

목차

2020년 11월 24일 화요일 오전 9:46

| 번호 | 수업 내용 |
|----|-------------------------------------|
| 1 | 대화식 모드로 프로그래밍하기 |
| 2 | 텍스트 에디터로 프로그래밍하기 |
| 3 | 변수명 만들기 |
| 4 | 변수에 값 대입하기 |
| 5 | 주석 처리하기(##) |
| 6 | 자료형 개념 배우기 |
| 7 | 자료형 출력 개념 배우기(print) |
| 8 | 들여쓰기 개념 배우기 |
| 9 | if문 개념 배우기 ① (if~else) |
| 10 | if문 개념 배우기 ② (if~elif) |
| 11 | for문 개념 배우기 ① (for) |
| 12 | for문 개념 배우기 ② (for~continue~break) |
| 13 | for문 개념 배우기 ③ (for~else) |
| 14 | while문 개념 배우기(while~continue~break) |
| 15 | None 개념 배우기 |
| 16 | 정수형 자료 이해하기 |
| 17 | 실수형 자료 이해하기 |
| 18 | 복소수형 자료 이해하기 |
| 19 | 대입 연산자 이해하기(=) |
| 20 | 사칙 연산자 이해하기(+, -, *, /, **) |
| 21 | 연산자 축약 이해하기(=, -=, *=, /=) |
| 22 | True와 False 이해하기 |
| 23 | 관계 연산자 이해하기(==, !=, <, <=, >, >=) |
| 24 | 논리 연산자 이해하기(and, or, not) |
| 25 | 비트 연산자 이해하기(&, , ~, ^, >>, <<) |
| 26 | 시퀀스 자료형 이해하기 |
| 27 | 시퀀스 자료 인덱싱 이해하기 |
| 28 | 시퀀스 자료 슬라이싱 이해하기 |

| | |
|----|---|
| 29 | 시퀀스 자료 연결 이해하기() |
| 30 | 시퀀스 자료 반복 이해하기(*) |
| 31 | 시퀀스 자료 크기 이해하기(len) |
| 32 | 멤버체크 이해하기(in) |
| 33 | 문자열 이해하기 |
| 34 | 문자열 포매팅 이해하기 |
| 35 | 이스케이프 문자 이해하기 |
| 36 | 리스트 이해하기([]) |
| 37 | 튜플 이해하기(()) |
| 38 | 사전 이해하기({ }) |
| 39 | 함수 이해하기(def) |
| 40 | 함수 인자 이해하기 |
| 41 | 지역변수와 전역변수 이해하기(global) |
| 42 | 함수 리턴값 이해하기(return) |
| 43 | 파이썬 모듈 이해하기 |
| 44 | 파이썬 패키지 이해하기 |
| 45 | 파이썬 모듈 임포트 이해하기 ① (import) |
| 46 | 파이썬 모듈 임포트 이해하기 ② (from~import) |
| 47 | 파이썬 모듈 임포트 이해하기 ③ (import~as) |
| 48 | 파일 열고 닫기(open, close) |
| 49 | 클래스 이해하기(class) |
| 50 | 클래스 멤버와 인스턴스 멤버 이해하기 |
| 51 | 클래스 메소드 이해하기 |
| 52 | 클래스 생성자 이해하기 |
| 53 | 클래스 소멸자 이해하기 |
| 54 | 클래스 상속 이해하기 |
| 55 | 예외처리 이해하기 ① (try~except) |
| 56 | 예외처리 이해하기 ② (try~except~else) |
| 57 | 예외처리 이해하기 ③ (try~except~finally) |
| 58 | 예외처리 이해하기 ④ (try~except Exception as e) |
| 59 | 예외처리 이해하기 ⑤ (try~except 특정 예외) |
| 60 | 사용자 입력받기(input) |
| 61 | 자료형 확인하기(type) |

| | |
|----|---|
| 62 | 나눗셈에서 나머지만 구하기(%) |
| 63 | 몫과 나머지 구하기(divmod) |
| 64 | Pandas 를 이용한 데이터 검색 |
| 65 | Pandas 를 이용한 조인 |
| 66 | Pandas 를 이용한 서브쿼리 |
| 67 | Pandas 와 오라클 그룹함수의 비교 |
| 68 | 반올림수 구하기(round) |
| 69 | 실수형 자료를 정수형 자료로 변환하기(int) |
| 70 | 정수형 자료를 실수형 자료로 변환하기(float) |
| 71 | 정수 리스트에서 소수만 걸러내기(filter) |
| 72 | 최대값, 최소값 구하기(max, min) |
| 73 | 판다스에서 결측치(NaN) 확인하기 |
| 74 | 판다스에서 파생변수 추가하는 방법 |
| 75 | 문자열에서 특정 위치의 문자 얻기 |
| 76 | 문자열에서 지정한 구간의 문자열 얻기 |
| 77 | 문자열에서 홀수 번째 문자만 추출하기 |
| 78 | 문자열을 거꾸로 만들기 |
| 79 | 두 개의 문자열 합치기() |
| 80 | 문자열을 반복해서 새로운 문자열로 만들기(*) |
| 81 | 문자열에서 특정 문자가 있는지 확인하기(in) |
| 82 | 문자열에서 특정 문자열이 있는지 확인하기(in) |
| 83 | 문자열 길이 구하기(len) |
| 84 | 문자열이 알파벳인지 검사하기(isalpha) |
| 85 | 문자열이 숫자인지 검사하기(isdigit) |
| 86 | 문자열이 알파벳 또는 숫자인지 검사하기(isalnum) |
| 87 | 문자열에서 대소문자 변환하기(upper, lower) |
| 88 | 문자열에서 좌우 공백 제거하기(lstrip, rstrip, strip) |
| 89 | 문자열에서 문자 개수 구하기 (count) |
| 90 | 문자열에서 특정 문자(열) 위치 찾기 (find) |
| 91 | 문자열을 특정 문자(열)로 분리하기(split) |
| 92 | 문자열을 특정 문자(열)로 결합하기(join) |
| 93 | 문자열에서 특정 문자(열)를 다른 문자(열)로 바꾸기(replace) |
| 94 | 문자열을 바이트 객체로 바꾸기(encode) |

| | |
|-----|---|
| 95 | 바이트 객체를 문자열로 바꾸기(decode) |
| 96 | 순차적인 정수 리스트 만들기(range) |
| 97 | 리스트에서 특정 위치의 요소 얻기 |
| 98 | 리스트에서 특정 요소의 위치 구하기(index) |
| 99 | 리스트에서 특정 위치의 요소를 변경하기 |
| 100 | 리스트에서 특정 구간에 있는 요소 추출하기 |
| 101 | 리스트에서 짝수 번째 요소만 추출하기 |
| 102 | 리스트 요소 순서를 역순으로 만들기 ① (reverse) |
| 103 | 리스트 요소 순서를 역순으로 만들기 ② (reversed) |
| 104 | 리스트 합치기() |
| 105 | 리스트 반복하기(*) |
| 106 | 리스트에 요소 추가하기(append) |
| 107 | 리스트의 특정 위치에 요소 삽입하기(insert) |
| 108 | 리스트의 특정 위치의 요소 제거하기(del) |
| 109 | 리스트에서 특정 요소 제거하기(remove) |
| 110 | 리스트에서 특정 구간에 있는 모든 요소 제거하기 |
| 111 | 리스트에 있는 요소 개수 구하기(len) |
| 112 | 리스트에서 특정 요소 개수 구하기(count) |
| 113 | 리스트 제거하기(del) |
| 114 | 리스트 요소 정렬하기 ① (sort) |
| 115 | 리스트 요소 정렬하기 ② (sorted) |
| 116 | 리스트 요소 무작위로 섞기(shuffle) |
| 117 | 리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate) |
| 118 | 리스트의 모든 요소의 합 구하기(sum) |
| 119 | 리스트 요소가 모두 참인지 확인하기(all, any) |
| 120 | 사전에 요소 추가하기 |
| 121 | 사전의 특정 요소값 변경하기 |
| 122 | 사전의 특정 요소 제거하기(del) |
| 123 | 사전의 모든 요소 제거하기(clear) |
| 124 | 사전에서 키만 추출하기(keys) |
| 125 | 사전에서 값만 추출하기(values) |
| 126 | 사전 요소를 모두 추출하기(items)20130 사전에 특정 키가 존재하는지 확인하기(in) |
| 127 | 사전 정렬하기(sorted) |

| | |
|-----|----------------------------------|
| 128 | 문자 코드값 구하기(ord) |
| 129 | 코드값에 대응하는 문자 얻기(chr) |
| 130 | 문자열로 된 식을 실행하기(eval) |
| 131 | 이름없는 한줄짜리 함수 만들기(lambda) |
| 132 | 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map) |
| 133 | 텍스트 파일을 읽고 출력하기(read) |
| 134 | 텍스트 파일을 한줄씩 읽고 출력하기 ① (readline) |
| 135 | 화면에서 사용자 입력을 받고 파일로 쓰기(write) |
| 136 | 텍스트 파일에 한줄씩 쓰기(writelines) |
| 137 | 텍스트 파일 복사하기(read, write) |
| 138 | 바이너리 파일 복사하기(read, write) |
| 139 | 파일을 열고 자동으로 닫기 (with ~ as) |
| 140 | HTML 기본문법 |
| 141 | beautiful soup 모듈 배우기 |
| 142 | 웹스크롤링 실전 1단계 (ebs 레یدی 버그 게시판) |
| 143 | 웹스크롤링 실전2 (중앙일보사) |
| 144 | 파이썬에서 워드 클라우드 그리기 |
| 145 | 웹스크롤링 실전3 (동아일보) |
| 146 | 웹에 있는 사진을 스크롤링 하는 방법 (구글 이미지) |
| 147 | 이미지 스크롤링 하기 (네이버 이미지 검색) |
| 148 | 파이썬과 oracle 연동 |
| 149 | 파이썬과 mySQL 연동 |
| 150 | 이미지를 숫자로 변환하는 방법 (폐사진) |
| 151 | 이미지를 숫자로 변환하는 방법 (개와 고양이) |
| 152 | 필수 알고리즘1 (합성곱 연산) |
| 153 | 필수 알고리즘2 (이진 탐색) |
| 154 | 필수 알고리즘3 (버블정렬) |
| 155 | 필수 알고리즘4 (탐욕알고리즘) |
| 156 | 필수 알고리즘5 (재귀알고리즘) |
| 157 | 필수 알고리즘6 (LRU 알고리즘) |
| 158 | 필수 알고리즘7 (자카드 유사도 알고리즘) |
| 159 | 필수 알고리즘8 (비밀지도) |
| 160 | 통계 마지막 문제 (도박사 이야기) |

20.11.24

2020년 11월 24일 화요일 오전 9:41

파이썬을 해서 우리가 얻어야할 지식

1. 빅데이터 수집을 위한 웹스크롤링을 할 수 있는 능력 키우기
2. SQL처럼 파이썬으로 데이터를 검색하고 데이터를 분석할 수 있는 능력 키우기
3. 딥러닝 개발자가 되기 위한 기본 코딩능력 키우기

파이썬 설치

아나콘다 최신 버전으로 설치

- 파이썬 기본 프로그램 + 여러가지 유용한 패키지

Spyder 툴로 파이썬 코드 연습

1. 대화식 모드로 프로그래밍하기

파이썬을 실행하는 방법 2가지

1. 대화식 모드 : 한라인씩 실행하는 모드

Ex) anaconda prompt 에서 python을 입력 후 실행

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

2. 배치 모드 : 여러개의 스크립트로 작성해서 한번에 실행하는 모드

Spyder에서 위와 같이 스크립트를 작성하고 전체선택 후 ctrl + enter 로 실행

문제1. 아래의 스크립트를 배치모드로 실행하시오.

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

2. 텍스트 에디터로 프로그래밍하기

Spyder 와 같은 텍스트 창에서 프로그램 코드를 작성하는 것이다.

Spyder 프로그램에서 ctrl + enter 키로 코드를 실행하거나 F5를 누르면 실행 된다.

문제2. 아래와 같은 결과가 나오게 spyder 에서 프로그래밍 하시오.

```
2
3
4
```

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

3. 변수명 만들기

어떤 값을 임시로 저장하는 변수의 이름을 만드는 방법과 규칙 학습

Ex)
A = 1

A라는 변수에 숫자 1을 할당한다.

Ex)
A = 1
B = 2
Print (a + b)

변수 이름을 생성할 때 주의할 사항

1. 변수 이름에는 다음 문자만 사용할 수 있다.
 - a. 소문자 (a ~ z)
 - b. 대문자 (A ~ Z)
 - c. 숫자 (0 ~ 9)
 - d. 언더스코어 (_)
2. 변수 이름은 숫자로 시작할 수 없다. (문자로 시작해야 한다.)
3. 예약어를 사용할 수 없다. (파이썬에서 이미 사용되고 있는 단어를 예약어라고 한다.)

Ex)
True, False, Class, Is, Return

문제3. 파이썬의 예약어가 무엇이 있는지 확인하시오.

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

문제4. 위의 예약어를 변수로 사용하면 에러가 나는지 직접 테스트 하

시오.

```
as = 1
```

4. 변수에 값 대입하기

다양한 값을 변수에 대입하는 방법을 학습한다.

파이썬은 변수에 값을 대입할 때 = (assingment) 기호를 사용한다.

Ex)

```
A = 1
```

```
B = 'scott'
```

```
Print (B)
```

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

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

파이썬의 코드는 심플함을 철학으로 삼고 있다.

파이썬의 기본 철학을 확인하는 방법

Import this

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

문제5. 변수의 자료형을 확인하시오.

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

<class 'str'> 문자형

문제6. 숫자형 변수의 자료형을 확인하시오.

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

<class 'int'> 숫자형

5. 주석 처리하기(#)

프로그램에서 주석부분은 인터프리터에 의해 무시되는 텍스트의 한 부분이다.

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

코드를 작성할 때 주석을 작 작성해두면 차후에 코드를 다시 보거나 타인이 코드를 검토할 때 매우 중요한 정보로 활용이 될 수 있다.

그래서 주석을 항상 달아두는 습관을 가지는 것이 좋다.

Ex)

1. 한줄 주석

A = 1 # 변수 a 에 숫자 1을 할당한다.

Print (a) # a의 결과를 출력한다.

2. 여러줄 주석

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

A = 1

Print(a)

6. 자료형 개념 배우기

자료형이란 프로그래밍을 할 때 쓰이는 숫자, 문자열등의 자료 형태로 사용되는 모든 것을 뜻한다.

파이썬에서 자주 다루게 되는 자료형 5가지

1. 숫자형 자료형 : 숫자를 표현하는 자료형

Ex)

A =1

2. 문자형 자료형 : 문자를 표현하는 자료형

Ex)

B = 'scott'

3. 리스트 자료형 : [] 대괄호 안에 임의 객체를 순서있게 나열한 자료형

Ex)

D = [1, 2, 3] # 파이썬은 시작을 0부터 시작한다.

Print (d)

Print (d[0]) # d리스트에서 첫번째 요소를 프린트

4. 튜플 자료형 : 리스트와 비슷하지만 요소값을 변경할 수 없다는 것이 리스트와 다르다.

Ex)

C = (1, 2, 3)

Print (c)

5. 사전 자료형 : {} 중괄호 안의 키:값으로 된 쌍이 요소로 구성된 순서가 없는 자료형이다.

Ex)

M = {'i':'나는','am':'입니다','boy':'소년'}

Print (m)

문제7. 아래의 리스트에서 문자 k 를 출력하시오.

```
mmm = ['a', 'b', 'd', 'e', 'k', 'm', 'n', 'z']
```

```
print (mmm[4])
```

문제8. 동전을 던져서 앞면이 나오는지 뒷면이 나오는지 확인하시오.

```
import random
```

```
coin = ['앞면', '뒷면']
```

```
print (random.choice(coin))
```

문제9. 주사위를 던져서 주사위의 눈이 뭐가 나오는지 확인하는 파이썬 코드를 작성하시오.

```
import random
```

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

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

7. 자료형 출력 개념 배우기(print)

Print 함수를 이용하면 다양한 자료형을 화면에 출력할 수 있다.

Ex)

A = 200

B = 'i love python'

C = ['a', 'b', 'c']

Print (a)

Print (b)

Print (c)

8. 들여쓰기 개념 배우기

파이썬에는 실행코드 부분을 묶어주는 ()괄호가 없다.

들여쓰기로 괄호를 대신한다.

파이썬은 다른 프로그래밍 언어와 달리 if, for, while 등과 같은 제어문, 루프문의 실행코드 부분을 구분해주는 괄호가 없다.

대신 들여쓰기로 괄호를 대신한다.

파이썬에서 제어문이나 함수이름, 클래스이름 뒤에 콜론(:) 으로 끝을 표시하며 콜론(:) 다음에 실행코드를 작성하는데 이때 들여쓰기를 해야한다.

Ex)

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

```
If 'a' in listdata:
```

```
    Print ('a가 listdata에 있습니다.')
```

Ex)

실행코드를 한 라인에 작성한 경우

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

```
If 'a' in listdata : print ('a가 listdata에 있습니다.')
```

콜론(:) 으로 if문의 끝을 알린다.

문제10. 주사위의 눈이 6개 있는 변수를 만들고 if문을 사용하여 "주사위의 눈에 숫자 5가 있습니다. " 라는 메시지가 출력되게 하시오.

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

```
if 5 in dice:
```

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

문제11. 위의 코드를 수정해서 숫자를 물어보게하고 숫자를 입력하면 위의 메시지가 출력되게 하시오.

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

```
a = int (input ('숫자를 입력하시오.'))
```

```
if a in dice:
```

```
    print ('주사위의 눈에 숫자', a, '가 있습니다.')
```

9. if문 개념 배우기 ① (if~else)

어떤 조건을 참과 거짓으로 판단할 때 if 문을 사용한다.

참과 거짓을 구분하여 코드를 실행하면 if ~ else를 사용한다.

코드를 작성하다 보면 조건을 따라 수행하는 일을 달리 해야하는 경우가 있다.

조건이 참인지 거짓인지 검사를 하고, 참인 경우 이 일을 하고 거짓인 경우에는 저 일을 해라 라는 식으로 처리할 수 있다.

Ex)

If 조건:

실행코드1

Else:

실행코드2

Ex)

```
A = int(input('숫자를 입력하시오.'))
```

```
If a%2 == 0:
```

```
    Print ('짝수 입니다.')
```

```
Else:
```

```
    Print ('홀수 입니다.')
```

%는 나눈 나머지값을 출력하는 연산자

==는 equal 연산자

=는 할당 연산자

문제12. 주사위의 눈을 담은 dice변수를 만들고 숫자를 물어보게 해서 해당 숫자가 주사위의 눈 중에 있으면 해당 숫자가 있습니다. 라는 메시지가 출력되게 하고 없으면 해당 숫자가 없습니다.라는 메시지가 출력되게 하시오.

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

```
a = int(input('숫자를 입력하시오.'))
```

```
if a in dice:
```

```
    print(a, '가 주사위의 눈에 있습니다.')
```

```
else:
```

```
    print(a, '가 주사위의 눈에 없습니다.')
```

문제13. 숫자 두개를 아래와 같이 각각 물어보게 하고 아래처럼 메시지가 출력되게하시오.

첫번째 숫자를 입력하세요. 2

두번째 숫자를 입력하세요. 3

2는 3보다 작습니다.

첫번째 숫자를 입력하세요. 3

두번째 숫자를 입력하세요. 2

3는 2보다 큼니다.

```
a = int(input('첫번째 숫자를 입력하세요.'))
```

```
b = int(input('두번째 숫자를 입력하세요.'))
```

```
if a > b:
```

```
    print(a, '는', b, '보다 큼니다.')
```

```
else:
```

```
    print(a, '는', b, '보다 작습니다.')
```

10. if문 개념 배우기 ② (if~elif)

여러개의 조건을 순차적으로 체크하고 해당 조건이 참이면 특정 로직을 수행하고자 할 때 if ~ elif문을 사용한다.

Ex)

If 조건1:

실행코드1

Elif 조건2:

실행코드2

Else:

실행코드3

위의 조건1과 조건2가 아니라면 실행코드3을 출력한다.

Ex)

문제13번을 다시 수행하는데 같은 숫자가 2개가 들어오면 서로 같습니다.라는 메시지가 출력되게 하시오.

```
A = int(input('첫번째 숫자를 입력하시오.'))
```

```
B = int(input('두번째 숫자를 입력하시오.'))
```

```
If a > b:
```

```
    Print(a, '는', b, '보다 큼니다.')
```

```
Elif a < b:
```

```
    Print(a, '는', b, '보다 작습니다.')
```

```
Else:
```

```
    Print(a, '는', b, '와 같습니다.')
```

11. for문 개념 배우기 ① (for)

특정코드를 반복적으로 수행하기 위해서는 반복문을 사용해야 하는데 파이썬에서는 for문이 반복문을 수행하기 위해 가장 많이 사용되는 문법이다.

Ex)

For 변수 in 범위:

반복적으로 실행할 코드

1. 리스트 범위인 경우

```
For i in [1,2,3]:
```

```
    Print(i)
```

2. 튜플 범위인 경우

```
For i in (1,2,3):
```

```
    Print(i)
```

3. Range() 범위인 경우

```
For i in range(10):
```

```
    Print(i)
```

4. 사형 범위인 경우

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

```
For i in m:
```

```
    Print (i)
```

- 키값만 출력된다.

5. 문자형 범위인 경우

```
For i in 'i am a boy':
```

```
    Print (i)
```

For loop문의 range 사용법 정리

```
For i in range(6):
```

```
    Print (i) # 0부터 5까지 출력한다.
```

```
For i in range(1,6):
```

```
    Print (i) # 0부터 5까지 출력한다.
```

```
For i in range(1,6,2):
```

```
    Print (i) # 1,3,5 를 출력한다.
```

```
For i in range(6,1,-1):
```

```
    Print (i) # 6부터 1씩 차감해서 2까지 출력한다.
```

문제14. ★을 5개 출력하시오.

★★★★★

```
print ('★'*5)
```

문제15. 아래와 같은 결과가 출력되는 파이썬 코드를 작성하시오.

숫자 입력하세요. 5

★

★★

★★★

★★★★

★★★★★

```
a = int(input('숫자를 입력하시오.'))
```

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

문제16. 주사위를 10번 던져서 출력되는 눈을 확인하시오.

```
import random
```

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

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

문제17. 동전을 10번 던지시오.

```
import random
```

```
coin = ['앞면', '뒷면']
```

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

문제18. 동전을 10번 던졌을 때 앞면이 나오는 횟수가 어떻게 되는가

```
import random
```

```
coin = ['앞면', '뒷면']
```

```
cnt = 0
```

```
for i in range(10):  
    result = random.choice(coin)
```

```
if result == '앞면':  
    cnt = cnt + 1  
  
print (cnt)
```

문제19. 주사위를 100번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오.

```
import random  
  
dice = [1, 2, 3, 4, 5, 6]  
cnt = 0  
  
for i in range(100):  
    result = random.choice(dice)  
    if result == 3:  
        cnt = cnt + 1  
  
print (cnt)
```

문제20. 주사위를 1000번 던지고 주사위의 눈이 5가 나올 확률을 구하시오.

```
import random  
  
dice = [1, 2, 3, 4, 5, 6]  
cnt = 0  
  
for i in range(1000):  
    result = random.choice(dice)  
    if result == 5:  
        cnt = cnt + 1  
  
print (cnt/1000)
```

문제21. 동전을 10000번 던져서 앞면이 나올 확률을 출력하시오.

```
import random  
  
coin = ['앞면', '뒷면']  
cnt = 0  
  
for i in range(10000):  
    result = random.choice(coin)  
    if result == '앞면':  
        cnt = cnt + 1  
  
print (cnt/10000)
```


문제22. 주사위 하나와 동전 한개를 동시에 던져서 주사위의 눈은 5가 나오고 동전은 앞면이 나올 확률을 구하시오.

```
import random

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

for i in range(100000):
    result1 = random.choice(dice)
    result2 = random.choice(coin)
    if result1 == 5 and result2 == '앞면':
        cnt = cnt + 1

print (cnt/100000)
```

20.11.25

2020년 11월 25일 수요일 오전 9:33

문제23. for loop문을 이용하여 구구단 2단을 출력하시오.

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

For loop 문은 특정 문장을 반복해서 수행하고 싶을 때 사용하는 파이썬 문법이다.

위의 문법은 1부터 9까지 9번 반복한다.

문제24. 구구단 전체를 파이썬으로 출력하시오.

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

J = 2 일때 i 가 1 ~ 9 까지 반복 실행

J = 3 일때 i 가 1 ~ 9 까지 반복 실행

J = 4 일때 i 가 1 ~ 9 까지 반복 실행

·
·
·

J = 9 일때 i 가 1 ~ 9 까지 반복 실행

- 중첩 for loop문

Loop문을 중첩시켜서 수행한다.

Ex)

```
For j in range (1, 10):  
    For i in range(1, 10):  
        Print ('2 x ', i, ' = ', 2*i)
```

2단을 출력하는 for loop문이 9번 실행되면서 2단이 9번 나온다.

문제25. 주사위를 10번 던져서 나오는 주사위의 눈을 출력하시오.

```
import random
```

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

```
for i in range(1, 11):  
    print (random.choice(dice))
```

문제26. 주사위를 10번 던졌을 때 짝수가 나오는 횟수를 출력하시오.

```
import random  
  
dice = [1, 2, 3, 4, 5, 6]  
cnt = 0  
  
for i in range (1, 11):  
    result = random.choice(dice)  
    if result%2 == 0:  
        cnt = cnt + 1  
  
print (cnt)
```

문제27. 위의 주사위를 10번 던져서 짝수가 나오는 횟수를 확인하는 작업을 5번 반복해서 짝수가 나오는 횟수가 5개가 나오게 하시오.

```
import random  
  
dice = [1, 2, 3, 4, 5, 6]  
  
for j in range(1, 6):  
    cnt = 0  
    for i in range(1, 11):  
        result = random.choice(dice)  
        if result%2 == 0:  
            cnt = cnt + 1  
  
    print (cnt)
```

주사위를 10 던져서 짝수가 나오는 횟수를 확인하는 for loop 문을 5번 반복하기 위해서 그 위에 for loop문을 쓰고 5번 반복하게 한다.

문제28. 동전을 100번 던져서 앞면이 나올 확률을 출력하시오.

```
import random  
  
coin = ['앞면', '뒷면']  
cnt = 0  
  
for i in range(1, 101):  
    result = random.choice(coin)  
    if result == '앞면':  
        cnt = cnt + 1  
  
print (cnt/100)
```

문제29. 동전을 100번 던져서 앞면이 나올 확률을 구하는 작업을 50번 반복하시오.

```
import random

coin = ['앞면', '뒷면']

for j in range(1,51):
    cnt = 0
    for i in range(1, 101):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt + 1

    print (cnt/100)
```

문제30. 파이썬 리스트의 append 함수를 이용하여 위의 확률 50개를 리스트에 담으시오.

```
import random

coin = ['앞면', '뒷면']
a = []

for j in range(1,51):
    cnt = 0
    for i in range(1, 101):
        result = random.choice(coin)
        if result == '앞면':
            cnt = cnt + 1

    a.append (cnt/100)
print (a)
```

리스트 append 함수의 기본 문법

A = [] # 비어있는 리스트 a를 생성한다.

A.append(7)

Print (a)

A.append(8)

Print (a)

문제31. 두개의 동전을 300번 던져서 동시에 둘 다 앞면이 나오는 횟수를 출력하시오.

```
import random
```

```

coin1 = ['앞면', '뒷면']
coin2 = ['앞면', '뒷면']
cnt = 0

for i in range(1,301):
    result1 = random.choice(coin1)
    result2 = random.choice(coin2)
    if result1 == '앞면' and result2 == '앞면':
        cnt = cnt + 1

print (cnt)

```

문제32. x라는 리스트를 만들고 x라는 리스트에 위의 문제에서 앞면이 나오는 횟수를 x 리스트에 담는데 위의 둘 다 앞면이 나오는 횟수를 출력하는 for loop문을 1000번 반복해서 x 리스트에 1000개를 입력하시오.

```

import random

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

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

print (x)

```

문제33. 확률변수 x 의 값들의 평균과 분산과 표준편차를 출력하시오.

```

import random
import numpy as np

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

for j in range(1, 1001):
    cnt = 0
    for i in range(1,301):

```

```

result1 = random.choice(coin1)
result2 = random.choice(coin2)
if result1 == '앞면' and result2 == '앞면':
    cnt = cnt + 1
x.append(cnt)

print(np.mean(x)) # 평균
print(np.var(x)) # 분산
print(np.std(x)) # 표준편차

```

문제34. 한개의 주사위를 360번 던져서 3의 배수의 눈이 나오는 횟수를 1000번 구하고 1000번 수행 하였을 때 x 변수를 넣고 x 변수의 평균과 분산과 표준편차를 출력하시오.

```

import random
import numpy as np

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

for j in range(1, 1001):
    cnt = 0
    for i in range(1, 361):
        result = random.choice(dice)
        if result%3 == 0:
            cnt = cnt + 1
    x.append(cnt)

print(np.mean(x))
print(np.var(x))
print(np.std(x))

```

12. for문 개념 배우기 ② (for~continue~break)

For 반복문 내에서 continue를 만나면 그 다음 반복 실행으로 넘어가고 break를 만나면 for 반복문을 완전히 벗어나게 된다.

Ex)

For 변수 in 범위:

실행코드

Continue # 다음 반복문을 수행

실행코드

Break # for 반복문을 탈출

문제35. 숫자 1 부터 10 까지 출력하시오.

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

문제36. 위의 결과에서 숫자 5를 빼고 출력하시오.

```
for i in range(1, 11):  
    if i == 5:  
        continue # 이 부분은 지나치고 다음 루프 실행문을 계속 실행  
    print (i)
```

문제37. 1부터 10까지의 숫자를 출력하는데 짝수만 출력하시오.

```
for i in range(1, 11):  
    if i%2 != 0:  
        continue  
    print (i)
```

문제38. 1부터 10까지의 합을 파이썬으로 구현하시오.

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

For 문에서 사용하는 break문

- 루프문을 중단시키는 역할을 하는 키워드

Ex)

```
For i in range(1, 11):  
    Print (i)  
    If i == 3:  
        break
```

문제39. 1부터 100번까지 출력하는 for loop 문을 작성하는데 다음과 같이 숫자를 물어보게 해서 입력된 숫자 까지만 출력되게 하시오.

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

문제40. 두 숫자를 각각 물어보게 하고 입력받은 두 숫자의 최대공약

수를 출력하시오.

```
a = int(input('첫번째 숫자를 입력하시오.'))
b = int(input('두번째 숫자를 입력하시오.'))

for i in range (max(a, b), 1, -1):
    if a%i == 0 and b%i == 0:
        break
print (a, '와', b, '의 최대공약수는', i, '입니다.')
```

13. for문 개념 배우기 ③ (for~else)

For 반복문을 완전히 수행했을 때만 실행하는 부분을 정의 하려면 for ~ else문을 사용해야 한다. For ~ else 뒤에 실행되는 코드는 for 반복문을 성공적으로 수행해야지만 실행된다.

```
Ex)
For i in range(1, 11):
    Print (i)
Else:
    Print ('Perfect') # break에 의해서 중단되면 else다음 실행문이 실행되지 않는다.
```

문제41. 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자 만큼 1번 부터 출력되게 하시오.

```
a = int(input('숫자를 입력하시오.'))

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

문제42. 위의 입력한 숫자까지 다 출력하면 아래에 perfect 이란 단어가 출력되게 하시오.

```
a = int(input('숫자를 입력하시오.'))

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

문제43. 위의 코드를 수정해서 중단할 숫자를 또 물어보게 해서 아래와 같이 실행되게 하시오.

숫자를 입력하시오. 10

중단할 숫자를 입력하시오. 5

```
a = int(input('숫자를 입력하시오.'))
b = int(input('중단할 숫자를 입력하시오.'))

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

14. while문 개념 배우기(while~continue~break)

For loop문 처럼 while loop문도 같은 반복문 이다.

For loop문은 특정 범위에서 반복 실행하게 하는 반면에 while loop문은 특정 조건에서 코드를 반복 실행하게 한다.

| For loop문 | While loop문 |
|----------------------------------|--|
| For i in range(10): Print (i) | X = 0 While x < 10: X = x + 1 Print (x) |

문제44. 아래와 같이 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오.

```
a = int(input('숫자를 입력하시오.'))
x = 0

while x < a:
    x = x + 1
    print(x * '★')
```

문제45. 아래와 같이 결과가 출력되게 하시오.

숫자를 입력하세요 5

```
★★★★★
★★★★★
★★★★★
★★★
★★
★
```

```
a = int(input('숫자를 입력하시오.'))
x = a
```

```
while x > 0:
    print (x * '★')
    x = x - 1
```

문제46. 주사위 2개를 동시에 던져서 두개의 주사위의 눈의 합을 구하는 실행문을 10000 번 반복하고 a 리스트에 넣으시오.

```
import random

dice1 = [1, 2, 3, 4, 5, 6]
dice2 = [1, 2, 3, 4, 5, 6]
x = 0
a = []

while x < 10000:
    x = x + 1
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    a.append (result1 + result2)

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

문제47. 위의 코드를 이용해서 두 개의 주사위의 눈의 합이 10이 되는 확률을 구하시오.

```
import random

dice1 = [1, 2, 3, 4, 5, 6]
dice2 = [1, 2, 3, 4, 5, 6]
x = 0
cnt = 0

while x < 10000:
    x = x + 1
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt = cnt + 1

print (cnt/10000)
```

15. None 개념 배우기

None는 아무것도 없다는 의미의 상수 이다.

아무것도 없다는 것을 나타내기 위해서 주로 활용 된다.

```
Ex)
X = 0
While x < 10:
    Print (x)
    X = x + 1
```

위의 while loop문에서의 x 변수는 숫자만 담기는 변수이다.
그런데 상황에 따라 숫자가 들어갈 수도 있고 문자가 들어갈 수도 있어서 변수 선언시 결정을 못하겠다면 아래와 같이 하면 된다.

```
X = none
A = 1
If a ==1:
    X = [1, 2, 3] # x 라는 변수에 리스트를 담았다.
Else:
    X = 'i love python' # x 라는 변수에 문자열을 담았다.
```

Print (x)

16. 정수형 자료 이해하기

자연수(1,2,3, ...) 와 음수(-1,-2,-3, ...) 와 0으로 이루어진 수의 체계를 정수라고 한다.

일반 프로그래밍에서 지원하는 정수형 상수의 범위는 -2,147,473,647 ~ 2,147,473,647 인데 (2의 31승)

파이썬은 메모리가 허용하는 범위에서 지원 가능한 수를 모두 사용할 수 있다.

```
import random
```

Print (random.uniform(0,1)) # 0 ~ 1 사이의 난수를 출력

문제48. for loop문으로 테스트해서 확인 하시오.

파이썬에서는 정수형 변수가 담을 수 있는 상수의 범위가 어떻게 되는지 테스트 하시오.

[illegible]

문제 49. 3의 2승을 파이썬으로 구하시오.

```
print (3**2)
```

문제50. 루트 4를 파이썬으로 구하시오.

```
import math

print (math.sqrt(4))
```

문제51. 몬테 카를로 알고리즘을 이용해서 원주율의 근사값을 파이썬으로 구하시오.

```
import random
cnt = 0

for i in range (10000):
    result1 = random.uniform(0, 1)
    result2 = random.uniform(0, 1)
    if result1**2 + result2**2 < 1:
        cnt = cnt + 1
print (cnt/ 10000 * 4)
```

17. 실수형 자료 이해하기

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

(무리수의 예 : 원주율, 오일러 자연상수 e)

파이썬은 실수를 지원하기 위해서 부동 소수형을 제공한다.

- 부동 소수형의 특징

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

Ex)

소수 43.2 에서 소수 43.1을 빼면 얼마인지 파이썬으로 구현하시오.

```
print (43.2 - 43.1)
```

```
0.1000000000000000142
```

부동 소수형은 위와 같이 정밀도의 한계를 가지고있다.

이것은 비단 파이썬만의 문제가 아니라 부동 소수형이 어떻게 구성되고 동작되는가를 정의하는 표준인 IEEE754를 따르는 모든 컴퓨터 시스템의 문제이다.

IEEE754는 제한된 메모리를 이용해서 (파이썬의 경우는 8바이트) 실수를 표현하기 때문에 제한된 정밀도를 가진다.

18. 복소수형 자료 이해하기

복소수

- 실수 : 정수, 유리수, 무리수
- 허수 : i 를 제공하면 -1 이 되는 수

복소수는 실수부와 허수부로 되어있고 허수부는 숫자 뒤에 문자 i 를 이용하는데 파이썬에서는 j 를 사용한다.

Ex)

$C1 = 1 + 7j$

- 1 : 실수부
- $7j$: 허수부

```
c1 = 1 + 7j
print (c1.real)
print (c1.imag)
c2 = complex(2,3)
print (c2)
```

`Print (c1.real)` # 복소수형 자료에서 실수부만 취한다.

`Print (c1.imag)` # 복소수형 자료에서 허수부만 취한다.

`C2 = complex(2,3)` # 실수부가 2이고 허수부가 3인 복소수를 만든다.

문제52. 아래의 수학식을 파이썬으로 구현해서 답을 출력하시오.

$$(1 - 2i)^2 - 2(1-2i) - 12$$

```
c1 = complex(1,-2)
```

```
result = c1**2 - 2*c1 -12
print (result)
```

20.11.26//

2020년 11월 26일 목요일 오전 9:42

19. 대입 연산자 이해하기(=)

변수에 값을 대입하는데 사용되는 기호는 = 이다.

수학에서 = 은 양쪽의 값이 서로 같다는 의미를 가지지만 파이썬을 포함한 컴퓨터 프로그래밍 언어에서는 왼쪽 변수에 오른쪽의 값을 대입하겠다는 의미이다.

