 !
(영화_마트_교통 Card를 생성하시오)

"죽음의 다이아몬드 상속"

예제 :



```
class Movie_mart_trans_card(Movie_card,Mart_card):  
    def consume(self, num, loc):  
        if loc == '교통':  
            num = num*0.7  
            self.card -= num
```

```

        if self.card <= 0:
            self.card += num
            print('잔액이 부족합니다.')
        else:
            print('{}에서 {}원 사용했습니다.'.format(loc, num))
    elif loc == '마트':
        Mart_card.consume(self, num, loc)
    elif loc == '영화관':
        Movie_card.consume(self, num, loc)
    else:
        Card.consume(self, num, loc)

```

문제 203. 위의 코드에 예외 처리 코드를 임혀서 분모를 0으로 입력해서 위와 같은 에러가 나오면 "나누기를 할 수 없습니다." 라는 메시지가 출력되게 하시오 !

```

def my_divide():
    try:
        x = input(' 분자의 숫자를 입력하세요 ~')
        y = input(' 분모의 숫자를 입력하세요 ~')
        return int(x)/int(y)
    except:
        return '나누기를 할 수 없습니다.'

print( my_divide() )

```

분자의 숫자를 입력하세요 ~10

분모의 숫자를 입력하세요 ~0
나누기를 할 수 없습니다.

설명 : try:
 # 문제가 없을 경우 실행할 코드
except:
 # 문제가 생겼을 경우 실행할 코드

문제 204. 이름을 물어보게 하고 이름을 입력하면 해당 사원의 월급이 출력되는 함수를 생성하시오 !
(emp.csv 와 pandas 를 이용해서)

```

def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')
    name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
    sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]    # 3000 만 가지고오려면 values[0]
을
    return sal                                                         써야함

print( find_sal())

```

문제 205. 위의 코드를 수정해서 사원명을 입력했을 때 없는 사원이름을 입력하면 에러가 나는게 아니라 아래의 메시지가 나오게끔 예외처리를 하시오 !

```

def find_sal():

```

```

import pandas as pd
emp = pd.DataFrame.from_csv('c:WWemp.csv')
name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
try:
    sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
    return sal
except:
    return '해당사원은 없습니다.'

print( find_sal())

```

월급을 알고 싶은 사원명을 입력하세요 ~ WJDTJDGH
해당사원은 없습니다.

아무것도 안넣고 프로그램을 돌리면 "해당사원은 없습니다." 라고 뜨는데 아무것도 안넣고 엔터를 치면 계속 물어보게 할 수 있는 코드를 많이 사용한다

문제 206. while loop 문을 이용해서 이름을 물어봤을 때 아무것도 안넣고 그냥 엔터를 치면 계속 물어보게 하시오 !

```

def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')
    while True:
        name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
        if name != '':
            break
    try:
        sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
        return sal
    except:
        return '해당사원은 없습니다.'

print( find_sal())

```

월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~
월급을 알고 싶은 사원명을 입력하세요 ~ a
해당사원은 없습니다.

문제 207. 위의 코드를 수정해서 이름을 물어볼 때 숫자를 입력하면 숫자는 입력할 수 없습니다. 라는 메시지가 출력되게 하시오 1

없는 사원 이름을 입력하면 "해당 사원 없습니다."
숫자를 입력하면 " 숫자는 입력할 수 없습니다."

내답 :

```

def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')

```

```

while True:
    name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
    if name != '':
        break
    try:
        sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
    return sal
    except:
        try:
            a = int(name)
        return '숫자는 입력할 수 없습니다.'
        except:
            return '해당사원은 없습니다.'

print( find_sal())

업그레이드(한솔) 답:
def find_sal():
    import pandas as pd
    emp = pd.read_csv("D:WWemp_pandas.csv")
    name = input('월급을 알고 싶은 사원명을 입력 하세요~')
    while name == '' or (name in i for i in range(0,10)):
        if name == '':
            name= input('월급을 알고 싶은 사원명을 입력 하세요~')
        else:
            print('숫자는 입력할 수 없습니다.')
    name = input('월급을 알고 싶은 사원명을 입력 하세요~')
    try:
        sal = emp[['sal']][emp['ename']==name.upper()].values[0]
        return sal
    except:
        return '해당 사원은 없습니다.'

find_sal()

```

문제 208. 문제 131 번의 번역할 수 없습니다를 예외처리로 구현하시오 !

번역할 한글을 입력하세요 ~나는 소년 이다
I am boy

번역할 한글을 입력하세요 ~나는 옷을 입습니다
번역할 수 없습니다.

```
dic = {}
```

```
import csv
file = open("c:WWSmt_dic.csv","r")
dic_csv = csv.reader(file)
```

```
for dic_list in dic_csv:
    dic[dic_list[1]] = (dic_list[3], int(dic_list[4]))
```

```
a = []
result = ''
input_kor = input ( '번역할 한글을 입력하세요 ~' )
input_list = input_kor.split(' ')
try:
```



```

for i in input_list:
    a.append([dic[i][1], dic[i][0]])
a.sort()
for i in a:
    result += i[1] + ' '
except:
    print('번역할 수 없습니다.')
print(result)

```

문제 209. 두 숫자를 물어보게 해서 두 수를 나누는 함수를 실행할 때 성공하던 실패하던 아래의 메시지가 무조건 출력되게 하시오 !

```

def my_divide():
    try:
        x = input(' 분자의 숫자를 입력하세요 ~')
        y = input(' 분모의 숫자를 입력하세요 ~')
        z = int(x)/int(y)
        print(z)
    except:
        print( '나누기를 할 수 없습니다.' )
    finally:
        print('정성호가 만든 프로그램입니다.')

my_divide()

```

문제 210. 문제 132 번 코드에서 코드상의 오류를 발생시키고 프로그램을 실행해 보시오 !

```

def my_divide():
    try:
        x = input(' 분자의 숫자를 입력하세요 ~')
        y = input(' 분모의 숫자를 입력하세요 ~')
        z = int(x)/int(y)
        return k
    except Exception as e:
        print(e)
    finally:
        print('정성호가 만든 프로그램입니다.')

my_divide()

```

문제 211. 이름을 물어보게 하고 이름을 입력하면 해당사원의 월급이 출력되게 하는 파이썬 코드를 작성하는데 월급이 3000 이상인 사원들은 아래의 메시지가 출력되게 하시오 !

```

" 해당 사원의 월급은 볼 수 없습니다."

def find_sal():
    import pandas as pd
    emp = pd.DataFrame.from_csv('c:WWemp.csv')
    name = input('월급을 알고 싶은 사원명을 입력하세요 ~ ')
    sal = emp [ ['sal'] ] [ emp['ename'] == name.upper() ].values[0]
    if sal >= 3000:
        raise Exception('해당 사원의 월급은 볼 수 없습니다.')
    else:

```

```

        return sal

print( find_sal())

월급을 알고 싶은 사원명을 입력하세요 ~ scott

Exception: 해당 사원의 월급은 볼 수 없습니다.

```

문제 212. 위의 코드를 수정해서 아래와 같이 결과를 출력하시오 !

```

첫번째 숫자를 입력하세요~ 11113

두번째 숫자를 입력하세요~ 23
11113를 23로 나누면 4가 나머지로 남습니다.

a = int(input('첫번째 숫자를 입력하세요~ '))
b = int(input('두번째 숫자를 입력하세요~ '))
ret = a%b
print('%d를 %d으로 나누면 %d가 나머지로 남습니다.' %(a,b,ret))

```

■ 64. 이진 탐색 알고리즘

문제 213. 순차 탐색 알고리즘을 파이썬으로 프로그래밍 하시오 !

```

data = [1, 7, 11, 12, 14, 23, 44, 47, 51, 64, 67, 77, 140, 672, 871]
        0  1  2    3  4  5    6  7  8  9  10  11  12  13  14

sequence_search(data,7)
'숫자 7이 있습니다.'
sequence_search(data,70)
'숫자 70이 없습니다.'

def sequence_search(data, num):
    for i in data:
        if i == num:
            result = "숫자 %d이 있습니다."%num
            break
        else:
            result = "숫자 %d이 없습니다."%num
    return result

```

문제 214. 이진 탐색 알고리즘을 파이썬으로 프로그래밍 하시오 !

```

binary_search(data,7)
'숫자 7이 있습니다.'
binary_search(data,70)
'숫자 70이 없습니다.'

def binary_search(data, num):
    import numpy as np                # 메디안쓰기위해 넘파이 로드
    data.sort()                       # 데이터 정렬
    while True:                       # 반복

```

```

if len(data) == 1:
    if data[0] == num:
        return '숫자 %d 이 있습니다.'%num
        break
    else:
        return '숫자 %d 이 없습니다.'%num
        break
a = np.array(data)
medi = int(np.median(a))
if len(data)%2 == 0:
    data_medi_index = int(len(data)/2)
    if num < medi:
        data = data[:data_medi_index]
    else:
        data = data[data_medi_index:]
else:
    data_medi_index = int(len(data)/2)
    if num < medi:
        data = data[:data_medi_index]
    elif num > medi:
        data = data[data_medi_index+1:]
    else:
        return '숫자 %d 이 있습니다.'%num
        break
# 만약 데이터리스트에 원소가 1 개라면
# 만약 그 데이터의 값이 num 과 같다면
# 값을 찾은것이니 출력
# 끝났으니 나가기
# 만약 그 데이터의 값이 num 과 같지않다면
# 값이 없는것이니 출력
# 끝났으니 나가기
# 중앙값 구하기 1
# 중앙값 구하기 2      중앙값 : medi
# 만약 데이터의 길이가 짝수라면
# 중앙 값의 인덱스
# 만약 데이터가 중앙값보다 작다면
# 중앙값 뒤의 데이터를 날림
# 만약 데이터가 중앙값보다 크다면
# 중앙값 앞의 데이터를 날림
# 만약 데이터의 길이가 홀수라면
# 중앙값의 인덱스
# 만약 데이터가 중앙값 보다 작다면
# 중앙값(포함) 뒤의 데이터를 날림
# 만약 데이터가 중앙값 보다 크다면
# 중앙값(포함) 앞의 데이터를 날림
# 만약 데이터가 중앙값과 일치한다면
# 값을 찾은것이니 출력
# 끝났으니 나가기

```

while True 문을 이용하되 결과가 출력되어야 하는 모든 부분에 break 를 걸어서 무한루프가 되지 않게 한다.

리스트의 길이가 만약 짝수라면 중앙값이 리스트 안에 있지 않기 때문에 짝수 일 경우와 홀수 일 경우로 나뉘야 한다.

짝수일 경우 중앙값이 데이터안에 있을 수 없으므로 같다는 조건은 생략이 가능하다.

홀수일 경우 크거나 같다, 작거나 같다 보다는 크다 작다를 이용해 중앙값까지 같이 날리는것이 효율이 더 좋다.

결과값이 출력되는 경우의수 1. 우연히 중앙값과 같게 되었을 때 2. 리스트의 원소를 1 개까지 줄인다음 확인

idea : 만약 리스트가 1 개라면? 모든 루프 돌지말고 바로 끝내기 위해 1 개일 경우를 제일 위로

■ 65. Pandas 를 이용해서 조건절 사용 데이터 검색 방법

오라클	vs	판다스
1. between ... and		1000 <= emp['sal'] and emp['age'] <= 3000 emp['sal'].between(1000,3000)
2. in		isin
3. is null		isnull()
4. like		apply 함수, lambda 표현식

문제 215. 우리반(emp7.csv)에서 이름이 이소진 학생의 이름과 나이를 출력하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\Wemp7.csv')

print (emp[['ename','age']][emp['ename'] == '이소진'])
```

문제 216. 나이가 25 에서 28 사이인 학생들의 이름과 나이를 출력하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\Wemp8.csv')
print (emp[ ['ename','age'] ] [ emp['age'].between(25,28) ] )

import pandas as pd
emp = pd.read_csv('c:\Wemp8.csv')
print (emp[ ['ename','age'] ] [ (25 <= emp['age']) & (emp['age'] <= 28) ] )
```

문제 217. 나이가 25 에서 28 사이가 아닌 학생들의 이름과 나이를 출력하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\Wemp8.csv')
print (emp[ ['ename','age'] ] [ ~emp['age'].between(25,28) ] )
```

문제 218. 7기 학생들 중에서 전공이 환경공학, 신학인 학생들의 이름과 전공을 출력하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\Wemp7.csv')
print (emp[ ['ename','major'] ] [ emp['major'].isin( [ '환경공학','신학' ] ) ] )

# 조건절에 괄호안에 리스트를 넣어야 함 !
```

■ 66. Pandas 를 이용한 조인

문제 219. Pandas 를 이용해서 이름과 부서위치를 출력하시오 !

```
import pandas as pd

emp = pd.read_csv('c:WWemp.csv')
dept = pd.read_csv('c:WWdept1.csv')

result = pd.merge( emp, dept, on = 'deptno')    # 조인을 해서 출력한다 연결고리(deptno)

print ( result[['ename','loc']])
```

문제 220. 부서위치가 DALLAS 인 사원들의 이름과 부서위치를 출력하시오 !

```
import pandas as pd

emp = pd.read_csv('c:WWemp.csv')
dept = pd.read_csv('c:WWdept1.csv')

result = pd.merge( emp, dept, on = 'deptno')

print ( result[['ename','loc']][result['loc'] == 'DALLAS'])
```

문제 221. 이름과 부서위치를 출력하는데 아래의 SQL 의 outer join 을 구현하시오 ~

```
SQL> select e.ename, d.loc
      from emp e, dept d
      where e.deptno = d.deptno (+);

import pandas as pd

emp = pd.read_csv('c:WWemp.csv')
dept = pd.read_csv('c:WWdept1.csv')

result = pd.merge( emp, dept, on = 'deptno', how = 'right')

print ( result[['ename','loc']])
```