Ex)

A = 7

대입 연산자와 일반 연산자를 비교

1. = : 대입연산자
2. == : 같다
3. In : 여러개의 값을 비교

문제53. 주사위 한 개를 10번 던지시오.

```
import random

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

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

    print(result)
```

문제54. 주사위를 10번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오.

```
import random

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

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

print(cnt)
```

문제55. 주사위를 10 번 던져서 주사위의 눈이 짝수가 나오는 횟수를

출력하시오.

```
import random

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

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

print (cnt)
```

문제56. 주사위 2개를 동시에 던져서 두개의 눈의 합이 10이 되는 횟수를 출력하시오. (주사위 2개를 20번 던지시오.)

```
import random

dice1 = [1, 2, 3, 4, 5, 6]
dice2 = [1, 2, 3, 4, 5, 6]
cnt = 0

for i in range (1, 21):
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt = cnt + 1

print (cnt)
```

문제57. 주사위 2개를 동시에 던져서 두 눈의 합이 10이 되는 확률을 구하시오. (주사위 100번 던지시오.)

```
import random

dice1 = [1, 2, 3, 4, 5, 6]
dice2 = [1, 2, 3, 4, 5, 6]
cnt = 0

for i in range (1, 101):
    result1 = random.choice(dice1)
    result2 = random.choice(dice2)
    if result1 + result2 == 10:
        cnt = cnt + 1

print (cnt/100)
```

20. 사칙 연산자 이해하기(, -, *, /, **)

사칙 연산자를 파이썬과 오라클과 비교

| 오라클 | 파이썬 |
|-------------|----------------------|
| + | + |
| - | - |
| * | * |
| / | / |
| Mod | % |
| Power(2, 3) | Power(2, 3), 2**3 |
| Sqrt | Sqrt |
| Log2 | Log2 (밑수가 2인 로그함수) |
| Log10 | Log10 (밑수가 10인 로그함수) |

Ex)

밑수가 10이고 진수가 10인 log 값을 출력하시오.

(log 함수를 사용하려면 math모듈을 import 해야한다.)

Import math # 수학에 관련한 함수들이 코딩되어 있는 모듈 math를 이 코드에서 사용한다.

Print(math.log10(10))

- 모듈이름.함수이름(매개변수값)

문제58. 아래의 수학식을 파이썬으로 구현하시오.

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

```
import math
```

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

21. 연산자 축약 이해하기(=, -=, *=, /=)

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

Ex)

Cnt = cnt + 1 을 다음과 같이 쓸 수 있다.

Cnt += 1

Ex)

Cnt = cnt + 1 == cnt += 1

Cnt = cnt - 1 == cnt -= 1


```
Cnt = cnt * 2 == cnt *= 2
Cnt = cnt / 4 == cnt /= 4
```

문제59. 주사위를 10번 던져서 주사위의 눈이 3이 나오는 횟수를 출력하시오. (축약 연산자를 이용해서 출력하시오.)

```
import random

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

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

print(cnt)
```

22. True와 False 이해하기

참을 나타내는 것이 true 이고 거짓을 나타내는 것이 false 이다.

조건을 판단해서 그 조건이 참이면 true, 거짓이면 false 를 리턴한다.

True는 1, False는 0 값을 가진다. 착과 거짓을 나타낼 때는 True와 False로 표현하면 더 직관적이고 프로그램 코드 가독성을 높일 수 있다.

Ex)

A = True

B = False

Print (1 == 1) # 결과가 True로 출력된다.

Print (1 == 2) # 결과가 False로 출력된다.

23. 관계 연산자 이해하기(==, !=, <, <=, >, >=)

| 오라클 | 파이썬 |
|-----|-----|
| > | > |
| >= | >= |
| < | < |
| <= | <= |
| = | == |
| != | != |
| in | in |

문제60. 주사위를 20번 던져서 주사위의 눈이 4이상 나오는 횟수를 출력하시오.

```
import random

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

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

print (cnt)
```

문제61. 주사위 한개를 288회 던져서 주사위 눈이 5 이상 나오는 횟수를 구하시오.

```
import random

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

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

print (cnt)
```

문제62. 주사위 한개를 288회 던져서 주사위 눈이 5 이상 나오는 확률을 구하시오.

```
import random

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

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

print (cnt/288)
```

문제63. 주사위 한개를 288회 던져서 주사위 눈이 5 이상 나오는 횟수를 출력하는데 이 작업을 100번 해서 횟수가 100개 출력되게 하시

오.

```
import random

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

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

    print (cnt)
```

문제64. 위에서 출력된 횟수 100개를 비어있는 리스트 a리스트에 담고 a 리스트의 갯수를 출력하시오.

```
import random

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

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

print (len(a))
```

위의 코드는 주사위를 288번 던져서 주사위의 눈이 5 이상 나올 횟수 100개를 a 리스트에 입력하는 코드이다.

파이썬은 들여쓰기를 통해서 실행 영역이 결정된다.

문제65. 동전을 100번 던져서 앞면이 나오는 횟수를 출력하시오.

```
import random

coin = ['앞면', '뒷면']
cnt = 0

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

```
print (cnt)
```

문제66. 동전 한개와 주사위 한개를 동시에 100 던져서 동전이 앞면이 나오고 주사위의 눈이 5가 나온 횟수를 출력하시오.

```
import random
```

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

```
coin = ['앞면', '뒷면']
```

```
cnt = 0
```

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

```
    result1 = random.choice(dice)
```

```
    result2 = random.choice(coin)
```

```
    if result1 == 5 and result2 == '앞면':
```

```
        cnt += 1
```

```
print (cnt)
```

문제67. 동전 한개와 주사위 한개를 동시에 100 던져서 동전이 앞면이 나오고 주사위의 눈이 5가 나온 횟수를 출력하는 행위를 50번 해서 횟수가 50개가 출력되게 하시오.

```
import random
```

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

```
coin = ['앞면', '뒷면']
```

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

```
    cnt = 0
```

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

```
        result1 = random.choice(dice)
```

```
        result2 = random.choice(coin)
```

```
        if result1 == 5 and result2 == '앞면':
```

```
            cnt += 1
```

```
    print (cnt)
```

문제68. 위의 횟수 50개를 비어 있는 리스트 a 에 담으시오.

```
import random
```

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

```
coin = ['앞면', '뒷면']
```

```
a = []
```

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

```

cnt = 0
for i in range(1, 101):
    result1 = random.choice(dice)
    result2 = random.choice(coin)
    if result1 == 5 and result2 == '앞면':
        cnt += 1
a.append(cnt)

print(a)

```

문제69. 담겨진 a 리스트의 요소들의 갯수를 확인하시오.

```

import random

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

for j in range(1, 51):
    cnt = 0
    for i in range(1, 101):
        result1 = random.choice(dice)
        result2 = random.choice(coin)
        if result1 == 5 and result2 == '앞면':
            cnt += 1
    a.append(cnt)

print(len(a))

```

문제70. 위의 a 리스트의 요소들의 평균값을 출력하시오.

(numpy 를 이용해서 구현)

```

import random
import numpy as np

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

for j in range(1, 51):
    cnt = 0
    for i in range(1, 101):
        result1 = random.choice(dice)
        result2 = random.choice(coin)
        if result1 == 5 and result2 == '앞면':
            cnt += 1
    a.append(cnt)

print(np.mean(a))

```

Numpy - 딥러닝에 필요한 수학 연산을 쉽게 구현하기 위해서 만들어 놓은 모듈 (특정 목적을

위해 만든 코드의 집합)

Ex)

아래의 리스트의 요소들의 평균값을 출력하시오.(numpy)

Import numpy as np # numpy 모듈을 사용 하는데 numpy의 별칭을 np라고 한다.

```
B = [7, 4, 5, 3, 2]
Print (np.mean(b))
```

Ex)

아래의 리스트의 요소들의 평균값을 출력하시오.(for loop)

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

```
For i in b:
    Cnt += i
Print (cnt/5)
```

```
Print (np.mean(a)) # 평균
Print (np.var(a)) # 분산
Print (np.std(a)) # 표준편차
```

문제71. 문제 64번 코드를 가져와서 a 리스트의 요소들의 평균과 분산과 표준편차를 출력하시오.

```
import random
import numpy as np

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

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

print (np.mean(a))
print (np.var(a))
print (np.std(a))
```

문제72. 위의 a 리스트에 담긴 요소 100개를 출력하시오.

```
import random
```

```

import numpy as np

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

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

print (a)

```

문제73. 위의 a 리스트의 요소들을 하나씩 빼내서 출력하시오.

```

import random
import numpy as np

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

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

for k in a:
    print (k)

```

문제74. a 리스트의 요소들의 숫자가 90이상 이고 106 이하일 횟수를 출력하시오.

```

import random

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

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

cnt1 = 0
for k in a:
    if 90 <= k <= 106:

```

```
cnt1 += 1

print (cnt1)
```

문제75. a 리스트의 요소들의 숫자가 90이상 이고 106 이하일 확률을 출력하시오.

```
import random

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

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

cnt1 = 0
for k in a:
    if 90 <= k <= 106:
        cnt1 += 1

print (cnt1/100)
```

판다스(pandas) 사용법

Pandas란

1. 데이터 분석을 위한 파이썬 모듈
2. 엑셀의 스프레드 시트와 같은 관계형 데이터베이스의 데이터 처리 능력이 뛰어나다.
3. 판다스는 dataframe 이라는 기본 자료구조를 사용한다. (오라클의 테이블과 판다스의 dataframe 이 서로 유사하다.)

Ex)

Emp3.csv를 파이썬을 이용하여 판다스의 데이터 프레임으로 만드시오.

Emp3.csv 와 dept3.csv를 c 드라이브 밑에 data 라는 폴더를 만들고 그 밑에 복사

Import pandas as pd # 판다스 모듈을 현 코드에서 사용

```
Emp = pd.read_csv("c:\data\emp3.csv")
```

```
Print (emp)
```

c 드라이브 밑에 emp3.csv를 읽어서 emp변수에 넣는다.

판다스의 read_csv 함수는 os의 csv파일을 읽어서 파이썬에서 판다스 데이터 프레임으로 만든다.

문제76. 사 테이블에서 이름과 월급을 출력하시오.

(SQL을 판다스 문법으로 변경)

SQL>

```
select ename, sal  
      From emp;
```

Pandas>

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

(데이터프레임명[['컬럼명1', '컬럼명2']][검색조건])

문제77. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, sal, job, hiredate  
      From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

문제78. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, sal  
      From emp  
      Where job = 'SALESMAN';
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

문제79. 월급이 3000 이상인 직원들의 이름과 월급을 출력하시오.

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')

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

문제80. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, sal
From emp
Where sal between 1000 and 3000;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'sal']][emp['sal'].between(1000,3000)])
```

문제81. 월급이 1000에서 3000사이가 아닌 직원들의 이름과 월급을 출력하시오.

(not은 판다스에서 ~ 이다.)

```
Select ename, sal
From emp
Where sal not between 1000 and 3000;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'sal']][~emp['sal'].between(1000, 3000)])
```

오라클 기타 비교 연산자와 파이썬 판다스 비교

| 오라클 | 판다스 |
|----------------|---|
| Between .. and | Emp['sal'].between(1000,3000) |
| In | Emp['job'].isin(['SALESMAN','ANALYST']) |
| Is null | Emp['comm'].isnull() |
| Like | Apply 함수 |

문제82. 직업이 CLERK, SALESMAN 인 직원들의 이름과 직업을 출력하시오.

```
Select ename, job
From emp
Where job in ('CLERK', 'SALESMAN');
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'job']][emp['job'].isin(['CLERK', 'SALESMAN'])])
```

문제83. 이번에는 직업이 CLERK, SALESMAN이 아닌 직원들의 이름과 직업을 출력하시오.

```
Select ename, job
  From emp
 Where job not in ('CLERK', 'SALESMAN');
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'job']][~emp['job'].isin(['CLERK', 'SALESMAN'])])
```

문제84. 커미션이 null 인 직원들의 이름과 커미션을 출력하시오.

```
Select ename, comm
  From emp
 Where comm is null;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'comm']][emp['comm'].isnull()])
```

문제85. 커미션이 null이 아닌 직원들의 이름과 커미션을 출력하시오.

```
Select ename, comm
  From emp
 Where comm is not null;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'comm']][~emp['comm'].isnull()])
```

문제86. 6개의 제품이 들어있는 상자가 있는데, 그 중에 2개가 불량품이라고 하자, 제품검사를 위해 서 3개를 추출 했을 때 적어도 한개가 불량품일 확률을 구하시오.

아래와 같이 상자 안에 제품 6개 넣고 랜덤으로 추출해서 수행하시오. (복원 추출)

Box = ['정상', '정상', '불량', '정상', '불량', '정상']

사건 a 의 여사건(나머지 사건)의 확률 : $1 - P(A)$

불량품이 하나도 없을 사건의 확률을 구한 다음에 1에서 빼면 된다.

```
import random
box = ['정상', '정상', '불량', '정상', '불량', '정상']
cnt = 0

for i in range(10000):
    result1 = random.choice(box)
    result2 = random.choice(box)
    result3 = random.choice(box)
    if result1 == '불량' or result2 == '불량' or result3 == '불량':
        cnt += 1

print (cnt/10000)
```

20.11.30

2020년 11월 30일 월요일 오전 9:42

1. 파이썬 기본 자료형 5가지

- a. 문자형 : a = 'scott'
- b. 숫자형 : cnt = 0
- c. 리스트형 : a = [1, 2, 3]
- d. 튜플 : b = (1, 2, 3)
- e. 딕셔너리 : c = {'apple':'사과','eye':'눈'}

2. 파이썬 연산자 : >, <, >=, <=, ==, !=

3. If문

If 조건:

실행문

4. Loop문 : 반복적으로 특정 실행문을 실행해야 할 때 loop문을 사용

a. For loop문

```
For i in range(1,11):  
    Print('a')
```

b. While loop문

While 조건:
실행문

5. 판다스 사용법 : 엑셀과 같은 데이터를 검색할 때 주로 사용

문제86. 이름이 scott인 사원의 이름과 월급을 출력하시오.

```
Select ename, sal  
From emp  
Where ename = 'SCOTT'
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

26. 시퀀스 자료형 이해하기

시퀀스 자료형은 어떤 객체가 순서를 가지고 나열되어 있는 것을 말한다.

예를 들어 문자열 'abcd' 는 문자 'a', 'b', 'c', 'd'가 순서를 가지고 차례대로 나열되어 있는 것이다.

Ex)

A = 'SCOTT'

Print (a) # SCOTT

위의 SCOTT라는 문자를 담은 a 변수에서 첫번째 요소만 출력하고 싶다면

Print (a[0]) # S

Print (a[1]) # C

문제87. 위의 SCOTT을 담은 문자형 변수 a 에서 알파벳 O를 출력하시오.

```
a = 'scott'
```

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

문자형 변수에서 특정 철자만 가져오고 싶으면 문자형변수 옆에 대괄호를 쓰고 순서에 해당하는 숫자를 적어주면 된다.

문제88. 아래의 문자형 변수에서 맨 끝의 철자인 H를 출력하시오.

B = 'SMITH'

```
b = 'smith'
```

```
print (b[-1])
```

문제89. 판다스를 이용해서 emp3.csv에서 이름을 출력하는데 이름의 첫 번째 철자만 출력하시오.

```
Select substr(ename, 1, 1)
  From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

```
for i in emp['ename']:
    print (i[0])
```

<class 'str'> 문자형으로 변경해줘야 문자형에서 특정 요소를 가져오는 문법을 사용할 수 있다.
판다스의 <class 'pandas.core.series.Series'> 를 문자형으로 변경하려면 for loop문을 이용해서 하나씩 가져오면 된다.

Ex)
For i in emp['ename']:
 Print (i)

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

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

for i in emp['ename']:
    print (i[-1])
```

문제91. 위의 결과에서 앞에 이름도 같이 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

for i in emp['ename']:
    print (i, i[-1])
```

문제92. 이름의 첫번째 철자가 S로 시작하는 직원들의 이름을 출력하시오.

```
Select ename
From emp
Where ename like 'S%';
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

for i in emp['ename']:
    if i[0] == 'S':
        print (i)
```

문제93. 이름의 끝글자가 T로 끝나는 직원들의 이름을 출력하시오.

```
Select ename
From emp
Where ename like '%T';
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
```

```
for i in emp['ename']:
    if i[-1] == 'T':
        print (i)
```

27. 시퀀스 자료 인덱싱 이해하기

인덱싱(indexing)이란 시퀀스 자료형에서 인덱스를 통해 해당하는 값을 얻는 방법이다.
파이썬에서는 인덱스를 0부터 시작하며 음수인 인덱스도 사용가능하다.
음수 인덱스는 -1 부터 시작한다.

Ex)
A = 'SALEMAN'
Print (a[2])

문제94. 이름의 두번째 철자가 M인 사원의 이름을 출력하시오.

```
Select ename
From emp
Where ename like '_M%';
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

for i in emp['ename']:
    if i[1] == 'M':
        print (i)
```

28. 시퀀스 자료 슬라이싱 이해하기

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

Ex)
A = 'SCOTT'
Print (a[1:3])

1번째 자리부터 3미만의 자리까지 요소들을 출력한다.

문제95. 아래의 SQL을 판다스로 구현하시오.

```
Select substr(ename, 1, 3)
From emp;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
```



```
for i in emp['ename']:
    print (i[0:3])
```

문제96. 아래의 SQL을 판다스로 구현하시오.

```
Select substr(ename, -2, 2)
From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

```
for i in emp['ename']:
    print (i[-2:])
```

문제97. 아래의 SQL을 판다스로 구현하시오.

```
Select substr(ename, 2, 3)
From emp;
```

29. 시퀀스 자료 연결 이해하기()

문제98. 아래의 두개의 리스트를 연결하시오.

```
A = [2, 3, 4, 5]
B = [9, 2, 4, 8]
```

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

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

문제99. 아래의 SQL을 판다스로 구현하시오.

```
Select ename || sal
From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

Zip함수를 for loop문에 사용하게 되면 두개의 범위 데이터를 한번에 받아서 loop를 수행할 수 있다.

문자형 + 문자형은 연결이 가능한데 문자형 + 숫자형은 연결이 안되기 때문에 숫자형을 문자형

으로 변환하기 위해서 str 함수를 사용한다.

30. 시퀀스 자료 반복 이해하기(*)

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

Ex)

Print ('a' * 7) # 알파벳 a 를 7번 출력한다.

결과 : aaaaaaa

Print ([1,2,3] * 5) # 리스트 [1,2,3] 안의 요소를 5개로 늘려서 출력한다.

결과 : [1,2,3,1,2,3,1,2,3,1,2,3,1,2,3]

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

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

```
dice100 = dice * 100
```

```
print (dice100)
```

문제101. 초등학생 10명의 키가 들어가 있는 아래의 tall 리스트의 요소 10개를 10000개로 증가시켜서 tall10000 리스트에 담고 출력하시오.

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

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

```
tall10000 = tall * 10000
```

```
print (tall10000)
```

문제102. 위의 모집단의 평균값, 분산, 표준편차를 출력하시오.

Np.mean(리스트) 를 사용하면 평균값이 출력된다.

```
import numpy as np
```

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

```
tall10000 = tall * 10000
```

```
print (np.mean(tall10000))
```

```
print (np.var(tall10000))
```

```
print (np.std(tall10000))
```

문제103. 위의 모집단 tall10000에서 표본을 20개를 랜덤으로 추출하시오.

```
import numpy as np
import random

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000

for i in range(20):
    result = random.choice(tall10000)
    print (result)
```

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

```
import numpy as np
import random

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
a = []

for i in range(20):
    result = random.choice(tall10000)
    a.append (result)
print (a)
```

문제105. 위의 표본의 평균값을 출력하시오.

```
import numpy as np
import random

tall = [129.3, 130.2, 132.5, 134.7, 136.3, 137.8, 138.1, 140.2, 142.3, 145.2]
tall10000 = tall * 10000
a = []

for i in range(20):
    result = random.choice(tall10000)
    a.append (result)
print (np.mean(a))
```

31. 시퀀스 자료 크기 이해하기(len)

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

Ex)
A = 'scott'

```
Print (len(a)) # 5
```

```
B = [2,3,4,8,1]
```

```
Print (len(b)) # 리스트의 요소의 갯수가 출력된다.
```

문제106. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, length(ename)  
From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

32. 멤버체크 이해하기(in)

In은 자료에 어떤 값이 있는지 없는지 확인할 때 사용하는 키워드이다.
활용하는 방법은 다음과 같다.

Ex)

값 in 자료

값이 자료에 있으면 True이고 없으면 False가 된다.

Ex)

```
Listdata = [1, 2, 3, 4]
```

```
A = 5 in listdata
```

```
B = 4 in listdata
```

```
Print (a) # False
```

```
Print (b) # True
```

문제107. 모평균이 30이고 모표준편차가 7인 모집단을 구성하시오.

Np.random.randn(숫자) 를 쓰면 이 숫자만큼 가우시안 정규분포에 따르는 난수들이 숫자만큼 생성된다.

```
import numpy as np
```

```
avg = 30
```

```
std = 7
```

```
N = 1000000
```

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

```
print (mo)
```

문제108. 위의 모집단의 모평균을 출력하시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg
print (np.mean(mo))
```

문제109. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, job
      From emp
      Where job in ('SALESMAN', 'ANALYST');

import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

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

문제110. 아래의 SQL을 판다스로 구현하시오.

```
Select ename, job
      From emp
      Where job not in ('SALESMAN','ANALYST');

import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'job']][~emp['job'].isin(['SALESMAN','ANALYST'])])
```

33. 문자열 이해하기

문자열은 문자나 기호 순서로 나열되어 있는 시퀀스 자료 이다.

문자열을 선언하는 방법 3가지

1. '문자열' : 싱글 쿼테이션 마크
2. "문자열" : 더블 쿼테이션 마크
3. """문자열""" : 더블 쿼테이션 마크 3개

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

```
A = "My son's name is John"
```

Print (a)

문자에 아래와 같이 더블 쿼테이션 마크(")가 포함되어져 있으면 위의 세번째인 더블 쿼테이션 마르를 3개 사용하면 되나.

```
B = """"My son's name is "John""""
```

Print (b)

문제111. 아래와 같은 글씨가 출력되게 하시오.

"모집단"의 모평균은

"모집단"의 모분산은

"모집단"의 모표준편차는

```
a = "" "모집단"의 모평균은 ""
```

```
b = "" "모집단"의 모분산은 ""
```

```
c = "" "모집단"의 모표준편차는 ""
```

```
print (a)
```

```
print (b)
```

```
print (c)
```

문제112. 모평균이 30이고 모표준편차가 7인 모집단 1000000개의 모평균과 모분산과 모표준편차를 아래와 같이 출력하시오.

"모집단"의 모평균은 29.990612681947418

"모집단"의 모분산은 48.934711270867325

"모집단"의 모표준편차는 6.995334964879618

```
import numpy as np
```

```
a = "" "모집단"의 모평균은 ""
```

```
b = "" "모집단"의 모분산은 ""
```

```
c = "" "모집단"의 모표준편차는 ""
```

```
avg = 30
```

```
std = 7
```

```
N = 1000000
```

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

```
print (a, np.mean(mo))
```

```
print (b, np.var(mo))
```

```
print (c, np.std(mo))
```

문제113. 모평균이 30이고 모표준편차가 7인 모집단(1000000개)에서 표본 49를 뽑으시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg

print (np.random.choice(mo, 49))
```

문제114. 위에서 뽑은 49개의 평균값을 출력하시오.

Numpy의 random 안의 choice 함수를 이용해서 모집단에서 49개를 표본샘플링 하고 그 표본의 평균값을 출력한다.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg

print (np.random.choice(mo, 49).mean())
```

문제115. 위에서 출력한 표본의 평균값을 하나가 아니라 100개가 출력되게 하시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg

for i in range(100):
    print (np.random.choice(mo, 49).mean())
```

문제116. 위의 결과를 출력하지 않고 비어있는 리스트 a에 담으시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg
a = []

for i in range(100):
    a.append (np.random.choice(mo, 49).mean())
print (len(a))
```

문제117. 위에서 구한 표본평균의 평균값과 표준편차를 아래와 같이 출력하시오.

"표본평균"의 평균값은 30.052069278313194

"표본평균"의 표준편차는 0.9452522700044904

```
import numpy as np
```

```
a = "" "모집단"의 모평균은 ""
```

```
b = "" "모집단"의 모표준편차는 ""
```

```
avg = 30
```

```
std = 7
```

```
N = 1000000
```

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

```
z = []
```

```
for i in range(100):
```

```
    z.append(np.random.choice(mo, 49).mean())
```

```
print(a, np.mean(z))
```

```
print(b, np.std(z))
```

34. 문자열 포매팅 이해하기

문자열 포매팅이란 변하는 값을 포함하는 문자열을 표현하기 위해 하나의 양식으로 문자열을 만드는 것이다.

문자열 포매팅에서는 변하는 값을 나타내기 위해 사용하는 기호를 '포맷 문자열' 이라고 하며, 자주 사용하는 포맷 문자열은 다음과 같다.

| 포맷 문자열 | 설명 |
|--------|---------------|
| %s | 문자열에 대응 |
| %c | 문자나 기호 한개에 대응 |
| %f | 실수에 대응 |
| %d | 정수에 대응 |
| %% | %라는 기호 자체를 표시 |

Ex)

```
Txt1 = '자바'
```

```
Txt2 = '파이썬'
```

```
Num1 = 5
```

```
Num2 = 10
```

```
Print('나는' %s 보다 %s 에 더 익숙합니다' %(txt1, txt2))
```


문제118. 아래의 변수를 이용해서 아래와 같이 결과가 출력되게 하시오.

```
Num1 = 5
Num2 = 10
```

5는 10보다 작습니다.

```
num1 = 5
num2 = 10

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

문제119. 문제 117번의 결과가 아래와 같이 출력되게 하시오.

표본평균의 평균값은 30.052069278313194 이고 분산값은 29.990612681947418 이고 표준편차는 0.9452522700044904 입니다.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg
z = []

for i in range(100):
    z.append(np.random.choice(mo, 49).mean())

print('표본평균의 평균값은 %f 이고 분산값은 %f 이고 표준편차는 %f 입니다.'%(np.mean(z),
np.var(z), np.std(z)))
```

문제120. 위의 결과가 소수점 두번째 자리까지만 나오게 하시오.

```
import numpy as np

avg = 30
std = 7
N = 1000000
mo = np.random.randn(N) * std + avg
z = []

for i in range(100):
    z.append(np.random.choice(mo, 49).mean())

print('표본평균의 평균값은 %.2f 이고 분산값은 %.2f 이고 표준편차는 %.2f 입니다.'
%(round(np.mean(z),2), round(np.var(z),2), round(np.std(z),2)))
```

문제121. <http://cafe.daum.net/oracleoracle/SfWb/235>

```
import numpy as np

mo = np.random.randn(1000000) * 3 + 18
cnt = 0

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

20.12.01

2020년 12월 1일 화요일 오전 9:35

복습

1. 파이썬의 자료형 5가지

- a. 문자형
- b. 숫자형
- c. 리스트형
- d. 튜플형
- e. 사전형

2. 파이썬의 연산자 : <, >, <=, >=, ==, !=

3. 파이썬에서의 들여쓰기

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

```
For i in range (1, 11):
```

```
    Print ('a') # for loop문의 관할하에 있는 실행영역
```

```
Print ('b') # for loop문과 상관없는 별도의 독립적인 실행영역
```

4. If문

```
If 조건:
```

```
    실행코드
```

```
Elif 조건:
```

```
    실행코드
```

```
Elif 조건:
```

```
    실행코드
```

```
Else:
```

```
    실행코드
```

5. 반복문 : for loop문, while loop문

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

7. 문자열 변수에서 특정 철자를 취하는 방법 (인덱싱)

```
A = 'SCOTT'
```

```
Print (a[0]) # S
```

8. 문자열 변수에서 특정 철자들을 취하는 방법 (슬라이싱)

```
A = 'SCOTT'
```

```
Print (a[0:2]) # SC
```

9. 문자열 변수의 길이 확인하는 방법 (len)

```
A = 'SCOTT'
```

```
Print (len(a)) # 5
```

10. 멤버 체크하는 방법(in)

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

```
B = 3 in a
```

```
Print (b) # True
```

11. 문자열 포맷 이해하기

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

35. 이스케이프 문자 이해하기

이스케이프 문자는 키보드로 입력하기 어려운 기호를 나타내기 위해 역슬래시 '₩' 로 시작하는 문자

파이썬에서 자주 사용되는 이스케이프 문자는 다음과 같다.

| 이스케이프 문자 | 설명 |
|----------|--------------|
| ₩n | 줄바꾸기 |
| ₩t | 탭 |
| ₩enter | 줄 계속 |
| ₩₩ | ₩ 기호 |
| ₩' 또는 ₩" | ' 기호 또는 " 기호 |

Ex)

```
Print ('나는 파이썬을 배웁니다. ₩n 파이썬은 자바보다 ₩  
훨씬 쉽습니다.')
```

36. 리스트 이해하기([])

리스트는 파이썬에서 가장 많이 활용되는 시퀀스 자료형 중 하나이다.

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

Ex)

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

```
Print (k)
```

```
Print (type(k))
```

문제122. 동전의 앞면과 뒷면을 포함하는 리스트 coin을 만드시오.

```
coin = ['앞면', '뒷면']
```

문제123. 위에서 만든 coin의 요소를 10000개로 늘려서 coin_10000 번

수에 담으시오.

```
coin = ['앞면', '뒷면']
coin_10000 = coin * 10000

print (coin_10000)
```

문제124. 위에서 만든 coin_10000 리스트에서 표본을 10개 추출하시오.

```
import numpy as np

coin = ['앞면', '뒷면']
coin_10000 = coin * 10000

print (np.random.choice(coin_10000, 10))
```

Np.random.choice은 numpy 모듈안에 random 코드안에 choice 함수를 사용한다.

Np.random.choice(모집단 리스트, 샘플 갯수)

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

```
import numpy as np

coin = ['앞면', '뒷면']
coin_10000 = coin * 10000
result = np.random.choice(coin_10000, 10)
cnt = 0

for i in result:
    if i == '앞면':
        cnt += 1
print (cnt)
```

37. 튜플 이해하기(())

튜플은 리스트와 비슷한 성질을 가지고있는 자료형이지만 요소의 값을 변경할 수 없다는 특징이 있다.

리스트는 대괄호 []로 요소들을 감싸는데 튜플은 소괄호 ()로 요소들을 감싼다.

튜플은 데이터가 변경이 안되므로 튜플로 만든 데이터에 대한 신뢰도가 높다.

Ex)

리스트에서 특정 요소를 변경하는 방법

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

```
Print (type(a)) # <class 'list'>
```

```
Print (a[0]) # 1
```

```
A[0] = 7 # a리스트의 0번째 요소를 7로 변경한다.
```

```
Print (a) # [7, 2, 3, 4, 5]
```

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

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

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

```
print (a)
```

Ex)

튜플은 데이터를 변경할 수 없다.

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

```
B[0] = 7
```

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

위와 같은 에러가 나면서 실행이 불가능하다.

변경이 되어서는 안되는 데이터는 프로그래밍할 때 튜플로 만들어서 관리한다.

Ex)

포인트 적립시 구매금액의 0.01% 적립을 해준다고 하면 0.01은 절대 변경이 어서는 안되는 데이터 이므로 튜플로 관리해야 한다.

문제127. 아래의 숫자데이터들을 튜플로 생성하시오.

튜플 변수 이름 : point

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

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

```
print (type(point)) # <class 'tuple'>
```

문제128. 위의 튜플 point의 요소중 0.03만 뽑아서 출력하시오.

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

```
print (point[2])
```

문제129. 위의 튜플의 모든 요소를 출력하시오.

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

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

38. 사전 이해하기({ })

사전은 키와 값을 하나의 요소로 하는 순서가 없는 집합이다.

그러므로 사전은 시퀀스 자료형이 아니며 인덱싱으로 값을 접근 할 수도 없다.

사전은 '키:값' 쌍이 하나의 요소이다.

Ex)

```
A = {'apple':'사과', 'banana':'바나나','peach':'복숭아','pear':'배'}
```

```
# 'apple' : 키
```

```
# '사과' : 값
```

```
Print (type(a)) # <class 'dict'>
```

```
Print (a.keys()) # 키값들만 출력
```

```
Print (a.values()) # 값들만 출력
```

```
A['grape'] = '포도' # 새로운 키와 값을 추가하는 방법
```

```
Print (a)
```

Ex)

```
B = {} #비어있는 딕셔너리를 생성
```

```
B['apple'] = '사과'
```

```
B['pear'] = '배'
```

```
B['grape'] = '포도'
```

```
Print (b)
```

문제130. 아래의 두개의 리스트를 각각 만들고 아래와 같이 fruit 라는 딕셔너리를 생성하시오.

```
A = ['사과','배','포도','복숭아','바나나']
```

```
B = ['apple','pear','grape','peach','banana']
```

```
a = ['사과', '배', '포도', '복숭아', '바나나']
```

```
b = ['apple', 'pear', 'grape', 'peach', 'banana']
```

```
fruit = {}
```

```
for i, k in zip(a,b):
```

```
    fruit[i] = k
```

```
print (fruit)
```

결과 :

```
print (fruit)
```

```
{'사과':'apple','배':'pear', ... , '바나나':'banana'}
```

39. 함수 이해하기(def)

```
Select max(sal)
```

```
From emp;
```

```
Select substr('smith', 1, 3)
```

```
From dual;
```

오라클 에서는 오라클 개발자가 만들어준 함수를 사용했지만 파이썬에서는 우리가 직접 함수를 생성한다.

각국의 개발자들이 파이썬 함수를 직접 만들어서 공유해주는데 우리는 import 명령어로 불러와서 만들어져 있는 함수를 사용 할 수 있다.

| 오라클 함수 | 파이썬 함수 |
|-----------|---------------------|
| Upper | Upper() |
| Lower | Lower() |
| Initcap | |
| Substr | |
| Replace | Replace() |
| Length | Len() |
| Rtrim | Rstrip() |
| Ltrim | Lstrip() |
| Rpad | |
| Lpad | |
| Instr | |
| Round | Round() |
| Trunc | Trunc() |
| Mod | % |
| Power | Pow |
| To_char | Str |
| To_number | Int, float |
| To_date | Datetime.strptime() |
| Nvl | |
| Decode | If 문 |

| | |
|------|------|
| Case | If 문 |
|------|------|

Ex)

아래의 SQL을 파이썬으로 구현하시오.

```
Select lower(ename)
From emp;
```

```
import pandas as pd
```

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

문제131. 아래의 SQL을 파이썬으로 구현하시오.

```
Select lower(ename), lower(job)
From emp;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
a = emp['ename']
b = emp['job']
```

```
for i, j in zip (a, b):
    print (i.lower(), j.lower())
```

- 함수를 생성하는 방법

Def 함수이름(매개변수):

실행문

Return 출력값이 있는 변수명

Ex)

#두개의 입력값을 받아서 더하는 함수를 생성

```
def add_number(n1, n2):
```

result = n1 + n2 # n1에 입력된 값과 n2에 입력된 값을 더해서 result에 입력한다.

return result # result에 입력된 값을 출력한다.

위에서 만든 함수를 실행

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

콜론(:) 으로 끝을 맺는 4가지

1. If 문 종료시

If 조건:

2. For 문 종료시

- ```
For i in range(1, 11):
```
3. 함수 생성시

```
Def 함수이름(입력 매개변수):
```
  4. 클래스 생성시

```
Class 클래스이름:
```

문제132. 1부터 10까지 출력하시오.

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

문제133. 1부터 10까지 다 더한 값을 출력하시오.

```
cnt = 0

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

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

```
Print (all_add(10))

55

def all_add(num):
 cnt = 0
 for i in range(1, num+1):
 cnt += i
 return cnt

print (all_add(10))
```

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

```
A = 'SCOTT'

a = 'SCOTT'

print (a.lower())
```

문제136. 아래의 문자열을 대문자로 출력하시오.

```
B = 'scott'
```

```
b = 'scott'
```

```
print (b.upper())
```

문제137. 아래의 문자열에서 첫번째 철자만 출력하시오.

```
A = 'scott'
```

```
a = 'scott'
```

```
print (a[0])
```

문제138. 위에서 출력한 첫번째 철자를 대문자로 출력하시오.

```
a = 'scott'
```

```
print (a[0].upper())
```

문제139. 아래의 문자열 변수에서 cott만 출력하시오.

```
A = 'scott'
```

```
a = 'scott'
```

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

문제140. 아래의 함수를 생성하시오.

```
Print (initcap('scott'))
```

```
Scott
```

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

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

## 40. 함수 인자 이해하기

인자란 위에서 사용한 입력 매개변수를 말한다.

Ex)

```
Def add_text(t1, t2):
 Return t1 + ' ' + t2
```

```
Print (add_text('파이썬', '자바'))
```

```
파이썬 자바
```

- 매개변수에 아무것도 입력하지 않고 실행하면 기본값이 출력되게 함수를 생성하는 방법

```
Def add_text(t1, t2 = '최고'):
```

```
 Return t1 + ' ' + t2
```

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

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

T2에 값을 아무것도 넣지 않으면 기본값으로 지정한 '최고' 가 출력된다.

## 41. 지역변수와 전역변수 이해하기(global)

변수는 자신이 생성된 범위(코드블럭) 안에서만 유효하다.

함수 안에서 만든 변수는 함수 안에서 살아있다가 함수 코드의 실행이 종료되면 그 생명을 다한다. 이것을 '**지역변수**' 라고 한다.

이와 반대로 함수 외부에서 만든 변수는 프로그램이 살아있는 동안에 함께 살아있다가 프로그램이 종료되면 같이 소멸된다.

이렇게 프로그램 전체를 유효범위로 가지는 변수를 '**전역변수**'라고 한다.

Ex)

```
Strdata = '전역변수' # func1 함수 외부에 있는 변수
```

```
Def func1():
```

```
 Strdata2 = '지역변수' # func1 함수 내부에 있는 변수
```

```
 Return strdata2
```

```
Print (func1())
```

대부분 많은 코드들이 지역변수를 사용하는데

프로그램 전체에서 공통적으로 사용되고 잘 변하지 않는 데이터는 전역변수를 사용한다.

Ex)

```
Pi = 3.141592653589793
```

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

```
 Global pi # global이라는 키워드를 사용하면 전역변수 pi를 함수내부로 가져올 수 있다.
```

```
 Return pi * num1 * num1
```

```
Def cycle_func2(num1): # 원의 1/4인 부채꼴의 넓이를 구하는 함수
 Global pi
 Return 1/4 * pi * num1 * num1

Print (cycle_func1(5))
Print (cycle_func2(5))
```

## 42. 함수 리턴값 이해하기(return)

모든 함수는 이름을 갖고 있는데 이 이름을 불러주면 파이썬은 그 이름 아래에 정의되어있는 코드를 실행한다.

이때 함수를 부르는 코드를 호출자라고 한다.

함수가 호출자에게 결과를 돌려주는것을 반환(return) 이라고 한다.