문제 222. 아래의 full outer join 을 pandas 로 구현하시오 !

```
SQL> select e.ename, d.loc
      from emp e full outer join dept d
      on ( e.deptno = d.deptno );

import pandas as pd

emp = pd.read_csv('c:WWemp.csv')
dept = pd.read_csv('c:WWdept1.csv')

result = pd.merge( emp, dept, on = 'deptno', how = 'outer')
```

```
print ( result[['ename','loc']])
```

※ right, left, outer

■ 67. Pandas 를 이용한 서브쿼리

문제 223. JONES 보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오 !

```
SQL> select ename, sal
      from emp
      where sal > (select sal
                   from emp
                   where ename = 'JONES');

import pandas as pd

emp = pd.read_csv('c:\\emp.csv')
jonessal = emp[['sal']][emp['ename'] == 'JONES'].values[0][0]
print(jonessal)
print (emp[['ename', 'sal']][emp['sal'] >= jonessal])

※ values[0]으로 출력하면 리스트형식
```

문제 224. SCOTT 의 직속상사의 이름을 출력하시오 !

```
SQL> select ename
      from emp
      where empno = (select mgr
                    from emp
                    where ename = 'SCOTT');

import pandas as pd

emp = pd.read_csv('c:\\emp.csv')

scottmgr = emp[['mgr']][emp['ename'] == 'SCOTT'].values[0][0]
print(scottmgr)
print (emp[['ename']][emp['empno'] == scottmgr])
```

문제 225. 관리자인 직원들의 이름을 출력하시오 !

```
SQL> select ename
      from emp
      where empno in (select mgr from emp);

import pandas as pd

emp = pd.read_csv('c:\\emp.csv')
print (emp[['ename']][emp['empno'].isin(emp['mgr'])])
```

문제 226. 관리자가 아닌 직원들의 이름을 출력하시오 !

```
SQL> select ename
      from emp
      where empno in (select mgr from emp);
```

```
import pandas as pd

emp = pd.read_csv('c:\\\\emp.csv')
print (emp[['ename']][~emp['empno'].isin(emp['mgr'])])
```

문제 227. 최대월급을 받는 사원의 이름과 월급을 출력하시오 !

```
SQL> select ename, sal
      from emp
      where sal = (select max(sal)
                  from emp);
```

```
import pandas as pd

emp = pd.read_csv('c:\\\\emp.csv')
max_sal = emp['sal'].max()
print (emp[['ename', 'sal']][emp['sal'] == max_sal ])
```

※ max_sal2 = max(emp['sal'])로 해도 됨

문제 228. 이름, 월급 * 0.15 를 출력하시오 !(판다스를 이용해서 출력)

```
import pandas as pd

emp = pd.read_csv('c:\\W\\emp.csv')
emp['sal'] = emp['sal']*0.15

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

문제 229. 위의 결과를 다시 출력하는데 월급을 출력할 때 소수점 첫째 자리에서 반올림 해서 출력하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\\W\\emp.csv')
emp['sal'] = round( emp['sal']*0.15, 0)
print(emp[['ename','sal']])
```

```
※ 7      JAMES  142.50  ----->  7      JAMES  142.0  # 2가 좋아서 2가 나옴
   8      WARD   187.50  ----->  8      WARD   188.0  # 8이 좋아서 8이 나옴
반올림이 안되었다! 왜???
```

```
"R 과 파이썬은 짝수를 좋아한다."
7.5 ----> 8
2.5 ----> 2
```

```
※ 설명 : 파이썬은 짝수를 좋아하기 때문에 142.5 ---> 142로 반올림 되어서 출력된다.
파이썬에서는 기본적으로 이것을 해결하는 함수가 없다.
0.5 일때는 짝수를 좋아하게끔 반올림 안되고 0.51 일 때는 반올림 된다.
```

문제 230. 문제 229 번을 제대로 반올림 되게 하시오 ! (오라클 처럼 반올림 되게 하시오 !

```
import pandas as pd
emp = pd.read_csv('c:\\W\\emp.csv')
emp['sal'] = round( emp['sal']*0.15+0.00001, 0)
print(emp[['ename','sal']])
```

문제 231. 판다스를 이용하지 말고 이름과 연봉 (sal*12)을 출력하시오 !

```
import csv

file = open('c:\\W\\emp2.csv','r')
emp_csv = csv.reader(file)

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

문제 232. 아래의 리스트에서 소수만 걸러내시오 !

```
a = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
결과 : 11, 13, 17, 19
```

선헤코드 :

```
for i in a:
    num=0
    for j in range(2,i):
        if i%j==0:
            num+=1
    if num==0:
        print(i)
```

문제 233. 선헤코드로 아래의 함수를 생성하시오 !

```
def getPrime(x):
    for i in x:
        num=0
        for j in range(2,i):
            if i%j==0:
                num+=1
        if num==0:
            print(i)
```

문제 234. getPrime 함수와 filter 함수를 이용해서 a 리스트에서 소수를 걸러내시오!

```
def getPrime(x):
    num=0
    for j in range(2,x):
        if x%j==0:
            num+=1
    if num==0:
        return x
```

```
a = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

```
result=filter(getPrime, a)
print(list(result))
```

※ filter 가 돌려면 요소를 추출하는 함수가 값으로 입력되게 만들어야 한다.
filter 자체에 for 문이 있어서 리스트의 요소마다 추출하는 것 같음!

문제 235. 사원테이블에서 최대월급을 출력하시오 !

1. Pandas 이용한 방법

```
import pandas as pd

emp = pd.read_csv('c:WWemp.csv')
print( emp['sal'].max() )
```

2. Pandas 이용하지 않은 방법

```
import csv

file = open('c:WWemp2.csv','r')
emp_csv = csv.reader(file)
```

```
a = []  
for i in emp_csv:  
    a.append(int(i[5]))  
print(max(a))
```

■ 73. 판다스와 오라클 그룹함수 비교

문제 236. 직업, 직업별 최대 월급을 출력하시오 !

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

```
import pandas as pd
```

```
emp = pd.read_csv('c:WWemp.csv')
result = emp.groupby('job')['sal'].max()
print(result)
```

문제 237. 부서번호, 직업, 부서번호별 직업별 토달월급을 출력하시오 !

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

```
import pandas as pd
```

```
emp = pd.read_csv('c:WWemp.csv')
result = emp.groupby(['deptno', 'job'])['sal'].max()
print(result)
```

deptno	job	max(sal)
10	CLERK	1300
	MANAGER	2450
	PRESIDENT	5000
20	ANALYST	3000
	CLERK	1100
	MANAGER	2975
30	CLERK	950
	MANAGER	2850
	SALESMAN	1600

문제 238. 직업, 직업별 토달월급을 출력하는데 직업이 SALESMAN 은 제외하고 출력하고
직업별 토달월급이 6000 이상인 것만 출력하시오 !

```
import pandas as pd
```

```
emp = pd.read_csv('c:WWemp.csv')
emp['job'] = emp['job'][emp['job'] != 'SALESMAN']
result = emp.groupby('job')['sal'].sum()
result1 = result >= 6000
print(result[result1 == True])
```

■ 74. pandas 와 오라클 분석함수 비교

"순위를 출력하는 분석함수를 판다스에서 구현"

문제 239. 이름, 월급, 순위를 출력하시오 !

```
import pandas as pd

emp = pd.read_csv('c:\wwemp.csv')
emp['rnk'] = emp['sal'].rank(ascending = False).apply(int)      #apply(함수) 함수 적용
emp = emp.sort_values('rnk',ascending=True)
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

문제 240. 직업, 직업별 토달월급, 순위를 출력하시오 !
순위는 직업별 토달월급이 높은 것에 대한 순위이다.

```
import pandas as pd

emp = pd.read_csv('c:\wwemp.csv')
emp = emp.groupby('job')['sal'].sum().reset_index()             # reset_index : DataFrame 으로 저장
emp['rnk'] = emp['sal'].rank(ascending = False).apply(int)
emp = emp.sort_values('rnk',ascending=True)
print(emp[['job','sal','rnk']])
```

■ (알고리즘 문제 3 번째) 버블 정렬을 파이썬으로 구현하시오 !

문제 245. 아래의 리스트를 버블 정렬하는 함수를 생성하시오!
(loop 문으로 구현)

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
print(bubble_sort(a))

def bubble_sort(data):
    dummy = 0 #자리를 바꾸기 위한 더미변수
    for j in range(len(data)):
        for i in range(len(data)-1):
            if data[i] > data[i+1]:
                dummy = data[i]
                data[i] = data[i+1]
                data[i+1] = dummy
    return data
```

-----한솔-----

```
def hansol(data, n=1, m=1):
    if m < len(data):
        if data[m-1] > data[m]:
            data[m-1], data[m] = data[m], data[m-1]
            return hansol(data, n, m+1)
        return hansol(data, n, m + 1)
    elif n == len(data):
        return data
    elif m == len(data):
        return hansol(data, n+1)

a = [5,4,3,2,1,7,8,10]
print( hansol(a) )
```

문제 246. 파이썬의 리스트가 스택구조임을 확인하시오 !

↓
후입선출

```
a = []

a.append(1)
a.append(2)
print(a)      # [1, 2]

a.append(3)
print(a)      # [1, 2, 3]
```

문제 247. 다시 숫자를 빼내시오 (나중에 들어온 숫자가 먼저 나가게끔 하시오!)

```
a.pop()
print(a)      # [1, 2]

a.pop()
print(a)      # [1]
```

문제 248. 아래와 같이 영화 대본에서 특정 단어가 몇개 있는지 찾는 스크립트를 구현하기 위해 두개의 질문을 물어보게 하시오 !

```
print( find_word() )

분석할 스크립트명을 입력하세요 ~ winter.txt
찾을 단어를 입력하세요 ~ elsa

결과 : c:\WWW\winter.txt
      elsa

def find_word():
    scr = input('분석할 스크립트명을 입력하세요 : ')
    an_word = input('찾을 단어를 입력하세요 ~ ')
    file = 'c:\WWW' + scr
    print(file)
    print(an_word)

find_word()
```

문제 249. 아래와 같이 겨울왕국 대본에서 elsa를 검색해서 몇건 나오는지 출력하는 함수를 생성하시오 !

```
x = input( 'input file name : ' )
y = input( 'input word      : ' )
from time import sleep

file = open("c:\WWW%s" %x, 'r')
sum=0

for winter_list in file:
    a = winter_list.split(' ')
    for b in a:
```

```

sum += b.lower().count(y)
msg = 'Wr %d' %sum
print(msg, end = '')
sleep(0.000001)

```

문제 250. 위의 함수를 클래스로 구성하여 아래와 같이 실행될 수 있게 하시오 !
(클래스 이름 : Text_mining())

```

tm = Text_mining()

print( tm.find_word() )

input file name : winter.txt

input word      : elsa

```

329

```

class Text_mining:
    def find_word(self):
        script = input( 'input file name : ' )
        search_word = input( 'input word      : ' )
        from time import sleep

        file = open("c:WWW%s" %script, 'r')
        sum=0

        for winter_list in file:
            a = winter_list.split(' ')
            for b in a:
                sum += b.lower().count(search_word)
            msg = 'Wr %d' %sum
            print(msg, end = '')
            sleep(0.000001)

```

문제 251. 겨울왕국 대본에서 단어들만 출력하시오!

```

def find_word():
    file = open('c:WWWwinter.txt','r')
    sum = 0
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            print(b)

find_word()

```

문제 252. 위의 출력된 결과중에 엔터를 정제하시오 !

```

import re                                # 데이터 정제시 중요한 정규식 모듈
def find_word():
    file = open('c:WWWwinter.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()

```

※ re.sub : 엔터 Wn 를 ''로 변경해라!

문제 253. 겨울왕국 대본에서 " - " 가 출력 안되게 하시오!

```

import re
def find_word():
    file = open('c:WWWwinter.txt','r')
    sum = 0
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            print( re.sub('[^A-z]', '', b) )

find_word()

```

문제 254. 긍정단어집을(positive-words.txt)를 위의 스크립트를 이용해서 한단어씩 출력되게 하시오 !

```

import re
def positive_words():
    file = open('c:WWWpositive-words.txt','r')
    sum = 0
    for positive_list in file:
        a = positive_list.split(' ')
        for b in a:
            print( re.sub('[^A-z]', '', b).lower() )

positive_words()

```

문제 255. 위의 긍정단어들을 p_list 라는 비어있는 리스트에 전부 append 시키도록 코드를 변경하시오 !

```

import re
def positive_words():
    file = open('c:WWWpositive-words.txt','r')
    sum = 0
    p_list = []
    for positive_list in file:
        a = positive_list.split(' ')
        for b in a:
            p_list.append( re.sub('[^A-z]', '', b).lower() )
    return p_list

p_list2 = positive_words()

```

문제 256. 위의 두 함수를 이용해서 find_word() 함수에서 겨울 왕국 대본에 긍정 단어가 몇 개 있는지

출력하도록 스크립트를 수정하시오.

```
import re
def positive_word():
    file = open('c:\\WWW\\positive-words.txt','r')
    sum = 0
    p_list = []
    for positive_list in file:
        a = positive_list.split(' ')
        for b in a:
            p_list.append( re.sub('[^A-z]', '', b).lower() )
    return p_list

p_list2 = positive_word()

import re
def find_word():
    file = open('c:\\WWW\\winter.txt','r')
    sum = 0
    p_list2 = positive_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in p_list2:
                sum = sum+1
    print(sum)

find_word()
```

문제 257. 겨울왕국에 부정단어가 얼마나 있는지 알아내기 위한 함수를 생성하시오 !

positive_word() # 긍정 단어를 리스트화 하는 함수
negative_word() # 부정 단어를 리스트화 하는 함수
p_count() # 긍정단어의 갯수 세는 함수
n_count() # 부정단어의 갯수 세는 함수
find_word() # 스크립트의 특정 단어의 갯수를 세는 함수

```
import re
def positive_word():
    file = open('c:\\WWW\\positive-words.txt','r')
    sum = 0
    p_list = []
    for positive_list in file:
        a = positive_list.split(' ')
        for b in a:
            p_list.append( re.sub('[^A-z]', '', b).lower() )
    return p_list

p_list2 = positive_word()

import re
def negative_word():
    file = open('c:\\WWW\\negative-words.txt','r')
    sum = 0
    n_list = []
    for negative_list in file:
        a = negative_list.split(' ')
        for b in a:
            n_list.append( re.sub('[^A-z]', '', b).lower() )
    return n_list
```

```

        for b in a:
            n_list.append( re.sub('Wn', '', b).lower() )
    return n_list

n_list2 = negative_word()

import re

def find_word():
    file = open('c:WWWwinter.txt','r')
    p_sum = 0
    n_sum = 0
    p_list2 = positive_word()
    n_list2 = negative_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in p_list2:
                p_sum = p_sum + 1
            if re.sub('[^A-z]', '', b).lower() in n_list2:
                n_sum = n_sum + 1
    print(p_sum)
    print(n_sum)

find_word()

```

문제 258.

```

tm = Text_mining()
tm.p_count()
tm.n_count()
positive_word()
negative_word()

class Text_mining:
    def __init__(self):
        self.script = input( 'input file name : ' )
        import re
        from time import sleep

    def find_word(self):
        search_word = input( 'input word      : ' )
        file = open("c:WWW%s" %self.script, 'r')
        sum=0
        for winter_list in file:
            a = winter_list.split(' ')
            for b in a:
                sum += b.lower().count(search_word)
            msg = 'Wr %d' %sum
            print(msg, end = '')
            sleep(0.000001)

    def positive_word(self):
        file = open('c:WWWpositive-words.txt','r')
        sum = 0
        p_list = []
        for positive_list in file:

```

```

        a = positive_list.split(' ')
        for b in a:
            p_list.append( re.sub('Wn', '', b).lower() )
    return p_list

def negative_word(self):
    file = open('c:WWnegative-words.txt','r')
    sum = 0
    n_list = []
    for negative_list in file:
        a = negative_list.split(' ')
        for b in a:
            n_list.append( re.sub('Wn', '', b).lower() )
    return n_list

def p_count(self):
    file = open('c:WW%s'%self.script,'r')
    p_sum = 0
    p_list2 = positive_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in p_list2:
                p_sum = p_sum + 1
    print(p_sum)

def n_count(self):
    file = open('c:WW%s'%self.script,'r')
    n_sum = 0
    n_list2 = negative_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in n_list2:
                n_sum = n_sum + 1
    print(n_sum)

```

문제 259. 인공지능이라는 키워드로 중앙일보에서 기사검색을 한 mydata3.txt 를 m_list 라는 비어있는 변수에 단어별로 쪼개서 담으시오 !

```

def hangle_list():
    file = open('c:WWmydata3.txt','r', encoding = 'UTF8')
    m_list = []
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            m_list.append( re.sub('[^가-힣 A-z]', '', b))
    return m_list

print(hangle_list())

```

* Wordcloud 모듈 설치 방법

아나콘다 프롬프트 창을 열고

```
> conda install -c conda-forge wordcloud
```

문제 260. 인공지능 기사 검색한 리스트를 워드 클라우드로 시각화 하시오 !

```
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://")
# 텍스트 파일이 있는 상위 디렉토리를 path로 지정
text = open(path.join(d, "mydata3.txt"), mode="r", encoding="UTF-8").read()
# 텍스트파일을 open 하는데 reading 만 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',
# 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_color 는 워드클라우드 배경색을 나타냅니다. 'black'으로하면 검은색이 됩니다.
                        height=800,
# width와 height 는 워드클라우드의 크기를 지정해 줍니다.
                        colormap='jet').generate(text)
# colormap 은 워드 색깔을 지정해 주는데 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")
```

```
# pyplot 에 x, y 축 표시를 없앱니다.
plt.show()
```

문제 261. EBS 프로그램중에 레이디버그 게시판 글들을 워드 클라우드로 시각화 하시오!

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

d = path.dirname("c://")
text = open(path.join(d, "ladybug3.txt"), mode="r", encoding="UTF-8").read()

wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs',
                        stopwords=STOPWORDS, background_color='red',
                        width=600,
                        height=500,
                        colormap='jet').generate(text)

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

문제 262. 워드 클라우드를 그리는 함수를 Text_mining() 클래스에 포함시켜서 아래와 같이 실행되게 하시오 !

```
tm = Text_mining()
tm.word_cloud()

input file name : ladybug3.txt

class Text_mining:
    def __init__(self):
        self.script = input( 'input file name : ')
        import re
        from time import sleep
        from wordcloud import WordCloud, STOPWORDS
        import matplotlib.pyplot as plt
        from os import path

    def find_word(self):
        search_word = input( 'input word      : ')
        file = open("c:WW%s" %self.script, 'r')
        sum=0
        for winter_list in file:
            a = winter_list.split(' ')
            for b in a:
                sum += b.lower().count(search_word)
                msg = 'Wr %d' %sum
                print(msg, end = '')
                sleep(0.000001)

    def positive_word(self):
        file = open('c:WWpositive-words.txt','r')
```

```

sum = 0
p_list = []
for positive_list in file:
    a = positive_list.split(' ')
    for b in a:
        p_list.append( re.sub('Wn', '', b).lower() )
return p_list

def negative_word(self):
    file = open('c:WWnegative-words.txt','r')
    sum = 0
    n_list = []
    for negative_list in file:
        a = negative_list.split(' ')
        for b in a:
            n_list.append( re.sub('Wn', '', b).lower() )
    return n_list

def p_count(self):
    file = open('c:WW%s'%self.script,'r')
    p_sum = 0
    p_list2 = positive_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in p_list2:
                p_sum = p_sum + 1
    print(p_sum)

def n_count(self):
    file = open('c:WW%s'%self.script,'r')
    n_sum = 0
    n_list2 = negative_word()
    for winter_list in file:
        a = winter_list.split(' ')
        for b in a:
            if re.sub('[^A-z]', '', b).lower() in n_list2:
                n_sum = n_sum + 1
    print(n_sum)

def word_cloud(self):
    d = path.dirname("c://")
    text = open(path.join(d, "%s"%self.script), mode="r", encoding="UTF-8").read()
    wordcloud = WordCloud(font_path='C://Windows//Fonts//BMHANNA_11yrs',
                           stopwords=STOPWORDS, background_color='white',
                           width=600,
                           height=500,
                           colormap='jet').generate(text)
    plt.figure(figsize=(13,13))
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.show()

tm = Text_mining()
tm.word_cloud()

```

활용하시오

1. 워드클라우드를 그릴려면 1번을 누르세요.
2. 스크립트에서 단어를 검색하려면 2번을 누르세요.
3. 스크립트에서 긍정단어의 갯수를 알려면 3번을 누르세요.
4. 스크립트에서 부정단어의 갯수를 알려면 4번을 누르세요.

숫자 1 입력하고 엔터를 치면

워드클라우드를 그릴 텍스트를 입력하세요 !

```
def print_menu():
    print("1. 워드클라우드를 그릴려면 1번을 누르세요.")
    print("2. 스크립트에서 단어를 검색하려면 2번을 누르세요.")
    print("3. 스크립트에서 긍정단어의 갯수를 알려면 3번을 누르세요.")
    print("4. 스크립트에서 부정단어의 갯수를 알려면 4번을 누르세요.")
    print("5. 종료")
    menu = input("메뉴선택 : ")
    return int(menu)

def run():
    import Text_mining
    tm = Text_mining.Text_mining()
    while 1:
        menu = print_menu()
        if menu == 1:
            tm.word_cloud()
        if menu == 2:
            tm.find_word()
        if menu == 3:
            tm.p_count()
        if menu == 4:
            tm.n_count()
        if menu == 5:
            break

run()
```

* 솔루션을 구현하기 위해서 저축해야하는 핵심 엔진 코드

1. text_mining() 클래스
2. 웹 스크롤링 클래스(네이버, 구글, 다음, 국내 메이저 신문사 데이터)
3. 머신러닝 클래스

문제 264. 위의 코드를 Comprehension 코드로 변경하시오 !

```
[ 출력 표현식 for 요소 in 입력 시퀀스 if 조건식 ]

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

print(newlist)
```


[1, 4, 9]

문제 265. 아래의 코드를 Comprehension 코드로 변경하시오 !

변경전 :

```
import csv

file = open('c:WWemp2.csv','r')
emp_csv = csv.reader(file)
emp_sal = []
for i in emp_csv:
    emp_sal.append(int(i[5]))
print( max(emp_sal))
```

변경후 :

```
import csv

file = open('c:WWemp2.csv','r')
emp_csv = csv.reader(file)
emp_sal = [int(i[5]) for i in emp_csv]
print( max(emp_sal))
```

문제 266. 위의 코드를 Comprehension 안쓴 코드로 작성하시오 !

```
oldset = [1, 2, 3, 3, 4]
newset = []
```

```
for i in oldset:
    newset.append(i*i)
```

```
newset = set(newset)
print(newset)
```

문제 267. 위의 코드를 Comprehension 안쓴 코드로 변경하시오 !

```
id_name = {1 : '김준구', 2: '장보겸', 3: '정지엽'}
a = {}
for key,val in id_name.items():
    a[val] = key
print(a)
```

■ 삽입정렬

문제 268. (오늘의 마지막 문제) 알고리즘 문제
삽입정렬 알고리즘을 파이썬으로 구현하시오 !

```
def insert_sort(data):
    sort_data = [min(data)]
    sort_data.append(max(data))
    data.remove(min(data))
    data.remove(max(data))
    for i in data:
        for j in sort_data:
            if j > i:
                ind = sort_data.index(j)
                sort_data.insert(ind,i)
                break
            elif j < i:
                continue
            elif sort_data.index(j) == len(sort_data)-1:
                sort_data.append(i)
    return sort_data

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

print(insert_sort(a))
```

좋은 방법(준하) :

```
a = [5,4,3,2,1,8,7,10, 9]
def insert_sort(data):
    for i in range(1,len(data)):
        for j in range(i):
            if data[i] < data[j]:
                tmp = data[i]
                del data[i]
                data.insert(j,tmp)
    return data
print(insert_sort(a))
```

문제 269. 겨울왕국 대본에 알파벳 문자의 갯수가 어떻게 되는가?

```
text_file = open('c:\WWWwinter.txt')
lines = text_file.readlines()      # 텍스트 파일을 한줄씩 읽기
total = 0

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

print(total)
```

82180

```
※ lines : [' "Final Shooting DraftWn',
            'byWn',
            'Jennifer LeeWn',
            '9/23/13Wn',
            'Property of the Walt Disney Animation StudiosWn' .....]
겨울왕국 대본을 한라인씩 리스트의 요소로 담아냄
```

문제 270. 겨울왕국 대본에 숫자가 몇개 있는지 출력하시오 !

```
text_file = open('c:\WWWwinter.txt')
lines = text_file.readlines()
total = 0

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

print(total)
```

264

문제 272. 겨울왕국 대본의 전체 문자의 갯수는 어떻게 되는가?

```
import re
text_file = open('c:\WWWwinter.txt')
text = text_file.read()
text = re.sub('\Wn', '', text)
print(len(text))
```

105199

문제 273. 겨울왕국 대본에서 특수문자의 개수를 출력하시오 !

전체문자의 개수 - 알파벳의 갯수 - 숫자의 개수 - 공백의 개수

```
text_file = open('c:\WWWwinter.txt')
lines = text_file.readlines()
```

```

total = 0
total1 = 0

for s in lines:
    cnt = len(s)
    cnt_alpha = sum( i.isalpha() for i in s)
    cnt_digit = sum( i.isdigit() for i in s)
    cnt_space = sum( i.isspace() for i in s)
    total += cnt
    total1 += cnt_alpha + cnt_digit + cnt_space

print( total - total1 )

7083

```

문제 274. 구구단 2단을 파이썬으로 구현하시오 ! (loop 문 사용)

```

2 x 10 = 20
2 x 9 = 18
2 x 8 = 16
2 x 7 = 14
2 x 6 = 12
2 x 5 = 10
2 x 4 = 8
2 x 3 = 6
2 x 2 = 4
2 x 1 = 2

for i in range(10,0,-1):
    print('2 x %d = %d' %(i, i*2))

```

문제 275. 위의 구구단을 재귀함수로 구현하시오!

```

countdown(10)

2 x 10 = 20
2 x 9 = 18
2 x 8 = 16
2 x 7 = 14
2 x 6 = 12
2 x 5 = 10
2 x 4 = 8
2 x 3 = 6
2 x 2 = 4
2 x 1 = 2

def countdown(n):
    if n != 0:
        print('2 x %d = %d' %(n,n*2))
        countdown(n-1)

```

★★ 재귀함수 내부적으로 일어나는 일★★

함수 내에서 다시 자신을 호출한 후 그 함수가 끝날 때 까지 함수 호출
이후의 명령문은 실행하지 않는다.

countdown(10)	#	실행순서
print('2 x %d = %d' %(10,10*2))	#	11
countdown(9)		
print('2 x %d = %d' %(9,9*2))	#	10
countdown(8)		
print('2 x %d = %d' %(8,8*2))	#	9
countdown		
print('2 x %d = %d' %(7,7*2))	#	8
countdown(6)		
print('2 x %d = %d' %(6,6*2))	#	7
countdown(5)		
print('2 x %d = %d' %(5,5*2))	#	6
countdown(4)		
print('2 x %d = %d' %(4,4*2))	#	5
countdown(3)		
print('2 x %d = %d' %(3,3*2))	#	4
countdown(2)		
print('2 x %d = %d' %(2,2*2))	#	3
countdown(1)		
print('2 x %d = %d' %(1,1*2))	#	2
countdown(0)		
출력할거 없음 끝	#	1