```
Pi = 3.141592653589793
```

```
Def cycle_func1(num1):
 Global pi
 Return pi * num1 * num1
```

```
Print (cycle_func1(5)) # 함수를 호출하는 호출자 코드
```

- 절대값을 출력하는 함수

```
Print (abs(-9)) # 9
Print (abs(9)) # 9
```

문제141. abs함수를 사용하지 않고 if 문을 이용하여 절대값을 출력하는 my\_abs라는 함수를 생성하시오.

```
def my_abs(num):
 if num >= 0:
 return num
 else:
 return -num
```

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

## 43. 파이썬 모듈 이해하기

파이썬에서는 각각의 소스파일을 일컬어 모듈이라고 한다.

이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게 하나의 파일로 묶어 놓은것을 모듈이라고 한다.

외부의 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로 가져와서 자유롭게 사용할 수 있도록 해야 하는데 이런일을 파이썬에서는 모듈을 import 한다라고 한다.

Ex)

Import time # time 모듈을 임포트 한다

Print ('5초간 프로그램을 정지합니다.')

Time.sleep(5)

Print ('5초가 지났습니다.')

Ex)

Import numpy as np

Avg = 30

Std = 5

N = 1000000

Mo = np.random.randn(N) \* std + avg

Print (mo)

문제142. 서울시 초등학생 백만명의 키를 모집단을 구성하는데 평균 키가 148.5이고 표준편차가 30인 모집단을 만드시오.

```
import numpy as np
```

```
avg = 148.5
```

```
std = 30
```

```
N = 100000
```

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

```
print (mo)
```

문제143. 위의 모집단에서 100명을 표본으로 추출하여 100명의 평균 키를 비어있는 리스트 a 에 입력하는 작업을 10000번 수행하여 a리스트에 10000개의 표본의 평균키가 입력되게 하시오.

```
import numpy as np
```

```
avg = 148.5
```

```
std = 30
```

```
N = 100000
```

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

```
a = []
```

```
for i in range(10000):
```

```
 a.append (np.random.choice(mo,100).mean())
```

```
print (a)
```

문제144. 통계를 전문으로 구현하는 모듈인 scipy 모듈을 임포트하여 위의 표본의 평균기 값에 대한 확률밀도값을 출력하시오.

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

avg = 148.5
std = 30
N = 1000000
a = []

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

for i in range (10000):
 a.append(np.random.choice(mo, 100).mean())

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

문제145. 데이터 시각화 전문 모듈인 matplotlib를 임포트 하여 위의 표본 평균값 10000개의 데이터를 시각화 하시오.

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt # matplotlib 모듈안에 pyplot라는 함수를 임포트

avg = 148.5
std = 30
N = 1000000
a = []

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

for i in range (10000):
 a.append(np.random.choice(mo, 100).mean())

x = np.arange(140, 160, 0.001)
y = norm.pdf(x, np.mean(a), np.std(a))
plt.plot(x,y,color = 'blue')
plt.show()
```

문제146. 위의 확률밀도함수 그래프의 아래쪽 영역도 색깔로 채우시오

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

```

import matplotlib.pyplot as plt

avg = 148.5
std = 30
N = 1000000
a = []

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

for i in range(10000):
 a.append(np.random.choice(mo, 100).mean())

x = np.arange(140, 160, 0.001)
y = norm.pdf(x, np.mean(a), np.std(a))
plt.plot(x, y, color = 'blue')
plt.fill_between(x, y, interpolate = True, color = 'skyblue', alpha = 0.7)
plt 모듈안의 fill_between 함수를 이용해서 확률밀도함수 그래프의 아래 영역을 색깔로
채우는데 interpolate = True 를 이용해서 아래의 영역이 색깔로 채워지게 되고,
alpha는 색깔 투명도인데 0.0 ~ 1.0 사이로 기술 할 수 있다.
plt.show()

```

문제147. 위에서 그린 확률밀도함수 그래프의 색깔을 변경하시오.

```

import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

avg = 148.5
std = 30
N = 1000000
a = []

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

for i in range(10000):
 a.append(np.random.choice(mo, 100).mean())

x = np.arange(140, 160, 0.001)
y = norm.pdf(x, np.mean(a), np.std(a))
plt.plot(x, y, color = 'black')
plt.fill_between(x, y, interpolate = True, color = 'gray', alpha = 0.5)
plt.show()

```



# 20.12.02

2020년 12월 2일 수요일 오전 9:35

직접 모듈 생성하기

지금 현재 작성한 코드가 어느 디렉토리에 어느 이름으로 저장되어있는지 확인 하는 방법

```
Def add_number(n1, n2):
 Result = n1 + n2
 Return n1 + n2
```

```
Print (add_number(1, 2))
```

C:\Users\WKJM\spyder-py3\temp.py

Temp.py - 파일명

.spyder-py3 - .이 붙어 있으면 숨김폴더이다.

```
Def add_number(n1, n2):
 Result = n1 + n2
 Return n1 + n2
```

위의 함수를 C:\Users\WKJM\ 밑에 my\_cal.py 라는 이름으로 저장

새로운 창에서 아래와 같이 임포트 한다.

```
Import my_cal # my_cal모듈을 임포트한다.
```

```
Print (my_cal.add_numb12(1,2)) # my_cal 모듈안에 add_number 함수를 실행
```

새로운 창에서는 def로 add\_number 함수를 만드는 코드는 없지만 my\_cal.py 에 있는 모듈을 새로운 창에서 사용할 수 있도록 import 했다.

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

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

문제149. 다른 새로운 창에서 my\_cal 모듈을 임포트 하고

gob\_number 함수를 실행하시오.

```
import my_cal
```

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

통계문제를 풀기 위해 사용한 import random에서 random은 여러개의 함수를 모아놓은 소스 코드이고 어딘가에 random.py라는 이름으로 저장되어 있는 것을 우리가 import한 것이다.

문제150. 나누기를 하는 함수를 my\_cal.py에 저장하고 다른 새로운 창에서 아래와 같이 import하고 실행될 수 있도록 하시오.

```
import my_cal
```

```
print (my_cal.devide_number(10,2))
```

## 44. 파이썬 패키지 이해하기

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

패키지 (폴더)

모듈 (스크립트)

파이썬 패키지를 만드는 단계

1. 아래의 디렉토리에 my\_loc라는 폴더를 생성한다.
  - C:\Users\WKJM\my\_loc
2. My\_loc 폴더 안에 my\_cal.py를 옮긴다.
3. My\_loc 폴더가 패키지로 인정을 받기 위해서는 반드시 \_\_init\_\_.py라는 파일을 가지고 있어야 한다.
4. 새로운 창에서 아래와 같이 스크립트를 수행한다.
  - From my\_loc import my\_cal # from 패키지 import 모듈

Print (my\_cal.add\_number(1,2)) # my\_cal 모듈안에 있는 add\_number 함수를 실행

Ex)

From scipy.stats import norm # scipy 패키지안의 stats라는 패키지에 norm 모듈을 임포트

Print (norm.pdf(x, 평균, 표준편차)) # norm 모듈에 pdf (확률밀도함수)를 실행

## 45. 파이썬 모듈 импорт 이해하기 ① (import)

이미 만들어져 있는 어떤 함수를 우리가 작성하는 코드에서 자유롭게 활용할 수 있으려면 해당 함수가 포함된 모듈을 импорт 해야한다.

임포트 하는 방법은 다음과 같다.

```
Import 모듈이름
From 패키지 import 모듈이름
Import 패키지이름.모듈이름
```

Ex) 아래와 같이 우리가 만든 모듈이 아닌 다른 사람이 만든 모듈을 импорт 해서 사용했는데 이 모듈은 어디에 있는지 확인

```
Import pandas
Import numpy
```

위와 같이 패키지 이름을 주지 않고 모듈만 импорт 했는데 실행이 되었다. 위와 같은 모듈은 파이썬 내장 모듈, sys.path에 정의되어 있는 모듈이다.

Import를 만나면 파이썬 모듈을 찾는 순서

1. 파이썬 내장 모듈에 있는지 확인
2. Sys.path에 정의되어 있는 디렉토리를 확인

파이썬 내장 모듈이 무엇이 있는지 확인하는 방법

```
Import sys
Print (sys.builtin_module_names)
```

Sys.path에 정의된 디렉토리가 무엇인지 확인하는 방법

```
Import sys
For i in sys.path:
 Print (i)
```

Site-packages란 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.

아래의 명령어가 수행되려면 스크립트가 C:\Users\WKJM 밑에 있어야 한다. 왜냐하면 my\_loc폴

더가 C:\Users\WKJM 밑에 있기 때문이다.

```
From my_loc import my_cal
```

C:\Users\WKJM 가 아니더라도 다른 디렉토리에서라도

From my\_loc import my\_cal 명령어를 자유롭게 실행하려면

C:\Users\KJM\anaconda3\lib\site-packages

밑에 있으면 된다.

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

## Numpy 모듈 이해하기

Numpy 모듈이란 python 언어에서 기본적으로 지원하지 않는 배열(array) 또는 행렬(matrix)의 계산을 쉽게 해주는 라이브러리이다.

딥러닝에서 많이 사용하는 선형대 수학에 관련된 수식들을 python에서 쉽게 프로그래밍 할 수 있게 해준다.

Ex)

아래의 행렬을 만드시오.

1 2

4 7

1. 리스트로만 했을 때

```
A = [[1,2],[4,7]]
```

```
Print (a)
```

2. Numpy array로 했을 때

```
Import numpy as np
```

```
A = [[1,2],[4,7]]
```

```
A2 = np.array(a)
```

```
Print (a2)
```

문제152. 아래의 행렬의 합을 출력하시오.

[1 2    3 1  
4 5    6 2]

```
import numpy as np
```

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

```
b = [[3,1],[6,2]]
```

```
a2 = np.array(a)
```

```
b2 = np.array(b)
```

```
print (a2 + b2)
```

문제153. 아래의 행렬의 합을 출력하시오.

$$\begin{bmatrix} 6 & 3 & 4 \\ 5 & 1 & 7 \end{bmatrix} + \begin{bmatrix} 4 & 5 & 7 \\ 9 & 20 & 4 \end{bmatrix}$$

```
import numpy as np
```

```
a = [[6,3,4],[5,1,7]]
b = [[4,5,7],[9,20,4]]
```

```
a2 = np.array(a)
b2 = np.array(b)
```

```
print (a2 + b2)
```

문제154. 아래의 행렬의 곱을 출력하시오.

$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$$
$$\begin{bmatrix} (1 \times 2 + 2 \times 3) & (1 \times 1 + 2 \times 4) \\ (4 \times 2 + 3 \times 3) & (4 \times 1 + 3 \times 4) \end{bmatrix}$$

```
import numpy as np
```

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

```
a2 = np.array(a)
b2 = np.array(b)
```

```
print (np.dot(a2, b2))
```

문제155. 아래의 행렬의 곱을 출력하시오.

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & 4 & 3 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 4 & 3 \\ 6 & 7 \end{bmatrix}$$

```
import numpy as np
```

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

```
a2 = np.array(a)
b2 = np.array(b)
```

```
print (np.dot(a2, b2))
```

## Numpy 모듈 이해하기2

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

Ex)

아래의 리스트에서 최대값을 출력하시오.

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

```
Import numpy as np
```

```
A2 = np.array(a)
```

```
Print (np.max(a2))
```

Ex)

아래의 리스트에서 최대값, 최소값, 평균값, 최빈값, 중앙값을 출력하시오.

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

```
Import numpy as np
```

```
from scipy.stats import mode
```

```
A2 = np.array(a)
```

```
Print (np.max(a2)) # 최대값
```

```
Print (np.min(a2)) # 최소값
```

```
Print (np.mean(a2)) # 평균값
```

```
Print (mode(a)) # 최빈값
```

```
Print (np.median(a2)) # 중앙값
```

## 48. 파일 열고 닫기(open, close)

파일은 텍스트 파일과 바이너리 파일 두가지 종류가 있다.

텍스트 파일은 사람이 읽을 수 있는 글자로 저장 된 파일이고

바이너리 파일은 컴퓨터가 읽고 이해할 수 있는 이진 데이터를 기록한 파일이다.

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

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

파이썬에서 파일을 다루기 위해서 가장 먼저 해야할 일은 파일을 오픈하는 것인데 파일을 오픈하기 위해서는 open() 함수를 이용해야 한다.

Open(파일이름,모드)

| 모드 | 설명          |
|----|-------------|
| R  | 텍스트 모드로 읽기  |
| W  | 텍스트 모드로 쓰기  |
| Rb | 바이너리 모드로 읽기 |
| Wb | 바이너리 모드로 쓰기 |

Ex)

이미지 파일을 파이썬에서 여는 방법

1. Lena.png 파일을 내려받아 c:\wwdata\ww 밑에 저장

2. 아래의 코드를 실행 합니다.

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

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

문제156. 폐사진을 파이썬에서 시각화 하시오.

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

im = pilimg.open('c:\\data\\1.png')
pix = np.array(im)
plt.imshow(pix)
```

## 49. 클래스 이해하기(class)

객체지향 프로그램에서 중요한 단어가 바로 클래스 이다.

클래스는 프로그래머가 지정한 이름을 만든 하나의 독립된 공간이며, 이름공간(name space)이라 부른다.

클래스를 구성하는 주요요소는 클래스에서 변수 역할을 하는 클래스 멤버와 함수와 동일한 역할을 하는 클래스 메소드 이다.

클래스는 설계도이고 객체는 설계도를 바탕으로 만든 제품이다.

파이썬 클래스를 만들때 self 키워드는 필수이다.

아래와 같이 총 설계도로 총을 만들고 총을 사용할 때

Charge 와 shoot 기능을 사용할 때 총의 소유가 나 라는 의미가 self 이다.

총 설계도에는 charge(self, num) 이라고 해서 입력 매개변수가 2개인데  
총을 충전할 때는 gun1.charge(100) 과 같이 매개변수 하나에 값을 입력한다.  
그러면 self 입력매개변수에는 자동으로 gun1이 들어간다.

Ex)

총 클래스(설계도) 를 생성하시오.

총의 기능 :

1. 장전
2. 격발

Class Gun(): # 클래스 이름은 첫번째 철자는 대문자로 쓰는 것을 권장한다.

```
 Def charge(self, num): # 총알을 충전하는 함수
 Self.bullet = num
```

```
 Def shoot(self, num): # 총을 쏘는 함수
 For i in range(num):
 If self.bullet > 0:
 Print('탕!')
 Self.bullet -= 1
 Elif self.bullet == 0:
 Print('총알이 없습니다.')
 break
```

Ex)

위에서 만든 총설계도를 가지고 총을 한정 만드시오.

```
Gun1 = Gun()
```

Ex)

Gun1이라는 제품에 총알 10발을 충전하시오.

```
Gun1.charge(10)
```

Ex)

총을 격발 하시오.

```
Gun1.shoot(3)
```



문제157. 총설계도를 수정해서 총알을 아래와 같이 충전하면 몇발이 충전 되었습니다. 라는 메시지가 출력되게 하시오.

```
Gun1.charge(10)
```

10발이 충전되었습니다.

```
class Gun():
 def charge(self, num):
 self.bullet = num
 print (num,'발이 충전되었습니다.')

 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

gun2 = Gun()
gun2.charge(10)
```

문제158. 총을 쏘면 총알이 탕! 하면서 아래에 몇발 남았습니다. 라는 메시지가 출력되게 하시오.

```
Gun1.charge(10)
```

```
Gun1.shoot(3)
```

탕!

탕!

탕!

7발 남았습니다.

```
class Gun():
 def charge(self, num):
 self.bullet = num
 print (num,'발이 충전되었습니다.')

 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
 print (self.bullet, '발 남았습니다.')

gun2 = Gun()
gun2.charge(10)

###
gun2.shoot(3)

```

문제159. 총을 처음생산했을 때 총알이 반드시 0발 충전되게 총 설계도를 수정하시오.

```

class Gun():
 def __init__(self): # 설계도를 가지고 제품을 처음 만들때 자동으로 작동되는 함수
 self.bullet = 0
 print ('총이 만들어졌습니다.',self.bullet, '발 충전되었습니다.')

 def charge(self, num):
 self.bullet = num
 print (num,'발이 충전되었습니다.')

 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
 print (self.bullet, '발 남았습니다.')

gun2 = Gun()

```

문제160. 총 클래스를 이용해서 카드클래스를 만들고 아래와 같이 카드를 충전하고 사용하시오.

```

Card1 = Card()
카드가 발급되었습니다.

Card1.charge(10000)
10000원이 충전되었습니다.

Card1.consume(1000)

```

1000원이 사용되었습니다.

```
class Card():
 def __init__(self):
 self.cash = 0
 print ('카드가 발급되었습니다.', self.cash, '원 충전되었습니다.')

 def charge(self, num):
 self.cash = num
 print (num, '원 충전되었습니다.')

 def consume(self, num):
 if self.cash > 0:
 self.cash -= num
 print (num, '원 사용되었습니다')
 print ('잔액이 ', self.cash, ' 원 남았습니다')
 elif self.cash ==0:
 print('잔액이 없습니다')

card1 = Card()
#%%
card1.charge(10000)
#%%
card1.consume(1000)
```

## 51. 클래스 메소드 이해하기

Card 클래스의 charge 와 consume 는 메소드(기능) 이다.

Ex)  
A = [1, 2, 3, 4, 5]  
Print (type(a)) # <class 'list'>

리스트가 클래스라고 출력이 되는데 리스트가 클래스라는 말은 리스트 클래스 안에 메소드(기능)가 존재할 것이다.

A = [1,2,3,4]  
A.append(5)

A - 객체  
Append(5) - 메소드(기능)

리스트 객체의 유용한 메소드

1. Append() : 리스트 맨 끝에 새로운 요소를 추가할 때 사용
2. Count() : 리스트에서 특정 요소의 개수를 카운트할 때 사용
3. Insert() : 리스트의 특정 위치에 요소를 입력할 때 사용

4. Remove() : 리스트의 특정 요소를 제거할 때 사용
5. Sort() : 리스트의 요소를 순차적으로 정렬할 때 사용
6. Reverse() : 리스트의 요소를 역순으로 정렬할 때 사용
7. Index() : 리스트의 특정 요소의 위치를 출력할 때 사용

A = [1,2,3,1,2,2,2,3,4] # a 라는 리스트 객체가 생성

Print(a.count(2)) # a 리스트에 숫자 2가 몇개가 있는지 조회

문제161. 초등학생 키에 대한 모집단을 생성하시오. (천만개로 구성)  
키의 평균값을 140, 표준편차 5로 해서 생성하시오.

```
import numpy as np
```

```
avg = 140
std = 5
N = 10000000
mo = np.random.randn(N) * std + avg

print (mo)
```

문제162. 위의 모집단에서 표본을 100개를 추출해서 표본의 평균값을 출력하시오.

```
import numpy as np
```

```
avg = 140
std = 5
N = 10000000
mo = np.random.randn(N) * std + avg

print (np.random.choice(mo,100).mean())
```

문제163. 위에서 표본 100개를 뽑아서 평균값을 구해서 a 라는 비어 있는 리스트에 담는 작업을 10000번 수행하시오.

```
import numpy as np
```

```
avg = 140
std = 5
N = 10000000
mo = np.random.randn(N) * std + avg
a= []

for i in range(10000):
 a.append (np.random.choice(mo,100).mean())
```

```
print (a)
```

문제164. 위에서 구한 표본 평균값들 10000개의 평균값과 표준편차를 s\_avg와 s\_std 변수에 각각 담으시오.

```
import numpy as np

avg = 140
std = 5
N = 10000000
mo = np.random.randn(N) * std + avg
a= []

for i in range(10000):
 a.append (np.random.choice(mo,100).mean())

s_avg = np.mean(a)
s_std = np.std(a)

print (s_avg)
print (s_std)
```

문제165. 초등학생 키 데이터를 120 부터 160 까지 0.001 간격으로 생성해서 x 라는 변수에 넣으시오.

```
x = np.arange(120, 160, 0.001)

print (x)
```

문제166. 위에서 만든 x 값을 x 축으로 두고 확률밀도함수 그래프를 생성하는데 y축의 확률밀도함수 값을 구할 때 문제 164번에서 구한 평균값, 표준편차를 이용하시오.

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

avg = 140
std = 5
N = 10000000
mo = np.random.randn(N) * std + avg
a= []

for i in range(10000):
 a.append (np.random.choice(mo,100).mean())

s_avg = np.mean(a)
s_std = np.std(a)
```

```
x = np.arange(130, 150, 0.001)
y = norm.pdf(x, s_avg, s_std)
```

```
plt.plot(x, y, color = 'blue')
plt.show()
```

문제167. <http://cafe.daum.net/oracleoracle/SfWb/261>

```
import numpy as np
```

```
avg = 140
std = 5
N = 100000
mo = np.random.randn(N) * std + avg
cnt = 0
```

```
for i in mo:
 if 145 <= i <= 150:
 cnt += 1
```

```
print (cnt/100000)
```

# 20.12.03

2020년 12월 3일 목요일 오전 9:38

클래스를 사용해서 파이썬 코딩을 해야하는 이유

- 객체 지향 언어의 장점인 상속을 활용할 수 있기 때문이다.
- 객체 지향 언어의 장점이 프로그램을 할 때 너무 유용하기 때문에 객체를 만드는 것을 지향(선호)하는 프로그램 언어이다.

상속 코드 :

```
Class Movie_Card(Card):
```

객체 지향 언어의 장점

- 상속

파이썬 클래스에서 사용하는 키워드 self

파이썬 클래스를 만들때 self 키워드는 필수이다.

아래와 같이 총 설계도로 총을 만들고 총을 사용할 때

Charge 와 shoot 기능을 사용할 때 총의 소유가 나 라는 의미가 self 이다.

```
Class Gun(): # 클래스 이름은 첫번째 철자는 대문자로 쓰는 것을 권장한다.
```

```
 Def charge(self, num): # 총알을 충전하는 함수
```

```
 Self.bullet = num
```

```
 Def shoot(self, num): # 총을 쏘는 함수
```

```
 For i in range(num):
```

```
 If self.bullet > 0:
```

```
 Print('탕!')
```

```
 Self.bullet -= 1
```

```
 Elif self.bullet == 0:
```

```
 Print('총알이 없습니다.')
```

```
 break
```

## 52. 클래스 생성자 이해하기

클래스의 인스턴스 객체가 생성될 때 자동적으로 호출되는 메소드가 클래스 생성자 이다.

클래스 생성자는 \_\_init\_\_(self): 이다.

카드의 기능

1. 발급 (0원으로 발급)
2. 충전

### 3. 소비

Ex)

```
Class Card():
 Def __init__(self):
 Self.cash = 0
 Print('카드가 발급되었습니다.',self.cash,'가 충전되어 있습니다.')
```

KJM\_card = Card() # 객체(제품)를 생성하는 명령어

문제167. 총 클래스를 생성하는데 총 클래스로 아래와 같이 총(제품)을 생성하면 총이 만들어졌습니다. 총알이 0 발 충전되었습니다. 라는 메시지가 출력되게 하시오.

```
class Gun():
 def __init__(self):
 self.bullet = 0
 print ('총알이', self.bullet, '발 충전 되었습니다.')
```

gun1 = Gun()

문제168. 문자열 포맷을 이용해서 위의 메시지를 자연스럽게 출력되게 하시오.

```
class Gun():
 def __init__(self):
 self.bullet = 0
 print ('총알이 %d발 충전 되었습니다.' %self.bullet)
```

gun1 = Gun()

## 53. 클래스 소멸자 이해하기

객체가 사라질 때 자동으로 호출되는 함수를 소멸자(\_\_del\_\_) 이라고 한다.

Ex)

총 클래스로 KJM\_gun을 만들고 총을 사용하다가 총을 폐기하고 싶으면 del 명령어로 총을 폐기하면 되는데 이때 자동으로 작동되는 메소드(함수)가 소멸자 함수이다.

Ex)

```
Class Gun():
 Def __init__(self):
 Self.bullet = 0
 Print('총 한정이 생성되었습니다.')
```



```
Def __del__(self):
 Print('총이 폐기 되었습니다.')
```

```
KJM_gun = Gun()
```

```
Del KJM_gun
```

## 54. 클래스 상속 이해하기

클래스는 상속이라는 특성을 가지고 있는 이름 공간이다.

클래스에서 상속이란 어떤 클래스 가지고 있는 멤버(변수)나 메소드(함수)를 상속받는 클래스가 모두 사용할 수 있도록 해주는 것이다.

상속을 해주는 클래스를 부모 클래스 또는 슈퍼 클래스라고 하고 상속을 받는 클래스를 자식 클래스 또는 서브 클래스 라고 한다.

부모 클래스로 부터 상속을 받아 자식 클래스를 정의하는 방법은 다음과 같다.

Ex)

Class 자식클래스이름(부모클래스이름):

자식 클래스는 부모 클래스에서 정의된 모든 멤버와 메소드를 그대로 상속 받는다.

1. 팀장님이 카드의 주요기능을 담고 있는 Card() 클래스를 생성
2. 팀원이 영화 할인 카드 클래스를 만들 것 인데 팀장님이 구현한 카드의 주요기능을 Card()클래스로 부터 상속받고 Card() 클래스 생성시 구현한 코드는 구현하지 않는다.

Ex)

1. 팀장님이 만든 카드 class (카드 설계도) # 부모 클래스
  - 카드가 발급되었을 때 0원이 충전되게 하는 기능 : \_\_init\_\_(self)
  - 카드를 충전하는 기능 : charge(self, num)
  - 카드를 사용하는 기능 : consume(self, num)

```
class Card():
```

```
 Def __init__(self):
 Self.cash = 0
 Print ('카드가 발급되었습니다.')
```

```
 Def charge (self, num):
 Self.cash += num
 Print (num, '원이 충전되었습니다.')
```

```
 Def consume (self, num):
 If self.cash >= num:
 Self.cash -= num
```

```
 Print (num, '원이 사용되었습니다.')
 Else:
 Print ('잔액이 부족합니다.')
```

문제169. 충전한 금액보다 더 많은 금액을 소비하면 어떻게 되는지 테스트 하시오.

```
class Card():
 def __init__(self):
 self.cash = 0
 print ('카드가 발급되었습니다.')

 def charge (self,num):
 self.cash += num
 print (num,'원이 충전되었습니다.')

 def consume (self,num):
 if self.cash >= num:
 self.cash -= num
 print (num,'원이 사용되었습니다.')
 else:
 print ('잔액이 부족합니다.')
```

```
KJM_card = Card()
KJM_card.charge(10000)
KJM_card.consume(12000)
```

문제170. 팀장님이 만든 Card() 클래스를 상속받아 영화 할인 카드를 생성하시오.

(영화할인 카드 클래스 : Movie\_Card())

```
class Movie_Card(Card): # 부모클래스인 Card 클래스를 상속받아 Movie_Card 클래스를 만든다.
 pass

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(8000)
```

문제171. 영화관에서 사용하면 할인이 될 수 있도록 영화 클래스를 생성하시오.

```
class Movie_Card(Card): # 부모에게 __init__, charge, consume 를 상속받는다.
 def consume(self, num, place): # 부모에게 받은 consume를 내가 만든 consume 로 덮어쓰기
 (overriding)다.
 if place == '영화관':
```

```

 num = 0.8 * num
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print ('잔액이 부족합니다.')
 else:
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print('잔액이 부족합니다.')

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(8000,'영화관')
m_card1.consume(8000,'편의점')

```

문제172. 위의 영화할인 카드에 할인 장소를 추가해서 주유소에서도 20% 할인 될 수 있도록 코드를 수정하시오.

```

class Movie_Card(Card):
 def consume(self, num, place):
 if place in ('영화관', '주유소'):
 num = 0.8 * num
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print ('잔액이 부족합니다.')
 else:
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print('잔액이 부족합니다.')

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(8000,'영화관')
m_card1.consume(10000,'주유소')
m_card1.consume(8000,'편의점')

```

문제173. 영화관과 주유소에서는 20% 할인되게 하고 스타벅스에서는 10% 할인되게 코드를 수정하시오.

```

class Movie_Card(Card):
 def consume(self, num, place):
 if place in ('영화관', '주유소'):
 num = 0.8 * num
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print ('잔액이 부족합니다.')
 elif place == '스타벅스':
 num = 0.9 * num
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print ('잔액이 부족합니다.')
 else:
 if self.cash >= num:
 self.cash -= num
 print (place,'에서',num,'원이 사용되었습니다.')
 else:
 print('잔액이 부족합니다.')

m_card1 = Movie_Card()
m_card1.charge(100000)
m_card1.consume(10000,'영화관')
m_card1.consume(10000,'주유소')
m_card1.consume(10000,'스타벅스')
m_card1.consume(10000,'편의점')

```

## 50. 클래스 멤버와 인스턴스 멤버 이해하기

클래스에서 선언된 변수는 클래스 멤버(변수)와 인스턴스 멤버(변수)가 있다.

클래스 멤버는 클래스 메소드 바깥에서 선언되고 인스턴스 멤버는 클래스 메소드 안에서 self 와 함께 선언되는 변수이다.

Ex)

사원이 입사하면 입사한 사원에 대한 이메일을 자동으로 생성하고 이름을 출력하는 함수와 월급을 인상하는 함수를 담은 클래스를 생성

```

class Employees: # __init__ 함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

 def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수
 self.first = first
 self.last = last
 self.pay = pay

```

```

self.email = first.lower() + '.' + last.lower() + '@gmail.com'
self.raise_amount = 1.1 # 인스턴스 변수

def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * self.raise_amount)

emp_chulsu = Employees('chulsu', 'kim', 5000000)
print(emp_chulsu.pay) # 5000000
emp_chulsu.apply_raise()
print(emp_chulsu.pay) # 5500000

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
emp_chulsu2.raise_amount = 1.2
print(emp_chulsu2.pay) # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay) # 6000000

```

문제174. 위에서 생선산 객체 emp\_chulsu의 email변수에 있는 내용을 출력하시오.

```

class Employees: # __init__함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

 def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수
 self.first = first
 self.last = last
 self.pay = pay
 self.email = first.lower() + '.' + last.lower() + '@gmail.com'
 self.raise_amount = 1.1 # 인스턴스 변수

 def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

 def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * self.raise_amount)

emp_chulsu = Employees('chulsu', 'kim', 5000000)

print (emp_chulsu.email)

```

문제175. 철수의 월급을 회사규정에 따라 인상시키시오.

```

class Employees: # __init__함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

```

```

def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수
 self.first = first
 self.last = last
 self.pay = pay
 self.email = first.lower() + '.' + last.lower() + '@gmail.com'
 self.raise_amount = 1.1 # 인스턴스 변수

def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * self.raise_amount)

emp_chulsu = Employees('chulsu', 'kim', 5000000)
print(emp_chulsu.pay) # 5000000
emp_chulsu.apply_raise()
print(emp_chulsu.pay) # 5500000

```

문제176. 새로운 사원 철수2로 emp\_chulsu2 객체를 생성하시오.

```

class Employees: # __init__ 함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

 def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수
 self.first = first
 self.last = last
 self.pay = pay
 self.email = first.lower() + '.' + last.lower() + '@gmail.com'
 self.raise_amount = 1.1 # 인스턴스 변수

 def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

 def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * self.raise_amount)

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)

```

문제177. 철수2 사원이 월급을 인상시키기 위한 인스턴스 변수를 알아내서 아래와 같이 emp\_chulsu2 객체의 멤버인 raise\_amount 라는 변수에 1.2를 할당하고 월급을 인상시키는 apply\_raise() 메소드를 실행 했다.

```

class Employees: # __init__ 함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

 def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수

```

```

self.first = first
self.last = last
self.pay = pay
self.email = first.lower() + '.' + last.lower() + '@gmail.com'
self.raise_amount = 1.1 # 인스턴스 변수

def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * self.raise_amount)

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
emp_chulsu2.raise_amount = 1.2
print(emp_chulsu2.pay) # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay) # 6000000

```

문제178. 위와 같이 악용이 될수 없도록 막는 방법을 찾으시오.

```

class Employees: # __init__함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.
 raise_amount = 1.1 # 클래스 변수(클래스 멤버)

 def __init__(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수
 self.first = first
 self.last = last
 self.pay = pay
 self.email = first.lower() + '.' + last.lower() + '@gmail.com'
 self.raise_amount = 1.1 # 인스턴스 변수

 def full_name(self): #사원의 전체이름을 출력하는 함수
 print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

 def apply_raise(self): # 월급을 인상하는 함수
 self.pay = int(self.pay * Employees.raise_amount) # self(인스턴스 변수) 에서 Employees(클래스 변수) 로 변경

emp_chulsu2 = Employees('chulsu2', 'kim', 5000000)
emp_chulsu2.raise_amount = 1.2
print(emp_chulsu2.pay) # 5000000
emp_chulsu2.apply_raise()
print(emp_chulsu2.pay) # 6000000

```

위와 같이 민감한 변수들은 인스턴스 변수가 아니라 클래스 변수를 사용해서 계산되게 코딩해야 한다.

클래스 내의 변수 2개

1. 클래스 변수 : 메소드 바깥의 변수
  - 객체 생성이후에 안의 값을 변경할 수 없다.

2. 인스턴스 변수 : 메소드 안의 변수
- 객체 생성이후에 안의 값을 변경할 수 있다.

class Employees: # \_\_init\_\_함수에 여러개의 입력 매개변수가 있기 때문에 ()를 쓰지 않았다.

raise\_amount = 1.1 # 클래스 변수(클래스 멤버)

def \_\_init\_\_(self, first, last, pay): # 객체가 만들어질때 바로 작동되는 함수

self.first = first

self.last = last

self.pay = pay

self.email = first.lower() + '.' + last.lower() + '@gmail.com'

self.raise\_amount = 1.1 # 인스턴스 변수

def full\_name(self): #사원의 전체이름을 출력하는 함수

print ('{} {}'.format(self.first, self.last)) # 중괄호에 각각 self.first와 self.last에 있는 값 입력

def apply\_raise(self): # 월급을 인상하는 함수

self.pay = int( self.pay \* Employees.raise\_amount) # self(인스턴스 변수) 에서 Employees(클래스 변수) 로 변경

emp\_chulsu2 = Employees('chulsu2', 'kim', 5000000)

emp\_chulsu2.raise\_amount = 1.2

print(emp\_chulsu2.pay) # 5000000

emp\_chulsu2.apply\_raise()

print(emp\_chulsu2.pay) # 6000000

## 55. 예외처리 이해하기 ① (try~except)

프로그램을 작성하다 보면 뜻하지 않은 오류가 발생하는 코드가 있다.

프로그램이 실행되는 동안 오류가 발생하면 프로그램이 더 이상 진행 될 수 없는 상태가 되는 데 이를 예외상황이라고 한다.

프로그램에 예외가 발생하더라도 프로그램을 중단 시키지 않고 예외에 대한 적절한 처리를 하여 프로그램을 계속 진행 시킬 수 있도록 하는 구문이 try ~ except 이다.

Try:

문제가 없을 경우 실행할 코드

Except:

문제가 생겼을 때 실행할 코드

Ex)

Try ~ except를 사용하지 않았을 때의 코드

Def my\_divide():

X = input('분자의 숫자를 입력하시오.')

Y = input('분모의 숫자를 입력하시오.')



```
Return int(x) / int(y)
```

```
Print (my_divide())
```

Ex)

Try ~ except를 사용했을 때의 코드

```
Def my_divide():
```

```
 Try:
```

```
 X = input('분자의 숫자를 입력하시오.')
```

```
 Y = input('분모의 숫자를 입력하시오.')
```

```
 Return int(x) / int(y)
```

```
 Except:
```

```
 Return '잘못된 값을 입력하여 나누기를 할 수 없습니다.'
```

```
Print (my_divide())
```

Try와 except 사이에 코드가 잘 실행이 된다면 except이후의 문장은 실행하지 않고 try와 except사이에 코드가 실행이 안된다면 except 이후의 문장을 실행한다.

문제179. 판다스를 이용해서 emp3.csv에서 이름과 월급을 출력하시오.

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

문제180. 이름이 SCOTT인 사원의 이름과 월급을 출력하시오.

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

문제181. input 함수를 이용해서 이름을 물어보게 하고 이름을 입력하면 해당사원의 이름과 월급이 출력되게 하시오.

```
import pandas as pd
```

```
name = input('이름을 입력하시오.')
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

문제182. 소문자로 이름을 입력해도 출력되게 하시오.

```
import pandas as pd

name = input('이름을 입력하시오.')
emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'sal']][emp['ename']==name.upper()]])
```

문제183. 위의 코드를 실행하는데 없는 사원이름을 입력하시오.

```
이름을 입력하시오.jack
Empty DataFrame
Columns: [ename, sal]
Index: []
```

문제184. 위의 코드를 예외처리를 해서 없는 사원이름을 입력하면 입력하신 이름의 사원은 존재하지 않습니다. 라는 메시지가 출력되게 하시오.

에러가 아니라서 예외처리가 되지 않는다.

문제185. 숫자를 입력하면 해당 숫자의 제곱값이 출력되는 코드를 구현하시오.