문제 276. loop 문을 이용하지 말고 재귀함수를 이용해서 아래의 결과를 출력하시오 !

```
def star_print(n):
    if n != 0:
        star_print(n-1)
        print(n*'★')
```

```
star_print(7)
```

```
★
★★
★★★
★★★★
★★★★★
★★★★★★
★★★★★★★
★★★★★★★★
```

문제 278. factorial 함수를 재귀함수로 구현하시오 !

```
def factorial(n):
    if n == 1:
        return 1
    elif n > 1:
        return n*factorial(n-1)
```

```
print(factorial(5))
```

문제 279. loop 문을 이용해서 최대공약수를 구하시오 !

```
def find_gcd(a, b):
    dummy = 0
    i = 1
    c = 1
    if a <= b:
        dummy = a
        a = b
        b = dummy
    while i < b:
        i += 1
        if a%i == 0:
            a = a / i
        if b%i == 0:
            b = b / i
            c = c * i
        i = 1
    return c

print(find_gcd(20,16))
```

문제 280. (오늘의 마지막 문제) 최대공약수 구하는 함수를 재귀함수로 구현하시오 !

```
def find_gcd(a, b):
    if a <= b:
        dummy = a
        a = b
        b = dummy
```

문제 281. 아래의 연설문에 '국민'이라는 단어가 몇번 나왔는지 count 하시오 !

```
x = '''안녕하십니까 안철수입니다.
```

```
.....
```

```
그렇습니다. 미래는 지금 우리 앞에 있습니다.
```

```
고맙습니다." '''
```

```
word_count1 = x.count('국민')
print(word_count1)
```

※ 안철수 연설문.txt 참고

문제 283. 아래의 log 변수를 가지고 아래의 결과를 출력하시오 !

```
log = 'name:홍길동 age:17 sex:남자 nation:조선'
log_split = log.split(' ')
for i in log_split:
    st = i.split(':')
    print('%s --> %s'%(st[0],st[1]))
```

문제 284. split 이용하지 말고 가급적 replace 를 이용해서 log 리스트에서 아래의 결과를 출력하시오 !

```
log = 'name:홍길동 age:17 sex:남자 nation:조선'
log_split = log.split(' ')
for i in log_split:
    st = i.replace(':', ' --> ')
    print(st)
```

문제 285. ascii 코드표를 파이썬으로 출력하시오 !

```
alpha = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
for i in range(len(alpha)):
    a_txt = alpha[i].encode()
    print (a_txt[0], '-->', alpha[i])
```

답 :

```
import string
for i in range(128):
    print( i, '-->', chr(i))
```

문제 288. (카카오 경력 공채 문제를 이해하기 위해서 알아야할 기본)
Wt (탭) 을 이용해서 아래의 결과를 출력하시오 !

```
10      11      12      13      14      15      16      17      18

for i in range(10,19):
    print( i, end = 'Wt')
```

문제 289. (카카오 경력 공채 문제를 이해하기 위해서 알아야할 기본)
아래의 탭으로 구분된 단어를 k_pop 리스트에 넣으시오!

```
song = 'YesterdayWtLet it beWtGee(지)Wt 소원을 말해봐Wt 다시 만난 세계'
```

결과 :

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

```
song = 'YesterdayWtLet it beWtGee(지)Wt 소원을 말해봐Wt 다시 만난 세계'
kpop_list = song.split('Wt')
print(kpop_list)
```

문제 290. 아래의 리스트를 무작위로 섞어서 3번 출력하시오 !

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

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

문제 302. 아래의 리스트에서 토성을 지우시오 !

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

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

문제 303. remove 로 토성과 화성을 같이 지우시오 !

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

print(a)
# TypeError: remove() takes exactly one argument (2 given)
# 답 : 안된다.
```

문제 306. 아래의 리스트를 ABCD 순으로 정렬하시오 !

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

print(namelist)
# 결과 :
# ['Aimy', 'Bob', 'Kelly', 'Mary', 'Michale', 'Same', 'Tom']
```

문제 307. namelist 의 요소를 완전히 정렬해 버리지 않고 namelist 의 요소를 정렬해서 b 라는 변수에 담아서 출력하시오 !

```
namelist = ['Mary', 'Same', 'Aimy', 'Tom', 'Michale', 'Bob', 'Kelly']
b = list(sorted(namelist))

print(b)
# ['Aimy', 'Bob', 'Kelly', 'Mary', 'Michale', 'Same', 'Tom']
```

※ sort 와 sorted 의 차이는?

namelist 의 데이터를 완전히 정렬하지 않고 정렬된 data 를 다른 변수에 담을 수 있는지 없는 지의 차이

문제 308. 아래의 리스트를 enumerate 를 이용해서 아래의 결과로 출력하시오 !

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
b = []
for i, name in enumerate(a):
    b.append((i,name))
print(b)
```



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

문제 311. 아래의 두개의 리스트를 이용해서 아래의 딕셔너리를 생성하시오 ! (enumerate를 사용하세요 ~)

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

for i, j in enumerate(solar1):
    sol[j] = solar2[i]

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

문제 312. (카카오 디퍼닝 파트 알고리즘 문제) 아래의 두개의 리스트를 아래의 딕셔너리 출력하시오 !

```
artist = ['Beatles', '소녀시대']
music = ['Yesterday', '소원을 말해봐']
art_mus = {}

for i, j in enumerate(artist):
    art_mus[j] = music[i]

print(art_mus)
# {'Beatles': 'Yesterday', '소녀시대': '소원을 말해봐'}
```

문제 313. 아래의 두개의 리스트로 아래의 딕셔너리를 생성하시오 !

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

from collections import defaultdict
artist_music = defaultdict(list)

for i in artist:
    artist_music[i]

for art_ind, art in enumerate(artist):
    artist_music[art].append(music[art_ind])

print(artist_music)
# {'Beatles': ['Yesterday', 'Let it be'], '소녀시대': ['소원을 말해봐', 'Gee']}
```

문제 314. 오늘 배운 shuffle 을 이용해서 수행될때마다 box 딕셔너리의 요소들이 shuffle 되게 하시오 !

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

```

from collections import defaultdict
from random import shuffle
artist_music = defaultdict(list)

for i in artist:
    artist_music[i]

for art_ind, art in enumerate(artist):
    artist_music[art].append(music[art_ind])
    shuffle(artist_music[art])

print(artist_music)

```

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.
- 10000 개 샘플 기준 3 초 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 Output 의 결과는 임의로 셔플한 결과이므로 반드시 Expected Output 과 같을 필요는 없습니다.
- 한글처리에 주의하세요.
- 시간제한은 240 분 입니다.

입력

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

```

문제 315. (오늘의 마지막 문제) k 사의 경력파트 알고리즘 문제를 해결하시오 !

```

# 변수생성
artist = ['Beatles', 'Beatles', '소녀시대', '소녀시대']
music = ['Yesterday', 'Let it be', '소원을 말해봐', 'Gee']

# 작업에 필요한 모듈로드와 diec 생성
from collections import defaultdict
from random import shuffle
artist_music = defaultdict(list)

```

```

#셔플 사전 생성(키값은 셔플 안됨) 사전이름 : artist_music
for i in artist:
    artist_music[i]
for art_ind, art in enumerate(artist):
    artist_music[art].append(music[art_ind])
    shuffle(artist_music[art])

# artist 변수 이용 키값(가수) 셔플 매개변수 생성 : shuffle_artist
shuffle_artist = list(set(artist))
shuffle(shuffle_artist)

# 출력할 플레이리스트 변수와 가수를 다시 섞을 때 쓸 더미변수 생성
playlist = []
dummy = ''

##만약 그 가수의 노래가 있다면 플레이리스트에 추가하고 다시 가수명을 섞는데
# 다음 플레이리스트에 동일 가수가 들어갈 수 없게 섞을 때 노래 넣은 가수 빼고섞는다
##만약 그 가수의 노래가 없다면 가수를 shuffle_artist 에서 지우고
#지운후에 만약 셔플 매개변수가 []이 된다면 더이상 넣을 노래가 없으므로 break
while True:
    if artist_music[shuffle_artist[0]] != []:
        playlist.append(artist_music[shuffle_artist[0]][0]) # 플레이리스트 추가
        del artist_music[shuffle_artist[0]][0] # 넣은 노래 제거(노래는 계속 줄어들것)
        dummy = shuffle_artist[0]
        del shuffle_artist[0]
        shuffle(shuffle_artist) # 플레이 리스트에 추가한 가수 빼고 다시 섞어
        shuffle_artist.append(dummy) # 플레이 리스트에 추가한 가수를 맨뒤로
    else:
        del shuffle_artist[0]
        if shuffle_artist == []:
            break

# 출력할 때 리스트 형식이 아닌 가로 스트링으로 뽑기위한 작업(문제 25 참고)
pr_playlist = ''
for i in playlist:
    pr_playlist += i + ' '
print(pr_playlist)

# 아마 내 코드가 제일 길듯;; ㅋㅋ 무튼 클리어!

```

문제 316. 아래의 두개의 리스트를 아래의 결과로 출력하시오 !

```

solar1 = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
solar2 = ['Sun', 'Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']
solar_union = []
for ind, name in enumerate(solar1):      #for i in solar1:
    solar_union.append(name)              #    solar_union.append(i)
    solar_union.append(solar2[ind])       #    solar_union.append(solar2[solar1.index(i)])
                                          #이렇게도 가능함
pr_solar = ''                            #for i in solar_union:
for i in solar_union:                    #print(i, end=' ')

```

```

pr_solar += i + ', '          #이렇게 할 수도 있음

print(pr_solar[:-2])
# 결과 : 태양, Sun, 수성, Mercury, 금성, Venus, 지구, Earth, 화성, Mars, 목성, Jupiter, 토성,
# Saturn, 천왕성, Uranus, 해왕성, Neptune

```

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

예제 :

```

for i, name in enumerate(['body', 'foo', 'bar']):
    print(i,name)

# 0 body
# 1 foo
# 2 bar
  ↑
index

```

문제 321. 아래의 딕셔너리에서 값만 추출하시오 !

```

dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']

for i in dic.values():
    for j in i:
        print(j, end = ', ')
    # 결과 : Yesterday let it be Hey Jude 소원을 말해봐 Gee 다만세

```

문제 322. 아래의 딕셔너리에서 아래의 값만 추출하시오 !

```

dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']

for i in dic.keys():
    print(dic[i][0], end = ' ')
    # 결과 : Yesterday 소원을 말해봐

```

문제 323. 아래의 딕셔너리에서 아래의 값들을 추출하시오 !

```

dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']

```

```

box2 = list(dic.values())
for i in range(len(box2[0])):
    for j in range(len(box2)):
        print(box2[j][i], end = ' ')
        # 결과 : Yesterday 소원을 말해봐 let it be Gee Hey Jude 다만세

```

문제 324. 문제 323 번 코드에 shuffle 기능을 추가해서 수행 될 때마다 곡이 섞여서 나오게 하시오 !

```

from random import shuffle
dic = {}
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세']
for i in dic:
    shuffle(dic[i])

box2 = list(dic.values())
shuffle(box2)
for i in range(len(box2[0])):
    for j in range(len(box2)):
        print(box2[j][i], end = ' ')
        # 결과 : Yesterday Gee let it be 다만세 Hey Jude 소원을 말해봐

```

문제 325. (점심시간 문제) 아래와 같이 소녀시대 곡을 하나 더 추가하고 잘 shuffle 되게 하시오 !

```

from random import shuffle
dic = {}
dic['소녀시대'] = ['소원을 말해봐', 'Gee', '다만세', 'Oh!']
dic['Beatles'] = ['Yesterday', 'let it be', 'Hey Jude']
for i in dic:
    shuffle(dic[i])

box2 = list(dic.values())
for i in range(len(box2[1])):
    for j in range(len(box2)):
        print(box2[j][0], end = ' ')
        del box2[j][0]

for i in box2:
    if i != []:
        print(i[0])
        del box2[box2.index(i)][0]

```

문제 326. 아래의 딕셔너리의 모든 요소를 다 출력하시오 !

```

dic = {'Beatles':['Yesterday','let it be','Hey Jude'], 'W
      '소녀시대':['소원을 말해봐','Gee','다만세'] }
dic.items()

# 결과 :
# dict_items([('Beatles', ['Yesterday', 'let it be', 'Hey Jude']), ('소녀시대', ['소원을 말해봐'
#                               , 'Gee', '다만세'])])

```

문제 327. 아래의 딕셔너리 데이터의 키값을 ABCD 순으로 정렬해서 출력하시오 !

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

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

문제 328. 아래의 딕셔너리 데이터의 키값을 ABCD 순으로 정렬하고 값도 출력하시오 !