```
num = int(input ('숫자를 입력하시오.'))

print (num**2)
```

문제186. try ~ except 예외처리를 이용해서 아래와 같이 문자를 입력하면 '잘못된 값을 입력하였습니다.' 라는 메시지가 출력되게 하시오.

```
try:
 num = int(input('숫자를 입력하시오.'))
 print (num**2)
except:
 print('잘못된 값을 입력하였습니다.')
```

Try와 except 사이에 있는 코드에서 에러가 나야지만 except 이후의 문장을 실행한다.

## 56. 예외처리 이해하기 ② (try~except~else)

어떤 로직을 수행할 때 오류 상황이 아닐 경우에만 어떤 작업을 수행하는 코드를 작성해야 할 때가 있는데 이런 경우에 try ~ except ~ else 구문을 사용한다.

Try:

실행할 코드 블록

Except:

예외처리할 코드 블록

Else:

Except 절을 만나지 않았을 경우 실행하는 코드 블록

Ex)

try:

```
num = int(input('숫자를 입력하시오.'))
```

```
print (num**2)
```

except:

```
print('잘못된 값을 입력하였습니다.')
```

else:

```
print('결과 출력에 성공했습니다.')
```

Try ~ except 사이의 코드에 에러가 나지 않으면 else: 이후의 문장을 실행한다.

문제187. 나누기 프로그램을 수정해서 나누기가 성공하면 성공적으로 나누기를 하였습니다. 라는 메시지가 출력되게 하시오.

try:

```
x = int(input('분자를 입력하시오.'))
```

```
y = int(input('분모를 입력하시오.'))
```

```
print (x/y)
```

except:

```
print ('잘못된 값을 입력하였습니다.')
```

else:

```
print ('성공적으로 나누기를 하였습니다.')
```

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

오류 발생 유무와 상관없이 어떤 코드를 무조건 실행 시키려면 try ~ except ~ finally 구문을 활용한다.

무조건 실행시키는 코드는 finally 부분에 작성하면 된다.

Ex)

Try:

```
Print('안녕하세요.')
```

Except:

```
Print('예외가 발생했습니다.')
Finally:
Print('저는 무조건 실행됩니다.')
```

문제188. 나누기 프로그램을 실행할 때 오류가 나도 무조건 아래의 메시지가 출력되게 하시오.

분자를 입력하시오.  
분모를 입력하시오.

결과  
김정민이 만든 프로그램입니다.

```
try:
 x = int(input('분자를 입력하시오.))
 y = int(input('분모를 입력하시오.))
 print (x/y)
except:
 print ('잘못된 값을 입력하였습니다.')
finally:
 print ('김정민이 만든 프로그램입니다.')
```

문제189. 동전을 10번 던져서 앞면이 2번 나올 확률을 출력하시오.  
(동전을 10번 던지는 작업을 10000번 수행되게 하시오.)

```
import random
coin = ['앞면','뒷면']
cnt = 0

for k in range (10000):
 a = []
 for i in range (10):
 result = random.choice(coin)
 a.append(result)
 if a.count('앞면') == 2:
 cnt += 1

print (cnt/10000)
```

문제190. 위의 코드를 함수를 생성하여 확률이 출력되게 하시오.

```
Coin_prob(2) # 0.04314
Coin_prob(4) # 0.2065
```

```
import random
coin = ['앞면', '뒷면']

def coin_prob(num):
 cnt = 0

 for k in range (10000):
 a = []
 for i in range (10):
 result = random.choice(coin)
 a.append(result)
 if a.count('앞면') == num:
 cnt += 1

 print (cnt/10000)

coin_prob(2)
coin_prob(4)
```

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

코드에서 예외가 발생하면 이에대한 자세한 내용을 파악하는 것이 중요하다.

파이썬은 발생

<http://docs.python.org/3/library/exceptions.html>

위의 사이트에 보면 여러가지 다양한 에러에 대해서 미리 에러 처리를 할 수 있도록 정의해 놓은 예외들을 확인할 수 있다.

Ex)

Try:

```
X = int(input('분자의 숫자를 입력하시오'))
```

```
Y = int(input('분모의 숫자를 입력하시오'))
```

```
Print (x/y)
```

Except:

```
Print('잘못된 값을 입력하여 나누기를 실행 할 수 없습니다.')
```

위의 경우에는 분모값을 입력할 때 숫자 0 을 입력했을때와 문자 a 를 입력했을 때 똑같이 '잘못된 값을 입력하여 나누기를 실행 할 수 없습니다.' 가 출력되는데 좀 더 구체화 해서 분모값으로 0을 입력하면 0으로 나눌 수 없습니다 와 같이 나오고 분모값으로 a 를 입력하면 잘못된 값을 입력하셨습니다 와 같이 나오게 할 수 있다.

Ex)

Try:

```
X = int(input('분자의 숫자를 입력하시오'))
```

```
Y = int(input('분모의 숫자를 입력하시오'))
```

```
Print (x/y)
```

Except ZeroDivisionError:

```
Print('0으로 나눌 수 없습니다.')
```

Except:

```
Print('잘못된 값을 입력하셨습니다.')
```

문제191. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 1번부터 숫자가 출력되게 하는 코드를 작성하시오.

```
num = int(input('숫자를 입력하시오.'))
```

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

문제192. 위의 코드에 예외처리를 해서 숫자를 물어볼때 문자를 입력하면 잘못된 값을 입력하셨습니다. 라고 메시지가 출력되게 하시오.

```
try:
 num = int(input('숫자를 입력하시오.'))
 for i in range(1,num+1):
 print (i)
except:
 print ('잘못된 값을 입력하셨습니다.')
```

위의 코드의 경우에는 숫자를 입력할 때 알파벳 a 를 넣으면 예외처리가 되어서 잘못된 값을 입력하셨습니다. 라는 말만 나오고 에러에 대한 정확한 원인파악은 하기가 어렵다.

잘못된 값을 입력하셨습니다. 말고 정확한 에러에 대한 원인을 파악하고 싶다면 아래와 같이 작성하면 된다.

```
try:
 a = int(input('숫자를 입력하시오.'))

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

except Exception as e:
 print ('잘못된 값을 입력하셨습니다.')
```

print (e) # 에러가 난 이유를 출력해준다.

## 59. 예외처리 이해하기 ⑤ (try~except 특정 예외)

파이썬 입장에서 봤을때는 오류가 아닌데 프로그래머가 오류라고 판단하고 일부러 프로그램이 돌지 않게 오류메세지를 출력하는 경우에 사용한다.

설명 : 금융권 프로그램에서는 금액이 안맞는 프로그램이 있다면 사고로 이어지게 되므로 금액이 안맞으면 프로그램을 종료하도록 예외처리를 할 수 있다.

실행코드

If 조건:

    Raise Exception('예외가 발생했습니다.') # 종료

Else:

    실행코드

문제193. 판다스를 이용해서 emp3.csv의 데이터를 로드하는데 이름을 물어보게하고 이름을 입력하면 해당 사원의 이름과 월급이 출력되게 하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
name = input('이름을 입력하시오.')

print (emp[['ename', 'sal']][emp['ename'] == name.upper()]
```

문제194. 위의 결과에서 월급만 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
name = input('이름을 입력하시오.')
sal = emp['sal'][emp['ename'] == name.upper()].values[0]

print (sal)
```

Emp 데이터 프레임에서 어떤 특정값 하나만 출력하려면 위와 같이 작성해야 한다.

문제195. 위의 코드에 사용자 정의 예외처리를 해서 월급이 고소득자는 해당 사원의 월급을 볼 수 없습니다. 라는 메시지가 출력되게 하시오.

(월급이 3000 이상인 사원들을 고소득자로 보고 작성)

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
name = input('이름을 입력하시오.')
sal = emp['sal'][emp['ename'] == name.upper()].values[0]

if sal >= 3000:
 raise Exception ('해당 사원의 월급을 볼 수 없습니다.')

else:
 print (sal)
```

문제196. 위의 코드를 수정해서 이름을 물어보게하고 이름과 직업을 출력하는 코드로 작성하는데 직업이 SALESMAN이면 해당 사원의 정



보는 볼 수 없습니다. 라는 메시지가 출력되면서 프로그램이 종료되게 하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
name = input('이름을 입력하시오.')
job = emp['job'][emp['ename'] == name.upper()].values[0]

if job == 'SALESMAN':
 raise Exception ('해당 사원의 정보를 볼 수 없습니다.')

else:
 print (emp[['ename', 'job']][emp['ename'] == name.upper()])
```

## 60. 값 입력받기(input)

파이썬 내장 함수 input은 사용자가 입력한 값을 문자열로 리턴한다.

Input() 의 인자는 사용자 입력을 돕기위한 안내 문구나 힌트 등을 표시하는 문자열이 된다.

Ex)

```
Name = input ('사원이름을 입력하시오.')
```

```
Num = int(input('숫자를 입력하시오.'))
```

문제197. 아래처럼 사원 이름을 물어보게하고 사원이름을 입력하면 해당 사원의 이름과 월급이 출력되게 하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
name = input('이름을 입력하시오.')

print (emp[['ename', 'sal']][emp['ename'] == name.upper()])
```

문제198. 없는 사원 이름을 입력하면 해당사원은 없습니다. 라는 메시지가 출력되게 하시오.

```
import pandas as pd

try:
 emp = pd.read_csv("c:\\data\\emp3.csv")
 name = input('사원이름을 입력하시오.')
 result = emp['ename'][emp['ename'] == name.upper()].values[0]
 print (emp[['ename', 'sal']][emp['ename'] == result])
```

```
except LookupError:
 print ('해당사원은 없습니다.')
```

```
result = emp['ename'][emp['ename'] == name.upper()].values[0]
```

이 코드에서 values[0] 을 사용하면 Series(컬럼) 가 아닌 값으로 출력이 되어서 result에 담기게 된다. 없는 사원 이름을 입력하면 result에 값이 입력되지 않게 되므로 LookupError 예외처리가 되어서 해당사원은 없습니다. 라는 메시지가 출력된다.

문제199. 직업을 물어보게하고 직업을 입력하면 해당사원의 이름과 직업과 월급이 출력되게 하는 코드를 작성하는데 없는 직업을 입력하면 해당 직업은 사원 테이블에 없습니다. 라고 출력되게 하시오.

```
import pandas as pd
```

```
try:
```

```
 emp = pd.read_csv('c:\\data\\emp3.csv')
 job = input('직업을 입력하십시오.')
 result = emp['job'][emp['job'] == job.upper()].values[0]
 print (emp[['ename', 'job', 'sal']][emp['job'] == result])
```

```
except LookupError:
```

```
 print ('해당 직업은 사원 테이블에 없습니다.')
```

## 61. 자료형 확인하기(type)

파이썬의 자료형은 하나의 클래스 이다. 파이썬은 숫자, 문자, 문자열, 리스트, 튜플, 사전, 함수 등을 각각의 하나의 클래스로 취급한다.

코드를 작성하다가 변수 이름만 보고 이 자료가 어떤 자료형인지 확인해야 하는 경우가 있는데 이때 파이썬 내장함수인 type()을 활용하면 자료형을 쉽게 확인 할 수 있다.

Ex)

```
Numdata = 57
```

```
Print (type(numdata)) # <class 'int'>
```

```
Numdata2 = 57.2
```

```
Print (type(numdata2)) # <class 'float'>
```

```
Strdata = '파이썬'
```

```
Print (type(strdata)) # <class 'str'>
```

```
A = [1, 2, 3, 4]
```

```
Print (type(a)) # <class 'list'>
```

문제200. 딕셔너리 자료형을 만들고 위와 같이 type을 확인하시오.

```
b = {'사과':'apple','배':'pear'}
print (type(b)) # <class 'dict'>
```

리스트 : 대괄호 []

튜플 : 소괄호 ()

딕셔너리 : 중괄호 {}

## 62. 나눗셈에서 나머지만 구하기(%)

나누기 연산에서 나머지만 구하는 연산은 %로 구한다.

Ex)

Print (12%3) # 0

Print (12%5) # 2

문제201. 아래와 같이 두개의 숫자를 각각 물어보게 하고 아래의 메시지가 출력되게 하시오.

첫번째 숫자를 입력하세요. 1113

두번째 숫자를 입력하세요. 23

1113을 23으로 나누면 9가 나머지로 남습니다.

```
a = int(input('첫번째 숫자를 입력하세요.'))
```

```
b = int(input('두번째 숫자를 입력하세요.'))
```

```
c = a % b
```

```
print ('%d을 %d로 나누면 %d가 나머지로 남습니다.'%(a,b,c))
```

## 63. 몫과 나머지 구하기(divmod)

Ex)

```
Result1, result2 = divmod(1113, 23)
```

몫     나머지

```
Print (result1, result2)
```

```
Print (divmod(1113,23))
```

Divmod함수가 출력하는 값이 2개이기 때문에 앞에 result1, result2 변수 두개를 사용

문제202. 아래와 같이 실행되게 코드를 실행하시오.

첫번째 숫자를 입력하시오. 1113

두번째 숫자를 입력하시오. 23

1113을 23으로 나눈 몫은 48 이고 나머지는 9 입니다.

try:

```
a = int(input('첫번째 숫자를 입력하시오.'))
```

```
b = int(input('두번째 숫자를 입력하시오.'))
```

```
result1, result2 = divmod(a, b)
```

```
print('%d 을 %d 으로 나눈 몫은 %d 이고 나머지는 %d 입니다.'%(a, b, result1, result2))
```

except ZeroDivisionError:

```
print('0으로는 나눌 수 없습니다.')
```

## 64. Pandas 를 이용한 데이터 검색

SQL, Pandas 를 자유롭게 사용할 수 있어야 한다.

Ex)

```
Emp[['ename','sal']] [emp['ename']=='SCOTT']
```

문제203. dept3.csv를 판다스로 로드해서 dept 데이터 프레임 전체를 출력하시오.

```
import pandas as pd
```

```
dept = pd.read_csv('c:\\data\\dept3.csv')
```

```
print (dept)
```

문제204. 부서위치가 DALLAS의 부서번호와 부서명(dname) 을 출력하시오.

```
import pandas as pd
```

```
dept = pd.read_csv('c:\\data\\dept3.csv')
```

```
print (dept[['deptno', 'dname']][dept['loc'] == 'DALLAS'])
```

## 65. Pandas 를 이용한 조인

Ex)

이름과 부서위치를 출력하시오.

```
Select e.ename, d.loc
 From emp e, dept d
 Where e.deptno = d.deptno;
```

```
Import pandas as pd
Emp = pd.read_csv("c:\\data\\emp3.csv")
Dept = pd.read_csv("c:\\data\\dept3.csv")
Result = pd.merge(emp,dept, on='deptno')
```

```
Print (result)
Print (result[['ename', 'loc']])
```

문제205. DALLAS에서 근무하는 직원들의 이름과 부서위치를 출력하시오.

```
Select e.ename, d.loc
 From emp e, dept d
 Where e.deptno = d.deptno
 And d.loc = 'DALLAS';
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')

result = pd.merge (emp,dept, on = 'deptno')

print (result[['ename', 'loc']][result['loc'] == 'DALLAS'])
```

문제206. 월급이 3000이상인 직원들의 이름과 월급과 부서위치를 출력하시오.

```
Select e.ename, e.sal, d.loc
 From emp e, dept d
 Where e.deptno = d.deptno
 And e.sal >= 3000;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')

result = pd.merge (emp,dept, on = 'deptno')

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

문제207. 부서번호가 10번, 20번인 직원들의 이름과 부서위치와 부서번호를 출력하시오.

```
Select e.ename, d.loc, e.deptno
 From emp e, dept d
 Where e.deptno = d.deptno
 And e.deptno in (10,20);
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')

result = pd.merge(emp, dept, on = 'deptno')

print (result[['ename', 'loc', 'deptno']][result['deptno'].isin([10, 20])])
```

문제208. 월급이 1000 에서 3000 사이인 직원들의 이름과 월급과 부서위치를 출력하시오.

```
Select e.ename, e.sal, d.loc
 From emp e, dept d
 Where e.deptno = d.deptno
 And e.sal between 1000 and 3000;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')

result = pd.merge(emp, dept, on = 'deptno')

print (result[['ename', 'sal', 'loc']][result['sal'].between(1000, 3000)])
```

문제209. 아래의 SQL을 pandas로 구현하시오.

```
Select e.ename, d.loc
 From emp e, dept d
 Where e.deptno (+) = d.deptno;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')

result = pd.merge(emp, dept, on = 'deptno', how = 'right')

print (result[['ename', 'loc']])
```

문제210. 아래의 SQL을 pandas로 구현하시오.

```
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:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")

result = pd.merge(emp,dept, on = 'deptno', how = 'outer')

print (result [['ename','loc']])
```

## 66. Pandas 를 이용한 서브쿼리

Ex)  
JONES 보다 많은 월급을 받는 직원들의 이름과 월급을 출력하시오.

```
Select ename, sal
 From emp
 Where sal > (select sal
 From emp
 Where ename = 'JONES');
```

Pandas>

```
import pandas as pd

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

jsal = emp['sal'][emp['ename']=='JONES'].values[0]
JONES 월급을 jsal 변수에 담는다.

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

문제211. 아래의 서브쿼리를 pandas로 구현하시오.

```
select ename, sal
 From emp
 Where job = (select job
 From emp
 Where ename = 'SCOTT');
```

import pandas as pd

```
emp = pd.read_csv("c:\\data\\emp3.csv")
sjob = emp['job'][emp['ename'] == 'SCOTT'].values[0]
```

```
print (emp[['ename', 'sal']][emp['job'] == sjob])
```

문제212. 아래의 서브쿼리를 pandas로 구현하시오.

```
select ename, sal
 From emp
 Where job = (select job
 From emp
 Where ename = 'SCOTT')
 And ename != 'SCOTT';
```

```
import pandas as pd
```

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

```
sjob = emp['job'][emp['ename'] == 'SCOTT'].values[0]
```

```
print (emp[['ename', 'sal']][(emp['job'] == sjob) & (emp['ename'] != 'SCOTT')])
```

판다스에서 and는 & 이고 or 는 | 이다.

& 와 | 를 사용 때는 괄호로 묶어줘야 한다.

## 67. Pandas 와 오라클 그룹함수의 비교

```
Select max(sal)
 From emp;
```

Pandas>

```
import pandas as pd
```

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

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

```
print (emp['sal'].max()) - 최대월급
```

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

```
print (emp['sal'].sum()) - 토탈월급
```

```
print (emp['sal'].var()) - 분산값
```

```
print (emp['sal'].std()) - 표준편차
```

문제213. 아래의 SQL을 판다스로 구현하시오.

```
Select max(sal)
 From emp
 Where deptno = 20;
```



```
import pandas as pd

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

print (emp['sal'][emp['deptno'] == 20].max())
```

문제214. 아래의 SQL을 판다스로 구현하시오.

```
Select min(sal)
 From emp
 Where job = 'SALESMAN';
```

```
import pandas as pd

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

print (emp['sal'][emp['job'] == 'SALESMAN'].min())
```

문제215. emp12.csv를 판다스 데이터 프레임으로 들어서 출력하시오.

```
import pandas as pd

emp12 = pd.read_csv("c:\\data\\emp122.csv")

print (emp12)
```

문제216. 우리반에서 최소나이를 출력하시오.

```
import pandas as pd

emp12 = pd.read_csv("c:\\data\\emp122.csv")

print (emp12['AGE'].min())
```

문제217. 아래의 SQL을 판다스로 구현하시오.

```
Select job, max(sal)
 From emp
 Group by job;
```

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('job')['sal'].max().reset_index()

print (result)
```

Reset\_index() 키워드는 Series(컬럼) 로 출력하는게 아니라 DataFrame(테이블) 으로 출력하는 키워드이다.

문제218. 아래의 SQL을 판다스로 구현하시오.

```
Select deptno, sum(sal)
 From emp
 Group by deptno;
```

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()

print (result)
```

문제219. 아래의 SQL을 판다스로 구현하시오.

```
Select deptno, sum(sal)
 From emp
 Where deptno != 20
 Group by deptno;
```

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].sum().reset_index()

print (result[result['deptno'] != 20])
```

문제220. <http://cafe.daum.net/oracleoracle/SfWb/316>

```
import random

coin = ['앞면','뒷면']

def coin_prob(num):
 cnt = 0
 for k in range (10000):
 a = []
 for i in range (10):
 result = random.choice(coin)
 a.append(result)
 if a.count('앞면') == num:
 cnt += 1
 return cnt/10000

for j in range(11):
 print ('동전을 10번 던져서', j, '개 앞면이 나올 확률', coin_prob(j))
```

# 20.12.07

2020년 12월 7일 월요일 오전 9:44

## 복습

1. 파이썬의 자료형 5가지
2. 파이썬 연산자
3. If문
4. Loop문
5. 문자열에서 특정 위치의 철자 검색, 슬라이싱 검색
6. 파이썬으로 함수 생성
7. 파이썬으로 클래스 생성
8. 파이썬의 예외처리
9. 판다스 사용법

클래스를 사용하는 이유 (객체지향 언어의 장점)

- 상속을 사용할 수 있기 때문

상속을 사용했을 때 부모 클래스가 작성한 기능(메소드)을 코딩하지 않아도 된다.

중요한 기능은 부모 클래스에서 하여 그 기능을 그대로 상속받아 상세코드를 작성 할 수 있다.

판다스로 조인하기 복습

판다스로 조인(join)을 하기 위해서는 merge를 사용하면 된다.

`Pandas.merge(emp, dept, how = 'inner', on = 'deptno')`

옵션 :

How = 'inner' : emp 와 dept 데이터 프레임에 공통적으로 존재하는 교집합일 경우에만 추출한다.

How = 'outer' : 열의 데이터가 양쪽 데이터 프레임에 공통적으로 존재하는 교집합이 아니어도 추출한다.

How = 'left' : 왼쪽 데이터 프레임의 키열에 속하는 데이터 값을 기준으로 병합한다.

How = 'right' : 오른쪽 데이터 프레임의 키열에 속하는 데이터 값을 기준으로 병합한다.

```
Import pandas as pd
```

```
Emp = pd.read_csv("c\\wwdata\\wemp3.csv")
```

```
Dept = pd.read_csv("c\\wwdata\\wdept3.csv")
```

```
Result = pd.merge(emp, dept, on = 'deptno', how = 'left')
```

```
Print (result[['emane', 'loc']])
```

문제221. 부서번호, 부서번호별 평균월급을 출력하시오.

```
Select deptno, avg(sal)
 From emp
 Group by deptno;
```

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('deptno')['sal'].mean().reset_index()

print (result)
```

Group by 부터 reset\_index()까지가 한 세트이므로 항상 같이 사용한다.

문제222. 위의 결과에서 평균월급을 출력할 때 정수부분만 출력 하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('deptno')['sal'].mean().reset_index().astype(int)

print (result)
```

Astype(int)는 출력되는 데이터에서 정수형으로 변환 해준다.

문제223. 직업과 직업별 토탈월급을 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('job')['sal'].sum().reset_index()

print (result)
```

문제224. 부서위치, 부서위치별 토탈월급을 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')
result1 = pd.merge(emp, dept, on = 'deptno')
result2 = result1.groupby('loc')['sal'].sum().reset_index()

print (result2)
```

문제225. 아래의 SQL을 판다스로 구현하시오.

```
Select d.loc, sum(e.sal)
 From emp e, dept d
 Where e.deptno (+) = d.deptno
 Group by d.loc;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')
result1 = pd.merge(emp, dept, how = 'right', on = 'deptno')
result2 = result1.groupby('loc')['sal'].sum().reset_index()
```

```
print (result2)
```

문제226. 아래의 SQL을 판다스로 구현하시오.

```
Select deptno, count(*)
 From emp
 Group by deptno;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('deptno')['empno'].count().reset_index()
```

```
print (result)
```

문제227. emp122.csv 를 내려받아 판다스 데이터 프레임으로 만들고 출력하시오.

```
import pandas as pd
```

```
emp12 = pd.read_csv('c:\\data\\emp122.csv')
```

```
print (emp12)
```

문제228. 통신사, 통신사별 인원수를 출력하시오.

```
import pandas as pd
```

```
emp12 = pd.read_csv('c:\\data\\emp122.csv')
result = emp12.groupby('TELECOM')['ENAME'].count().reset_index()
```

```
print (result)
```

문제229. 우리반 테이블에서 통신사가 kt이고 나이가 30살 이상인 학생들의 이름과 나이와 통신사를 출력하시오.

```
import pandas as pd

emp12 = pd.read_csv('c:\\data\\emp122.csv')

print (emp12[['ENAME','AGE','TELECOM']][(emp12['TELECOM'] == 'kt') & (emp12['AGE'] >= 30)])
```

And는 판다스에서 & 이고 or는 판다스에서 | 이다.

그리고 &, | 를 사용할 때는 양쪽 조건에 소괄호를 둘러줘야 한다.

## 68. 반올림수 구하기(round)

파이썬 내장함수 round()는 인자로 입력된 숫자형 자리수 에서 반올림한 결과를 리턴 한다.

Ex)

```
Print(round(16.554)) # 소수점 첫번째 자리에서 반올림 17
Print(round(16.554, 0)) # 소수점 첫번째 자리에서 반올림 17.0
Print(round(16.554, 1)) # 소수점 두번째 자리에서 반올림 16.6
Print(round(16.554, 2)) # 소수점 세번째 자리에서 반올림 16.55
```

파이썬에서 반올림할때 중요하게 알아야 할 내용

Print(round(142.5)) # 143이 나오지 않고 142가 출력된다.

R과 파이썬은 짝수를 출력한다.

142.5 -> 142

187.5 -> 188

파이썬은 기본적으로 이것을 해결하는 함수가 없다.

0.5일때는 짝수로 반올림되고 0.51 일때는 그냥 반올림된다.

Ex)

판다스를 이용하지 않고 파이썬으로만 emp3.csv에서 이름과 월급을 출력하시오.

```
import csv
file = open("c:\\data\\emp2.csv") # os에 있는 emp2.csv를 읽어서 file이라는 변수에 넣는다.
emp_csv = csv.reader(file) # file 변수에 있는 csv 파일을 읽어서 emp_csv변수에 넣는다.

print (emp_csv) # 메모리 주소만 나온다.
```

```
for emp_list in emp_csv: # csv파일의 내용을 한행씩 리스트에 담아서 출력한다.
 print (emp_list[1], emp_list[6])
```

문제230. 판다스를 이용하지 않고 이름과 월급\*12.3를 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
 print (emp_list[1], int(emp_list[5]) * 12.3)
```

Csv파일의 내용을 읽어서 출력할 때 기본적으로 월급도 문자형으로 출력되므로 산술 연산을 하려면 int() 함수를 이용해서 숫자형으로 변환 해주어야 한다.

문제231. 위의 결과를 다시 출력하는데 소수점 첫번째 자리에서 반올림되게 하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
 print (emp_list[1], round(int(emp_list[5]) * 12.3))
```

문제232. 직업이 SALESMAN인 직원들의 이름과 직업을 출력하는데 판다스를 이용하지 않고 emp2.csv을 읽어서 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

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

문제233. 부서번호가 20번인 직원들의 이름과 월급과 부서번호를 출력하시오.

```
판다스
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp[['ename', 'sal', 'deptno']][emp['deptno'] == 20]

print(result)
```

```

판다스 x
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
 if emp_list[-1] == '20':
 print (emp_list[1], emp_list[5], emp_list[-1])

```

## 69. 실수형 자료를 정수형 자료로 변환하기(int)

파이썬의 변환함수 정리

|      | 데이터 유형 | 변환함수    |
|------|--------|---------|
| 정수형  | Int    | Int()   |
| 문자형  | Str    | Str()   |
| 실수형  | Float  | Float() |
| 리스트형 | List   | List()  |
| 튜플형  | Tuple  | Tuple() |

코드 작성시 수학연산을 하다보면 정수끼리만 계산해야하는 경우가 있는데 이때 우리가 가진 데이터가 실수형이라면 실수형 자료를 정수형으로 변환한 후에 계산을 해주어야 한다.

파이썬 내장함수 int()는 인자로 입력된 실수형 자료를 정수형 자료로 변환해준다.

Int()는 입력된 실수형 자료의 소수부분은 버리고 정수부분만 취하여 정수값으로 리턴한다.

Ex)

```
Print(int(-5.4)) # -5
```

문제234. 판다스를 이용해서 emp3.csv를 읽어서 이름과 월급을 출력하는데 월급을 출력할 때 소수점 이하는 버리고 정수부분만 출력하시오.

```

import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

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

```

Int()함수를 따로 쓰지 않아도 정수형으로 출력된다.

판다스를 이용하지 않았을때와는 다르게 숫자는 바로 숫자형으로 출력해준다.



## 70. 정수형 자료를 실수형 자료로 변환하기(float)

이미지 처리나 공학용 프로그램을 작성할 때 실수형 끼리만 계산해야하는 경우가 많은데 이때 우리가 가진 데이터가 정수형이라면 정수형 자료를 실수형 자료로 변환한 후에 계산해 주어야 한다.

파이썬 내장함수 float()은 인자로 입력된 정수형 자료를 실수형으로 변환해 준다.

Ex)

```
Print (float(10)) # 10.0
```

문제235. 판다스를 이용해서 이름과 월급을 출력하는데 월급을 출력할 때 실수형으로 출력하시오.

```
import pandas as pd
```

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

```
emp['sal'] = emp['sal'].apply(float)
```

```
emp 데이터프레임의 sal컬럼의 데이터를 float(실수형)으로 변환해서
```

```
emp 데이터 프레임에 sal 컬럼의 데이터로 변경
```

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

emp['sal'].apply는 emp 데이터 프레임에 sal 시리즈에 apply 함수를 적용해서 데이터 유형을 변경할 수 있다.

emp['sal'].apply(float) 이렇게 하면 데이터 유형을 실수형으로 변경하는 것이다.

## 71. 정수 리스트에서 소수만 걸러내기(filter)

파이썬 내장함수인 filter() 는 리스트와 같은 자료형에서 특정 조건을 만족하는 값만 편리하게 추출할 수 있는 방법을 제공한다.

Filter()의 첫번째 인자는 특정 조건의 값을 추출하는 함수가 입력되면 두번째 인자(입력매개변수)리스트와 같은 자료형이 입력된다.

Ex)

숫자가 나열되어 있는 리스트에서 짝수만 추출해내는 코드를 작성

1. 숫자를 입력하면 짝수이면 결과를 출력하고 홀수면 출력하지 않는 함수를 생성

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

```
Def get_even(num):
```

```
 If num%2 == 0:
```

```
 Return num
```

Else:

Return # 리턴 다음에 아무것도 쓰지 않아서 리턴되지 않는다.

Print (get\_even(2)) # 2

Print (get\_even(7)) # None

2. Filter 함수와 get\_even 함수를 이용해서 위의 a 리스트에서 짝수를 추출하기

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

Result = filter(get\_even, a) # result = filter(함수이름, 리스트이름)

Print (list(result)) # list 함수로 리스트 변환해야 결과가 출력된다.

문제236. 아래의 리스트에서 숫자가 300 이상이면 출력하고 300 미만이면 출력되지 않게 하시오.

b = [100, 352, 254, 456, 123, 234, 567, 903]

b = [100, 352, 254, 456, 123, 234, 567, 903]

```
def get_over(num):
```

```
 if num >= 300:
```

```
 return num
```

```
 else:
```

```
 return
```

```
result = filter(get_over, b)
```

```
print (list(result))
```

문제237. 우리반 데이터에서 나이가 30살 이상인 나이만 따로 리스트로 출력하시오.

```
import pandas as pd
```

```
emp12 = pd.read_csv('c:\\data\\emp122.csv')
```

```
print (list(emp12['AGE'][emp12['AGE'] >= 30]))
```

## 72. 최대값, 최소값 구하기(max, min)

Max()와 min()은 인자로 입력된 자료에서 최대, 최소값을 구해주는 함수이다.

Max()와 min()의 인자로 리스트와 같은 시퀀스 자료가 입력되면 시퀀스 자료의 요소 가운데 최대값 또는 최소값을 구하여 리턴한다.

| 오라클   | 파이썬  |
|-------|------|
| Max   | Max  |
| Min   | Min  |
| Count | Len  |
| Sum   | Sum  |
| Avg   | Mean |

Ex)

A = [8, 7, 12, 55, 21, 34, 15, 9, 22]

Print (max(a))

문제238. 사원테이블에서 최대월급을 출력하시오.

판다스

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
```

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

판다스 x

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
```

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

```
a = []
```

```
for emp_list in emp_csv:
```

```
 a.append (int(emp_list[5]))
```

```
print (max(a))
```

문제239. 우리반 데이터(emp122.csv) 에서 최소나이를 출력하시오.

판다스

```
import pandas as pd
```

```
emp12 = pd.read_csv('c:\\data\\emp122.csv')
```

```
print (emp12['AGE'].min())
```

판다스 x

```
import csv
```

```
file = open('c:\\data\\emp1222.csv')
```

```
emp12 = csv.reader(file)
```

```
a = []
```

```
for emp12 in emp12:
 a.append (int(emp12[2]))

print (min(a))
```

emp122.csv를 복사해서 emp1222.csv로 붙여넣는다.

## 73. 판다스에서 결측치(NaN) 확인하기

NaN : Not a Number

Emp3.csv에 comm처럼 결측치(NaN)를 처리하는 판다스 함수를 배운다.

Ex)

판다스를 이용해서 이름과 커미션을 출력하시오.

```
Import pandas as pd
```

```
Emp = pd.read_csv("c:\\data\\emp3.csv")
Print (emp[['ename', 'comm']])
```

문제240. 커미션이 결측치(NaN)인 직원들의 이름과 커미션을 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'comm']][emp['comm'].isnull()])
```

문제241. 커미션이 결측치(NaN)이 아닌 직원들의 이름과 커미션을 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')

print (emp[['ename', 'comm']][~emp['comm'].isnull()])
```

데이터 받았으면 데이터 분석을 하기 전에 데이터 전처리를 해야 하는데 전처리중에서 결측치 확인하는 단계가 있다.

문제242. emp3.csv에 결측치가 있는지 확인하시오.

```
import pandas as pd
```

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

print (emp.isnull())
```

문제243. emp3.csv에 결측치가 몇개인지 확인하시오.

```
import pandas as pd

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

print (emp.isnull().sum())
```

결측치가 있으면 데이터 분석하기가 어렵고 머신러닝을 이용한 데이터 분석인 경우 좋은 머신러닝 모델이 나오기 어렵기 때문에 결측치를 처리해줘야 한다.

문제244. 타이타닉 데이터에 결측치가 어느 컬럼에 많은지 확인하시오.

```
import pandas as pd

tat = pd.read_csv("c:\\data\\train.csv")

print (tat.isnull().sum())
```

## 74. 판다스에서 파생변수 추가하는 방법

파생변수란 기존 데이터를 가지고 새롭게 가공해서 만든 새로운 컬럼이다.

Ex)  
판다스에서 컬럼 추가하는 방법

Emp 데이터 프레임에 sal 의 데이터와 똑같은 데이터로 sal2라는 컬럼을 추가하시오.

```
Import pandas as pd
Emp = pd.read_csv("c:\\data\\emp3.csv")
Emp['sal2'] = emp['sal']
emp 데이터 프레임에 sal2 컬럼을 추가하는데 데이터는 emp 데이터 프레임에 sal로 한다.

Print (emp)
```

문제245. 판다스를 이용해서 이름과 부서위치를 출력하시오.

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
dept = pd.read_csv('c:\\data\\dept3.csv')
result = pd.merge(emp, dept, on = 'deptno')

print (result[['ename', 'loc']])
```

문제246. emp 데이터 프레임에 loc 컬럼을 추가하고 해당 사원의 부서 위치로 값을 갱신하시오.

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
dept = pd.read_csv("c:\\data\\dept3.csv")
result = pd.merge(emp, dept, how = 'inner', on = 'deptno')
emp['loc'] = result['loc']

print (emp)
```

파생변수를 추가하는 이유

- 머신러닝이 예측을 잘하려면 좋은 데이터를 주고 학습시켜야 한다.

Ex)

Emp 테이블에서 퇴사할 것 같은 사원이 누구인지 예측하시오.

자기의 월급이 자기가 속한 직업의 평균월급보다 더 작은 월급을 받는 사원이면 퇴사할 가능성이 높다.

직업별 평균월급이 emp 데이터 프레임에 추가되어 있으면 머신러닝이 예측하기 좋은 데이터가 추가된 것이다.

문제247. 직업, 직업별 평균 월급을 판다스로 출력하시오.

```
import pandas as pd

emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('job')['sal'].mean().reset_index()
result['sal'] = result['sal'].apply(int)

print (result)
```

문제248. emp와 result를 서로 조인해서 조인된 전체 데이터 프레임을 출력하시오.

```
import pandas as pd

emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('job')['sal'].mean().reset_index()
result['sal'] = result['sal'].astype(int)
```

```
result2 = pd.merge(emp, result, how='inner', on='job')
```

```
print (result2)
```

Emp에도 sal이 있고 result 에도 sal 이 있어서 emp의 sal의 컬럼명이 sal\_x로 변경되었고 result 의 sal은 sal\_y로 변경 되었다.

Sal\_y는 해당 직업의 평균 월급이다.

문제249. emp 데이터 프레임에 컬럼을 추가하는데 job\_avgsal 로 추가 하고 문제 248번에서 구한 직업별 평균 월급인 result2[sal\_y]의 값으로 값을 갱신하시오.

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('job')['sal'].mean().reset_index()
result['sal'] = result['sal'].astype(int)
result2 = pd.merge(emp, result, how='inner', on='job')
emp['job_avgsal'] = result2['sal_y']
```

```
print (emp)
```

문제250. 데이터 프레임에 해당 사원이 근무하는 부서번호의 평균월급 을 sal\_avg라는 이름으로 파생변수를 생성하시오.

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.groupby('deptno')['sal'].mean().reset_index()
result['sal'] = result['sal'].astype(int)
result2 = pd.merge(emp, result, how='inner', on='deptno')
emp['sal_avg'] = result2['sal_y']
```

```
print (emp)
```

문제251. 판다스로 emp데이터 프레임을 출력하는데 월급이 높은 사원 부터 출력하시오.

```
import pandas as pd
```

```
emp = pd.read_csv("c:\\data\\emp3.csv")
result = emp.sort_values('sal',ascending=False)
```

```
print (result)
```

판다스를 이용해서 데이터를 정렬하려면 위와 같이 sort\_values 함수를 이용하면 된다.

Ascending = True : 오름차순

Ascending = False : 내림차순

문제252. 아래의 SQL을 판다스로 구현하시오.

```
Select job, sum(sal)
 From emp
 Group by job
 Having sum(sal) >= 6000
 Order by sum(sal) desc;
```

```
import pandas as pd
```

```
emp = pd.read_csv('c:\\data\\emp3.csv')
result = emp.groupby('job')['sal'].sum().reset_index()
result1 = result.sort_values('sal', ascending = False)
```

```
print (result1[result1['sal'] >= 6000])
```



# 20.12.08

2020년 12월 8일 화요일 오전 9:33

## 75. 문자열에서 특정 위치의 문자 얻기

문자열에서 특정 위치의 문자를 얻는 방법은 인덱싱을 이용하는 것이다.  
인덱스는 0부터 시작한다. 파이썬 인덱스는 음수도 가능하다.

```
Ex)
Print('scott'[0]) # s
Print('scott'[2]) # o
```

```
A = 'scott'
Print (a[0]) # s
Print (a[2]) # o
```

문제253. 아래의 txt 변수에서 w를 출력하시오.

```
txt = 'A tale that was not right'

txt = 'A tale that was not right'

print (txt[12])
```

문제254. 위의 txt변수에서 맨 끝의 철자인 t를 출력하시오.

```
txt = 'A tale that was not right'

print (txt[-1])
```

문제255. emp3.csv에서 이름만 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

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

문제256. 위의 출력된 이름의 데이터 유형이 무엇인지 확인하시오.

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
```

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

문제257. 문제255번을 다시 수행해서 emp2.csv에서 이름을 출력하는데 이름의 첫번째 철자만 출력하시오.

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
 print (emp_list[1][0])
```

문제258. 위의 결과를 다시 출력하는데 첫번째 철자가 소문자로 출력되게 하시오.

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
```

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

## 76. 문자열에서 지정한 구간의 문자열 얻기(슬라이싱)

문자열에서 특정 구간에 있는 문자열을 얻으려면 슬라이싱을 이용하면 된다.

Ex)

Print ('scott'[0:2]) # sc 0부터 2 미만까지 출력

A = 'scott'

Print (a[0:2]) # sc

Print (a[3:]) # tt 3부터 문자 끝까지 출력

Print (a[:3]) # sco 문자 처음부터 3 미만까지 출력

문제259. 아래의 SQL을 파이썬으로 구현하시오.

(판다스 이용 x)

```
Select ename, substr(ename, 1, 3)
From emp;
```

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

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

## 77. 문자열에서 홀수 번째 문자만 추출하기

주어진 문자열에서 홀수 번째 문자만 추출하는 방법은 슬라이싱의 스텝을 이용하면 된다.

```
Ex)
Txt = 'aAbBcCdDeEfGhHiJjKkK'
Result = txt[0:] # 0번째 부터 문자 끝까지 출력
Print (result)
Result2 = txt[0: :2] # 문자열에서 2칸씩 건너뛰면서 출력
```

문제260. 위의 txt문자열에서 짝수번째 철자들만 출력하시오.

```
txt = 'aAbBcCdDeEfGhHiJjKkK'

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

## 78. 문자열을 거꾸로 만들기

슬라이싱을 이용하면 매우 간단하게 거꾸로 된 문자열을 얻을 수 있다.  
문자열 txt를 처음부터 끝까지 스텝 -1로 슬라이싱 하면된다.

```
Ex)
Txt = 'aAbBcCdDeEfGhHiJjKkK'

Print (txt[:]) # 문자열 전체 출력
Print (txt[: :]) # 문자열 전체 출력
Print (txt[: :1]) # 문자열 전체 출력

Result = txt[: :-1] # 문자열 전체를 뒤에 철자부터 앞으로 출력
Print (result)
```

문제261. 이름을 출력하는데 이름의 철자를 거꾸로 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
```

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

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

## 79. 두 개의 문자열 합치기(+)

두개의 문자열을 합치는 방법은 매우 간단하다.

문자열 1 + 문자열 2 + 문자열 3 으로 + 연산자를 이용하면 된다.

Ex)

A = 'scott'

B = 'king'

Print (a+b) # scottking

문제262. 아래의 SQL을 파이썬으로 구현하시오.

```
Select ename || job
 From emp;
```

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
 print (emp_list[1] + emp_list[2])
```

문제263. 아래와 같이 결과가 출력되게 하시오.

```
Select ename || '의 직업은 ' || job || ' 입니다.'
 From emp;
```

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
 print (emp_list[1] + '의 직업은 ' + emp_list[2] + ' 입니다.')
```

## 80. 문자열을 반복해서 새로운 문자열로 만들기(\*)

Ex)

Print ('★'\*10) # ★★★★★★★★★★

Print ('여러분'\*10) # 여러분 여러분 ... 여러분

문제264. 숫자 1부터 5까지 출력하시오.

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

문제265. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자까지 숫자가 출력되게 하시오.

```
num = int(input('숫자를 입력하시오.'))
```

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

문제266. 위의 코드를 수정해서 숫자를 물어보게하고 아래와 같이 ★이 출력되게 하시오.

```
num = int(input('숫자를 입력하시오.'))
```

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

문제267. 아래와 같이 사각형이 출력되게 하시오.

가로의 숫자를 입력하시오. 3

세로의 숫자를 입력하시오. 5

```
★★★
★★★
★★★
★★★
★★★
```

```
num1 = int(input('가로의 숫자를 입력하시오.'))
```

```
num2 = int(input('세로의 숫자를 입력하시오.'))
```

```
for i in range (1, num2 + 1):
 print (num1 * '★')
```

**81. 문자열에서 특정 문자가 있는지 확인하기(in)**

문자열에서 특정 문자가 있는지 없는지 확인하려면 in 키워드를 이용한다.

Ex)

```
Txt = 'abcdefghijklmnopqr'
```

```
If 'b' in txt:
```

```
 Print ('존재합니다.')
```

```
Else:
```

```
 Print ('존재하지 않습니다.')
```

문제268. emp2.csv에서 이름을 출력하는데 이름에 S를 포함하고 있는  
사원들의 이름을 출력하시오.

```
Select ename
```

```
From emp
```

```
Where ename like '%S%';
```

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
```

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

```
for emp_list in emp_csv:
```

```
 if 'S' in emp_list[1]:
```

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

문제269. 문제268번 코드를 수정하여 출력하는데 이름에 S자가 포함된  
사원들 이름이 몇명인지 출력하시오.

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
```

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

```
a = []
```

```
for emp_list in emp_csv:
```

```
 if 'S' in emp_list[1]:
```

```
 a.append (emp_list[1])
```

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

## 82. 문자열에서 특정 문자열이 있는지 확인하기(in)

겨울왕국 스크립트에서 elsa 라는 단어(문자열)가 몇 번 나오는지 카운트 하시오.

Ex)

겨울왕국(winter.txt) 스크립트를 파이썬으로 불러와서 출력하시오.

```
winter = open ("c:\\data\\winter.txt")
for i in winter:
 print (i)
```

Ex)

출력되는 겨울왕국 스크립트를 전부 소문자로 출력하시오.

```
winter = open ("c:\\data\\winter.txt")
for i in winter:
 print (i.lower())
```

Ex)

위에서 소문자로 출력된 스크립트 한행을 하나 가져와서 아래와 같이 실행하시오.

```
if 'elsa' in 'elsa then freezes the fountain in a beautiful design and adds':
 print ('존재합니다.')
else:
 print ('존재하지 않습니다.')
```

Ex)

겨울왕국 스크립트에 elsa가 몇번 나오는지 카운트 하시오.

잘못된 코드

```
winter = open ("c:\\data\\winter.txt")
cnt = 0
```

```
for i in winter:
 if 'elsa' in i.lower():
 cnt += 1
print (cnt) # 318
```

위의 코드는 한 행에 elsa가 두번 나와도 cnt는 한개만 카운트 된다.

Ex)

Winter.txt를 한행씩 읽어서 어절 단위로 출력되게 하시오.

```
winter = open("c:\\data\\winter.txt")
winter2 = winter.read().split(' ') # 스크립트를 공백단위로 분리
for i in winter2:
 print (i)
```

Ex)

위의 어절들이 소문자로 출력되게 하시오.

```
winter = open("c:\\data\\winter.txt")
winter2 = winter.read().split(' ') # 스크립트를 공백단위로 분리
for i in winter2:
 print (i.lower())
```

문제270. 겨울왕국 스크립트에서 elsa가 몇번 나오는지 확인하시오.

#문제 270.1

```
winter = open("c:\\data\\winter.txt") # winter.txt 를 winter로 불러온다.
```

```
winter2 = winter.read().split("\n") # 행단위로 분리하는 코드
```

```
cnt = 0
```

```
for i in winter2:
```

```
 for k in i.split(' '):
```

```
 if 'elsa' in k.lower():
```

```
 cnt += 1
```

```
print (cnt)
```

#문제 270.2

```
winter = open("c:\\data\\winter.txt") # winter.txt 를 winter로 불러온다.
```

```
winter2 = winter.read().split() # split() 안에 아무것도 쓰지않으면 어절별로 나뉜다.
```

```
cnt = 0
```

```
for i in winter2:
```

```
 if 'elsa' in i.lower():
```

```
 cnt += 1
```

```
print (cnt)
```

스크립트를 행단위로 먼저 분리를 고나서 어절별로 분리를 해야한다.

## 83. 문자열 길이 구하기(len)

Ex)

```
A = 'scott'
```

```
Print (len(a)) # 5
```

문제271. 이름, 이름의 길이를 출력하시오.

```
Select ename, length(ename)
```

```
From emp;
```

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
```

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

```
for emp_list in emp_csv:
```

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

## 84. 문자열이 알파벳인지 검사하기(isalpha)



문자열은 문자나 숫자, 기호들로 구성이 된다. 코드를 작성하다 보면 특정 문자열이 한글이나 알파벳과 같이 사람의 언어를 표현하기 위해 사용되는 문자로만 구성되어 있는지 확인해야하는 경우가 있다. 파이썬 문자열 객체가 제공하는 메소드인 `isalpha()` 는 주어진 문자열이 사람의 언어 문자로만 구성되어 있는지 확인해준다.

Ex)

```
txt1 = 'Warcarft three'
```

```
txt2 = '안녕'
```

```
txt3 = '3PO'
```

```
print(txt1.isalpha()) # False 공백이 있어서 False가 출력
```

```
print(txt2.isalpha()) # True 한글이나 다른나라 언어도 True가 출력
```

```
print(txt3.isalpha()) # False 숫자 3때문에 False가 출력
```

문제272. jobs.txt를 한행씩 출력하시오.

```
file = open("c:\\data\\jobs.txt", encoding = 'UTF8')
```

```
stev = file.read().split("\n") # 한 행씩 분리
```

```
for i in stev:
```

```
 print(i)
```

위의 코드를 실행했는데 cp949 에러가 나면 encoding 에 UTF8, ANSI, CP949를 작성한다.

문제273. 위에서 한행씩 출력하고 있는 스크립트를 철자 하나씩 출력하시오.

```
file = open("c:\\data\\jobs.txt", encoding = 'UTF8')
```

```
stev = file.read().split("\n") # 한 행씩 분리
```

```
for i in stev:
```

```
 for k in i: # 스크립트 한 행을 읽어서 철자를 하나씩 불러온다.
```

```
 print(k)
```

문제274. 위에서 출력된 철자가 알파벳이면 cnt를 증가시켜서 스티븐 잡스의 연설문에 알파벳이 몇개가 있는지 확인하시오.

```
file = open("c:\\data\\jobs.txt", encoding = 'UTF8')
```

```
stev = file.read().split("\n")
```

```
cnt = 0
```

```
for i in stev:
```

```
 for k in i:
```

```
 if k.isalpha() == True:
```

```
 cnt += 1
```

```
print (cnt)
```

스티븐 잡스 연설문에서 숫자, 공백문자, 마침표와 같은 구두점을 뺀 알파벳 철자만 카운트 한다.

## 85. 문자열이 숫자인지 검사하기(isdigit)

문자열 객체의 `isdigit()` 메소드는 문자열을 구성하는 요소가 모두 숫자인지 체크하고 `True` 또는 `False` 로 리턴한다.

|   | 내장함수                   | 설명              |
|---|------------------------|-----------------|
| 1 | <code>isalpha()</code> | 알파벳(한글)이 있는지 확인 |
| 2 | <code>isdigit()</code> | 숫자가 있는지 확인      |
| 3 | <code>isspace()</code> | 공백이 있는지 확인      |

문제275. 스티븐잡스의 연설문에서 숫자가 몇 개 있는지 확인하시오.

```
file = open("c:\\data\\jobs.txt", encoding = 'UTF8')
stev = file.read().split('\n')
cnt = 0
```

```
for i in stev:
 for k in i:
 if k.isdigit() == True:
 cnt += 1
```

```
print (cnt)
```

## 86. 문자열이 알파벳 또는 숫자인지 검사하기(isalnum)

알파벳(한글)과 숫자를 동시에 확인하는 문자열 함수는 `isalnum`이다.

```
Ex)
a = 'A story is 2003'
for i in a:
 if i.isalnum() == True:
 print (i)
```

캐냐 은행의 데이터 분석 사례를 파이썬 코드로 구현  
(긍정단어집, 부정단어집을 다운로드)

```
Ex)
긍정단어 (positive-words.txt)를 파이썬으로 읽어서 한행씩 출력하시오.
```

```
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n")

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

Ex)

아래의 pos를 프린트하시오.

```
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n")

print(pos)
```

긍정단어들이 리스트에 담겼으며 리스트 이름은 pos 이다.

Ex)

아래의 단어가 긍정단어 리스트들 중에 있는지 확인하시오.

```
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n")

if 'wonderful' in pos:
 print('존재합니다.')
else:
 print('존재하지 않습니다.')
```

Ex)

스티븐 잡스 연설문(jobs.txt)를 읽어서 한단어씩 출력하시오.

```
stev = open("c:\\data\\jobs.txt", encoding = 'UTF8')
stev2 = stev.read().split()
for i in stev2:
 print(i)
```

문제276. 스티븐 잡스 연설문에는 긍정단어가 몇개가 있는지 확인하시오.

```
file = open("c:\\data\\jobs.txt", encoding = 'UTF8')
stev = file.read().split()
positive = open("c:\\data\\positive-words.txt")
pos = positive.read().split("\n")
cnt = 0

for i in stev:
 if i.lower() in pos:
 cnt += 1

print(cnt)
```

## 87. 문자열에서 대소문자 변환하기(upper, lower)

문자열 객체의 upper() 메소드는 문자열에 있는 모든 알파벳을 대문자로 변환한 결과를 리턴한다.  
문자열 객체의 lower() 메소드는 문자열에 있는 모든 알파벳을 소문자로 변환한 결과를 리턴한다.

```
Ex)
txt = 'A lot of things occur each day'
result1 = txt.upper()
result2 = txt.lower()

print(result1) # A LOT OF THINGS OCCUR EACH DAY
print(result2) # a lot of things occur each day
```

문제277. emp2.csv에서 이름과 월급을 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

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

문제278. 이름을 물어보게하고 이름을 입력하면 해당사원의 이름과 월급이 출력되게 하는데 소문자로 입력하던 대문자로 입력하던 출력되게 하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
name = input('이름을 입력하시오.')

for emp_list in emp_csv:
 if emp_list[1].lower() == name.lower():
 print(emp_list[1], emp_list[5])
```

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

| 함수     | 설명                    |
|--------|-----------------------|
| Lstrip | 문자열에서 존재하는 왼쪽 공백을 제거  |
| Rstrip | 문자열에서 존재하는 오른쪽 공백을 제거 |
| Strip  | 문자열에서 존재하는 양쪽 공백을 제거  |

```
Ex)
txt7 = ' 양쪽에 공백이 있는 문자열 '
print (txt7.lstrip()) # 양쪽에 공백이 있는 문자열
print (txt7.rstrip()) # 양쪽에 공백이 있는 문자열
print (txt7.strip()) # 양쪽에 공백이 있는 문자열
```

문제279. 스티븐 잡스 연설문에서는 정관사 the가 몇 번 나오는지 확인 하시오.

```
file = open('c:\\data\\jobs.txt', encoding = 'UTF8')
stev = file.read().split()
cnt = 0

for i in stev:
 if i.lower().strip() == 'the':
 cnt += 1

print (cnt)
```

## 89. 문자열에서 문자 개수 구하기 (count)

문자열 객체의 count() 메소드는 문자열에서 특정 문자의 갯수를 리턴한다.

```
txt = 'A lot of things occur each day. Today is beautiful day'

print (txt.count('day')) # txt에서 day단어의 건수를 확인
```

문제280. 안철수 연설문에서는 국민 이라는 단어가 몇번 나오는지 확인 하시오.

```
file = open("c:\\data\\ahn.txt", encoding = 'UTF8')
ahn = file.read()

print (ahn.count('국민'))
```

## 파이썬에서 막대 그래프 그리는 방법

```
import matplotlib.pyplot as plt

y_value = [21.6, 23.6, 45.8, 77.0]
x_index = [0, 1, 2, 3]

plt.bar(x_index, y_value, color = 'red')
plt.show()
```

문제281. 동전을 10번 던져서 앞면이 2개가 나오는 확률 10개를 가지고 막대그래프를 그리시오.

```
import matplotlib.pyplot as plt

y_value = [0.00191, 0.01, 0.07, 0.16]
x_index = [0, 1, 2, 3]

plt.bar(x_index, y_value, color = 'red')
plt.show()
```

문제282. 위의 그래프의 결과에 제목, x 축 라벨과 y 축 라벨을 같이 출력하시오.

```
import matplotlib.pyplot as plt

y_value = [0.00191, 0.01, 0.07, 0.16]
x_index = [0, 1, 2, 3]

plt.bar(x_index, y_value, color = 'gray')
plt.title('coin Probability')
plt.xlabel('cnt')
plt.ylabel('probability')

plt.show()
```

문제283. <http://cafe.daum.net/oracleoracle/SfWb/345>

```
import random
import matplotlib.pyplot as plt

coin = ['앞면', '뒷면']

def coin_prob(num):
 cnt = 0
 for k in range(10000):
 a = []
 for i in range(10):
 result = random.choice(coin)
 a.append(result)
 if a.count('앞면') == num:
 cnt += 1
 return cnt/10000

for j in range(11):
 y_value = coin_prob(j)
 x_index = j

plt.bar(x_index, y_value, color = 'gray')
plt.title('coin Probability')
```

```
plt.xlabel('cnt')
plt.ylabel('probability')
plt.show()
```

# 20.12.09

2020년 12월 9일 수요일 오전 9:41

## 파이썬 복습

1. 파이썬 자료형 5가지
2. 파이썬 연산자
3. If문 과 loop문
4. 문자열 다루는 함수1
5. 파이썬에서 모듈 사용법
6. 파이썬에서 클래스 사용법
7. 판다스 사용법
8. 예외처리
9. 숫자형 자료를 다루는 함수
10. 문자열 다루는 함수2

## 90. 문자열에서 특정 문자(열) 위치 찾기 (find)

문자열 객체의 find() 함수는 문자열에서 특정 문자나 문자열이 위치하는 인덱스를 리턴한다. Find()에서 찾고자 하는 문자를 입력하면 해당 문자나 문자열이 최초로 나타나는 위치에 대한 인덱스를 리턴한다.

Ex)

```
txt = 'A lot of things occur each day'
word_count1 = txt.find('o')
print (word_count1) # 3
```

3이 출력되는데 3은 o 라는 철자가 위치하는 인덱스 번호를 리턴한다.

```
txt = 'A lot of things occur each day'
word_count2 = txt.find('day')
print (word_count2) # 27
```

Find는 오라클의 instr함수처럼 해당 자리의 인덱스 번호를 출력하는 문자열 함수 이다.

문제284. emp122.csv를 파이썬으로 로드해서 이메일만 출력하시오.

```
import csv
```

```
file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
```

```
for emp12_list in emp12_csv:
```



```
print (emp12_list[6])
```

문제285. 위에서 출력한 이메일에서 @ 의 위치 인덱스 번호를 출력하십시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 print (emp12_list[6].find('@'))
```

문제286. 이메일에서 . 의 위치 인덱스를 출력하십시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 print (emp12_list[6].find('.'))
```

문제287. 이메일을 출력하는데 이메일의 첫번째 철자부터 세번째 철자까지만 출력하십시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 print (emp12_list[6][:3])
```

문제288. 우리반 데이터 이메일을 출력하는데 이메일에서 도메인만 출력하십시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 a = emp12_list[6].find('@')
 b = emp12_list[6].find('.')
 print (emp12_list[6][a+1:b])
```

문제289. 미승이의 이메일인 miseung.hailey@gmail.com 도 같이 출력하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 a = emp12_list[6].find('@')
 b = emp12_list[6].find('.', emp12_list[6].find('@'))
 print (emp12_list[6][a+1:b])
```

문제290. @domain.co.kr 도 domain이 출력되게 하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12_list in emp12_csv:
 a = emp12_list[6].find('@')
 b = emp12_list[6].find('.', emp12_list[6].find('@'))
 print (emp12_list[6][a+1:b])
```

## 91. 문자열을 특정 문자(열)로 분리하기(split)

코드를 작성할 때 가장 많이 접하게 되는 경우가 구분자(separator) 로 구분되어 있는 문자열 파싱(parsing)을 하는 일이다. 이 때 문자열 객체의 split()를 활용하면 구분자를 기준으로 문자열을 쉽게 분리하여 파싱할 수 있다.

```
a = "1,"이준혁",29,"남","경영학과"
print (a) # 1,"이준혁",29,"남","경영학과"
result = a.split(',') # 콤마(,)로 구분해서 리스트에 담는다
print (result) # ['1', '"이준혁"', '29', '"남"', '"경영학과"']
```

1. A = 문자열.split(',') : 문자열을 콤마(,)로 구분해서 구분하는 문자들을 요소로 해서 a 리스트에 담는다.
2. A = 문자열.split(' ') : 문자열을 공백으로 구분해서 구분하는 문자들을 요소로 해서 a 리스트에 담는다.
3. A = 문자열.split() : 문자열을 어절별로 구분해서 구분하는 문자들을 요소로해서 a 리스트에 담는다.

문제291. 아래의 url 변수에 있는 문자열은 슬래시(/)로 구분되어 있는데

이 문자열의 요소를 아래의 리스트처럼 구성하시오.

```
url = 'http://www.naver.com/news/today=20191204'

['http:', '', 'www.naver.com', 'news', 'today=20191204']

url = 'http://www.naver.com/news/today=20191204'
result = url.split('/')

print (result)
```

## 92. 문자열을 특정 문자(열)로 결합하기(join)

문자열 객체의 join()메소드는 split()와는 반대로 문자열이 요소인 리스트를 인자로 받아 리스트 모든 요소를 특정 문자열로 연결하여 새로운 문자열로 만들어 러턴합니다.

```
Ex)
a = ['http:', '', 'www.naver.com', 'news', 'today=20191204']
bond = '/'
url = bond.join(a)
print(url)
```

```
http://www.naver.com/news/today=20191204
```

문자열 함수인 join을 이용해서 리스트의 요소를 불러와서 다시 문자열로 만들 수 있다.

문제292. 우리반 데이터에서 이름만 출력하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)

for emp12 in emp12_csv:
 print (emp12[1])
```

문제293. 위에서 출력된 이름을 a라는 비어있는 리스트에 담고 a 리스트를 출력하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[1])
print (a)
```

문제294. 위의 a 리스트에 담겨진 이름을 하나씩 뽑아서 콤마(,)로 연결해서 아래와 같이 출력되게 하시오.

이준혁,한결,현지연, ... ,홍재연

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[1])
result = ','.join(a)

print (result)
```

문제295. 위의 결과가 김씨부터 출력되게 하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[1])
 a.sort()
result = ','.join(a)

print (result)
```

### 93. 문자열에서 특정 문자(열)를 다른 문자(열)로 바꾸기(replace)

문자열 객체의 replace() 메소드는 문자열에서 특정 문자나 문자열을 다른 문자나 문자열로 변경한다.

```
Ex)
txt = 'My password is 1234'
result = txt.replace('1','0') # txt 문자열의 숫자 1을 숫자 0으로 변경
print (result)

My password is 0234
```

문제296. emp2.csv에서 이름과 월급을 출력하는데 월급을 출력할 때 0 대신에 \* 로 출력하시오.

```
import csv
```

```
file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

for emp in emp_csv:
 print (emp[1], emp[5].replace('0', '*'))
```

문제297. 아래의 리스트를 가지고 아래와 같이 출력하시오.

```
a = ['name:홍길동', 'age:17', 'major:경영학', 'nation:한국']
```

결과 :

name ---> 홍길동

age ---> 17

major ---> 경영학

nation ---> 한국

```
a = ['name:홍길동', 'age:17', 'major:경영학', 'nation:한국']
```

```
for i in a:
 print (i.replace(':', ' ---> '))
```

## 94. 문자열을 바이트 객체로 바꾸기(encode)

```
file = open("c:\\data\\emp122.csv", encoding = 'UTF8')
```

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

문자집합(컴퓨터에서 문자를 담는 저장소)이 미국을 기준으로 제정되었는데 미국에서 제정된 ACSII(미국 정보 교환 표준부호)는 1960년대에 제정된 문자집합으로 이후 개발된 문자 집합의 토대를 이루고 있다. ASCII는 7비트 만을 이용해서 음이 아닌수 (0~127)의 문자집합내의 문자 할당 약속이다.

Ex)

A는 숫자 65 이고, a는 숫자 97 이다.

Ex)

chr함수를 이용해서 숫자 65부터 127까지 어떤 문자인지 출력하시오.

```
for i in range(65,128):
 print (i, chr(i)) # 컴퓨터가 알고있는 숫자가 문자로 무엇인지 출력
```

서유럽이나 아시아에서는 7비트가 맞지 않아서 더 큰 문자셋이 필요했는데 16비트로 된 문자집합을 제공해서 해결했다.

그 문자 집합셋이 바로 CP949, EUC-KR 이다.

# 과 같은 문자를 해결하기 위해 나온 집합셋은 UTF8, UTF16 이다.

문제298. 우리반 데이터에서 이름을 출력하는데 아래와 같이 나이도 같이 출력되게 하시오.

구윤모(26),권세원(36),권준환(28), ... ,황나현(27)

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[1] + '(' + emp12[2] + ')')
 a.sort()
result = ','.join(a)

print (result)
```

## 95. 바이트 객체를 문자열로 바꾸기(decode)

인코딩(encoding)

- 사람이 알아 볼 수 있는 언어를 컴퓨터가 알아 볼 수 있도록 변경

디코딩(decoding)

- 컴퓨터가 알아 볼 수 있는 언어를 사람이 알아 볼 수 있도록 변경

```
Ex)
txt = 'A'
b_txt = txt.encode()
print (b_txt) # binary(이진) 'A'
print (txt) # A
c_txt = b_txt.decode()
print (c_txt) # A
```

file = open("c:\\data\\emp122.csv", encoding = 'UTF8')  
Emp122.csv를 파이썬으로 로드하는데 파이썬으로 load하려면 컴퓨터가 알아 볼 수 있는 언어로 변경해야 한다.

이때 문자집합셋을 유니코드 문자집합셋인 UTF8을 이용해서 인코딩 즉 컴퓨터가 알아 볼 수 있는 언어로 변경하겠다는 뜻이다.

## 96. 순차적인 정수 리스트 만들기(range)

[0, 1, 2, 3] 이나 [100, 101, 102, 103] 과 같이 순차적인 정수 리스트를 만드는 가장 간단한 방법은 파이썬 내장함수인 range()를 이용하는 것이다.

Ex)

```
print (list(range(1, 11)))
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

문제299. 아래와 같이 주사위의 눈을 담은 리스트를 만드시오.

```
print (dice)

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

dice = list(range(1, 7))

print (dice)
```

문제300. 주사위 2개를 만들고 주사위 2개를 동시에 던져서 두 주사위의 눈의 합이 10이 나오는 확률을 구하시오.

(주사위 2개를 동시에 10000번 던지시오.)

```
import random

dice1 = list(range(1, 7))
dice2 = list(range(1, 7))
cnt = 0

for i in range (10000):
 result1 = random.choice(dice1)
 result2 = random.choice(dice2)
 if result1 + result2 == 10:
 cnt += 1

print (cnt/10000)
```

문제301. 아래의 결과를 출력하시오.

[2, 4, 6, 8, 10, 12, 14, 16, 18]

```
a = list(range(2, 20, 2))

print (a)
```

## 97. 리스트에서 특정 위치의 요소 얻기

리스트의 특정 위치에 있는 요소값을 구하려면 인덱싱을 이용해야 한다.  
인덱스는 0부터 시작한다.

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6]]
```

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

Ex)

위의 리스트 a 에서 숫자 2를 출력하시오.

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6]]
```

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

Ex)

위의 리스트 a 에서 숫자 4를 출력하시오.

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6]]
```

```
print (a[5][0]) # 4
```

문제302. 아래의 리스트에서 숫자 7을 출력하시오.

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

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

```
print (b[4][0])
```

## 98. 리스트에서 특정 요소의 위치 구하기(index)

리스트 객체의 index() 메소드는 리스트에서 요소의 값을 알고 있을 때 그 요소가 최초로 나타나는 위치의 인덱스를 리턴한다.

Ex)

```
a = [1, 2, 'a', 'b', 'c', [1, 2, 3]]
```

```
print (a.index(2)) # 1
print (a.index('a')) # 2
```

문제303. 아래의 리스트에서 숫자 4의 인덱스 번호를 출력하시오.

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6]]
```

```
a = [1, 2, 'a', 'b', 'c', [4, 5, 6]]
```

```
print (a.index('4'))
```



ValueError: 4 is not in list

리스트 안에 있는 리스트의 인덱스 번호는 출력할 수 없다.

## 99. 리스트에서 특정 위치의 요소를 변경하기

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print (list_a[0]) # a

list_a[0] = 'z'
print (list_a) # ['z', 'b', 'c', 'd', 'e', 'f', 'g']
```

문제304. list\_a의 알파벳 d를 k로 변경하시오.

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

list_a[list_a.index('d')] = 'k'

print (list_a)
```

## 100. 리스트에서 특정 구간에 있는 요소 추출하기

리스트에서 특정 구간에 있는 요소를 추출하여 새로운 리스트를 만드려면 슬라이싱을 이용하면 된다.

```
Ex)
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print (list_a[0:4]) # ['a', 'b', 'c', 'd']
print (list_a[2:]) # ['c', 'd', 'e', 'f', 'g']
print (list_a[:3]) # ['a', 'b', 'c']
```

문제305. 알파벳 a 부터 z 까지 담은 리스트를 list\_a로 생성하시오.

```
import string # string 모듈 안에 알파벳이 들어있다.

print (string.ascii_lowercase)

list_a = []
for i in string.ascii_lowercase:
 list_a.append(i)

print (list_a)
```

문제306. 위에서 만든 list\_a에서 요소를 검색하는데 맨 끝에 알파벳 z 만 빼고 모두 출력하시오.

```
import string

list_a = []
for i in string.ascii_lowercase:
 list_a.append(i)

print (list_a[:list_a.index('z')])
```

## 101. 리스트에서 짝수 번째 요소만 추출하기

리스트에서 짝수번째 요소를 추출하는 방법은 두번째 요소부터 끝까지 스텝 2 로 슬라이싱 하면 된다.

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
print (list_a[1: : 2]) # ['b', 'd', 'f', 'h']
```

인덱스 번호 1번째 요소부터 끝까지 2스텝으로 읽는다.

문제307. 1부터 100까지의 숫자중에 짝수만 list\_a에 담아서 출력하시오.

```
list_a = list(range(1, 101))

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

## 102. 리스트 요소 순서를 역순으로 만들기 ① (reverse)

리스트 객체의 reverse() 메소드는 리스트의 모든 요소 순서를 거꾸로 만든다.

```
list_a = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
list_a.reverse()
print (list_a)
```

문제308. 우리반 테이블에서 이름을 출력하는데 ㅎ 부터 ㄱ 순으로 출력되게 하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[1])
 a.sort()
 a.reverse()
 result = ','.join(a)
```

```
print (result)
```

문제309. emp2.csv 에서 이름과 월급을 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)

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

문제310. 위의 데이터중 월급을 비어있는 리스트 a 에 담으시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
a = []

for emp in emp_csv:
 a.append (emp[5])

print (a)
```

문제311. 위의 a 리스트에 있는 월급을 월급이 높은 순서대로 정렬해서 a 리스트에 담고 출력하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
a = []

for emp in emp_csv:
 a.append (int(emp[5]))
 a.sort()
 a.reverse()

print (a)
```

## 103. 리스트 요소 순서를 역순으로 만들기 ② (reversed)

리스트 요소 순서를 역순으로 만드는 또 다른 방법은 파이썬 내장 함수인 reversed() 를 이용하는 방법이다.

Reverse()는 인자로 입력된 시퀀스 자료형의 요소 순서를 역순으로 새로운 시퀀스 자료형을 만들어 리턴한다.

Reverse()함수 : 원본 데이터를 역순으로 만든다.

Reversed()함수 : 역순으로 만들지만 원본 데이터를 역순으로 만들지 않는다.

```
list_a = ['a', 'b', 'c', 'd', 'e']
list_a.reverse()
print(list_a) # list_a의 요소를 역순으로 만든다

list_b = ['a', 'b', 'c', 'd', 'e']
result = reversed(list_b) # list_b의 요소를 역순으로 만들어서 result에 저장
print(list(result))
print(list_b)
```

## 104. 리스트 합치기()

두개의 리스트를 연결하여 새로운 리스트를 만드는 방법은 + 연산자를 이용해 두개의 리스트를 더하면 된다.

```
Ex)
listdata1 = ['a', 'b', 'c', 'd']
listdata2 = ['f', 'g', 'h', 'i']
listdata3 = listdata1 + listdata2

print(listdata3)

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

문제312. 아래와 같이 엄마와 아기가 함께하는 수영교실 나이 리스트를 생성하시오.

```
[34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

```
listdata1 = [34]
listdata2 = [2]

print(listdata1 * 10 + listdata2 * 10)
```

## 105. 리스트 반복하기(\*)

List \* n 은 리스트를 n번 반복해서 새로운 리스트를 만든다.

예를들어 list \* 2는 list + list 와 동일한 결과가 출력된다.

```
Ex)
listdata = list(range(3))
result = listdata * 3

print(result)
```

```
[0, 1, 2, 0, 1, 2, 0, 1, 2]
```

문제313. 아래의 엄마와 아기가 함께하는 수영교실의 나이의 평균값을 출력하시오.

```
[34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

```
import numpy as np

listdata1 = [34]
listdata2 = [2]
swim_age = (listdata1 * 10 + listdata2 * 10)
a = np.array(swim_age) # numpy array(배열) 로 만든다.
b = np.mean(a)

print (b)
```

문제314. 아래의 엄마와 아기가 함께하는 수영교실의 나이의 중앙값을 출력하시오.

```
[34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

```
import numpy as np

listdata1 = [34]
listdata2 = [2]
swim_age = (listdata1 * 10 + listdata2 * 10)
a = np.array(swim_age)
b = np.median(a)

print (b)
```

문제315. 아래의 엄마와 아기가 함께하는 수영교실의 나이의 최빈값을 출력하시오.

```
[34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 34, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

```
import numpy as np
from scipy import stats

listdata1 = [34]
listdata2 = [2]
swim_age = (listdata1 * 10 + listdata2 * 10)
a = np.array(swim_age)
b = stats.mode(a)

print (b)
```

문제316. 우리반 나이 데이터를 비어있는 리스트 a 에 담으시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (emp12[2])

print (a)
```

문제317. 우리반 나이데이터의 평균값과 중앙과 최빈값을 출력하시오.

```
import csv
import numpy as np
from scipy import stats

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append (int(emp12[2]))

a1 = np.array(a)

print (np.mean(a1))
print (np.median(a1))
print (stats.mode(a1))
```

## 106. 리스트에 요소 추가하기(append)

리스트 객체의 append() 메소드는 인자로 입력된 값을 리스트의 맨 마지막 요소로 추가한다.