```
names = {'Mary':10999, 'Sams':2111, 'Aimy':9778, 'Tom':20245,
'Michale':27115, 'Bob':5887, 'Kelly':7855}
name_items = list(names.items())
print(sorted(name_items))

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

문제 329. 알파벳 대문자 A ~ Z까지의 ascii 코드값을 출력하시오 !

```
import string
alpha = string.ascii_uppercase
for i in alpha:
    print(i, '-->', ord(i))

# 결과 :
# A --> 65
# B --> 66
# C --> 67
#      :
# Y --> 89
# Z --> 90
```

문제 330. eval 을 이용해서 구구단 2 단을 출력하시오 !

```
a = '2*i'
for i in range(1,10):
    print('2 * %d = %d' %(i, eval(a)))
```

```
다른방식 :
for i in range(1,10):
    a = '2*%d'%i
    print(a, eval(a))
# 2 * 1 = 2
# 2 * 2 = 4
#      :
#      :
# 2 * 9 = 18
```

문제 331. 아래의 리스트의 결과가 아래와 같이 수행될 수 있게 하시오 !
(lambda를 이용해서 구현하시오 !)

```
a = [1, 2, 3, 4, 5]
mul = lambda x : x*x
print([mul(i) for i in a])
```

문제 332. 위의 결과를 map을 이용하지 말고 출력하시오 !

```
a = [1,2,3,4,5]
multi = lambda x : x * x
for i in range(len(a)):
    a[i] = multi(a[i])
print( a )
```

문제 333. 몬테카를로 알고리즘을 이용해서 원주율 구하는 함수를 반복수행해서 점의 갯수가 많을 수록 원주율이 3.14에 가까워짐을 테스트 하시오 !

```
def find_cycle(total):
    import random
    cnt = 0
    for i in range(total):
        x = random.uniform(0, 1)
        y = random.uniform(0, 1)
        if x*x + y*y <= 1:
            cnt += 1
    return(cnt/total)*4

print(find_cycle(1000))
# 결과 : 3.036
```

문제 334. 몬테카를로 알고리즘을 이용해서 원주율 구하는 함수를 반복 수행해서 점의 갯수가 많을 수록 원주율이 3.14에 가까워짐을 테스트하시오 !

```
a = [1000, 10000, 100000, 1000000, 10000000]

def find_cycle(total):
    import random
    cnt = 0
    for i in range(total):
        x = random.uniform(0, 1)
        y = random.uniform(0, 1)
        if x*x + y*y <= 1:
            cnt += 1
    return(cnt/total)*4

fin = map(find_cycle, a)
print( list(fin))

# 결과 : [3.156, 3.178, 3.1394, 3.14404, 3.1423888]
```

문제 335. 위의 예제를 이용해서 기업명을 물어보게 하고 기업명을 입력하면 주식코드가 출력되게 하시오!

```
ask = input('기업명을 입력하세요 ! ')

f = open('c:WWWstockcode_part.txt','r')
data = f.readlines()
for i in data:
    if ask in i:
        print (i[:6])
        break
f.close()

# 기업명을 입력하세요 ! 대동공업
# 000490
```

문제 336. 카카오 경력파트 알고리즘 문제를 문제에서 제시한 url 알고리즘에 충실해서 문제를 푸시오 !

```
artist = ['beatles','beatles','beatles','beatles','소녀시대','소녀시대','소녀시대','버즈','버즈','버즈','버즈']
music = ['Yesterday','Let it be','something','please please me','Gee','소원을 말해봐','다만세','가시','겁쟁이','사랑하지 않은 것처럼','활주']
```

```
from collections import defaultdict
from random import shuffle, uniform
artist_music = defaultdict(list)
```

```
for i in artist:
    artist_music[i]
for art_ind, art in enumerate(artist):
    artist_music[art].append(music[art_ind])
    shuffle(artist_music[art])
play_list = []
```

```
for i in artist_music.keys():
    percent = []
    for j in range(len(artist_music[i])):
        if j == 0:
            song_loc = uniform(0,1/len(artist_music[i]))
            p = song_loc + uniform(-0.15,0.15)
            q = artist_music[i][j]
            r = i
            percent.append([p,q,r])
        else:
            p = song_loc + (1/len(artist_music[i]))*j
            q = artist_music[i][j]
            r = i
            percent.append([p,q,r])
    play_list += percent
play_list.sort()
```

```
for i in range(1,len(play_list)-1):
    if play_list[i-1][2] == play_list[i][2]:
        if i == len(play_list) - 1:
            dummy = play_list[i-1]
```



```

        play_list[i - 1] = play_list[i - 2]
        play_list[i - 2] = dummy
    else:
        dummy = play_list[i]
        play_list[i] = play_list[i + 1]
        play_list[i + 1] = dummy

for i in play_list:
    print(i[1], end = '  ')

```

문제 337. 위의 스크립트는 한줄 쓰고 엔터를 치면 바로 종료가 되는데 그렇게 하지 말고 한줄 쓰고 엔터를 쳐도

계속 물어보게 하고 계속 쓸 수 있게 하되 아무것도 안쓰고 그냥 엔터를 치면 종료되게 하시오

```

text_save = []
while True:
    text = input('파일에 저장할 내용을 입력하세요 ~ ')
    text_save.append(text + '\n')
    if text == '':
        break
f = open('c:\Wmydata.txt', 'w')
for i in text_save:
    f.write(i)
f.close()

- 더 간단한 답
f = open('c:\Wmydata.txt', 'w')
while True:
    text = input('파일에 저장할 내용을 입력하세요 ~ ')
    f.write(text + '\n')
    if text == '':
        break

f.close()

```

문제 338. 아래의 개사진 전부를 c 드라이브 밑에 image 라는 폴더를 만들고 거기에 복사하는 파이썬 코드를 작성하시오 !

```

import os
path_dir = 'c:\Wdog_images'
file_list = os.listdir(path_dir)
file_list.sort()
bufsize = 1024
for items in file_list:
    f = open('C:\Wdog_images\W' + items, 'rb')
    h = open('c:\Wimage\W' + items, 'wb')
    data = f.read(bufsize)
    while data:
        h.write(data)
        data = f.read(bufsize)
    f.close()
    h.close()

```

문제 339. 위의 예제를 파일을 닫는 f.close()를 명시하지 않고 자동으로 닫히게 하시오 !

```
with open('c:WWstockcode_part.txt', 'r') as f:
    data = f.readlines()
    print(data)
```

※ 설명 : with ~ as 절을 사용하게 되면 f.close()를 명시하지 않아도 되므로 프로그래머가 실수로 f.close()를 작성하지 않아서 코드 실행할 때 메모리 부족 오류가 나는 것을 예방할 수 있다.

문제 340. (점심시간문제) 위의 코드를 가지고 파일명을 물어보게하고 파일명을 입력하면 해당 파일이 삭제가 되는 코드를 작성하시오 !

```
def delete_file():
    target_file = input('삭제하고 싶은 파일을 입력하세요 ~ ')
    remove(target_file)
    print ('[%s]를 삭제했습니다.' %target_file)

delete_file()

# 삭제하고싶은 파일을 입력하세요 ~ c:WWmydata.txt
# [c:WWmydata.txt]를 삭제했습니다.
```

문제 341. 폐사진 폴더에 있는 이미지들을 불러오는 함수를 생성하시오 !

```
import os

test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    return file_list
print(image_load(test_image) )
```

문제 342. 위의 스크립트를 수정해서 png 떼어 버리고 아래와 같이 출력되게 하시오 !

```
import os
import re
test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    f_list = []
    for i in file_list:
        st = re.sub('[^0-9]', '', i)
        f_list.append(st)
    return f_list
print(image_load(test_image) )
# ['1', '10', '100', '1000', '1001', '1002', ..... ]
```

문제 343. 아래의 결과를 1, 2, 3, 4, 5 ... 순으로 정렬되게 하시오 !

```

import os
import re
test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    file_list.pop()
    f_list = []
    for i in file_list:
        st = re.sub('[^0-9]', '', i)
        f_list.append(int(st))
        f_list.sort()
    return f_list
print(image_load(test_image) )
# [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 .....]

```

문제 344. 위의 예제를 이용해서 아래와 같이 결과가 출력되게 하시오 !

```

import os
import re
test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    file_list.pop()
    f_list = []
    for i in file_list:
        st = re.sub('[^0-9]', '', i)
        f_list.append(int(st))
        f_list.sort()
    f_sort_list = []
    for i in f_list:
        f_sort_list.append(str(i) + '.png')
    return f_sort_list
print(image_load(test_image) )
# ['1.png', '2.png', '3.png', '4.png', '5.png', '6.png', '7.png', '8.png', .....]

```

문제 345. 이미지 앞에 절대 경로가 아래처럼 붙게 하시오 !

```

import os
import re
test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    file_list.pop()
    f_list = []
    for i in file_list:
        st = re.sub('[^0-9]', '', i)
        f_list.append(int(st))
        f_list.sort()
    f_sort_list = []
    if test_image[-1] == 'WW' :
        for i in f_list:

```

```

        f_sort_list.append(test_image + str(i) + '.png')
    else:
        for i in f_list:
            f_sort_list.append(test_image + 'WW' + str(i) + '.png')

    return f_sort_list
print(image_load(test_image) )
#['c:WWlungsWWimageWW1.png', 'c:WWlungsWWimageWW2.png', 'c:WWlungsWWimageWW3.png', .....]

```

문제 346. 위의 이미지들을 딥러닝 신경망에 넣기 전의 데이터인 숫자 배열로 만들기 위해서 cv2.imread 함수를 이용해서 list로 변환하시오 !

```

# 아나콘다 프롬프트 창 열고
# (base) C:\Users\WAdministrator>conda install opencv

import cv2
import os
import re
test_image = 'c:WWlungsWWimage'

def image_load(path):
    file_list = os.listdir(path)
    file_list.pop()
    f_list = []
    for i in file_list:
        st = re.sub('[^0-9]', '', i)
        f_list.append(int(st))
    f_list.sort()
    f_sort_list = []
    if test_image[-1] == 'WW' :
        for i in f_list:
            f_sort_list.append(test_image + str(i) + '.png')
    else:
        for i in f_list:
            f_sort_list.append(test_image + 'WW' + str(i) + '.png')
    image = []
    for j in f_sort_list:
        img = cv2.imread(j)
        image.append(img)
    return image

print(image_load(test_image) )

결과 :   빨강 녹색 파랑
[array([[ 16,  16,  16],
        [ 16,  16,  16],
        [ 16,  16,  16],
        ...,
        [182, 182, 182],
        [187, 187, 187],
        [191, 191, 191]],

```

문제 347. 위의 스크립트를 수정해서 폴더를 생성했으면 아래와 같이 결과가 출력되게 하시오 !

```

# 새로 생성할 디렉토리 이름을 입력하세요 : dddd

```

```
# C:\Users\Administrator\PycharmProjects 밑에 dddd 라는 폴더를 생성했습니다.
```

```
import os
pdir = os.getcwd()
newfolder = input('새로 생성할 디렉토리 이름을 입력하세요 : ')
os.mkdir(newfolder)
print('%s 밑에 %s 라는 폴더를 생성했습니다.' %(pdir, newfolder))
```

문제 346. 다시 data7 폴더를 생성하는데 이번에는 폴더 생성후에 파일을 몇개 넣은 후에 파이썬 코드로 삭제해 보시오 !

```
import os
target_folder = 'c:\\data7'
os.rmdir(target_folder)
```

```
# OSError: [WinError 145] 디렉터리가 비어 있지 않습니다: 'c:\\data7'
```

문제 349. 아래와 같이 폴더를 생성하는 스크립트를 작성하는데 폴더가 이미 존재하면 "이미 존재합니다." 라는 메시지가 출력되게 하시오 !

```
from os.path import exists
import os
newfolder = input('새로 생성할 디렉토리 이름을 입력하세요 : ')
if exists(newfolder) == True:
    os.mkdir(newfolder)
else:
    print('폴더가 이미 존재합니다.')
```

```
# 새로 생성할 디렉토리 이름을 입력하세요 : d:\\data7
# 폴더가 이미 존재합니다.
```

문제 350. (오늘의 마지막 문제) 오라클 next_day 함수를 파이썬으로 구현하시오 !

```
def next_day(date, week):
    week_dic = {}
    week_dic['월요일'] = 0
    week_dic['화요일'] = 1
    week_dic['수요일'] = 2
    week_dic['목요일'] = 3
    week_dic['금요일'] = 4
    week_dic['토요일'] = 5
    week_dic['일요일'] = 6
    import datetime as d
    from datetime import datetime as da
    date1 = d.datetime.strptime(date, '%Y-%m-%d')
    date_week = da.weekday(date1)
    result = date1 + d.timedelta(days = (7 - date_week) + week_dic[week])
    return result

print(next_day('2018-12-10', '수요일'))
print(next_day('2018-12-10', '월요일'))

# 2018-12-19 00:00:00
```

2018-12-17 00:00:00

문제 351. 아래의 간단한 html 문서를 만들어 보시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body>
<p class="title"> 재혁이형은 오늘 결석을 하셨습니다. </p>
</body>
</html>
```

문제 352. 문제 351 번에 나온 글씨를 진하게 하시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body>
<p class="title"><b> 재혁이형은 오늘 결석을 하셨습니다. </b></p>
</body>
</html>
```

문제 353. 위의 글씨에 밑줄을 그어보시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body>
<p class="title"><b><u> 재혁이형은 오늘 결석을 하셨습니다. </u></b></p>
</body>
</html>
```

문제 354. 위의 글씨체를 이탤릭체(기울임)로 변경하시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body>
<p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>
</body>
</html>
```

문제 355. 위의 글씨의 색깔을 파란색으로 변경하시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body text = "blue">
<p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>
</body>
</html>
```

문제 356. p tag 를 추가해서 제목과 내용을 나누시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body text = "blue">
<p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>
<p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다.
```

커트는 2만원, 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다.
그래서 커트로 머리를 올렸다. </p>

</body>
</html>

문제 357. 위의 html 문서에 링크를 거시오 !

```
<html><head><title> 재혁이형의 오늘 일정 </title></head>
<body text = "blue">
<p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>
<p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다.
    커트는 2만원, 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다.
    그래서 커트로 머리를 올렸다.
<a href="http://cafe.daum.net/oracleoracle" class="cafe1" id="link1">다음카페</a></p>
</body>
</html>
```

문제 358. beautiful soup 모듈을 이용해서 test_page.html 문서의 title을 검색하시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")    # html 문서를 컴퓨터가 알아볼수 있게 기계어로
                                                # 바꾸고 로드
print(soup.title)

# <title> 재혁이형의 오늘 일정 </title>
```

문제 359. 위의 결과에서 html 코드는 나오지 않고 텍스트만 출력되게 하시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")

print(soup.title.string)                # 이부분 수정 !

# 재혁이형의 오늘 일정
```

문제 360. test_page.html 문서에서 p 태그에 대한 html을 검색하시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")
```



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

```
# <p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>
# 설명 : p tag 가 두개인데 하나만 나왔네?
```

문제 361. test_page.html 문서에서 p 태그에 대한 html 을 모두 검색하시오 !