```
Ex)
listdata = []
listdata.append('a')
listdata.append('b')

print (listdata)

['a', 'b']
```

## 파이썬으로 히스토그램 그래프 그리고

히스토그램 그래프는 계급을 가로축에 도수를 세로축에 나타낸 뒤 각 계급의 크기를 가로의 길이로

하고 도수를 세로의 길이로 하는 직사각형을 차례대로 그려서 나타낸 그래프

Ex)

# 1. 평균이 150이고 표준편차가 5인 초등학교 10만명의 키 데이터를 생성

```
import matplotlib.pyplot as plt
import numpy as np
height = np.random.randn(100000) * 5 + 150
```

# 2. 계급의 크기를 나타내는 가로의 길이를 설정

```
bins = [142, 144, 146, 148, 150, 152, 154, 156, 158, 160]
```

# 3. 히스토그램 그래프의 세로축에 해당하는 도수(건수)를 구하는 코드

# 도수 확인을 위한 코드

```
'''
hist, bins = np.histogram(height, bins)
print (hist, bins)
'''
```

# 4. 위의 데이터를 가지고 히스토그램 그래프를 생성

```
plt.grid() # 격자 모양이 생긴다.
plt.hist (height, bins, rwidth = 0.9, alpha = 0.6, color = 'gray')
```

# rwidth : 히스토그램 그래프의 넓이

# alpha : 색깔 투명도

문제318. 우리반 나이 데이터를 가지고 히스토그램 그래프를 그리시오.

X 축의 계급(간격)은 24 ~ 44 (2살 간격)

```
import csv
import matplotlib.pyplot as plt
import numpy as np

file = open('c:\data\emp1222.csv')
emp12_csv = csv.reader(file)
x = list(range(24,45,2))
y = []

for emp12 in emp12_csv:
 y.append(int(emp12[2]))

plt.grid()
plt.hist (y, x, rwidth = 0.9, alpha = 0.6, color = 'gray')
```

20.12.10

2020년 12월 10일 목요일 오전 9:41

## 107. 리스트의 특정 위치에 요소 삽입하기(insert)

리스트 객체의 insert() 메소드는 리스트의 특정 위치의 새로운 요소를 입력한다.

Ex)

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

```
list_a.append('f')
```

```
print (list_a) # ['a', 'b', 'c', 'd', 'e', 'f']
```

```
list_a.insert(3, 'k')
```

```
print (list_a) # ['a', 'b', 'c', 'k', 'd', 'e', 'f']
```

문제319. 아래의 a 리스트의 요소 화성 다음에 소행성을 요소로 입력하시오.

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

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

```
a.insert(a.index('화성') + 1, '소행성')
```

```
print (a)
```

## 108. 리스트의 특정 위치의 요소 제거하기(del)

del 메소드와 remove 메소드의 차이

둘 다 리스트 안의 요소를 삭제하는 것인데 remove는 요소 명으로 삭제하는 것이고 del은 리스트의 인덱스 번호로 삭제하는 것이다.

Ex)

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

```
del(a[4]) # a 리스트에 인덱스 번호 4번에 해당하는 요소를 삭제
```

```
print (a)
```

문제320. 아래의 a 리스트에서 목성을 지우시오.



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

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

```
del(a[a.index('목성')])
```

```
print (a)
```

## 109. 리스트에서 특정 요소 제거하기(remove)

리스트에서 특정 요소의 값을 알고 있을때 그 요소를 제거하는 방법은 리스트 객체의 remove() 메소드를 이용하면 된다.

Ex)

아래의 a 리스트에서 목성을 지우시오

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

```
a.remove('목성')
```

```
print (a)
```

문제321. 아래의 빨간공 2개와 파란공 3개가 들어있는 주머니에서 공을 하나 랜덤으로 빼면 그 공이 주머니에서 지워지게 하시오.

```
box = ['red', 'blue', 'red', 'blue', 'blue']
```

```
import random
```

```
box = ['red', 'blue', 'red', 'blue', 'blue']
```

```
result = random.choice(box)
```

```
if result == 'red':
```

```
 box.remove('red')
```

```
else:
```

```
 box.remove('blue')
```

```
print (result)
```

```
print (box)
```

문제322. 아래의 주머니에서 임의로 하나의 공을 뽑았을 때 그 공이 파란색일 확률을 구하시오. (복원추출)

(공을 뽑는 작업을 10000번 수행하시오.)

```
box = ['red', 'blue', 'red', 'blue', 'blue']
```

```
import random
```

```
box = ['red', 'blue', 'red', 'blue', 'blue']
cnt = 0
```

```
for i in range(10000):
 result = random.choice(box)
 if result == 'blue':
 cnt += 1
```

```
print (cnt/10000)
```

문제323. 파란공 70개와 빨간공 30개가 들어있는 주머니를 만드시오.

```
red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
```

```
print (box)
```

문제324. 위의 box에 요소가 몇개인지 확인하시오.

```
red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
```

```
print (len(box))
```

문제325. 위의 box에있는 요소들을 무작위로 섞으시오.

```
from random import shuffle
```

```
red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
```

```
shuffle(box)
```

```
print (box)
```

문제326. 위의 box에서 공3개를 추출했을때 3개가 모두 파란색일 확률을 구하시오.

(공을 뽑는 작업을 10000번 수행하시오.)

1)

```
from random import shuffle
```

```
red = ['red']
blue = ['blue']
```

```

box = red * 30 + blue * 70
cnt = 0

shuffle(box)

for i in range(10000):
 result1 = random.choice(box)
 result2 = random.choice(box)
 result3 = random.choice(box)
 if result1 == 'blue' and result2 == 'blue' and result3 == 'blue':
 cnt += 1

print (cnt/10000)

```

```

2)
from random import shuffle
import random

red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
cnt = 0

shuffle(box)

for i in range(10000):
 result = random.sample(box, 3) # box에서 공을 3개씩 추출
 if result.count('blue') == 3:
 cnt += 1 # result 리스트 요소에서 blue가 3개이면 cnt의 값을 누적

print (cnt/10000)

```

문제327. 위의 box에서 공을 3개를 뽑았을 때 그중 2개가 blue일 확률을 출력하시오. (비복원추출)

```

from random import shuffle
import random

red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
cnt = 0

shuffle(box)

for i in range(10000):
 result = random.sample(box, 3)
 if result.count('blue') == 2:
 cnt += 1

print (cnt/10000)

```

Random 패키지의 sample 함수는 비복원 추출이다.

문제328. 위의 문제를 복원추출로 수행하시오.

```
import numpy as np

red = ['red']
blue = ['blue']
box = red * 30 + blue * 70
cnt = 0
cnts = 0

for i in range(10000):
 result = list(np.random.choice(box, 3, replace = True))
 # replace = True 복원, replace = False 비복원
 cnts += 1
 if result.count('blue')==2:
 cnt += 1

print(cnt/cnts)
```

문제329. 6개의 제품이 들어있는 상자가 있는데 그 중에 2개가 불량품이라고 했을 때 제품 검사를 위해 3개를 추출했을 때 1개가 불량품일 확률을 출력하시오. (복원추출)

```
import numpy as np

a = ['정상']
b = ['불량']
box = a * 4 + b * 2
cnt = 0
cnts = 0

for i in range(10000):
 result = list(np.random.choice(box, 3, replace = True))
 cnts += 1
 if result.count('불량') == 1:
 cnt += 1

print(cnt/cnts)
```

## 110. 리스트에서 특정 구간에 있는 모든 요소 제거하기

Ex)

```
a = ['태양', '수성', '금성', '지구', '화성', '목성', '토성', '천왕성', '해왕성']
del(a[2:6]) # 인덱스번호 2번부터 6번 미만까지 요소를 지운다
print(a)
```

```
#['태양', '수성', '토성', '천왕성', '해왕성']
```

문제330. 위의 a 리스트의 요소를 다 지우시오.

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

```
del(a[:])
```

```
print(a)
```

문제331. 초등학생 10만명의 체중 데이터를 가지고 히스토그램 그래프를 그리시오.

(초등학생 10만명의 평균 체중은 45kg 이고 표준편차는 5 입니다.  
30kg ~ 60kg 까지 2 kg 간격으로 x 축의 간격(계급)을 정하시오.)

```
import numpy as np
```

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

```
y = np.random.randn(100000) * 5 + 45
```

```
x = list(range(30,61,2))
```

```
plt.grid()
```

```
plt.hist(y, x, rwidth = 0.9, alpha = 0.6, color = 'gray')
```

```
plt.xticks(x)
```

## 111. 리스트에 있는 요소 개수 구하기(len)

파이썬 내장함수인 len()은 시퀀스 자료형의 크기를 구하는 함수이다.

Len()을 리스트에 적용하면 리스트의 모든 요소의 개수를 리턴한다.

Ex)

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

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

```
9
```

문제332. 331번에서 만든 데이터를 모수로 보고 10만명의 체중 데이터에서 100명을 샘플링하여 평균을 구하시오.

```
import numpy as np
```

```
y = np.random.randn(100000) * 5 + 45
```

```
result = np.random.choice(y, 100).mean()
```

```
print (result)
```

문제333. 위의 모집단에서 표본 100명을 추출하여 표본평균을 구하는 작업을 1000번 수행해서 1000개의 표본평균을 a 리스트에 담고 출력하시오.

```
import numpy as np

y = np.random.randn(100000) * 5 + 45
a = []

for i in range(1000):
 result = np.random.choice(y, 100).mean()
 a.append(result)

print(a)
```

문제334. 위의 a 리스트의 요소의 갯수를 출력하시오.

```
import numpy as np

y = np.random.randn(100000) * 5 + 45
a = []

for i in range(1000):
 result = np.random.choice(y, 100).mean()
 a.append(result)

print(len(a))
```

문제335. 위의 표본평균 1000개를 가지고 히스토그램 그래프를 그리시오.

```
import numpy as np
import matplotlib.pyplot as plt

x = list(range(40,51,1))
y = np.random.randn(100000) * 5 + 45
a = []

for i in range(1000):
 result = np.random.choice(y, 100).mean()
 a.append(result)

plt.grid()
plt.hist(a, x, rwidth = 0.9, alpha = 0.6, color = 'gray')
plt.xticks(x)
```

## 112. 리스트에서 특정 요소 개수 구하기(count)

리스트.count('요소') 은 객체 리스트에서 '요소'의 갯수를 구하는 메소드이다.

```
box = ['yellow', 'yellow', 'red', 'red', 'red']
print(box.count('red'))
```

# 3

문제336. 우리반 데이터의 나이를 비어있는 리스트 a 에 입력하고  
우리반 나이 데이터 중에서 27살이 몇명인지 출력하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp12_csv = csv.reader(file)
a = []

for emp12 in emp12_csv:
 a.append(int(emp12[2]))

print(a.count(27))
```

## 113. 리스트 제거하기(del)

리스트 자체를 메모리에서 제거하려면 다음과 같이 del 명령어를 수행한다.

```
Ex)
a = [1, 2, 3, 3, 3, 4]
del a
print(a)
```

```
NameError: name 'a' is not defined
메모리에서 완전히 지우는 명령어
```

문제337. 파이썬에서 정의된 모든 변수들을 확인하시오.

```
print(dir())
```

## 114. 리스트 요소 정렬하기 ① (sort)

List.sort()는 리스트 객체 list의 모든 요소를 정렬한다.

```
Ex)
a = [2, 4, 1, 3, 5]
a.sort()
print(a)
```

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

```
역순으로 정렬
a = [2, 4, 1, 3, 5]
a.sort(reverse=True)
print(a)
```

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

문제338. emp2.csv에서 월급을 리스트 a에 담고 월급을 역순으로 정렬하시오.

```
import csv

file = open('c:\\data\\emp2.csv')
emp_csv = csv.reader(file)
a = []

for emp in emp_csv:
 a.append(int(emp[5]))
 a.sort(reverse = True)

print(a)
```

## 115. 리스트 요소 정렬하기 ② (sorted)

파이썬 내장함수 sorted()를 이용하여 리스트를 정렬할 수 있다.

리스트 객체의 sort() 메소드와 다른점은 리스트 객체의 sort()는 원본 리스트를 정렬한 형태로 변경하지만 sorted()는 원본 리스트는 그대로 두고 정렬한 결과 리스트만 리턴한다.

```
Ex)
a = [2, 1, 3, 5, 4]
b = sorted(a)

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

## 116. 리스트 요소 무작위로 섞기(shuffle)

파이썬 기본 모듈인 random은 난수를 발생하는 모듈이다.

이 모듈이 제공하는 shuffle은 리스트의 요소를 무작위로 섞는데 활용된다.

```
Ex)
from random import shuffle

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
shuffle(a)
```



```
print(a)
```

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

문제339. 아래의 artist라는 리스트에 가수 이름들을 추가하고 music 리스트에 음악 리스트를 추가하시오.

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

```
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
```

문제340. 위의 artist를 키로하고 music를 값으로 해서 아래와 같은 딕셔너리를 구성하시오.

```
{'비틀즈': 'yesterday', '아이유': '너랑나', '마이클 잭슨': 'beat it'}
```

```
gini = {} # 비어있는 딕셔너리를 gini라는 이름으로 생성한다.
```

```
gini['비틀즈'] = 'yesterday'
```

```
gini['아이유'] = '너랑나'
```

```
gini['마이클 잭슨'] = 'beat it'
```

```
print (gini)
```

문제341. 아래의 2개의 리스트를 이용해서 for 문과 zip을 이용해서 gini 딕셔너리를 구성하시오.

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
```

```
music = ['yesterday', '너랑나', 'beat it']
```

```
{'비틀즈': 'yesterday', '아이유': '너랑나', '마이클 잭슨': 'beat it'}
```

```
artist = ['비틀즈', '아이유', '마이클 잭슨']
```

```
music = ['yesterday', '너랑나', 'beat it']
```

```
gini = {}
```

```
for i, k in zip (artist, music):
 gini[i] = k
```

```
print (gini)
```

## 117. 리스트의 모든 요소를 인덱스와 쌍으로 추출하기(enumerate)

파이썬 내장함수 enumerate() 는 시퀀스 자료형을 인자로 받아 각 요소를 인덱스와 함께 쌍

으로 추출할 수 있는 반복가능한 자료인 enumerate 객체를 리턴한다.

Enumerate 객체는 주로 for 문과 함께 자주 사용되고 list()를 이용해서 리스트 객체를 변환할 수 있다.

Ex)

```
emp12 = ['김주원', '김미승', '김홍비', '권세원']
result = list(enumerate(emp12))
```

```
print(result)
```

```
[(0, '김주원'), (1, '김미승'), (2, '김홍비'), (3, '권세원')]
```

Ex)

```
emp12 = ['김주원', '김미승', '김홍비', '권세원']
```

```
for i, k in enumerate(emp12):
 print(i, k)
```

```
#
```

```
0 김주원
```

```
1 김미승
```

```
2 김홍비
```

```
3 권세원
```

문제342. 아래의 music리스트에서 요소를 하나씩 빼내는데 앞에 인덱스 번호도 같이 출력하시오.

```
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
```

```
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
```

```
for i, k in enumerate(music):
 print(i, k)
```

문제343. aritist리스트를 가지고 아래와 같이 결과를 출력하시오.

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

결과 :

```
비틀즈 0
```

```
비틀즈 1
```

```
아이유 2
```

```
아이유 3
```

```
마이클 잭슨 4
```

마이클 잭슨 5

```
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

```
for i, k in enumerate(artist):
 print (k, i)
```

문제344. 아래의 두개의 리스트를 가지고 아래와 같이 결과를 출력하시오.

```
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

결과 :

비틀즈 yesterday

비틀즈 imagine

아이유 너랑나

아이유 마시멜로우

마이클 잭슨 beat it

마이클 잭슨 smooth criminal

```
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
```

```
for i, k in zip (artist, music):
 print (i, k)
```

최종

```
{'비틀즈':['yesterday', 'imagine'], '아이유':['너랑나', '마시멜로우'],
'마이클잭슨':['beat it', 'smooth criminal']}
```

위와 같은 딕셔너리를 구성하려면 그냥 딕셔너리인 {} 하나로 비어있는 딕셔너리를 만들면 안 되고 defaultdict를 이용해야 한다. defaultdict는 collections 패키지에 있어서 아래와 같이 임포트해야 한다.

```
from collections import defaultdict
gini = defaultdict(list)
```

```
gini['비틀즈'].append('yesterday')
print(gini) # defaultdict(<class 'list'>, {'비틀즈': ['yesterday']})
```

```
gini['비틀즈'].append('imagine')
print(gini) # defaultdict(<class 'list'>, {'비틀즈': ['yesterday', 'imagine']})
```

문제345. 위에서 만든 default 딕셔너리에 아이유를 키로 하고 너랑 나, 마시멜로우를 값으로 해서 추가하시오.

```
from collections import defaultdict
gini = defaultdict(list)

gini['아이유'].append('너랑나')
gini['아이유'].append('마시멜로우')

print(gini)
```

문제346. enumerate를 이용해서 gini 딕셔너리를 구현하시오.

```
from collections import defaultdict
gini = defaultdict(list)

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']

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

print (gini)
```

문제347. gini 딕셔너리에서 음악만 추출하시오.

딕셔너리에서 키값을 추출할 때 : 딕셔너리이름.keys()  
딕셔너리에서 값만 추출할 때 : 딕셔너리이름.values()

```
from collections import defaultdict
gini = defaultdict(list)

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']

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

for i in gini.values():
 for k in i:
 print (k)
```

문제348. 위의 음악들을 비어있는 리스트인 a에 담으시오.

```
from collections import defaultdict
```

```

gini = defaultdict(list)

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
a = []

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

for i in gini.values():
 for k in i:
 a.append(k)

print(a)

```

문제349. 아래의 코드를 수행해서 gini.values()의 요소들을 프린트 하시오.

```

from collections import defaultdict
gini = defaultdict(list)

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']

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

for j in gini.values():
 print(j[0])

```

문제350. 아래와 같이 결과를 출력하시오.

Yesterday, 너랑나, beat it, imagine, 마시멜로우, smooth criminal

```

from collections import defaultdict
gini = defaultdict(list)

artist = ['비틀즈', '비틀즈', '아이유', '아이유', '마이클 잭슨', '마이클 잭슨']
music = ['yesterday', 'imagine', '너랑나', '마시멜로우', 'beat it', 'smooth criminal']
a = []
b = []

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

for j in gini.values():
 a.append(j[0])
 b.append(j[1])

```

```
print(','.join(a) + ','.join(b))
```

# 20.12.14

2020년 12월 14일 월요일 오전 9:26

## 118. 리스트의 모든 요소의 합 구하기(sum)

파이썬 내장 함수 `sum()`을 이용해 list의 모든 요소의 합을 출력한다.

Ex)

```
listdata = [2, 2, 1, 3, 8, 4, 3, 9, 2, 20]
```

```
result = sum(listdata)
```

```
print (result)
```

문제351. 아래의 리스트의 짝수 번째 요소의 합을 출력하시오.

```
listdata = [2, 2, 1, 3, 8, 4, 3, 9, 2, 20]
```

```
listdata = [2, 2, 1, 3, 8, 4, 3, 9, 2, 20]
```

```
result = sum(listdata[1::2])
```

```
print (result)
```

문제352. 아래의 리스트에 1부터 10까지의 숫자중에 없는 숫자가 하나 있는데 그 없는 숫자는 무엇인지 확인하시오.

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

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

```
result = sum(range(1,11)) - sum(a)
```

```
print (result)
```

## 119. 리스트 요소가 모두 참인지 확인하기(all, any)

리스트의 모든 요소가 참인지 또는 모든 요소가 거짓인지 판단해야하는 경우 파이썬 내장함수 `all()`이나 `any()`를 사용하면 된다.

`all()`은 인자로 입력되는 리스트의 모든 요소가 참인 경우에만 `True`를 리턴하고 거짓이 하나라도 포함되어 있으면 `False`를 리턴한다.

이와는 달리 `any()`는 인자로 입력되는 리스트의 모든 요소가 거짓인 경우에만 `False`를 리턴하고

참이 하나라도 존재하면 True를 리턴한다.

Ex)

```
listdata1 = [True, True, True]
listdata2 = [True, False, True]
```

```
print (all(listdata1)) # True
print (all(listdata2)) # False
print (any(listdata1)) # True
print (any(listdata2)) # True
```

Ex)

```
listdata1 = [1, 1, 1]
listdata2 = [1, 0, 1]
```

```
print (all(listdata1)) # True
print (all(listdata2)) # False
print (any(listdata1)) # True
print (any(listdata2)) # True
```

True는 숫자로 1이고 False는 숫자로 0으로 표현할 수 있다.

문제353. while loop문을 이용해서 무한루프문을 수행하시오.

```
while True:
 print ('aaa')
```

문제354. 위의 True대신에 숫자 1을 넣고 무한루프문을 수행하시오.

```
while 1:
 print ('aaa')
```

## 120. 사전에 요소 추가하기

1. 문자형과 문자열 함수들
2. 숫자형과 숫자형 함수들
3. 리스트형과 리스트형 함수들
4. 사전형과 사전형 함수들

+ if와 loop문, 모듈과 클래스, 예외처리, 판다스 사용법, 웹스크롤링

사전형은 키 : 값으로 되어진 요소로 구성되어 있다.

사전형은 리스트형처럼 인덱스번호로 요소를 접근하는게 아니라 키값으로 요소의 값에 접근한다.

Ex)



sol = {} # 중괄호를 써서 비어있는 딕셔너리를 생성한다.

sol['태양'] = 'sun'

print (sol)

sol['수성'] = 'mercury'

sol['금성'] = 'venus'

print (sol)

문제355. 아래의 2개의 리스트를 가지고 sol 딕셔너리를 생성하시오.

sol\_eng = ['sun', 'mercury', 'venus', 'earth', 'mars']

sol\_kor = ['태양', '수성', '금성', '지구', '화성']

sol\_eng = ['sun', 'mercury', 'venus', 'earth', 'mars']

sol\_kor = ['태양', '수성', '금성', '지구', '화성']

sol = {}

for i, j in zip(sol\_kor, sol\_eng):

sol[i] = j

print (sol)

## 121. 사전의 특정 요소값 변경하기

사전 자료형의 특정 요소값을 변경하는 방법은 다음과 같다.

Ex)

# 사전형이름['키값'] = '값'

sol['태양'] = 'aaa'

print (sol)

문제356. 아래의 딕셔너리의 값 중에 Fire를 피땀눈물로 변경하시오.

dict = {'방탄소년단':'Fire', '소녀시대':'Gee'}

dict = {'방탄소년단':'Fire', '소녀시대':'Gee'}

dict['방탄소년단'] = '피땀눈물'

print (dict)

문제357. 아래의 딕셔너리의 값중에서 Fire를 피땀눈물로 변경하시오.

```
dict = {'소녀시대':['다시만난세계','Gee'], '방탄소년단':['DNA','Fire']}
```

```
dict = {'소녀시대':['다시만난세계','Gee'], '방탄소년단':['DNA','Fire']}
```

```
dict['방탄소년단'][1] = '피땀눈물'
```

```
print (dict)
```

## 122. 사전의 특정 요소 제거하기(del)

사전 dict에서 특정요소 key:value를 제거하는 방법은 다음과 같다.

Ex)

```
sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```
del sol['태양']
```

```
print (sol)
```

```
{'수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

문제358. 아래의 딕셔너리에서 다시만난세계의 값만 지우시오.

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
del dict['소녀시대'][0]
```

```
print (dict)
```

## 123. 사전의 모든 요소 제거하기(clear)

사전의 모든 요소를 제거하여 빈 사전으로 만드는 방법은 사전 객체의 clear()메소드를 이용한다.

Ex)

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
dict.clear()
```

```
print(dict)
```

```
{}
```

문제359. 위와 같이 딕셔너리의 요소를 지우는게 아니라 딕셔너리 자체를 메모리에서 지우시오.

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
dict = {'소녀시대': ['다시만난세계', 'Gee'], '방탄소년단': ['DNA', '피땀눈물']}
```

```
del dict
```

```
print (dict)
```

## 124. 사전에서 키만 추출하기(keys)

아래의 사전에서 key값만 추출하고 싶으면 사전이름.keys() 라고 하면 된다.

```
sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```
print (sol.keys())
```

```
dict_keys(['태양', '수성', '금성', '지구', '화성'])
```

문제360. 위의 sol 사전에 있는 키를 추출해서 아래와 같이 결과를 출력하시오.

태양,수성,금성,지구,화성,

```
sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}
```

```
for i in sol.keys():
 print (i, end=',')
```

Print의 end 옵션에 준 구분자로 값들을 가로로 출력하면서 구분자로 구분해준다.

문제361. 위의 결과에서 맨끝에 있는 콤마(,)는 출력되지 않게 하시오.

## 125. 사전에서 값만 추출하기(values)

아래의 사전형에서 값에 해당하는 부분만 추출하려면 사전형이름.values()를 사용하면 된다.

```
sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

print (sol.values())

dict_values(['aaa', 'mercury', 'venus', 'earth', 'mars'])
```

문제362. 아래의 sol 딕셔너리에서 값들만 아래와 같이 출력하시오.

```
sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

aaa,mercury,venus,earth,mars,

sol = {'태양': 'aaa', '수성': 'mercury', '금성': 'venus', '지구': 'earth', '화성': 'mars'}

for i in sol.values():
 print (i, end=',')
```

문제363. 아래의 gini 딕셔너리에서 값만 추출하시오.

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'],
'마이클 잭슨': ['beat it', 'smooth criminal']}

['yesterday', 'imagine']
['너랑나', '마슈멜로우']
['beat it', 'smooth criminal']

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it',
'smooth criminal']}

for i in gini.values():
 print (i)
```

문제364. 위의 결과를 다시 출력하는데 결과 리스트 3개중에서 0번째 요소만 출력하시오.

```
yesterday
너랑나
beat it

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it',
'smooth criminal']}

for i in gini.values():
 print (i[0])
```

문제365. gini 딕셔너리에서 음악만 출력하는데 아티스트별로 노래가 겹치지 않게 출력하시오.

```
from collections import defaultdict
artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it', 'smooth criminal']
gini = defaultdict(list)
a = []
b = []

for i,k in enumerate(artist):
 gini[k].append(music[i])
for j in gini.values():
 a.append(j[0])
 b.append(j[1])
bond = ','
result = bond.join(a+b)
print (result)
```

문제366. 아래의 gini 딕셔너리의 값들을 리스트로 만드시오.

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```
['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth criminal']
```

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
gini2 = list(gini.values())

print(gini2)
```

문제367. 위에서 출력된 gini2리스트의 요소들을 shuffle로 섞으시오.

```
from random import shuffle

gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
gini2 = list(gini.values())
shuffle(gini2)

print(gini2)
```

문제368. 위 문제의 리스트 요소를 무작위로 섞는 shuffle 함수를 이용하여 코드를 수행할 때마다 곡이 무작위로 섞여서 출력되게 하시오.  
(단 아티스트별로 노래가 겹치면 안된다.)

```

from collections import defaultdict
from random import shuffle

artist = ['비틀즈','비틀즈','아이유','아이유','마이클 잭슨','마이클 잭슨']
music = ['yesterday','imagine','너랑나','마슈멜로우','beat it', 'smooth criminal']
gini = defaultdict(list)
a = []
b = []

for i,k in enumerate(artist):
 gini[k].append(music[i])
for j in gini.values():
 a.append(j[0])
 b.append(j[1])

bond = ','
shuffle (a)
shuffle (b)
result = bond.join(a+b)

print (result)

```

## 126. 사전 요소를 모두 추출하기(items) 20130 사전에 특정 키가 존재하는지 확인하기(in)

딕셔너리 자료형에서 데이터를 추출하는 방법은 다음과 같다.

1. Key 추출 : dict.keys()
2. 값만 추출 : dict.values()
3. 둘다 추출 : dict.items()

Ex)

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
```

```

print (gini.keys())
print (gini.values())
print (gini.items())

```

```
dict_keys(['비틀즈', '아이유', '마이클 잭슨'])
```

```
dict_values([['yesterday', 'imagine'], ['너랑나', '마슈멜로우'], ['beat it', 'smooth criminal']])
```

```
dict_items([('비틀즈', ['yesterday', 'imagine']), ('아이유', ['너랑나', '마슈멜로우']), ('마이클 잭슨', ['beat it', 'smooth criminal'])])
```

문제369. 위의 gini 딕셔너리의 items를 뽑아서 리스트에 담으시오.

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it',
```

```
'smooth criminal']}]
result = list(gini.items())

print (result)
```

문제370. 위의 result에서 음악 첫번째 곡들만 아래와 같이 출력하시오.

```
Yesterday
너랑나
beat it
```

```
gini = {'비틀즈': ['yesterday', 'imagine'], '아이유': ['너랑나', '마슈멜로우'], '마이클 잭슨': ['beat it', 'smooth criminal']}
result = list(gini.items())

for i in result:
 print (i[1][0])
```

## 127. 사전 정렬하기(sorted)

파이썬 내장함수 sorted()는 사전 자료를 인자로 입력받아 정렬할 수 있습니다. 사전의 각 요소는 키:값 으로 구성되어 있고 sorted()함수에 사전을 인자로 입력하면 기본적으로 사전의 키를 오름차순으로 정렬한 결과를 리스트로 반환한다.

```
Ex)
dict2 = {'소녀시대': '소원을 말해봐', '방탄소년단': 'DNA', '오마이걸': '살짝 설렘'}
```

```
1. dict2 에서 키만 출력
print (dict2.keys()) # dict_keys(['소녀시대', '방탄소년단', '오마이걸'])
```

```
2. dict2의 키값을들 정렬해서 출력
result = dict2.keys()
print (sorted(result)) # ['방탄소년단', '소녀시대', '오마이걸']
```

문제371. 위의 결과를 reverse 되게 해서 출력하시오.

```
dict2 = {'소녀시대': '소원을 말해봐', '방탄소년단': 'DNA', '오마이걸': '살짝 설렘'}
result = dict2.keys()

print (sorted(result, reverse = True))
```

문제372. emp12데이터에서 이름과 통신사를 출력하시오.

```
import csv

file = open('c:\\data\\emp1222.csv')
emp_csv = csv.reader(file)

for emp12 in emp_csv:
 print (emp12[1], emp12[5])
```

Defaultdictionary를 이용해서 딕셔너리의 값을 리스트로 구성하는 코드

```
from collections import defaultdict

telecom = defaultdict(list)
telecom['kt'].append('허혁')
telecom['kt'].append('정다희')
telecom['sk'].append('박혜진')
telecom['sk'].append('김승순')

print(telecom)
```

문제373. emp12데이터에서 통신사를 key로 하고 학생이름을 값으로 해서 defaultdict를 생성하시오.

```
import csv
from collections import defaultdict

telecom = defaultdict(list)
file = open("c:\\data\\emp1222.csv", encoding = "CP949")
emp_csv = csv.reader(file)

for emp12 in emp_csv:
 telecom[emp12 [5]].append(emp12 [1])
print(telecom)
```

문제374. 위에서 구성한 telecom 딕셔너리에서 통신사가 kt인 학생들을 출력하는데 이름이 ㄱ ㄴ ㄷ 순으로 출력되게 하시오.

```
import csv
from collections import defaultdict

telecom = defaultdict(list)
file = open("c:\\data\\emp1222.csv", encoding = "CP949")
emp_csv = csv.reader(file)

for emp12 in emp_csv:
 telecom[emp12 [5]].append(emp12 [1])
result = telecom['kt']

print(sorted(result))
```



## 128. 문자 코드값 구하기(ord)

파이썬 내장함수 ord()는 문자를 컴퓨터가 인식하는 코드값으로 변환한다.

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

문제375. 알파벳 대문자를 출력하시오.

```
import string

print(string.ascii_uppercase)
```

문제376. 위의 코드와 ord() 함수를 이용해서 아래의 결과를 출력하시오.

```
A ---> 65
B ---> 66
C ---> 67
:
Z ---> 90
```

```
import string

result = string.ascii_uppercase

for i in result:
 print (i, '--->', ord(i))
```

## 129. 코드값에 대응하는 문자 얻기(chr)

파이썬 내장함수 chr()은 ord() 함수의 반대기능을 수행한다.

chr()의 인자로 정수값을 입력하면 이 정수값이 해당하는 문자를 리턴한다.

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

문제377. 아래와 같이 결과를 출력하시오.

```
65 ---> A
66 ---> B
67 ---> C
:
90 ---> Z
```

```
for i in range(65, 91):
 print(i, '---->', chr(i))
```

### 130. 문자열로 된 식을 실행하기(eval)

코드를 작성하다 보면 파일에서 읽은 수식이나 문자열을 그대로 실행해야하는 경우가 있다. 예를 들어 '2 + 3' 과 같이 두개의 숫자를 더하는 문자열을 텍스트 파일에서 읽어 이를 실행하여 이 수식의 결과인 5를 구하는 경우이다.

```
Ex)
a = '2 + 3'

print(a) # 2 + 3
print(eval(a)) # 5
```

문제378. 아래의 결과를 출력하시오.

$34 + 256 - 71 * 34 = ?$

```
a = '34 + 256 - 71 * 34'
```

```
print(a, '=', eval(a))
```

문제379. 아래의 리스트의 숫자를 뽑아서 아래의 문자열을 생성하시오.

```
a = [34, 22, 45, 27, 31, 33, 55]
```

$34 + 22 + 45 + 27 + 31 + 33 + 55$

```
a = [34, 22, 45, 27, 31, 33, 55]
b = []
```

```
for i in a:
 b.append(str(i))
result = ' + '.join(b)
```

```
print(result)
```

문제380. 아래의 리스트를 가지고 아래의 결과를 출력하시오.

```
a = [34, 22, 45, 27, 31, 33, 55]
```

$34 + 22 + 45 + 27 + 31 + 33 + 55 = 247$

```
a = [34, 22, 45, 27, 31, 33, 55]
b = []
```

```
for i in a:
 b.append(str(i))
result = ' + '.join(b)

print (result, '=', eval(result))
```

문제381. emp12데이터에서 나이를 읽어들이어 아래와 같이 결과가 출력되게 하시오.

29 + 31 + 35 + 29 + 28 + 28 + 25 + 28 + 27 + 27 + 27 + 26 + 33 + 30 + 27 + 27 + 25 + 28 + 29 + 24 + 24 + 29 + 36 + 26 + 26 + 26 + 44 + 28 + 29 + 28 = ?

```
import csv

file = open("c:\\data\\emp1222.csv", encoding = "CP949")
emp_csv = csv.reader(file)
a = []

for emp12 in emp_csv:
 a.append (emp12[2])
result = ' + '.join(a)

print (result, '=', eval(result))
```

## 131. 이름없는 한줄짜리 함수 만들기(lambda)

파이썬에서 함수를 정하는 방법은 일반적으로 다음과 같다.

```
def 함수이름(입력매개변수):
 실행코드
```

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

```
def add_func(a,b):
 return a + b
```

```
print (add_func(2,3))
```

```
5
```

함수를 만들때 위와 같이 함수 이름을 주고 생성

함수를 생성하는 이유는 함수를 한번 만들면 다음부터는 함수이름을 가지고 함수만 호출 하면 되기 때문에 함수를 생성해서 코드를 작성했다.

지금부터 만들 함수는 함수 이름 없이 함수를 생성하는 코드이다.

함수 이름을 주지 않고 함수를 생성한다는 것은 그 함수는 작성코드에서 한번만 실행하고 앞으로 쓸 일이 없는 경우 함수 이름없이 함수를 생성한다.

변수이름 = lambda 입력매개변수 : 실행문

```
k = lambda a, b : a + b
print (k(2,3)) # 5
```

문제382. 아래의 함수를 이름없는 한 줄 짜리 함수로 만드시오.

```
def gob_func(a,b):
 return a*b
```

```
k = lambda a, b : a * b
```

```
print (k(2, 3))
```

문제383. 아래의 함수를 이름없는 한 줄 짜리 함수로 만드시오.

```
def odd_func(a):
 if a%2 == 0:
 return '짝수 입니다.'
 else:
 return '홀수 입니다.'
```

```
print(odd_func(2)) # 짝수 입니다.
```

```
print(odd_func(3)) # 홀수 입니다.
```

```
k = lambda a : '짝수 입니다.' if a%2 == 0 else '홀수 입니다.'
```

```
print(k(2))
```

```
print(k(3))
```

문제384. 아래의 함수를 lambda로 구현하시오.

```
def high_income(a):
 if a >= 3000:
 return '고소득자 입니다.'
 else:
 return '일반 소득자 입니다.'
```

```
k = lambda a : '고소득자 입니다.' if a >= 3000 else '일반 소득자 입니다.'
```

```
print (k(4000))
```

```
print (k(2000))
```

문제385. 아래의 함수를 lambda로 구현하시오.

```
def child_tall(num):
 if 140 - 1.96 * 5 <= num <= 140 + 1.96 * 5:
 return '신뢰구간 95%안에 있습니다.'
 else:
 return '신뢰구간 95%안에 없습니다.'

print (child_tall(147)) # 신뢰구간 95%안에 있습니다.
print (child_tall(157)) # 신뢰구간 95%안에 없습니다.
```

(평균값 140, 표준편차 5)

신뢰구간 68% = 평균  $\pm$  1 \* 표준편차  
신뢰구간 95% = 평균  $\pm$  1.96 \* 표준편차  
신뢰구간 99% = 평균  $\pm$  2.58 \* 표준편차

```
k = lambda num : '신뢰구간 95%안에 있습니다.' if 140 - 1.96 * 5 <= num <= 140 + 1.96 * 5 else '신뢰구간 95%안에 없습니다.'
```

```
print (k(147))
print (k(157))
```

문제386. 초등학생 키 데이터를 120 ~ 160 사이로 해서 0.001 간격으로 생성하시오.

```
import numpy as np

x = np.arange(120, 160, 0.001)

print (x)
```

문제387. 위에서 만든 40000건의 데이터에 대한 확률밀도 함수값을 출력하시오.

(평균값 140, 표준편차 5)

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

x = np.arange(120, 160, 0.001)
y = norm.pdf(x, 140, 5) # 확률밀도 함수

print (y)
```

문제388. 위의 데이터 중 x값과 y 값을 가지고 시각화 하시오.

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
```

```
x = np.arange (120, 160, 0.001)
y = norm.pdf (x, 140, 5)
```

```
plt.plot (x, y, color = 'red')
plt.show()
```

문제389. 위의 그래프에서 신뢰구간 95%에 해당하는 부분만 색깔로 칠하시오.

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
```

```
x = np.arange (120, 160, 0.001)
y = norm.pdf (x, 140, 5)
```

```
plt.plot (x, y, color = 'red')
plt.fill_between (x, y, where = (x >= 140 - 1.96 * 5) & (x <= 140 + 1.96 * 5), color = 'red', alpha = 0.5)
plt.show()
```

Where에 색칠하는 구간을 지정한다.

문제390. 위의 코드를 이용해서 confidence\_interval 함수를 생성하고 아래와 같이 신뢰구간을 넣으면 해당 신뢰구간에 해당하는 그래프가 출력되게 하시오.

신뢰구간 68% = 평균  $\pm$  1 \* 표준편차

신뢰구간 95% = 평균  $\pm$  1.96 \* 표준편차

신뢰구간 99% = 평균  $\pm$  2.58 \* 표준편차

```
confidence_interval(68)
confidence_interval(95)
confidence_interval(99)
```

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt
```

```
def confidence_interval(num):
```

```
x = np.arange(120, 160, 0.001)
y = norm.pdf(x, 140, 5)
plt.plot(x, y, color = 'red')

if num == 68:
 return plt.fill_between(x, y, where = (x >= 140 - 1 * 5) & (x <= 140 + 1 * 5), color = 'red', alpha = 0.5)
elif num == 95:
 return plt.fill_between(x, y, where = (x >= 140 - 1.96 * 5) & (x <= 140 + 1.96 * 5), color = 'red', alpha
= 0.5)
else:
 return plt.fill_between(x, y, where = (x >= 140 - 2.58 * 5) & (x <= 140 + 2.58 * 5), color = 'red', alpha
= 0.5)
plt.show()

confidence_interval(68)
confidence_interval(95)
confidence_interval(99)
```

# 20.12.15

2020년 12월 15일 화요일 오전 9:46

## 132. 인자를 바꾸어 함수를 반복 호출하여 결과값 얻기(map)

파이썬 내장함수 map()은 리스트 A와 함수 f가 주어지면 리스트 A의 요소를 함수 f에 입력해서 출력하는 결과를 간단하게 출력해 준다.

map(함수명, 함수에 제공할 매개변수 값들)

매개변수의 값들을 바꿔가면서 함수를 반복 실행할 때 유용하게 사용한다.

```
ex)
def gob(x):
 return x*x
```

```
a = [1, 2, 3, 4, 5]
result = map(gob,a)
print (list(result))
```

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

map 함수가 리턴해주는 결과를 보려면 반드시 list함수로 리턴된 값을 변환해서 리스트 형태로 봐야한다.

리스트의 요소 하나하나를 함수에 입력값으로 입력한다(mapping)

문제391. 숫자를 입력해서 해당 숫자가 3000보다 크면 숫자 1을 리턴하게 하고 입력하는 숫자가 3000보다 작다면 0을 리턴하는 함수를 생성하시오.

```
def high_income(x):
 if x >= 3000:
 return 1
 else:
 return 0

print (high_income(5000))
print (high_income(2000))
```

문제392. 위에서 만든 함수를 map함수에 적용해서 아래의 리스트의 요소를 high\_income함수의 입력 매개변수로 입력해서 아래의 결과가



출력되게 하시오.

```
a = [4000, 5000, 2000, 3500, 1000]
```

```
[1, 1, 0, 1, 0]
```

```
a = [4000, 5000, 2000, 3500, 1000]
```

```
def high_income(x):
 if x >= 3000:
 return 1
 else:
 return 0
```

```
result = map(high_income, a)
```

```
print (list(result))
```

문제393. 동전을 지정된 숫자만큼 던져서 앞면이 나오는지 뒷면이 나오는지 출력하는 함수를 생성하시오.

```
import random
```

```
def coin_cnt(x):
 coin = ['앞면', '뒷면']
 for i in range(1, x+1):
 result = random.choice(coin)
 print(result)
```

```
coin_cnt(5)
```

문제394. 위의 함수를 수정해서 숫자를 입력하면 해당 숫자만큼 동전을 던져서 동전이 앞면이 나오는 확률을 출력하시오.

```
import random
```

```
def coin_cnt(x):
 coin = ['앞면', '뒷면']
 cnt = 0
 for i in range(1, x+1):
 result = random.choice(coin)
 if result == '앞면':
 cnt += 1
 print(cnt/x)
```

```
coin_cnt(10000)
```

문제395. 위에서 만든 coin\_cnt함수에 아래의 a 리스트 요소들을 적용해서 결과로 확률이 출력되게 하시오.

```
a = [10, 100, 1000, 10000, 100000]
```

```
import random
```

```
a = [10, 100, 1000, 10000, 100000]
```

```
def coin_cnt(x):
 coin = ['앞면', '뒷면']
 cnt = 0
 for i in range(1, x+1):
 result = random.choice(coin)
 if result == '앞면':
 cnt += 1
 return (cnt/x)
```

```
result1 = map(coin_cnt, a)
```

```
print (list(result1))
```

동전을 던진 횟수가 많아질수록 0.5에 근사한 값이 출력된다.

문제396. 주사위를 던져서 주사위의 눈이 5가 나올 확률을 출력하는 함수를 만들고 아래의 a 리스트에 주사위를 던지는 횟수를 map함수로 적용해서 확률이 1/6으로 근사해지는지 실험하시오.

```
a = [10, 100, 1000, 10000, 100000]
```

```
import random
```

```
a = [10, 100, 1000, 10000, 100000]
```

```
def dice_cnt(x):
 dice = list(range(1, 7))
 cnt = 0
 for i in range(1, x+1):
 result = random.choice(dice)
 if result == 5:
 cnt += 1
 return (cnt/x)
```

```
result1 = map(dice_cnt, a)
```

```
print (list(result1))
```

문제397. 아래의 불량품이 들어있는 박스에서 제품 3개를 뽑았을 때 3개중에 2개가 불량품일 확률을 출력하는 함수를 만드시오.

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

```
import numpy as np

def box_cnt(x):
 box = ['정상', '정상', '불량', '정상', '불량', '정상', '정상', '불량']
 cnt = 0
 for i in range(1, x+1):
 result = list(np.random.choice(box, 3, replace = True))
 if result.count('불량') == 2:
 cnt += 1
 return (cnt/x)

print (box_cnt(10000))
```

문제398. 위에서 만든 box\_cnt 함수에 아래의 a 리스트의 횟수를 map 함수로 적용해서 확률이 출력되게 하시오.

```
a = [10, 100, 1000, 10000, 100000]
```

```
import numpy as np

a = [10, 100, 1000, 10000, 100000]

def box_cnt(x):
 box = ['정상', '정상', '불량', '정상', '불량', '정상', '정상', '불량']
 cnt = 0
 for i in range(1, x+1):
 result = list(np.random.choice(box, 3, replace = True))
 if result.count('불량') == 2:
 cnt += 1
 return (cnt/x)

result1 = map(box_cnt, a)

print (list(result1))
```

### 133. 텍스트 파일을 읽고 출력하기(read)

텍스트 파일을 읽고 그 내용을 화면에 출력하고자 하면 제일 먼저 텍스트 읽기 모드로 파일을 연다.

텍스트 파일을 오픈하면 텍스트 파일을 읽어 내용을 화면에 출력하면 된다.

```
ex)
f = open('c:\\data\\jobs.txt', encoding = 'UTF8')
data = f.read() # 파일을 한번에 전부 읽어오는 함수
```

```
print(data)
f.close()
```

# 파일을 닫는다. 닫는 코드를 쓰지 않으면 스파이더를 실행하고 있으면 파일이 계속 열려있어서 메모리를 계속 차지하고 있게 된다.

스티븐 잡스 연설문은 길지 않아 한번에 읽어와도 관계 없지만 텍스트 파일의 용량이 매우 클 경우 read()로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기시킬수 있다.

웹에서 스크롤링한 데이터를 분석하고자 할 때 위와 같이 open 함수를 이용해서 파이썬 데이터로 불러온다.

문제399. 중앙일보에서 인공지능으로 검색한 기사인 mydata3.txt를 파이썬으로 불러서 출력하시오.

```
f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.read()

print(data)

f.close()
```

문제400. 위의 중앙일보 기사에서 빅데이터라는 단어가 몇번 나오는지 count 하시오.

```
f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.read()
result = data.count('빅데이터')

print(result)

f.close()
```

## 134. 텍스트 파일을 한줄씩 읽고 출력하기 ① (readline)

텍스트 파일의 용량이 매우 클 경우 read()로 한꺼번에 파일의 내용을 읽어들이는 것은 메모리 문제를 야기할 수 있다.

이 경우 텍스트 파일 내용을 한 단위로 읽고 작업을 수행하면 된다.

readline()은 텍스트 파일에서 한 줄을 읽는다.

한 줄을 읽고 나면 파일을 읽기 시작하는 위치는 그 다음줄의 맨 처음이 된다.

```
ex)
f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.readline()

print(data)

f.close()
```

read()함수는 텍스트 파일 전체를 읽어오는 반면 readline()은 텍스트 파일에서 한줄만 읽어 온다.

문제401. 위의 스티븐 잡스 연설문 데이터를 모두 읽어 오시오.

(readline() 함수를 이용)

```
f = open('c:\\data\\jobs.txt', encoding = 'UTF8')
data = f.readline()

while data: # data 변수안에 data가 있으면 True 없으면 False
 print (data)
 data = f.readline() # 그 다음줄을 읽어서 data 변수에 입력

f.close()
```

맨 위의 data = f.readline() 코드는 처음에 한번만 실행되고 그 다음에는 실행되지 않는다. 그 다음부터는 while 문 안에 있는 data = f.readline() 가 작동되어서 반복적으로 스티븐 잡스의 연설문을 한줄씩 읽어서 data 변수에 넣는다.

문제402. 위에서 한줄씩 읽은 데이터에서 인공지능이라는 단어가 나오면 count하게 하시오.

```
f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.readline()

while data:
 print (data.count('인공지능'))
 data = f.readline()

f.close()
```

문제403. 위에서 count한 건수를 누적시켜서 출력하시오.

```
f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.readline()
a = []

while data:
 a.append (data.count('인공지능'))
 data = f.readline()
print (sum(a))

f.close()
```

문제404. 위의 코드를 수정해서 단어를 물어보게 하고 단어를 입력하면 해당 단어가 몇건 나오는지 출력하시오.

```

f = open('c:\\data\\mydata3.txt', encoding = 'UTF8')
data = f.readline()
a = []
x = input('단어를 입력하시오.')

while data:
 a.append (data.count(x))
 data = f.readline()
print ('이 기사에서', x, '단어는', sum(a), '번 나왔습니다.')

f.close()

```

## 135. 화면에서 사용자 입력을 받고 파일로 쓰기(write)

사용자로 부터 입력받은 텍스트를 파일로 저장하려면 파일을 텍스트 쓰기 모드로 열고 파일 객체의 write()를 이용해 데이터를 파일에 기록한다.

```

ex)
text = input('파일에 저장할 내용을 입력하시오.')
f = open('c:\\data\\mydata.txt','w')
w 는 write의 약자이다. c드라이브 밑에 mydata3.txt 파일로 쓰겠다는 뜻

f.write(text)
c.close()

```

위의 코드는 웹에서 스크롤링한 데이터를 파일로 저장할 때 응용되게 된다.

문제405. 아래의 파란색 공과 빨간색 공이 들어있는 박스에서 5개의 공을 뽑았을 때 그 중 2개가 파란색 공일 확률을 출력하는 함수를 만들고 공을 뽑 횟수를 10, 100, 1000, 10000, 100000 했을때의 확률을 map 함수를 이용해서 출력하시오.

파란색 공 : 24개, 빨간색 공 : 26개

```

import numpy as np

box = ['blue'] * 24 + ['red'] * 26
a = [10, 100, 1000, 10000, 100000]

def box_cnt(x):
 cnt = 0
 for i in range(1, x+1):
 result = list(np.random.choice(box, 5, replace = True))
 if result.count('blue') == 2:
 cnt += 1
 return cnt/x

```

```
result = map(box_cnt, a)
```

```
print (list(result))
```

ex)

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

```
result = sorted(listdata) # 리스트의 요소를 정렬
```

```
print(result) # [1, 2, 2, 3, 5, 7, 8]
```

```
f = open('c:\\data\\mydata2.txt', 'w') # mydata2.txt 를 생성
```

```
f.write(str(result)) # result에 있는 내용을 문자로 변환하고 mydata2.txt 로 생성
```

```
f.close()
```

문제406. 아래의 리스트를 mydata9.txt.로 저장하시오.

```
listdata = ['a', 'b', 'c', 'd', 'd', 'f', 'g']
```

```
listdata = ['a', 'b', 'c', 'd', 'd', 'f', 'g']
```

```
result = sorted(listdata)
```

```
print (result)
```

```
f = open('c:\\data\\mydata9.txt', 'w')
```

```
f.write(str(result))
```

```
f.close()
```

## 136. 텍스트 파일에 한줄씩 쓰기(writelines)

파일 객체의 writelines()는 텍스트 문자열이나 텍스트 문자열이 요소로 되어있는 리스트를 인자로 받아 파일에 한 줄씩 기록한다.

리스트가 인자인 경우 writelines()는 리스트의 요소를 하나의 문자열로 결합하여 파일을 한꺼번에 기록한다.

writelines는 리스트 자료형도 파일에 저장할 수 있다.

ex)

```
data = [] # data라는 비어있는 리스트 생성
```

```
while True: # 무한 루프 수행
```

```
 text = input('저장할 내용을 입력하시오.')
```

```
 if text == "": # text에 아무것도 입력하지 않으면 break문을 실행해서 무한루프를 종료
```

```
 break
```

```
 data.append(text + '\n') # text가 엔터와 함께 data 리스트에 append
```

```
f = open('c:\\data\\mydata99.txt', 'w') # mydata99.txt를 생성
```

```
f.writelines(data) # data리스트의 내용을 mydata99.txt에 저장
```

```
f.close()
```

open의 옵션 중 w는 쓰기이고, a는 추가이다.

## 137. 텍스트 파일 복사하기(read, write)

텍스트 파일을 복사하는 방법을 파이썬으로 구현하려면 read와 write를 이용하면 된다.  
read()로 텍스트 파일을 읽고 write로 읽은 내용을 다른 파일 이름으로 저장하면 된다.

```
ex)
###
f = open('c:\\data\\mydata.txt', 'r') # r 은 read
h = open('c:\\data\\mydata_copy.txt', 'w') # w 는 write
data = f.read()
h.write(data)
f.close()
h.close()
```

## 138. 바이너리 파일 복사하기(read, write)

이미지나 동영상도 파이썬으로 복사 붙여넣기 할 수 있다.  
이미지나 동영상을 복사 붙여넣기할때는 파일의 크기가 크므로 한번에 읽어들이 수 없고 일부  
분만 일정한 크기로 읽으면서 복사 붙여넣기를 한다.

```
ex)
bufsize = 1024 # 1Kb 크기의 버퍼 사이즈를 지정
f = open('c:\\data\\lena.png', 'rb') # rb는 read binary의 약자
h = open('c:\\data\\lena_copy.png', 'wb') # wb는 write binary의 약자
data = f.read(bufsize) # 이미지를 1Kb을 읽어서 data에 저장
while data: # data에 data가 발견되는 동안에 루프문을 수행
 h.write(data) # lena_copy.png에 1Kb의 데이터씩 write
 data = f.read(bufsize) # lena.png 에서 1Kb의 데이터를 read
f.close()
h.close()
```

lena.png의 총크기가 334Kb 이므로 1Kb씩 읽어들이어서 쓰는 작업을 334번 수행

## 139. 파일을 열고 자동으로 닫기 (with ~ as )

with ~ as 절을 사용하게 되면 f.close()를 명시하지 않아도 되므로 프로그래머가 실수로  
f.close()를 작성하지 않아서 발생하는 메모리 부족오류를 예방할 수 있다.

1. with ~ as 절을 사용하지 않았을 때

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



```
data = f.readlines()
print (data)
f.close()
```

2. with ~ as 절을 사용했을 때

```
with open('c:\\data\\mydata.txt','r') as f:
 data = f.readlines()
 print(data)
```

with ~ as 절을 사용하면 자동으로 파일 close를 해준다.

## 140. HTML 기본문법

데이터 분석 순서

데이터 수집 - 데이터 유형 및 속성 파악 - 데이터 변환 - 데이터 저장 - 데이터 정제 -  
데이터 분석

html이란 Hyper Text Markup Language의 약자이고 여러개의 태그(tag)를 연결해서 모아놓은 문서이다.

ex)

메모장 열고 아래와 같이 코딩 하시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </p>
</body>
</html>
```

메모장의 파일이름을 a.html로 저장(파일형식은 모든파일로 변경)

ex)

위의 글씨를 진하게 하시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </p>
</body>
</html>
```

글씨를 진하게 하려면 b 태그를 사용한다.

ex)

위의 글씨에 밑줄을 그어보시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"><u> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </u>
</p>
</body>
</html>
```

글씨에 밑줄을 사용하려면 u 태그를 사용한다.

ex)

위의 글씨를 이탤릭체로 변경하시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"><u><i> 양건준님은 오늘 점심시간에 우육탕을 먹었습니다. </i></u>
</p>
</body>
</html>
```

글씨를 이탤릭체로 변경하려면 i 태그를 사용한다.

ex)

p 태그를 추가해서 제목과 내용을 나누시오.

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"><u><i> 양건준님은 오늘 점심 </i></u></p>
<p class = "content"> 양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육
탕을 먹었습니다. 코로나에 대한 걱정도 있었지만 용기내서 먹었습니다. </p>
</body>
</html>
```

제목은 class title에 적고 내용은 class content에 적는다.

ex)

위에서 만든 html문서의 본문에 링크를 거시오.

(링크는 우리반 카페주소로 하시오.)

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"><u><i> 양건준님은 오늘 점심 </i></u></p>
```

```
<p class = "content"> 양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육
탕을 먹었습니다. 코로나에 대한 걱정도 있었지만 용기내서 먹었습니다.
 다음카페
</p>
</body>
</html>
```

class는 html문서의 특정 단위이라고 보면 되고 id는 링크를 줄 때 부여하는 제목인데 id는 값이 unique 하다.

문제407. 다음과 같이 내용과 링크를 수정하시오.

양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육탕을 먹었습니다. 코로나에 대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이지는 바로가기를 누르세요. 바로가기

```
<html><head><title> 양건준님의 오늘 일정 </title></head>
<body>
<p class = "title"><u><i> 양건준님은 오늘 점심 </i></u></p>
<p class = "content"> 양건준님은 오늘 점심에 지하식당으로 내려가서 반 친구들과 같이 우육
탕을 먹었습니다. 코로나에 대한 걱정도 있었지만 용기내서 먹었습니다. 우육탕집 식당 홈페이
지는 바로가기를 누르세요.
 바로가기
</p>
</body>
</html>
```

## 141. beautiful soup 모듈 배우기

beautiful soup 모듈이란 파이썬 코드를 복잡하게 작성하지 않아도 편하게 웹스크롤링 할 수 있도록 여러 함수들을 제공하는 웹스크롤링 전문 모듈이다.

카페에서 ecologicalpyramid.html 문서를 내려받아 c 드라이브 밑에 data 폴더에 저장하시오.

ex)

ecologicalpyramid.html 코드를 beautiful soup 모듈에서 사용할 수 있도록 파싱 하고 파싱된 내용을 출력하시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
print (soup)
```

ex)

ecologicalpyramid.htm 코드에서 class 이름 name에 접근해서 데이터를 긁어오시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_ = "name")

print (result)
```

```
#
[<div class="name">plants</div>,
<div class="name">algae</div>,
<div class="name">deer</div>,
<div class="name">rabbit</div>,
<div class="name">fox</div>,
<div class="name">bear</div>,
<div class="name">lion</div>,
<div class="name">tiger</div>]
```

ex)

위에서 긁어온 데이터에서 html코드 말고 text만 출력하시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_ = "name")

for i in result:
 print (i.get_text())
```

```
#
plants
algae
deer
rabbit
fox
bear
lion
tiger
```

get\_text() 함수를 사용하면 html코드 말고 텍스트만 가져온다.

문제408. ecologincalpyramid.html 문서에서 number 클래스에 있는 숫자들만 출력하시오.

```
from bs4 import BeautifulSoup
```

```
f = open ("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_ = "number")

for i in result:
 print (i.get_text())
```

문제409. 위에서 긁어온 숫자들을 a 라는 비어있는 리스트에 저장한 후 a 안의 요소들을 정렬하고 a 리스트를 출력하시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\ecologicalpyramid.html")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_ = "number")
a = []

for i in result:
 a.append (int(i.get_text()))
 a.sort()

print (a)
```

문제410. 중앙일보사 홈페이지로 가서 신문기사 하나를 보시오.

문제411. aa77.html을 beautifulsoup 모듈의 함수를 쓸 수 있도록 파싱하고 출력하시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\aa77.html", encoding = "UTF8")
soup = BeautifulSoup(f, "html.parser")

print (soup)
```

문제412. 위의 기사의 본문 가져오기 위해서 기사 본문의 class 이름이 무엇인지 확인하시오.

개발자 모드(F12)로 접속 후 기사본문을 클릭하면 html문서의 class 이름을 확인할 수 있다.

```
article_body mg fs4
```

문제413. 클래스 이름 article\_body mg fs4 로 접근하여 text를 긁어오시오.

```
from bs4 import BeautifulSoup

f = open ("c:\\data\\aa77.html", encoding = "UTF8")
soup = BeautifulSoup(f, "html.parser")
result = soup.find_all(class_ = "article_body mg fs4")

for i in result:
 print (i.get_text())
```

문제414. 위에서 스크롤링한 중앙일보 기사를 c드라이브 밑에  
mytext23.txt로 저장하시오.

(반드시 저장해야 기사를 워드클라우드로 시각화 할 수 있다.)

```
from bs4 import BeautifulSoup

a = []
f = open ("c:\\data\\aa77.html", encoding = "UTF8")
soup = BeautifulSoup(f, "html.parser")
result1 = soup.find_all(class_ = "article_body mg fs4")

for i in result1:
 a.append (i.get_text())

bond = '\n'
text = bond.join(a)
h = open('c:\\data\\mytext23.txt', 'w', encoding = 'UTF8')

h.write(text)
f.close()
h.close()
```

# 20.12.16

2020년 12월 16일 수요일 오전 9:41

## 142. 웹스크롤링 실전 1단계 (ebs 레이디 버그 게시판)

회사의 신제품이 출시 되었을 때 그 제품에 대한 사람들의 반응을 데이터 분석을 하고자 할 때 웹 스크롤링과 데이터 시각화를 이용하면 된다.

ex1)

ebs 레이디 버그 시청자 게시판의 url을 가지고 직접 html문서를 내려 받을 수 있도록 코드를 구현 하시오.

```
from bs4 import BeautifulSoup
import urllib.request # 웹상의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

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

url = urllib.request.Request(list_url) # 위의 url을 파이썬이 알아볼 수 있도록 변환

result = urllib.request.urlopen(url).read().decode("utf-8") # 위의 url의 html문서들을 result변수에 담는다.

print(result) # 위의 url 의 html 전체 문서가 모두 출력된다.
```

ex2)

위에서 긁어온 html 코드를 Beautiful soup의 함수를 이용해서 웹스크롤링을 할 수 있도록 Beautiful soup를 쓸 수 있게 파싱하시오.

```
from bs4 import BeautifulSoup
import urllib.request

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

url = urllib.request.Request(list_url)

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

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

print(soup)
```

ex3)

지금 페이지의 시청자 게시판의 글 내용에 해당하는 부분의 태그와 클래스 이름을 알아내시오.

ex4)

p태그중에 class 가 con에 해당하는 부분을 스크롤링 하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

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

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

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

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

```
result2 = soup.find_all('p', class_ = 'con')
```

```
print (result2)
```

find 함수는 맨 처음 하나만 가져오는데 find\_all은 p 태그에 class 이름 con 에 해당하는 부분을 모두 가져온다.

ex5)

위의 결과에서 한글 텍스트만 가져오시오.

(위의 결과는 result2는 리스트이다.)

```
from bs4 import BeautifulSoup
import urllib.request
```

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

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

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

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

```
result2 = soup.find_all('p', class_ = 'con')
```

```
for i in result2:
 print (i.get_text())
```

ex6)

위에서 출력되고 있는 텍스트들이 좀 더 깔끔하게 나오게 하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

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

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

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



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

```
result2 = soup.find_all('p', class_ = 'con')
```

```
for i in result2:
 print (i.get_text(" ", strip = True))
```

ex7)

위에서 출력되고 있는 텍스트들이 params 라는 비어있는 리스트에 담기게 하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

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

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

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

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

```
result2 = soup.find_all('p', class_ = 'con')
```

```
params = []
```

```
for i in result2:
 params.append (i.get_text(" ", strip = True))
print (params)
```

ex8)

게시글을 올린 날짜를 스크롤링하기 위해서 게시글 날짜가 있는 html 문서의 태그 이름과 클래스 이름을 확인하시오.

태그 이름은 span 이고 클래스 이름은 date 이다.

ex9)

위의 날짜를 모두 스크롤링서 params2라는 리스트에 담으시오.

# 1. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱

```
from bs4 import BeautifulSoup
import urllib.request
```

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

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

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

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

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = 'date')
```

```
result2 = soup.find_all('p', class_ = 'con')
```

# 3. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.

```
params1 = []
```

```
params2 = []
```

```
for i in result1:
```

```
 params1.append(i.get_text(" ", strip = True))
```

```
for i in result2:
```

```
 params2.append(i.get_text(" ", strip = True))
```

```
print (params1)
```

```
print (params2)
```

ex10)

위의 날짜와 본문 내용이 아래와 같이 출력되게 하시오.

(zip과 + 를 이용)

2020.12.11 19:52 재미있 다

2020.12.11 19:06 어제 흐름상 종영 같지 않았는데 갑자기 끝이네요??

# 1. 웹에서 html 문서를 가져와 BeautifulSoup 으로 파싱

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

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

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

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

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

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = 'date')
```

```
result2 = soup.find_all('p', class_ = 'con')
```

# 3. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.

```
params1 = []
```

```
params2 = []
```

```
for i in result1:
```

```
 params1.append(i.get_text(" ", strip = True))
```

```
for i in result2:
```

```
 params2.append(i.get_text(" ", strip = True))
```

# 4. 날짜와 본문을 같이 출력

```
for k, j in zip (params1, params2):
```

```
 print (k + ' ' + j)
```

ex11)

레이디 버그 게시판 전체의 글을 모두 스크롤링해서 예제 10번 결과처럼 출력되게 하시오.

[http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuId=106  
&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=](http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=)

0&  
<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&>  
0&  
<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=3&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&>  
0&

위의 url을 보면 페이지 번호만 다르고 나머지는 모두 같다는 것을 확인할 수 있다.

ex12)

아래의 결과를 출력하는데 페이지 번호가 1 번 부터 22까지 변경되어서 출력되게 하시오.

<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=1&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&>  
<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=2&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&>  
0&  
<http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=3&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819&searchKeyword=&searchCondition=&searchConditionValue=0&>  
0&

```
for i in range (1, 23):
```

```
 print ('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' + str(i) +
'&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819
&searchKeyword=&searchCondition=&searchConditionValue=0&')
```

ex13)

예제 12번 코드를 예제 10번 코드에 적용해서 레이디 버그 전체 게시판의 글들이 날짜와 함께 출력되게 하시오.

1. 예제 10 번 코드를 가져온다.
2. 예제 10 번 코드에서 url의 페이지 번호가 1번 부터 22번 까지 변경되면서 예제 날짜와 게시글을 가져올 수 있도록 코드를 수정

# 1. 웹에서 html 문서를 가져와 BeautifulSoup 으로 파싱

```
from bs4 import BeautifulSoup
import urllib.request
```

```
for i in range (1, 23):
```

```
 list_url = ('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' + str(i) +
'&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819
&searchKeyword=&searchCondition=&searchConditionValue=0&')
```

```
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
```

# 2. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = 'date')
result2 = soup.find_all('p', class_ = 'con')
```

# 3. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.

```
params1 = []
params2 = []
```

```
for i in result1:
 params1.append (i.get_text(" ", strip = True))
for i in result2:
 params2.append (i.get_text(" ", strip = True))
```

# 4. 날짜와 본문을 같이 출력

```
for k, j in zip (params1, params2):
 print (k + ' ' + j)
```

위의 params1 과 params2 리스트가 for in range(1, 23) 안쪽에 있기 때문에 페이지 번호가 돌 때마다 params 리스트는 새로운 내용과 날짜로 변경된다.

ex14)

params1 과 params2에 레이디버그 시청자 게시판의 모든 게시날짜와 본문 내용이 들어가게 코드를 수정하시오.

# 0. 웹스크롤링에 필요한 모듈을 임포트한다.

```
from bs4 import BeautifulSoup
import urllib.request
```

# 1. 레이디 버그 게시판 게시날짜와 게시글 전체를 담는 리스트 생성

```
params1 = []
params2 = []
```

for i in range (1, 23):

# 2. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱

```
list_url = ('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' + str(i) +
'&hmpMnuld=106&searchKeywordValue=0&bbsId=10059819
&searchKeyword=&searchCondition=&searchConditionValue=0&')
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

# 3. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = 'date')
result2 = soup.find_all('p', class_ = 'con')
```

# 4. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.

```
for i in result1:
 params1.append (i.get_text(" ", strip = True))
for i in result2:
 params2.append (i.get_text(" ", strip = True))
```

```
5. 날짜와 본문을 같이 출력
for k, j in zip (params1, params2):
 print (k + ' ' + j)
```

예제13번은 레이디 버그 게시판 전체글을 화면에 출력하는 코드이고 예제14번은 레이디 버그 게시판 전체글을 params1과 params2 리스트에 모두 담고 화면에 출력하는 코드이다.

문제415. 레이디 버그 게시판의 전체 게시글은 총 몇건인지 확인하시오.

# 0. 웹스크롤링에 필요한 모듈을 임포트한다.

```
from bs4 import BeautifulSoup
import urllib.request
```

# 1. 레이디 버그 게시판 게시날짜와 게시글 전체를 담는 리스트 생성

```
params1 = [] # 게시날짜 리스트
params2 = [] # 게시글 리스트
```

for i in range (1, 23): # 페이지 번호를 1번 부터 22번까지 변경

# 2. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱

```
list_url = ('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' + str(i) +
'&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819
&searchKeyword=&searchCondition=&searchConditionValue=0&')
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

# 3. 시청자 게시판의 날짜와 본문 내용을 가져온다.

```
result1 = soup.find_all('span', class_ = 'date')
result2 = soup.find_all('p', class_ = 'con')
```

# 4. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.

```
for i in result1:
 params1.append (i.get_text(" ", strip = True))
for i in result2:
 params2.append (i.get_text(" ", strip = True))
```

# 5. 게시글의 건수 출력

```
print (len(params1))
```

문제416. 게시글 429건 전체를 c 드라이브 밑에 mytext34.txt 라는 이름으로 저장하시오.

텍스트 파일을 저장하는 파이썬 코드

```
text = 'abcdefghijkl'
```

```
f = open('c:\\data\\mytext32.txt', 'w', encoding = 'UTF8')
f.write(text)
f.close()
```

# 0. 웹스크롤링에 필요한 모듈을 임포트한다.

```
from bs4 import BeautifulSoup
import urllib.request
```

# 1. 레이디 버그 게시판 게시날짜와 게시글 전체를 담은 리스트 생성

```
params1 = [] # 게시날짜 리스트
```

```
params2 = [] # 게시글 리스트
```

```
f = open('c:\\data\\mytext32.txt', 'w', encoding = 'UTF8')
```

```
for i in range (1, 23): # 페이지 번호를 1번 부터 22번까지 변경
```

```
 # 2. 웹에서 html 문서를 가져와 beautifulsoup 으로 파싱
```

```
 list_url = ('http://home.ebs.co.kr/ladybug/board/6/10059819/oneBoardList?c.page=' + str(i) +
 '&hmpMnuId=106&searchKeywordValue=0&bbsId=10059819
 &searchKeyword=&searchCondition=&searchConditionValue=0&')
```

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

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

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

```
 # 3. 시청자 게시판의 날짜와 본문 내용을 가져온다.
```

```
 result1 = soup.find_all('span', class_ = 'date')
```

```
 result2 = soup.find_all('p', class_ = 'con')
```

```
 # 4. 시청자 게시판의 날짜와 본문을 params1 와 params2 리스트에 담는다.
```

```
 for i in result1:
```

```
 params1.append (i.get_text(" ", strip = True))
```

```
 for i in result2:
```

```
 params2.append (i.get_text(" ", strip = True))
```

# 5. 텍스트 파일을 저장

```
for k, j in zip (params1, params2):
```

```
 f.write (k + ' ' + j + '\n')
```

```
f.close()
```

## 143. 웹스크롤링 실전2 (중앙일보사)

중앙일보사 홈페이지에서 인공지능으로 검색 했을 때 나오는 기사들을 전부 웹스크롤링 한다.

ex1)

중앙일보사 홈페이지에서 인공지능으로 검색했을 때 나오는 url을 가져오시오.

<https://news.join.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews>

ex2)

위의 사이트에서 보이는 상세 기사를 클릭하고 그 기사의 url을 가져오시오.

<https://news.joins.com/article/23947044>

<https://news.joins.com/article/23946979>

<https://news.joins.com/article/23946876>

ex3)

인공지능으로 검색했을 때 나오는 상세기사들의 url을 스크롤링 하시오.

# 1. 웹스크롤링에 필요한 모듈을 임포트

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

# 2. 중앙일보에서 인공지능으로 검색했을 때 나오는 첫페이지의 html 코드를 beautiful soup에서 활용할 수 있도록 파싱

```
list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
```

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

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

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

# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.

```
result1 = soup.find_all('h2', class_='headline mg')
```

# 4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼내면서 href의 값을 얻어낸다.

```
<h2 class="headline mg">
```

```
for i in result1: # result1 리스트의 요소를 하나씩 빼내는 코드
```

```
 for k in i: # h2 태그안의 a 태그의 html 코드를 가져오기 위한 코드
```

```
 print(k.get('href')) # a 태그의 href의 값을 출력
```

ex4)

위의 상세 기사 url을 params 라는 비어있는 리스트에 append 시키시오.

# 1. 웹스크롤링에 필요한 모듈을 임포트

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

```
params = []
```

# 2. 중앙일보에서 인공지능으로 검색했을 때 나오는 첫페이지의 html 코드를 beautiful soup에서 활용할 수 있도록 파싱

```
list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

# 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.

```
result1 = soup.find_all ('h2', class_ = 'headline mg')
```

# 4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼내면서 href의 값을 얻어낸다.

```
for i in result1:
 for k in i:
 params.append (k.get('href'))
```

```
print (params)
```

ex5)

상세기사 url 중에 하나를 복사해 오고 그 상세기사 url의 웹페이지로 접속해서 본문 기사의 태그 이름과 클래스 이름이 무엇인지 확인하시오.

태그 이름은 div 이고 클래스 이름은 article\_body mg fs4 이다.

ex6)

위의 상세기사의 본문 텍스트를 스크롤링해서 출력하시오.

# 1. 웹스크롤링에 필요한 모듈을 импорт

```
from bs4 import BeautifulSoup
import urllib.request
```

```
params2 = []
```

# 2. 상세 기사 url로 검색했을 때 나오는 첫페이지의 html 코드를 beautiful soup에서 활용할 수 있도록 파싱

```
list_url = 'https://news.joins.com/article/23946032'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

# 3. 상세기사를 가져올 수 있도록 태그와 클래스를 찾는다.

```
result1 = soup.find_all ('div', class_ = 'article_body mg fs4')
```

# 4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼내면서 본문의 텍스트를 얻는다.

```
for i in result1:
 params2.append(i.get_text(" ", strip = True))
```

```
print (params2)
```



ex7)

상세기사 url을 가져와서 params 리스트에 넣고 출력하는 예제4번 코드를 j\_scroll 함수로 생성하시오.

```
###
```

```
1. 웹스크롤링에 필요한 모듈을 импорт
```

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

```
5. j_scroll 함수 생성
```

```
def j_scroll():
```

```
 # 2. 중앙일보에서 인공지능으로 검색했을 때 나오는 첫페이지의 html 코드를 beautiful soup에서
 활용할 수 있도록 파싱
```

```
 list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
```

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

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

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

```
 # 3. 상세기사 url을 가져올 수 있도록 태그와 클래스를 찾는다.
```

```
 result1 = soup.find_all('h2', class_='headline mg')
```

```
 params = []
```

```
 # 4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼내면서
 href의 값을 얻어낸다.
```

```
 for i in result1:
```

```
 for k in i:
```

```
 params.append(k.get('href'))
```

```
 return params
```

```
print(j_scroll())
```

ex8)

상세기사 url로 본문 기사를 스크롤링해서 리스트에 담았던 예제6번을 j\_detail\_scroll() 함수로 만드시오.

```
1. 웹스크롤링에 필요한 모듈을 импорт
```

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

```
5. j_detail_scroll 함수 생성
```

```
def j_detail_scroll():
```

```
 # 2. 상세 기사 url로 검색했을 때 나오는 첫페이지의 html 코드를 beautiful soup에서 활용할 수 있
 도록 파싱
```

```
 list_url = 'https://news.joins.com/article/23946032'
```

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

```
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

# 3. 상세 기사를 가져올 수 있도록 태그와 클래스를 찾는다.

```
result1 = soup.find_all('div', class_='article_body mg fs4')
```

```
params2 = []
```

# 4. 위의 result1은 리스트 이므로 for loop 문을 이용해서 리스트에 있는 요소를 하나씩 빼내면서  
본문의 텍스트를 얻는다.