```
from bs4 import BeautifulSoup
```

```
with open("c:WWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")
```

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

```
# [<p class="title"><b><u><i> 재혁이형은 오늘 결석을 하셨습니다. </i></u></b></p>,
# <p class="content"> 오늘 나는 강남의 더킹 미용실에 갔다.
#           커트는 2만원, 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다.
#           그래서 커트로 머리를 올렸다.
# <a class="cafe1" href="http://cafe.daum.net/oracleoracle" id="link1">다음카페</a></p>]
```

문제 362. test_page.html 문서에서 a 태그에 대한 html 을 모두 검색하시오 !

```
from bs4 import BeautifulSoup
```

```
with open("c:WWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")
```

```
print(soup.find_all('a'))
```

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

문제 363. test_page.html 문서에서 a 태그에 href 링크의 url 만 읽어오시오 !

```
from bs4 import BeautifulSoup
```

```
with open("c:WWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")
```

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

문제 364. test_page.html 문서에서 html 소스 말고 text 만 출력하시오 !

```
from bs4 import BeautifulSoup
```

```
with open("c:WWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")
```

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

```
# 재혁이형의 오늘 일정
#
# 재혁이형은 오늘 결석을 하셨습니다.
# 오늘 나는 강남의 더킹 미용실에 갔다.
#           커트는 2만원, 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다.
#           그래서 커트로 머리를 올렸다.
# 다음카페
```

문제 365. 위의 텍스트를 한줄로 나오게 하시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWtest_page.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")

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

# 재혁이형의 오늘 일정재혁이형은 오늘 결석을 하셨습니다.오늘 나는 강남의 더킹 미용실에 갔다.
#           커트는 2만원, 파마는 10만원, 파마를 하고 싶었지만 돈이 없었다.
#           그래서 커트로 머리를 올렸다.다음카페

# 설명 : 엔터를 직접 친 부분은 한줄로 나오지 않는다.
```

문제 366. ecologicalpyramid.html 문서에서 text 만 출력하시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWecologicalpyramid.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")

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

# plants100000algae100000deer 1000rabbit2000fox100bear 100lion80tiger50
```

문제 367. ecologicalpyramid.html 문서에서 number 클래스에 있는 모든 텍스트를 가져오시오 !

```
from bs4 import BeautifulSoup

with open("c:WWWecologicalpyramid.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")

print(soup.find_all(class_="number"))

# [<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>]

- 소스빼고 텍스트만 출력하고 싶다면 ?
from bs4 import BeautifulSoup
```

```

with open("c:WWecologicalpyramid.html", encoding = 'UTF-8') as a:
    soup = BeautifulSoup( a, "html.parser")

result = soup.find_all(class_="number")

for i in result:
    print(i.get_text())

# 100000
# 100000
# 1000
# 2000
# 100
# 100
# 80
# 50

```

문제 368. 위의 문제에서 1000 만 출력하시오 !

```

(1)
    from bs4 import BeautifulSoup

    with open("c:WWecologicalpyramid.html", encoding = 'UTF-8') as a:
        soup = BeautifulSoup( a, "html.parser")

    result = soup.find_all(class_="number")
    for i in result:
        if i.get_text() == '1000':
            print(i.get_text())

(2)
    from bs4 import BeautifulSoup

    with open("c:WWecologicalpyramid.html", encoding = 'UTF-8') as a:
        soup = BeautifulSoup( a, "html.parser")

    result = soup.find_all(class_="number")

    print(result[2].get_text())

```

문제 369. ebs 의 레이디 버그 시청자 게시판의 html 문서 전체를 스크롤링 하는 함수를 작성하라.

```

import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuId=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    return result

print (ebs_scroll() )

```

※ 설명 : urllib.request.urlopen(url).read().decode("utf-8") == ebs 의 html 문서를 utf-8 로 decoding 하겠다.

문제 370. (점심시간 문제) 위의 html 문서를 BeautifulSoup 모듈로 파싱을 하고 이 html 문서중에서 텍스트만 가져와서 출력하시오 !

```
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    return result

soup = BeautifulSoup( ebs_scroll() , "html.parser")

print(soup.get_text())
```

문제 371. ebs html 문서중에서 p 태그에 해당하는 부분을 모두 가져오시오 !

```
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&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_scroll())
```

문제 372. 위의 p 태그중에 class 가 con 에 해당하는 부분만 스크롤링하시오 !

```
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&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_scroll())
```

문제 373. 위의 결과에서 html 말고 텍스트만 가져오시오 !

```
import urllib.request
from bs4 import BeautifulSoup
```

```

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    result2 = soup.find_all("p", class_="con")
    for i in result2:
        print(i.get_text())

ebs_scroll()

```

문제 374. 위의 텍스트를 좀 더 깔끔하고 예쁘게 출력되게 하시오 !

```

import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    result2 = soup.find_all("p", class_="con")
    for i in result2:
        print(i.get_text().strip())

ebs_scroll()

```

문제 375. 위의 게시판의 글들을 하나의 리스트에 하나 하나의 요소로 담으시오 !

```

import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuld=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    result2 = soup.find_all("p", class_="con")
    scroll_list = []
    for i in result2:
        scroll_list.append(i.get_text().strip())
    return scroll_list

ebs_scroll()

```

문제 376. 위의 결과에서 Wr 과 Wn 을 삭제하고 params 리스트에 담기게 하시오 !

```

import re
import urllib.request

```

```

from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuId=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    result2 = soup.find_all("p", class_="con")

    scroll_list = []
    for i in result2:
        scroll_list.append(i.get_text().strip())

    scroll_list2 = []
    for i2 in scroll_list:
        scroll_list2.append(re.sub('[\n\r]', '', i2))

    return scroll_list2

ebs_scroll()

```

문제 377. 위의 게시판에 글 앞에 게시날짜를 앞에 출력되게 하기 위해 게시판의 글올린 날짜만 가져오는 코드를

작성하시오 !

```

import re
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuId=106"
    url = urllib.request.Request(list_url)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    result2 = soup.find_all("span", class_="date")

    scroll_list = []
    for i in result2:
        scroll_list.append(i.get_text().strip())

    return scroll_list

ebs_scroll()

```

문제 378. 위의 게시판에 글 앞에 게시날짜를 앞에 출력되게 하시오!

```

import re
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?
c.page=1&hmpMnuId=106"

```

```

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result , "html.parser")

scroll_list = []
for x, y in zip(soup.find_all("span", class_="date"), soup.find_all("p", class_="con")):
    scroll_list.append(x.get_text().strip() + y.get_text().strip())

scroll_list2 = []
for i2 in scroll_list:
    scroll_list2.append(re.sub('[\n\r]', ' ', i2))

return scroll_list2

print(ebs_scroll())

```

문제 379. 레이디버그 게시판 전체의 글들을 스크롤링 하시오 !

```

import re
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    for i in range(1, 17):
        list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=
%d&hmpMnuId=106"%i
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")

        scroll_list1 = []
        scroll_list = []
        for x, y in zip(soup.find_all("span", class_="date"), soup.find_all("p", class_="con")):
            scroll_list.append(x.get_text().strip())
            scroll_list1.append(re.sub('[\n\r]', ' ', y.get_text().strip()))

        for i1, i2 in zip(scroll_list, scroll_list1):
            print(i1, end = ' ')
            print(i2)

ebs_scroll()

```

문제 380. 변수의 내용과 리스트의 내용을 파일로 저장되게 하는 방법은 ?

- 스트링 저장방법


```

text = 'asdnjfnasdiuvqaeprktmngkewptmng0adifjasdkvm'
f = open("c:\\\\mydata.txt", 'w', encoding = 'UTF-8')
f.write(text)
f.close()

```
- 리스트 저장방법


```

text = ['aaa', 'bbb', 'ccc', 'ddd']
f = open('c:\\\\mydata2.txt', 'w', encoding = 'UTF-8')
for item in text:
    f.write('%s\n'%item)
f.close()

```

문제 381. 레이디버그 게시판의 날짜와 글을 mydata7.txt 에 저장하시오 !

```
import re
import urllib.request
from bs4 import BeautifulSoup

def ebs_scroll():
    scroll_list = []
    for i in range(1, 17):
        list_url = "http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=
%d&hmpMnuId=106"%i
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup(result, "html.parser")
        f = open('c:WWWmydata7.txt', 'w', encoding="UTF-8")

        for x, y in zip(soup.find_all("span", class_="date"), soup.find_all("p", class_="con")):
            scroll_list.append(x.get_text().strip() + y.get_text().strip())

    scroll_list2 = []
    for i2 in scroll_list:
        scroll_list2.append(re.sub('[\n\r]', '', i2))

    for item in scroll_list2:
        f.write('%s\n' %item)
    f.close()
    print('저장되었습니다.')

ebs_scroll()
```

문제 382. Text_mining 클래스의 워드클라우드 그리는 메소드를 실행해서 mydata7.txt 를 워드클라우드에 그리시오!

```
import Text_mining
tm = Text_mining.Text_mining()
tm.word_cloud()
```

문제 383. 한겨레 신문사 홈페이지로 들어가서 인공지능으로 기사검색을 한 url 을 가져오시오 !

<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=2018.12.11&pageseq=0>

<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=2018.12.11&pageseq=1>

문제 384. 한겨레 신문사에서 인공지능으로 기사검색을 했을 때 나오는 html 문서에서 상세기사 제목을 클릭했을 때

잡히는 url 만 검색하시오 !

 이마트 의왕점 13일 개점... 종이↓ 디지털↑

상세기사 url

```
import re
import urllib.request
from bs4 import BeautifulSoup

def hani_scroll():
    scroll_list = []
    for i in range(0, 10):
        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=2018.12.11&pageseq="+ str(i)
        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('dt'):
            for i2 in j:
                print(i2.get('href'))

hani_scroll()
```

문제 385. 위의 상세기사 url 들을 params 라는 빈 리스트에 담으시오 !

```
import re
import urllib.request
from bs4 import BeautifulSoup

def hani_scroll():
    params = []
    for i in range(0, 10):
        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=2018.12.11&pageseq="+ str(i)
        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('dt'):
            for i2 in j:
                params.append(i2.get('href'))
    return params

print(hani_scroll())
```

문제 386. 아래의 상세 url 의 기사 내용의 텍스트를 스크롤링 하시오 !

#http://www.hani.co.kr/arti/economy/consumer/873902.html

```
def hani_scroll2():
    list_url = "http://www.hani.co.kr/arti/economy/consumer/873902.html"
    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('div', 'text'): # 기사 본문 내용이 있는 태그와 클래스 명시
    print(i.get_text())

```

hani_scroll2()

문제 387. hani_scroll() 함수와 hani_scroll2() 함수를 가지고 인공지능으로 검색한 전체 기사를 다 스크롤링하는 함수를 완성시키시오 !

```

import re
import urllib.request
from bs4 import BeautifulSoup

def hani_scroll():
    params = []
    for i in range(0, 10):
        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=2018.12.11&pageseq="+ str(i)
        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('dt'):
            for i2 in j:
                params.append(i2.get('href'))
    return params

def hani_scroll2():
    list_url = hani_scroll()
    for i in list_url:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")
        for i in soup.find_all('div', 'text'): # 기사 본문 내용이 있는 태그와 클래스 명시
            print(i.get_text())

```

hani_scroll2()

문제 388. 위의 함수를 가지고 인공지능 기사로 검색한 기사들이 전부 mytext8 이라는 텍스트에 생성되게 하시오 !

(오늘의 마지막 문제)

워드 클라우드 그림을 반 카툰에 올리고 검사받으세요~

```

import re
import urllib.request
from bs4 import BeautifulSoup

def hani_scroll():
    params = []
    f = open('c:WWWmytext8.txt', 'w', encoding = 'UTF-8')
    for i in range(0, 10):
        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=2018.12.11&pageseq="+ str(i)
        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('dt'):
    for i2 in j:
        params.append(i2.get('href'))
params2 = []
for i in params:
    url = urllib.request.Request(i)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    for i in soup.find_all('div', 'text'): # 기사 본문 내용이 있는 태그와 클래스 명시
        params2.append(i.get_text())
for item in params2:
    f.write('%s\n' %item)
print('저장되었습니다.')

```

hani_scroll()

```

import Text_mining
tm = Text_mining.Text_mining()
tm.word_cloud()

```

문제 389. 중앙일보 신문사 사이트에서 인공지능으로 검색한 url 을 알아내시오 !

<https://search.join.com/TotalNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews> #

<https://search.join.com/TotalNews?page=2&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews> #

-
- 어제 만들었던 한겨레 사이트 웹스크롤 함수 2 개 가져온다
 1. hani_scroll() : 상세기사 url 가져와서 params 리스트에 담는 함수
 2. hani_scroll() : 상세기사 url 로 상세기사 본문을 스크롤링하는 함수

```

import re
import urllib.request
from bs4 import BeautifulSoup

def chuan_scroll():
    params = []
    f = open('c:WWWmytext9.txt', 'w', encoding = 'UTF-8')
    for i in range(1, 5):
        list_url = "https://search.join.com/TotalNews?page="+ str(i) + "&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews"
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")
        for j in soup.find_all('strong', class_='headline mg'):
            for i2 in j:
                params.append(i2.get('href'))
    params2 = []
    for i in params:

```

```

url = urllib.request.Request(i)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup( result , "html.parser")
for i in soup.find_all('div', id="article_body"):      #찾는 태그로 id를 이용
    params2.append(i.get_text())
for item in params2:
    f.write('%s\n' %item)
print('저장되었습니다.')

```

chuan_scroll()

※ 설명 : html 코드에서 id란?

잘 설계된 html 코드 문서라면 id의 값은 html 문서 내에 단 한개만 존재한다.

문제 390. 위에서 끌어온 데이터를 mydata9.txt로 생성하시오 !

```

import re
import urllib.request
from bs4 import BeautifulSoup

def chuan_scroll():
    params = []
    f = open('c:WWWmytext9.txt', 'w', encoding = 'UTF-8')
    for i in range(1, 5):
        list_url = "https://search.joins.com/TotalNews?page="+ str(i) + "&Keyword=%EC%9D%B8%EA
%B3%B5 W
                %EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=TotalNews"
        url = urllib.request.Request(list_url)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")
        for j in soup.find_all('strong', class_='headline mg'):
            for i2 in j:
                params.append(i2.get('href'))
    params2 = []
    for i in params:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")
        for i in soup.find_all('div', id="article_body"):
            params2.append(i.get_text())
    for item in params2:
        f.write('%s\n' %item)
    print('저장되었습니다.')

```

chuan_scroll()

문제 391. 동아일보 신문사에서 딥러닝으로 검색한 기사를 mytext10.txt에 담은 함수 2개를 생성하시오 !

```

import urllib.request
from bs4 import BeautifulSoup

def donga_scroll():

```

```

params = []
f = open('c:WWmytext10.txt', 'w', encoding = 'UTF-8')
for i in range(1, 100, 15):
    list_url = "http://news.donga.com/search?p="+ str(i) + "&query=%EB%94%A5%EB%9F%AC%EB%8B
%9D&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")
    for i in soup.find_all('p', class_='tit'):
        result = i.find('a').get('href')
        params.append(result)
params2 = []
for i in params:
    url = urllib.request.Request(i)
    result = urllib.request.urlopen(url).read().decode("utf-8")
    soup = BeautifulSoup( result , "html.parser")
    for i in soup.find_all('div', class_="article_txt"):
        for i2 in i(["script"]):
            i2.decompose()
            params2.append(i.get_text())
for item in params2:
    f.write('%s\n' %item)
print('저장되었습니다.')

donga_scroll()

```

문제 392. 위의 문제를 워드클라우드로 출력하시오 !

```

import urllib.request
from bs4 import BeautifulSoup

def donga_scroll():
    params = []
    f = open('c:WWmytext10.txt', 'w', encoding = 'UTF-8')
    for i in range(1, 100, 15):
        list_url = "http://news.donga.com/search?p="+ str(i) + "&query=%EB%94%A5%EB%9F%AC%EB%8B
%9D&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")
        for i in soup.find_all('p', class_='tit'):
            result = i.find('a').get('href')
            params.append(result)
    params2 = []
    for i in params:
        url = urllib.request.Request(i)
        result = urllib.request.urlopen(url).read().decode("utf-8")
        soup = BeautifulSoup( result , "html.parser")
        for i in soup.find_all('div', class_="article_txt"):
            for i2 in i(["script"]):
                i2.decompose()
                params2.append(i.get_text())
    for item in params2:
        f.write('%s\n' %item)
    print('저장되었습니다.')

donga_scroll()

```

```
import Text_mining
tm = Text_mining.Text_mining()
tm.word_cloud()
```

문제 393. (이미지 스크롤링) 네이버에서 아이폰맨으로 이미지 검색을 했을 때 나오는 이미지들을 모두 스크롤링하시오 !

* 이미지 스크롤링 하려면 필요한 내용 2 가지

1. selenium 설치 # 사람이 손으로 스크롤링하는것을 파이썬으로 구현하는 것
* 아나콘다 프롬프트 창 열고
(base) C:\Users\WAdministrator>conda install selenium

2. 크롬 드라이버 (크롬으로 스크롤링)
c:\WWWchromedriver <---- 폴더 생성후에 여기에 chromedriver.exe 를 넣으시오

3. c 드라이브 밑에 naver_images 라는 폴더를 생성하시오 !

```
import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time #중간중간에 sleep 을 넣기 위해
```

```
binary = 'c:\Wchromedriver\chromedriver.exe' #크롬 여는 장치
browser = webdriver.Chrome(binary)
browser.get("https://search.naver.com/search.naver?where=image&sm=stb_nmr&") #이미지
```

검색 url

```
elem = browser.find_element_by_id("nx_query") # 검색창의 id
#find_elements_by_class_name("")
```

```
# 검색어 입력
elem.send_keys("조로")
elem.submit()
```

```
# 반복할 횟수
```

```
for i in range(1,20):
    browser.find_element_by_xpath("//body").send_keys(Keys.END) #바디를 누르고 END 키를 눌러 아
```

래로 내려감

```
time.sleep(5) #5 초 슬립 이유 : 인터넷 환경에 따라 빨리 이미지 안나옴
```

```
time.sleep(5) #마지막 스크롤바 내리고 5 초 쉬고
html = browser.page_source
soup = BeautifulSoup(html,"html.parser")
print(soup)
print(len(soup))
```

```
def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="_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:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/naver_images/"+ str(a) + ".jpg" )
        a += 1

fetch_detail_url()

browser.quit()

```

문제 394. (이미지 스크롤링) 구글에서 이미지 검색하는 웹 스크롤링 코드를 작성하시오 !

```

import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

binary = 'c:\chromedriver\chromedriver.exe'
browser = webdriver.Chrome(binary)
browser.get("https://www.google.co.kr/imghp?hl=ko")
elem = browser.find_element_by_class_name("gLfyf")
#find_elements_by_class_name("")

# 검색어 입력
elem.send_keys("조로")
elem.submit()

# 반복할 횟수
for i in range(1,10):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    try:
        browser.find_element_by_id("smb").click()
        time.sleep(5)
    except:
        time.sleep(5)

time.sleep(10)
html = browser.page_source
soup = BeautifulSoup(html,"html.parser")
#print(soup)
#print(len(soup))

def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="rg_ic rg_i")
    for im in imgList:
        try:
            params.append(im["src"])
        except KeyError:
            params.append(im["data-src"])
    return params

def fetch_detail_url():

```

```

    params = fetch_list_url()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/naver_images/" + str(a) + ".jpg" )
        a += 1
    fetch_detail_url()

browser.quit()

```

문제 395. (이미지 스크롤링) 다음 이미지 검색에서 햄버거로 검색했을 때 나오는 이미지들을 c드라이브 밑에 daumImages 라는 폴더에 저장되게 하시오 !

```

import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

binary = 'c:\\chromedriver\\chromedriver.exe'
browser = webdriver.Chrome(binary)
browser.get("https://search.daum.net/search?w=img&nil_search=btn&DA=NTB&enc=utf8&q=")
elem = browser.find_element_by_id("q")
#find_elements_by_class_name("")

# 검색어 입력
elem.send_keys("조로")
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,"html.parser")
#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:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/daum_images/" + str(a) + ".jpg" )
        a += 1

```



```
fetch_detail_url()
```

```
browser.quit()
```

문제 396. (이미지 스크롤링) Bing 에서 이미지 검색을 조회했을 때 나오는 이미지들을 c 드라이브 밑에 bing 이 라는

폴더에 저장하시오 !

```
import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

binary = 'c:\\chromedriver\\chromedriver.exe'
browser = webdriver.Chrome(binary)
browser.get("https://www.bing.com/?scope=images&FORM=Z9LH1")
elem = browser.find_element_by_id("sb_form_q")
#find_elements_by_class_name("")

# 검색어 입력
elem.send_keys("조로")
elem.submit()

# 반복할 횟수
for i in range(1,10):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5)

time.sleep(5)
html = browser.page_source
soup = BeautifulSoup(html,"html.parser")
#print(soup)
#print(len(soup))

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()
    #print(params)
    a = 1
    for p in params:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/Bing_Images/"+ str(a) + ".jpg" )
        a += 1
    fetch_detail_url()

browser.quit()
```

문제 397. (오늘의 마지막 문제) instagram 에서 이미지를 다운로드 받는 웹 스크롤링 코드를 작성하시오 !

```

import urllib.request
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

binary = 'c:\Wchromedriver\chromedriver.exe'
browser = webdriver.Chrome(binary)
browser.get("https://www.instagram.com/instagram/")
browser.find_element_by_class_name("TqC_a").click()
browser.find_element_by_css_selector("input.XTCLo.x3qfX").clear()
browser.find_element_by_css_selector("input.XTCLo.x3qfX").send_keys('#에일리')
time.sleep(2)
browser.find_element_by_css_selector("input.XTCLo.x3qfX").send_keys(Keys.ENTER)
browser.find_element_by_css_selector("input.XTCLo.x3qfX").send_keys(Keys.ENTER)
time.sleep(4)
# 반복할 횟수
for i in range(1,4):
    browser.find_element_by_xpath("//body").send_keys(Keys.END)
    time.sleep(5)

time.sleep(5)
html = browser.page_source
soup = BeautifulSoup(html, "html.parser")
#print(soup)
#print(len(soup))

def fetch_list_url():
    params = []
    imgList = soup.find_all("img", class_="FFVAD")
    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:
        # 다운받을 폴더경로 입력
        urllib.request.urlretrieve(p, "c:/instagram_Images/" + str(a) + ".jpg" )
        a += 1
    fetch_detail_url()

browser.quit()

```

※ 참고

html 개발 지원 창 콘솔창 활용하기

- document.querySelectorAll(" ")
- 상위 엘리먼트는 띄어쓰기
- 클래스는 .
 - 엘리먼트.클래스이름
- 아이디는 #
 - 엘리먼트#아이디이름

* 콘솔 언어 파이썬에서 활용

- find_element_by_css_selector(" ")

문제 397. 서울시 응답소 게시판의 글을 웹스크롤링 하기 위해 페이지마다 html 코드를 가져올 수 있도록 하시오 !

```
<a href="javascript:commonPagingPost('2','10','Shr01_lis.jsp');" title="2 페이지">2</a>
```

commonPagingPost <=----- 자바 스크립트 함수

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    params = []
    for j in range(1,30):
        list_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header = urllib.parse.urlencode({"page": j})
        print (request_header)

fetch_list_url()

#javascript:commonPagingPost('2','10','Shr01_lis.jsp');
#
# function commonPagingPost (Page, Block, Action) {
#     var Frm = document.mainForm;
#     Frm.RCEPT_NO.value = "";
#     Frm.page.value = Page;
#     Frm.action = Action;
#     Frm.submit ();
# }
# -->
# </script>
```

문제 398. 아래의 서울시 응답소 게시판 url 과 페이지번호를 commonPagingPost 자바스크립트 함수에 제공을 해야

해당 페이지의 html 코드를 내려받을 수 있다.

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    for j in range(1,30):
        list_url = "http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header = urllib.parse.urlencode({"page": j})
        # print(request_header) page = 1, page=2 ... 가 출력됨

        request_header = request_header.encode("utf-8")          # 컴퓨터가 알아볼 수 있는 언어로

        # page=1 을 컴퓨터가 이해할수 있는 이진 코드로 변환
        # print (request_header) # b'page=1' 출력

        url = urllib.request.Request( list_url, request_header )    # 페이지번호 같이 제공
```

인코딩

```
res = urllib.request.urlopen(url).read().decode("utf-8") # res 에 html 코드가 담김
print (res)
```

```
fetch_list_url()
```

문제 399. 문재인 대통령 기사를 이름은 최재혁으로 사진도 최재혁으로 변경하시오 !

```
http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp
```

```
page=1, page=2, page=3 .....
```

↓

자바 스크립트 함수 : commonPagingPost ----> html 코드를 보내준다

↓ 게시글을 클릭했을 때 바뀌는 url

```
http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_vie.jsp
```

자바 스크립트 함수 : JavaScript:onView

문제 400. 게시글의 상세 게시판글을 스크롤링 하기 위해서 필요한 상세 url 에 해당하는 코드(javascript 함수 에 제공해야 하는 코드)를 스크롤링하시오 !

```
<a href="JavaScript:onView ('20181126905228')" title="도시기반시설본부 면허과다 제한 및 공동도급
불허 개선 "> 도시기반시설본부 면허과다 제한 및 공...</a>
```

```
JavaScript:onView ('20180801905041')
```

↑

이 부분을 스크롤링 해야한다.

```
import urllib.request
from bs4 import BeautifulSoup
import os
import re
```

```
def fetch_list_url():
    for j in range(1,10):
        list_url ="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header = urllib.parse.urlencode({"page": j})
        # print(request_header) page = 1, page=2 ... 가 출력됨

        request_header = request_header.encode("utf-8")
        # page=1 을 컴퓨터가 이해할수 있는 이진 코드로 변환
        # print (request_header) # b'page=1' 출력

        url = urllib.request.Request( list_url, request_header )
        res = urllib.request.urlopen(url).read().decode("utf-8")
```

```

        #print (res)

        soup=BeautifulSoup(res, "html.parser")
        soup2 = soup.find_all('li', class_="plist_list_tit42")
        for i in soup2:
            print(i.find('a')["href"])

    fetch_list_url()

```

문제 401. 20181129905727 <----- 이부분만 추출하시오 !

```

import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    for j in range(1,10):
        list_url ="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header = urllib.parse.urlencode({"page": j})
        # print(request_header) page = 1, page=2 ... 가 출력됨

        request_header = request_header.encode("utf-8")
        # page=1 을 컴퓨터가 이해할수 있는 이진 코드로 변환
        # print (request_header) # b'page=1' 출력

        url = urllib.request.Request( list_url, request_header )
        res = urllib.request.urlopen(url).read().decode("utf-8")
        #print (res)

        soup=BeautifulSoup(res, "html.parser")
        soup2 = soup.find_all('li', class_="plist_list_tit42")
        for i in soup2:
            soup3 = re.sub('[^0-9]', '', i.find('a')["href"])
            # soup3 = re.search('[0-9]{14}', i.find('a')["href"]) 로 해도 됨
            print(soup3)

    fetch_list_url()

```

문제 402. params 라는 비어있는 리스트변수에 위의 data 들을 append 시키시오 !

```

import urllib.request
from bs4 import BeautifulSoup
import os
import re

def fetch_list_url():
    params = []
    for j in range(1,10):
        list_url ="http://eungdapso.seoul.go.kr/Shr/Shr01/Shr01_lis.jsp"
        request_header = urllib.parse.urlencode({"page": j})
        # print(request_header) page = 1, page=2 ... 가 출력됨

        request_header = request_header.encode("utf-8")

```

```

# page=1 을 컴퓨터가 이해할수 있는 이진 코드로 변환
# print (request_header) # b'page=1' 출력

url = urllib.request.Request( list_url, request_header )
res = urllib.request.urlopen(url).read().decode("utf-8")
#print (res)

soup=BeautifulSoup(res, "html.parser")
soup2 = soup.find_all('li', class_="plist_list_tit42")

for i in soup2:
    soup3 = re.sub('[^0-9]', '', i.find('a')['href'])
    params.append(soup3)
print(params)

fetch_list_url()

```

문제 403. JavaScript:onView 함수에 위의 번호들이 어떻게 제공이 되어야 하는지 확인하시오 !

```
JavaScript:onView('20181129905727', '상세 게시판 글 url')
```

↓

상세 게시판 글이 있는 html 문서를 리턴해준다

```

-----
function onView (Seq) {
    var Frm = document.mainForm;
    Frm.RCEPT_NO.value = Seq;                # '20181129905727'
    Frm.action = "/Shr/Shr01/Shr01_vie.jsp";    # '상세 게시판 글 url'
    Frm.submit ();
}

# RCEPT_NO : 20181129905727
-----

```

문제 404. RCEPT_NO : 20180820907144 형태로 onView 함수에 /Shr/Shr01/Shr01_vie.jsp 와 같이 제공을 해서 상세 게시판

글의 html 코드를 가져오는 fetch_list_url2()함수를 생성하시오 !

```

import urllib.request
from bs4 import BeautifulSoup
import os
import re

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) page = 1, page=2 ... 가 출력됨

        request_header = request_header.encode("utf-8")
        # page=1 을 컴퓨터가 이해할수 있는 이진 코드로 변환
        # print (request_header) # b'page=1' 출력

```

```

url = urllib.request.Request( list_url, request_header )
res = urllib.request.urlopen(url).read().decode("utf-8")
#print (res)

soup=BeautifulSoup(res, "html.parser")
soup2 = soup.find_all('li', class_="plist_list_tit42")

for i in soup2:
    soup3 = re.sub('[^0-9]', '', i.find('a')['href'])
    params.append(soup3)
return params

def fetch_list_url2():
    params2 = fetch_list_url()

    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")
        print (soup)

fetch_list_url2()

```

문제 405. (점심시간 문제) 위의 코드를 이용해서 상세 게시판을 글들을 스크롤링 하는 코드를 완성시키시오 !

```

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 : " )
    save_path = save_path.replace("WW", "/")
    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, 30):
        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("c:WWresult.txt" , '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 + "Wn")
        f.write(title + "Wn")
        f.write(date + "Wn")
        f.write(question + "Wn")
        f.write(answer + "Wn")
        f.write("==" * 30 + "Wn")

    f.close()

fetch_list_url2()

```

문제 406. 위의 결과를 워드클라우드에 출력하시오 !

```

import Text_mining
tm = Text_mining.Text_mining()
tm.word_cloud()

```

```
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 크기 및 색깔
        self.canvas.move(self.id, 245, 100) #공을 캔버스 중앙으로 이동
        starts = [-3, -2, -1, 1, 2, 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 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.(공이 화면에서 사
라지지 않기위해)
        self.canvas_width = self.canvas.winfo_width() #캔버스의 현재 넓이를 반환한다.(공이 화면에서 사라
지지 않기위해)
        self.hit_bottom = False

    def hit_paddle(self,pos): # 패들에 공이 튀기게 하는 함수
        paddle_pos = self.canvas.coords(self.paddle.id)
        if pos[2] >= paddle_pos[0] and pos[0] <= paddle_pos[2]: # 공이 패들이 좌우로 겹친다.
            if pos[3] >= paddle_pos[1] and pos[1] <= paddle_pos[3]: # 공이 패들이 위아래로 겹친다.
                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) # 볼의 현재 좌표를 출력해준다. 공 좌표( 서쪽(0) , 남쪽(1) ,
동쪽(2), 북쪽(3) )
        # [ 255,29,270,44]
        if pos[1] <= 0: # 공의 남쪽이 가리키는 좌표가 0보다 작아진다면 공이 위쪽 화면 밖으로 나가버리므
로
            self.y = 3 # 공을 아래로 떨어뜨린다. (공이 위로 올라갈수록 y 의 값이 작아지므로 아래로 내
리려면 다시 양수로)
            if pos[3] >= self.canvas_height: #공의 북쪽이 가리키는 좌표가 캔버스의 높이보다 더 크다면 화면
아래로 나가버려서
                self.y = -3 #공을 위로 올린다. (공이 아래로 내려갈수록 y 값이 커지므로 공을 위로 올릴려면 다
시 음수로)
            if pos[0] <= 0: # 공의 서쪽이 가리키는 좌표가 0보다 작으면 공이 화면 왼쪽으로 나가버리므로
                self.x = 3 # 공을 오른쪽으로 돌린다.
            if pos[2] >= self.canvas_width: #공의 동쪽이 가리키는 좌표가 공의 넓이보다 크다면 공이 화면 오른
쪽으로 나가버림
                self.x = -3 #공을 왼쪽으로 돌린다.
            if self.hit_paddle(pos) == True: #패들 판에 부딪히면 위로 튕겨올라가게
                self.y = -3 # 공을 위로 올린다.
```

```
class Paddle:
```

```
    def __init__(self,canvas,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, 300) # 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 밖으로 패들이 나
        가지 않도록
        self.canvas.bind_all('<KeyPress-Left>',self.turn_left) # 왼쪽 화살표 키를 '<KeyPress-Left>' 라
        는 이름로 바인딩
        self.canvas.bind_all('<KeyPress-Right>',self.turn_right) # 오른쪽도 마찬가지로 바인딩한다.

    def draw(self):
        self.canvas.move(self.id, self.x, 0) #시작할때 패들이 위아래로 움직이지 않도록 0 으로 설정
        pos = self.canvas.coords(self.id)
        print(pos)
        if pos[0] <= 0: #공의 서쪽이 가리키는 좌표가 0보다 작으면 공이 화면 왼쪽으로 나가버리므로
            self.x = 0 # 패들을 멈춰버린다.
        elif pos[2] >= self.canvas_width: #공의 동쪽이 캔버스의 넓이 보다 크면 공이 화면 오른쪽으로 나가
        버리므로
            self.x = 0 # 패들을 멈춰버린다

#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

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 알려준
다.

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 알려주는것이다.
paddle = Paddle(canvas,'blue')
ball = Ball(canvas, paddle, 'red')
start = False
#공을 약간 움직이고 새로운 위치로 화면을 다시 그리며, 잠깐 잠들었다가 다시 시작해 ! "
while 1:
    if ball.hit_bottom == False:
        ball.draw()
        paddle.draw()
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그려라 !
    tk.update() # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고 알려주는것이다.
    time.sleep(0.01) # 무한 루프중에 100 분의 1초마다 잠들어라

```

★ 파이썬으로 ping pong 게임 구현

"tkinter 라는 모듈을 사용해서 구현"

1. 캔버스(canvas) ---> 게임 바탕
2. 공(ball) 클래스
 - init 함수 : 공의 색깔과 크기를 설정, 캔버스에서의 공의 시작위치를 설정
 - 공을 움직이게 하는 함수 : 공이 캔버스 밖으로 안나가게 하는 코드 구현
 - 패들에 공이 부딪히면 공이 튕기게 하는 함수
3. 패들(paddle) 클래스
 - init 함수 : 패들의 색깔과 크기를 설정.
 캔버스에서의 패들의 시작위치를 설정
 - 패들을 움직이게 하는 함수 :
 패들이 캔버스 밖으로 안나가게 하는 코드 구현
 - 왼쪽으로 패들을 움직이게 하는 함수
 - 오른쪽으로 패들을 움직이게 하는 함수

```

from tkinter import *
import random
import time

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=500, bd=0, highlightthickness=0)
canvas.pack()
tk.update()
tk.mainloop()

```

1. tk 를 인스턴스화 한다.
 # 2. tk 객체의 title 메소드(함수)로 게임창에 제목을 부여한다.
 # 3. 게임창의 크기는 가로나 세로로 변경될수 없다라고 말하는것이다.
 # 4. 다른 모든 창들 앞에 캔버스를 가진 창이 위치할것을 tkinter 에
 알려준다
 # bg=0,highlightthickness=0 은 캔버스 외곽에 둘러싼
 # 외곽선이 없도록 하는것이다. (게임화면이 좀더 좋게)
 # 앞의 코드에서 전달된 폭과 높이는 매개변수에 따라 크기를 맞추라
 # 에 말한다.
 # tkinter 에게 게임에서의 애니메이션을 위해 자신을 초기화하라고
 알려주는것
 # 이다.
 # 창이 바로 꺼지는것을 막으려면 ? mainloop() 라 불리는 애니메
 이션 루프를
 # 추가해야 한다.

```
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
#타입)

        #          ↑      ↑      ↑      ↑
        #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()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=500, bd=0, highlightthickness=0)

canvas.pack()
tk.update()

ball = Ball(canvas, 'red')
while 1:
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그리고
    tk.update()
    time.sleep(0.01)

```

문제 409. 캔버스를 검정색으로 하고 공을 하얀색으로 변경하시오 !

```

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)

        #           ↑   ↑   ↑   ↑
        #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()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=500, bd=0, highlightthickness=0)

canvas.configure(background='black')
canvas.pack()
tk.update()

ball = Ball(canvas, 'white')
while 1:
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그리고
    tk.update()
    time.sleep(0.01) # 100 분의 1 초마다 잠들어라 !

```

문제 410. 캔버스의 공의 첫 시작위치가 천정 위가 되게 하시오 !

```

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)
                                #      ↑   ↑   ↑   ↑
                                #좌측상단 구석에서 시작하는 x y   우측 상단 구석에서 시작하는 x y
        self.canvas.move(self.id, 245, 0) #공을 캔버스 중앙으로 이동
    def draw(self):
        pass

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=500, bd=0, highlightthickness=0)

canvas.configure(background='black')
canvas.pack()
tk.update()

ball = Ball(canvas, 'white')
while 1:
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그리고
    tk.update()
    time.sleep(0.01) # 100 분의 1 초마다 잠들어라 !

```

문제 411. 공이 키보드의 방향키로 움직여지게 하려면?

```

관련코드 : self.canvas.move( self.id, self.x, self.y)
                                ↓       ↓       ↓
                                공     공의 x 방향   공의 y 방향
                                (좌, 우)   (위 아래)

```

키보드 방향키와 공을 움직이는 함수를 매칭시켜준다.

```

self.canvas.bind_all('<KeyPress-Left>', self.turn_left)
# 키보드 방향키 <- 를 누르면 turn_left 함수가 실행되게해라 !

```

```

self.canvas.bind_all('<KeyPress-Right>', self.turn_right)
# 키보드 방향키 -> 를 누르면 turn_right 함수가 실행되게해라 !

```

```

def turn_left(self, evt ):
    self.x = -9

```

```

def turn_right(self, evt):
    self.y = 9

```

```

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)
self.x=0
self.y=0
self.canvas.move(self.id, 245, 0)
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)

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
    pass

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=500, bd=0, highlightthickness=0)

canvas.configure(background='black')
canvas.pack()
tk.update()

ball = Ball(canvas,'white')
while 1:
    ball.draw()
    tk.update_idletasks() # 우리가 창을 닫으라고 할때까지 계속해서 tkinter 에게 화면을 그리고
    tk.update()
    time.sleep(0.01) # 100분의 1 초마다 잠들어라 !

```

문제 412. (오늘의 마지막 문제) 공이 아래로 떨어졌을 때 게임이 끝나는게 아니라 공이 다시 위로 올라가게 하시오 !

```

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)
        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.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

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)
```


문제 414. (탐욕 알고리즘)탐욕 알고리즘을 이용하여 금액과 화폐가 주어졌을 때 가장 적은 화폐로 지불하시오 !

"탐욕 알고리즘이 어떤 알고리즘이냐면, 매 순간마다 최선의 선택하는것 입니다.

즉, 선택할때마다 가장 좋다고 생각되는 것을 선택해나가며

최종적인 해답을 구하는 알고리즘 입니다.

이 알고리즘을 설계할 때 유의할 점은 전체를 고려하는게 아니라 문제를 부분적으로 나누어, 나누어진 문제에 대한 최적의 해답을 구하므로 전체적인 최적의 해가 될 수 있는 경우가 존재합니다. "

최단 거리 알고리즘 구현하기 위해서 필요한 알고리즘이

"탐욕(greedy) 알고리즘 "

액수 입력: 362

화폐단위를 입력하세요 : 1 50 100

결과 :

100 원 :3 개

50 원 : 1 개

1 원 : 12 개

참고 스크립트 :

```
money = int(input('액수입력 : '))
cash_type = [int(x) for x in input('화폐단위를 입력하세요 : ').split(' ')]
res = coinGreedy(money,cash_type)
for key in res:
    print('{0}원 : {1}개'.format(key,res[key]))
```

답 :

```
def greedy():
    money = int(input('액수입력 : '))
    cash_type = [int(x) for x in input('화폐단위를 입력하세요 : ').split(' ')]
    cash_type.sort(reverse = True)
    for cash in cash_type:
        x = int((money - (money%cash))/cash)
        money = money - x*cash
        print('%d 원 : %d 개'%(cash, x))

greedy()
```