```
for i in result1:
 params2.append(i.get_text(" ", strip = True))
```

```
return params2
```

```
print(j_detail_scroll())
```

ex9)

j\_detail\_scroll() 함수는 상세기사 한개의 본문을 출력하는 함수인데 이 함수에 j\_scroll() 함수를 실행했을 때 나오는 상세 url 여러개를 제공 할 수 있도록 코드를 수정하시오.

# 웹스크롤링에 필요한 모듈을 импорт

```
from bs4 import BeautifulSoup
import urllib.request
```

# j\_scroll 함수 생성

```
def j_scroll():
```

```
 list_url = 'https://news.joins.com/Search/JoongangNews?page=1&Keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&SortType=New&SearchCategoryType=JoongangNews'
```

```
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
```

```
 result1 = soup.find_all('h2', class_='headline mg')
```

```
 params = []
```

```
 for i in result1:
 for k in i:
 params.append(k.get('href'))
```

```
 return params
```

# j\_detail\_scroll 함수 생성

```
def j_detail_scroll():
```

```
 list_url = j_scroll()
```

```
 params2 = []
```

```
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
```

```

result1 = soup.find_all ('div', class_ = 'article_body mg fs4')

for i in result1:
 params2.append(i.get_text(" ", strip = True))

return params2

print (j_detail_scroll())

```

내용 정리

중앙일보의 기사를 스크롤링하는 내용 총 정

1. 검색 키워드(예 : 인공지능)를 가지고 검색을 한 후 url을 얻어낸다.
2. 상세기사 url을 리스트에 담는 j\_scroll() 함수를 생성한다.
3. 상세기사 url로 기사 본문을 스크롤링하는 j\_detail\_scroll() 함수를 생성한다.
4. j\_detail\_scroll() 함수 안에서 j\_scroll() 함수를 호출하여 상세기사 url 여러개를 받아오게 코딩 하고 받아온 상세기사 url로 본문 기사들을 params2 에 담게 코드를 수정한다.

문제417. 동아일보에서 검색 키워드(예 : 인공지능)를 가지고 검색을 한 후 url을 얻어내시오.

[https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check\\_news=1&more=1&sorting=1&search\\_date=1&v1=&v2=&range=1](https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1)

[https://www.donga.com/news/search?p=16&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check\\_news=1&more=1&sorting=1&search\\_date=1&v1=&v2=&range=1](https://www.donga.com/news/search?p=16&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1)

[https://www.donga.com/news/search?p=31&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check\\_news=1&more=1&sorting=1&search\\_date=1&v1=&v2=&range=1](https://www.donga.com/news/search?p=31&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=1&more=1&sorting=1&search_date=1&v1=&v2=&range=1)

문제418. 상세기사 url을 리스트에 담는 d\_scroll() 함수를 생성한다.

```

from bs4 import BeautifulSoup
import urllib.request

def d_scroll():
 list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=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')

 result1 = soup.find_all ('p', class_ = 'txt')

 params = []

```

```

for i in result1:
 for k in i:
 params.append(k.get('href'))

return params

print(d_scroll())

```

문제419. 상세기사 url로 기사 본문을 스크롤링하는 d\_detail\_scroll() 함수를 생성한다.

```

from bs4 import BeautifulSoup
import urllib.request

def d_scroll():
 list_url = 'https://www.donga.com/news/search?p=1&query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&check_news=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')

 result1 = soup.find_all('p', class_='txt')

 params = []

 for i in result1:
 for k in i:
 params.append(k.get('href'))

 return params

def d_detail_scroll():
 list_url = d_scroll()

 params2 = []
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')

 result1 = soup.find_all('div', class_='article_txt')

 for i in result1:
 params2.append(i.get_text(" ", strip=True))

 return params2

print(d_detail_scroll())

```

문제420. 한겨레 신문사에서 인공지능으로 검색했을 때 나오는 기사 본

문을 스크롤링하는 함수 두개를 생성하시오.

```
h_scroll()
h_detail_scroll()

from bs4 import BeautifulSoup
import urllib.request
#import time

def h_scroll():
 list_url = 'http://search.hani.co.kr/Search?command=query&keyword=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&media=news&submedia=&sort=d&period=all&datefrom=1988.01.01&dateto=2020.12.16&pageseq=0'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('dt')
 params = []

 for i in result1:
 for k in i:
 params.append('http:' + k.get('href'))
 return params

def h_detail_scroll():
 list_url = h_scroll()
 params2 = []
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='text')

 for i in result1:
 #time.sleep(10)
 params2.append(i.get_text(" ", strip = True))
 return params2

print(h_detail_scroll())
```

문제421. 전자신문사에서 스크롤링 하고 싶은 키워드를 넣고 검색했을 때 나오는 본문 기사들을 수집하는 함수 두개를 생성하시오.

(페이지 번호 3 까지만 스크롤링)

```
jj_scroll()
jj_detail_scroll()

from bs4 import BeautifulSoup
import urllib.request
#import time

def jj_scroll():
```

```

for i in range(1, 4):
 list_url = 'https://search.etnews.com/etnews/search.php?category=CATEGORY1&kwd=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&pageNum=' + str(i) + '&pageSize=10&reSrchFlag=false&sort=1&startDate=&endDate=&sitegubun=&jisikgubun=&preKwd%5B0%5D=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('dt')
 params = []

 for i in result1:
 for k in i:
 params.append('http:' + k.get('href'))
 return params

def jj_detail_scroll():
 list_url = jj_scroll()

 params2 = []
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all(class_='article_body')

 for i in result1:
 #time.sleep(10)
 params2.append(i.get_text(" ", strip = True))
 return params2

print(jj_detail_scroll())

```

# 20.12.17

2020년 12월 17일 목요일 오전 9:42

## 파이썬 코드의 큰 틀

1. 필요한 모듈 импорт
2. 데이터 발생 또는 수집 (ex : 주사위를 던지거나 웹스크롤링을 해서 데이터 수집)
3. 수집된 데이터를 리스트에 append
4. 리스트에 저장한 데이터를 이용하여 통계분석
5. 결과 출력

## 데이터 분석가가 갖추어야할 필수 기술

1. 웹스크롤링 기술
2. 통계로 데이터 분석
3. 머신러닝으로 데이터 분석
4. 딥러닝으로 데이터 분석

## beautiful soup 문법중에 좀 더 알아야할 내용

```
from bs4 import BeautifulSoup
import urllib.request # 웹사이트의 url 을 파이썬이 인식할 수 있도록 해주는 모듈

def c_scroll():
 for i in range(1,4):
 list_url = 'https://futurechosun.com/page/'+str(i)+'?s=%EB%B4%89%EC%82%AC'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, "html.parser")
 # result1=soup.find('div',class_='elementor-post__title').find('a')
 result1=soup.find_all('div',class_='elementor-post__title')
 for i in result1:
 print (i.find_all('a')[0].get("href"))
 #print(result1.get("href"))
 ""
 for i in result1:
 for k in i:
 print(type(k))
 ""

c_scroll()
```

ex1)

웹스크롤링을 할 사이트에 접속

<https://futurechosun.com/page1?s=%EB%B4%89%EC%82%AC>  
<https://futurechosun.com/page2?s=%EB%B4%89%EC%82%AC>  
<https://futurechosun.com/page3?s=%EB%B4%89%EC%82%AC>

ex2)

위의 첫페이지의 html코드를 파이썬으로 불러오는 작업

# 1. 웹스크롤링에 필요한 모듈 импорт

```
from bs4 import BeautifulSoup
import urllib.request
```

# 2. 첫 페이지의 url을 파싱한다.

```
list_url = 'https://futurechosun.com/page1?s=%EB%B4%89%EC%82%AC'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
```

# 3. 웹스크롤링 전문 모듈인 beautiful soup 모듈을 이용할 수 있게 html코드를 파싱한다.

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

# 4. 첫페이지 12개의 기사들의 기사 url을 알아내기 위해 태그이름과 클래스 이름을 확인

# (태그 : div, 클래스 : elementor-post\_\_title)

# 5. beautiful soup으로 파싱된 html 코드를 긁어온다.

```
result1=soup.find_all('div',class_='elementor-post__title')
```

# 6. result1 리스트의 요소(beautiful soup 으로 파싱된 html 코드)들을 for loop문으로 빼낸다.

```
'''
```

```
for i in result1:
```

```
 print (i)
```

```
'''
```

# 7. 빼낸 요소인 위의 html 코드에서 a 태그에 해당하는 부분만 가져오게 한다.

```
'''
```

```
for i in result1:
```

```
 print(i.find_all('a'))
```

```
'''
```

# 8. 위의 a태그의 html 코드를 담은 리스트안의 요소만 출력

```
'''
```

```
for i in result1:
```

```
 print(i.find_all('a')[0])
```

```
'''
```

# 9. 뽑아낸 a 태그의 html 문서에서 href의 값만 출력

```
'''
```

```
for i in result1:
```

```
 print(i.find_all('a')[0].get('href'))
```

```
'''
```

# 10. 위의 상세기사 url을 params 리스트에 append 시키기

```
params = []
```

```
for i in result1:
```

```
 params.append (i.find_all('a')[0].get('href'))
```

```
my_result = set(params)
```

```
my_result2 = list(my_result)
```

```
print (my_result2)
```



ex3)

1페이지 부터 3페이지 까지 36개의 기사 url을 params 리스트에 append 시키기

# 1. 웹스크롤링에 필요한 모듈 импорт

```
from bs4 import BeautifulSoup
import urllib.request
import time
```

```
params = []
```

```
for i in range (1, 4):
```

```
 time.sleep(5)
```

```
 # 2. 첫 페이지의 url을 파싱한다.
```

```
 list_url = 'https://futurechosun.com/page'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
```

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

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

```
 # 3. 웹스크롤링 전문 모듈인 beautiful soup 모듈을 이용할 수 있게 html코드를 파싱한다.
```

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

```
 # 4. 첫페이지 12개의 기사들의 기사 url을 알아내기 위해 태그이름과 클래스 이름을 확인
```

```
 # (태그 : div, 클래스 : elementor-post__title)
```

```
 # 5. beautiful soup으로 파싱된 html 코드를 긁어온다.
```

```
 result1=soup.find_all('div',class_='elementor-post__title')
```

```
 for i in result1:
```

```
 params.append (i.find_all('a')[0].get('href'))
```

```
my_result = set(params) # set 함수를 이용해서 중복 제거
```

```
my_result2 = list(my_result) # 중복을 제거한 요소들을 다시 리스트화
```

```
print (my_result2)
```

위의 사이트는 사진 이나 기사제목을 클릭했을때 상세기사로 가는 구조로 되어있어서 상세기사 url이 중복되어서 스크롤링이 되었다. 따라서 params의 요소들의 중복을 제거해야 한다.

ex4)

위의 코드를 bs\_scroll() 이라는 함수로 생성하시오.

```
from bs4 import BeautifulSoup
```

```
import urllib.request
```

```
import time
```

```
def bs_scroll():
```

```
 params = []
```

```
 for i in range (1, 4):
```

```
 time.sleep(5)
```

```
 list_url = 'https://futurechosun.com/page'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
```

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

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

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

```
 result1=soup.find_all('div',class_='elementor-post__title')
```

```
 for i in result1:
```

```

 params.append (i.find_all('a')[0].get('href'))
my_result = set(params)
my_result2 = list(my_result)

return my_result2

```

```
print (bs_scroll())
```

ex5)

상세기사 url 하나를 가지고 본문 기사를 출력하시오.

```

from bs4 import BeautifulSoup
import urllib.request

```

```

list_url = 'https://futurechosun.com/archives/52491'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
result1 = soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')

```

```

for i in result:
 print(i.find_all('p'))

```

ex6)

위의 코드를 bs\_detail\_scroll() 함수로 생성하시오.

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def bs_detail_scroll():
 list_url = 'https://futurechosun.com/archives/52491'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')

```

```

 for i in result1:
 return i.find_all('p')

```

```
print (bs_detail_scroll())
```

ex7)

bs\_detail\_scroll() 함수의 코드 안에 bs\_scroll()함수를 호출하는 코드를 추가해서 bs\_scroll()함수가 리턴하는 36개의 상세기사 url에 대한 기사본문이 전부 출력되게 하시오.

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def bs_scroll():
 params = []
 for i in range (1, 4):
 list_url = 'https://futurechosun.com/page'+ str(i) +'?s=%EB%B4%89%EC%82%AC'

```

```

url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
result1=soup.find_all('div',class_='elementor-post__title')

for i in result1:
 params.append (i.find_all('a')[0].get('href'))

my_result = set(params)
my_result2 = list(my_result)
return my_result2

def bs_detail_scroll():
 list_url = bs_scroll()
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-widget__width-initial elementor-widget elementor-widget-theme-post-content')

 for i in result1:
 print(i.find_all('p'))

print (bs_detail_scroll())

```

ex8)

bs\_detail\_scroll() 함수의 params2 리스트를 추가해서 params2 리스트에 36개의 기사가 append 되게 하시오.

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def bs_scroll():
 params = []
 for i in range (1, 4):
 list_url = 'https://futurechosun.com/page'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1=soup.find_all('div',class_='elementor-post__title')

 for i in result1:
 params.append (i.find_all('a')[0].get('href'))

 my_result = set(params)
 my_result2 = list(my_result)
 return my_result2

def bs_detail_scroll():
 list_url = bs_scroll()
 params2 = []
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")

```

```

soup = BeautifulSoup(result, 'html.parser')
result1 = soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-
widget__width-initial elementor-widget elementor-widget-theme-post-content')

for i in result1:
 params2.append(i.find_all('p'))
return params2

print(bs_detail_scroll())

```

ex9)

위에서 수집한 기사가 들어있는 params2 리스트의 내용을 c 드라이브 밑에 data 밑에 bongsa.txt 로 생성되게 하시오.

```

from bs4 import BeautifulSoup
import urllib.request

```

```

def bs_scroll():
 params = []
 for i in range(1, 2):
 list_url = 'https://futurechosun.com/page'+ str(i) +'?s=%EB%B4%89%EC%82%AC'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1=soup.find_all('div',class_='elementor-post__title')

 for i in result1:
 params.append(i.find_all('a')[0].get('href'))

 my_result = set(params)
 my_result2 = list(my_result)
 return my_result2

def bs_detail_scroll():
 list_url = bs_scroll()
 f = open('c:\\data\\bongsa.txt', 'w', encoding = 'UTF8')
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='elementor-element elementor-element-24e82692 elementor-
widget__width-initial elementor-widget elementor-widget-theme-post-content')

 for i in result1:
 for j in i.find_all('p'):
 f.write(str(j.get_text()) + '\n')
 f.close()

bs_detail_scroll()

```

## 웹스크롤링 실전 (이데일리)

ex1)

이데일리에서 첫눈 으로 검색한 기사를 검색하시오.

<https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1>

ex2)

위의 url 첫페이지의 html 코드를 BeautifulSoup 으로 파싱하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

```
print (soup)
```

ex3)

상세기사 url을 찾기 위한 태그 이름과 클래스 이름을 찾으시오.

ex4)

위에서 찾은 태그와 클래스로 href의 상세기사 url을 출력하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
result1 = soup.find_all('div', class_='newsbox_04')
```

```
for i in result1:
 print ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))
```

ex5)

위의 상세기사 url을 params 리스트에 담으시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
result1 = soup.find_all('div', class_='newsbox_04')
params = []
for i in result1:
 params.append ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))
```

```
print (params)
```

ex6)

위의 코드들을 가지고 eda\_scroll() 이라는 함수를 생성하시오.

```
from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
 list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=1'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='newsbox_04')
 params = []
 for i in result1:
 params.append ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))

 return params

print (eda_scroll())
```

ex7)

위의 상세기사 url 하나를 가지고 기사 본문을 출력하시오.

<https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258>

```
from bs4 import BeautifulSoup
import urllib.request

list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
result1 = soup.find_all('div', class_='news_body')

for i in result1:
 print (i.get_text())
```

ex8)

위의 스크립트로 eda\_detail\_scroll() 함수를 생성하시오.

```
from bs4 import BeautifulSoup
import urllib.request

def eda_detail_scroll():
 list_url = 'https://www.edaily.co.kr/news/read?newsId=01423526625998520&mediaCodeNo=258'
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='news_body')

 for i in result1:
 return i.get_text()

print (eda_detail_scroll())
```

ex9)

eda\_scroll() 함수를 수정하여 3개 페이지의 상세기사 url이 params에 담겨지게 수정하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def eda_scroll():
 params = []
 for i in range(1, 4):
 list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=' +
 str(i)
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='newsbox_04')
 for i in result1:
 params.append ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))

 return params

print (eda_scroll())
```

ex10)

eda\_detail\_scroll() 함수 안에서 eda\_scroll() 함수를 호출하여 상세기사 url 60개를 가져와서 본문을 출력하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
def eda_scroll():
 params = []
 for i in range(1, 4):
 list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=' +
 str(i)
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='newsbox_04')
 for i in result1:
 params.append ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))

 return params

def eda_detail_scroll():
 list_url = eda_scroll()
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='news_body')

 for i in result1:
 print (i.get_text())

eda_detail_scroll()
```

ex11)

위에 출력되고 있는 기사본문이 c 드라이브 밑에 data 밑에 eda.txt로 저장되게 하시오.

```
from bs4 import BeautifulSoup
import urllib.request

def eda_scroll():
 params = []
 for i in range(1, 4):
 list_url = 'https://www.edaily.co.kr/search/news/?keyword=%ec%b2%ab%eb%88%88&page=' +
 str(i)
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='newsbox_04')
 for i in result1:
 params.append ('https://www.edaily.co.kr' + i.find_all('a')[0].get('href'))

 return params

def eda_detail_scroll():
 list_url = eda_scroll()
 f = open('c:\\data\\eda.txt', 'w', encoding = 'UTF-8')
 for i in list_url:
 url = urllib.request.Request(i)
 result = urllib.request.urlopen(url).read().decode("utf-8")
 soup = BeautifulSoup(result, 'html.parser')
 result1 = soup.find_all('div', class_='news_body')

 for i in result1:
 f.write (str(i.get_text()) + '\n')
 f.close()

eda_detail_scroll()
```

## 웹스크롤링 실전 (연합뉴스)

ex1)

연합 뉴스에서 인공지능으로 검색했을 때 url을 확인하시오.

[https://www.yna.co.kr/search/index?query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&ctype=A&page\\_no=1](https://www.yna.co.kr/search/index?query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5&ctype=A&page_no=1)

ex2)

위의 url의 html 코드를 beautiful soup 으로 파싱하여 출력하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
list_url = 'https://www.yna.co.kr/search/index?query=%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%
```



```
8A%A5&ctype=A&page_no=1'
```

```
url = urllib.request.Request(list_url)
result = urllib.request.urlopen(url).read().decode("utf-8")
soup = BeautifulSoup(result, 'html.parser')
```

```
print (soup)
```

```
ex3)
```

상세기사 url을 찾기 위한 태그 이름과 클래스 이름을 찾으시오.

연합뉴스는 셀레니움을 이용해서 웹스크롤링을 해야한다.

(셀레니움 손으로 클릭해서 웹의 기사를 긁어 오는 것을 컴퓨터에게 시키는 모듈이름이다.)

문제422. 네이버 영화 평점 게시판의 게시글들을 스크롤링해서 영화 이름을 텍스트 파일로 해서 저장하시오.

```
from bs4 import BeautifulSoup
import urllib.request
```

```
params1 = []
params2 = []
f = open('c:\\data\\titanic.txt', 'w', encoding = 'UTF-8')
```

```
for i in range(1, 6):
 list_url = 'https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=18847&page=' + str(i)
 url = urllib.request.Request(list_url)
 result = urllib.request.urlopen(url).read().decode("cp949")
 soup = BeautifulSoup(result, "html.parser")
 result1 = soup.find_all('td', class_='num')
 result2 = soup.find_all('td', class_='title')
```

```
for i in result1:
 params1.append (i.get_text(" ", strip = True))
```

```
for i in result2:
 params2.append (i.get_text(" ", strip = True))
```

```
5. 날짜와 본문을 같이 출력합니다.
```

```
for k, j in zip (params1, params2):
 f.write (k + ' ' + j + '\n')
```

```
f.close()
```

## 144. 파이썬에서 워드 클라우드 그리기

웹스크롤링한 데이터를 분석을 하는데 분석하는 방법

1. 우리회사의 신제품이 출시되었을때 소비자들의 반응을 살펴보고자 할 때 ( 기업 ) ---> 감정분석
2. 시기별 사회현상을 파악하고자 할 때 (국가)
3. 인공지능 상담원(딥러닝의 RNN)을 만들기 위한 자연어 처리 학습 데이터로 웹스크롤링한 데이터를 활용
4. 인공지능의 눈이라고 할 수 있는 딥러닝의 CNN 의 신경망의 학습 자료로 이미지 데이터가 활용

\* 빅데이터 기사 문제: 데이터의 구조적 관점에서 3가지로 나뉜다.(P1-4)

1. 정형 데이터 : 정형화된 스키마 구조. DBMS 에 저장될 수 있는 구조  
예: Oracle 의 emp 테이블, MySQL, MSSQL 의 테이블
2. 반정형 데이터: 데이터 내부의 데이터 구조에 대한 메타정보가 포함된 구조의 데이터  
예: html 문서, xml 문서
3. 비정형 데이터 : 웹스크롤링 기술로 수집해서 모은 데이터  
예: 텍스트 문서, 이미지, 동영상

문제423. 조선일보에서 봉사로 검색했을 때 아누 받았던 기사에서 가장 많이 나오는 단어는 무엇인지 확인하시오.

```
stev = open('c:\\data\\bongsa.txt', encoding = 'UTF8')
stev2 = stev.read().split() # 어절별로 분리해서 리스트에 담는다.
stev2.sort() # 리스트 안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) # stev2 리스트 안의 어절별로 각각 몇 건씩 있는지 정리

print (count_list)
```

```
result = count_list.most_common(30) # top 10개만 추출

print (result)
```

문제424. 위의 결과를 for loop 문을 이용해서 아래와 같이 출력하십시오.

```
stev = open('c:\\data\\bongsa.txt', encoding = 'UTF8')
stev2 = stev.read().split() # 어절별로 분리해서 리스트에 담는다.
stev2.sort() # 리스트 안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) # stev2 리스트 안의 어절별로 각각 몇 건씩 있는지 정리

result = count_list.most_common(30) # top 10개만 추출

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

문제425. 문제 422 번에서 스크롤링했던 영화 평가 게시글들에서 위와 같이 가장 많이 나오는 어절이 무엇인지 출력하십시오.

```
stev = open('c:\\data\\titanic.txt', encoding = 'UTF8')
stev2 = stev.read().split() # 어절별로 분리해서 리스트에 담는다.
stev2.sort() # 리스트 안의 요소들을 정렬

from collections import Counter
count_list = Counter(stev2) # stev2 리스트 안의 어절별로 각각 몇 건씩 있는지 정리

result = count_list.most_common(30) # top 10개만 추출

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

문제426. 웹스크롤링을 한 데이터로 감정분석(긍정적, 부정적)을 하기 위해 신성이의 라라랜드 영화 평가를 다운로드 해서 c밑에 data밑에 두고 가장 많이 나오는 단어가 무엇인지 확인하십시오.

```
stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.read().split() # 어절별로 분리해서 리스트에 담는다.
stev2.sort() # 리스트 안의 요소들을 정렬
```

```

from collections import Counter
count_list = Counter(stev2) # stev2 리스트 안의 어절별로 각각 몇 건씩 있는지 정리

result = count_list.most_common(30) # top 30개만 추출

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

```

문제427. 영화 라라랜드에서 긍정적인 단어가 몇건이 나오는지 출력하시오.

```

문제 276 코드
stev = open("c:\\data\\laland.txt", encoding = 'UTF8')
stev2 = stev.read().split()
positive = open("c:\\data\\positive-words2.txt")
pos = positive.read().split("\n")
cnt = 0

for i in stev2:
 if i.lower() in pos:
 cnt += 1
 print (i,cnt)

print (cnt)

```

문제428. 라라랜드 영화 평가에 부정단어는 몇개가 있고 단어들은 무엇인지 출력하시오.

```

stev = open("c:\\data\\laland.txt", encoding = 'UTF8')
stev2 = stev.read().split()
positive = open("c:\\data\\negative-words2.txt")
pos = positive.read().split("\n")
cnt = 0

for i in stev2:
 if i.lower() in pos:
 cnt += 1
 print (i,cnt)

print (cnt)

```

문제429. 영화 라라랜드의 평가글들을 워드 클라우드로 시각화 하시오.

ex) 파이썬에서 워드 클라우드 그리기

1. 아나콘다 프롬프트 창을 열고 wordcloud 패키지 설치

```
conda install wordcloud
```

또는

```
pip install wordcloud
```

2. c 드라이브 밑에 project 폴더를 생성

3. project 폴더 밑에 4가지 파일을 둡니다.

- usa\_im.png
- s\_korea.png
- word.txt
- 중앙일보 스크롤링했던 기사 파일 my\_text21.txt

'''

# 텍스트마이닝 데이터 정제

```
from wordcloud import WordCloud, STOPWORDS # 구두점 데이터 정제
```

```
import matplotlib.pyplot as plt # 그래프 그리는 모듈
```

```
from os import path # os 에 있는 파일을 파이썬에서 인식
```

```
import re # 데이터 정제를 위해서 필요한 모듈
```

```
import numpy as np
```

```
from PIL import Image # 이미지 시각화를 위한 모듈
```

# 워드 클라우드의 배경이 되는 이미지 모양을 결정

```
usa_mask = np.array(Image.open("c:\\project\\usa_im.png"))
```

# 워드 클라우드를 그릴 스크립트 이름을 물어봅니다.

```
script = 'laland.txt'
```

# 워드 클라우드 그림이 저장될 작업 디렉토리를 설정

```
d = path.dirname("c:\\project\\")
```

# 기사 스크립트와 os 의 위치를 연결하여 utf8로 인코딩해서 한글 텍스트를

# text 변수로 리턴한다.

```
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()
```

# 파이썬이 인식할 수 있는 한글 단어의 갯수를 늘리기 위한 작업

```
file = open('c:\\project\\word.txt', 'r', encoding = 'utf-8')
```

```
word = file.read().split(' ') # 어절별로 분리해서 word에 담아 리스트로 구성
```

```
for i in word:
```

```
 text = re.sub(i,"",text) # re.sub('있다', '', '있다') 라라랜드 게시글에 '있다'를 ""으로 대체
```

```
 # 일반적인 문장에서 자주 나오는 단어들을 전부 ""로 대체
```

# 워드 클라우드를 그린다.

```
wordcloud = WordCloud(font_path='C:\\Windows\\Fonts\\gulim', # 글씨체
```

```
 stopwords=STOPWORDS, # 마침표, 느낌표,싱글 쿼테이션 등을 정제
```

```
 max_words=1000, # 워드 클라우드에 그릴 최대 단어갯수
```

```

background_color='white', # 배경색깔
max_font_size = 100, # 최대 글씨 크기
min_font_size = 1, # 최소 글씨
mask = usa_mask, # 배경 모양
colormap='jet').generate(text).to_file('c:\\project\\lala_cloud.png')
c 드라이브 밑에 project 폴더 밑에 생성되는 워드 클라우드 이미지 이름

```

```

plt.figure(figsize=(15,15)) # 가로 15, 세로 15 사이즈
plt.imshow(wordcloud, interpolation='bilinear') # 글씨가 퍼지는 스타일
plt.axis("off") # 축 표시 없음

```

문제430. 영화 평가 또는 스크롤링했던 기사들 중 하나 선택해서 워드 클라우드로 그리시오.

```

from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from os import path
import re
import numpy as np
from PIL import Image

usa_mask = np.array(Image.open("c:\\project\\usa_im.png"))
script = 'donsaNews.txt'
d = path.dirname("c:\\project\\")
text = open(path.join(d, "%s"%script), mode="r", encoding="utf-8").read()
file = open('c:\\project\\word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ')

for i in word:
 text = re.sub(i,"",text)

wordcloud = WordCloud(font_path='C:\\Windows\\Fonts\\gulim',
 stopwords=STOPWORDS,
 max_words=1000,
 background_color='white',
 max_font_size = 100,
 min_font_size = 1,
 mask = usa_mask,
 colormap='jet').generate(text).to_file('c:\\project\\bongsa_cloud.png')

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")

```

문제431. 라라랜드 평가 게시글에서 평가 점수만 출력하시오.

```

stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.readlines()

for i in stev2:

```

```
print (int(i[6:8]))
```

문제432. 라라랜드 평가 글 중에 평점이 6 점 이상인 글들만 출력하시오.

```
stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.readlines()

for i in stev2:
 if int(i[6:8]) >= 6:
 print (i, end = '')
```

문제433. 라라랜드 평가글 중에 평점이 6 점이상이면 pos 라는 비어있는 리스트에 해당 평가글을 append 되게 하고 6점 미만이면 nag 라는 비어있는 리스트에 append 되게하시오.

```
stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.readlines()
pos = []
nag = []

for i in stev2:
 if int(i[6:8]) >= 6:
 pos.append (i[9:])
 else:
 nag.append (i[9:])

print (pos)
```

문제434. 위의 pos 에 들어있는 긍정글들을 c드라이브 밑에 project 밑에 pos\_lala.txt 로 저장하시오.

```
stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.readlines()
pos = []
nag = []

for i in stev2:
 if int(i[6:8]) >= 6:
 pos.append (i[9:])
 else:
 nag.append (i[9:])

f = open ('c:\\project\\pos_lala.txt', 'w')
f.writelines(pos)
f.close()
```

문제435. 6점 미만인 글들은 nag\_lala.txt 로 c드라이브 밑에 project 밑

에 저장하시오.

```
stev = open('c:\\data\\laland.txt', encoding = 'UTF8')
stev2 = stev.readlines()
pos = []
nag = []

for i in stev2:
 if int(i[6:8]) >= 6:
 pos.append (i[9:])
 else:
 nag.append (i[9:])

f = open ('c:\\project\\pos_lala.txt', 'w')
f.writelines(pos)
f.close()

f2 = open ('c:\\project\\nag_lala.txt', 'w')
f2.writelines(nag)
f2.close()
```

문제436. 위의 pos\_lala.txt 와 nag\_lala.txt의 글을 워드 클라우드로 각각 시각화 하시오.

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from os import path
import re
import numpy as np
from PIL import Image

usa_mask = np.array(Image.open("c:\\project\\usa_im.png"))

script = 'nag_lala.txt'

d = path.dirname("c:\\project\\")

text = open(path.join(d, "%s"%script), mode="r",encoding="CP949", errors='ignore').read()
file = open('c:/project/word.txt', 'r', encoding = 'utf-8')
word = file.read().split(' ')
for i in word:
 text = re.sub(i,"",text)

wordcloud = WordCloud(font_path='C://Windows//Fonts//gulim',
 stopwords=STOPWORDS,
 max_words=1000,
 background_color='white',
 max_font_size = 100,
 min_font_size = 1,
 mask = usa_mask,
 colormap='jet').generate(text).to_file('c:\\project\\lala_nag_cloud.png')

plt.figure(figsize=(15,15))
plt.imshow(wordcloud, interpolation='bilinear')
```



plt.axis("off")

## 146. 웹에 있는 사진을 스크롤링 하는 방법 (구글 이미지)

딥러닝 기술

1. cnn : 인공지능의 눈
  - 관련된 학습 빅데이터 : 이미지
2. rnn : 인공지능의 입과 귀
  - 관련된 학습 빅데이터 : 자연어 처리를 위한 문장들

셀레니움을 사용하여 마치 손으로 클릭해서 이미지를 저장하는데 컴퓨터를 시켜서 자동화 시키는 방법으로 스크롤링을 한다.

구글에서 이미지 스크롤링 하기

1. 크롬 웹브라우저가 설치 되어져 있어야 한다.
2. c드라이브 밑에 chromedriver 폴더를 생성하고 거기에 chromedriver.exe를 넣어야 한다.
3. c드라이브 밑에 gimages 폴더를 생성한다.
4. 다운받을 이미지 키워드를 결정한다.
5. 아나콘다 프롬프트 창을 열고 selenium을 설치한다.

conda install selenium

conda list selenium

또는

pip install selenium

6. 구글에서 햄버거로 검색했을 때 웹스크롤링 코드

문제438. c 드라이브 밑에 gimages2 라는 폴더를 만들고 이미지를 다운 받으시오.

# 구글

#####

import urllib.request # 웹 url을 파이썬이 인식 할 수 있게하는 패키지

from bs4 import BeautifulSoup # html에서 데이터 검색을 용이하게 하는 패키지

from selenium import webdriver

# selenium : 웹 애플리케이션의 테스트를 자동화하기 위한 프레임 워크

# 손으로 클릭하면서 컴퓨터가 대신하면서 스크롤링하게 하는 패키지

from selenium.webdriver.common.keys import Keys

```

import time # 중간중간 sleep 을 걸어야 해서 time 모듈 import

url 받아오기

웹브라우저로 크롬을 사용할거라서 크롬 드라이버를 다운받아 아래 파일경로의 위치에 둔다
팬텀 js로 하면 백그라운드로 실행할 수 있음
binary = 'C:\\chromedriver\\chromedriver.exe'

브라우저를 인스턴스화
browser = webdriver.Chrome(binary)

구글의 이미지 검색 url 받아옴(아무것도 안 쳤을때의 url)
browser.get("https://www.google.co.kr/imghp?hl=ko&tab=wi&ei=l1AdWbegOcra8QXvtr-4Cw&ved=0EKouCBUoAQ")

구글의 이미지 검색에 해당하는 input 창의 id 가 ' ? ' 임(검색창에 해당하는 html코드를 찾아서
elem 사용하도록 설정)
input창 찾는 방법은 원노트에 있음

elem = browser.find_elements_by_class_name('gLfyf gsfi') # Tip : f12누른후 커서를 검색창에 올려
두고 id를 찾으면 best

elem = browser.find_element_by_xpath("//*[@class='gLfyf gsfi']") # 위의 코드대로 하거나 이렇게 하
거나 둘 중 하나 select

검색어 입력

elem 이 input 창과 연결되어 스스로 햄버거를 검색
elem.send_keys("치킨") # 여기에 스크롤링하고싶은 검색어를 입력

웹에서의 submit 은 엔터의 역할을 함
elem.submit()

현재 결과 더보기는 구현 되어있지 않은상태 -> 구글의 경우 400개 image가 저장됨.

반복할 횟수

스크롤을 내리려면 브라우저 이미지 검색결과 부분(바디부분)에 마우스 클릭 한번 하고 End키
를 눌러야함

for i in range(1, 6): # 5번 스크롤 내려가게 구현된 상태 range(1,5)
 browser.find_element_by_xpath("//body").send_keys(Keys.END)
 time.sleep(10) # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5
 번 누르는데 end1번누르고 10초 쉼

```

```
time.sleep(10) # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지
가 다운받아질 수 있음.)
```

```
html = browser.page_source # 크롬브라우저에서 현재 불러온 소스 가져옴
```

```
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정
```

```
browser.find_element_by_xpath("//*[class='mye4qd']").click() # 여기가 결과 더보기 코드입니다
```

```
for i in range(1, 5): # 4번 스크롤 내려가게 구현된 상태 range(1,5)
```

```
 browser.find_element_by_xpath("//body").send_keys(Keys.END)
```

```
 time.sleep(10) # END 키 누르고 내려가는데 시간이 걸려서 sleep 해줌 / 키보드 end키를 총 5
번 누르는데 end1번누르고 10초 쉼
```

```
time.sleep(10) # 네트워크 느릴까봐 안정성 위해 sleep 해줌(이거 안하면 하얀색 이미지
가 다운받아질 수 있음.)
```

```
html = browser.page_source # 크롬브라우저에서 현재 불러온 소스 가져옴
```

```
soup = BeautifulSoup(html, "lxml") # html 코드를 검색할 수 있도록 설정
```

```
그림파일 저장
```

```
검색한 구글 이미지의 url을 따오는 코드
```

```
def fetch_list_url():
```

```
 params = []
```

```
 imgList = soup.find_all("img", class_="rg_i Q4LuWd") # 구글 이미지 url 이 있는 img 태그의 _img 클
래스에 가서 (f12로 확인가능.)
```

```
 for im in imgList:
```

```
 try :
```

```
 params.append(im["src"]) # params 리스트에 image url 을 담음.
```

```
 except KeyError:
```

```
 params.append(im["data-src"])
```

```
 return params
```

```
except부분
```

```
이미지의 상세 url 의 값이 있는 src 가 없을 경우
```

```
data-src 로 가져오시오 ~
```

```
def fetch_detail_url():
```

```
 params = fetch_list_url()
```

```
 for idx,p in enumerate(params,1):
```

```
 # 다운받을 폴더경로 입력
```

```
 urllib.request.urlretrieve(p, "C:/gimages2/" + str(idx) + "_google.jpg")
```

```
enumerate 는 리스트의 모든 요소를 인덱스와 쌍으로 추출
```

```
하는 함수 . 숫자 1은 인덱스를 1부터 시작해라 ~
```

```
fetch_detail_url()
```

```
끝나면 브라우저 닫기
```

```
browser.quit()
```

## 147. 이미지 스크롤링 하기 ( 네이버 이미지 검색)

### # 네이버

```
import urllib.request # 웹 url을 파이썬이 인식할 수 있게 하는 패키지
```

```
from bs4 import BeautifulSoup # html 코드에서 원하는 지점을 빨리 찾을 수 있게 만든 모듈
```

```
from selenium import webdriver # 손으로 클릭하는 것을 컴퓨터가 대신 하게 하는 모듈
```

```
from selenium.webdriver.common.keys import Keys
```

```
import time # sleep
```

```
binary = 'C:\chromedriver/chromedriver.exe' # 크롬드라이버 위치 지정
```

```
browser = webdriver.Chrome(binary) # browser 객체 생성
```

```
browser.get("https://search.naver.com/search.naver?where=image&sm=stb_nmr&")
```

```
elem = browser.find_element_by_id("nx_query") # 검색창 지점을 알아내서 elem에 담는다.
```

```
#find_elements_by_class_name("") 클래스 이름으로 찾을때 필요한 코드
```

```
검색어 입력
```

```
elem.send_keys("헐크") # 컴퓨터가 아이언맨을 직접 적는다.
```

```
elem.submit() # 엔터
```

```
반복할 횟수
```

```
for i in range(1,5): # end키를 누르면서 아래로 내린다.
```

```
 browser.find_element_by_xpath("//body").send_keys(Keys.END)
```

```
 time.sleep(10) # sleep을 5초 주면서 5번 수행한다.
```

```
time.sleep(10) # 5초간 멈춘다.
```

```
html = browser.page_source # 현 페이지의 html코드를 불러온다.
```

```
soup = BeautifulSoup(html,"lxml") # beautifulsoup 을 이용할 수 있도록 파싱한다.
```

```
#print(soup)
```

```
#print(len(soup))
```

```
def fetch_list_url(): # 이미지의 상세 url 가져오는 함수
```

```
 params = []
```

```
 imgList = soup.find_all("img", class_="_img") # img 태그의 클래스 이름 _img로 접근
```

```
 for im in imgList:
```

```
 params.append(im["src"]) # src의 값을 가져와서 params에 append한다.
```

```
 return params
```

```

def fetch_detail_url(): # 상세 이미지 url을 가져와서 이미지를 다운로드 하는 함수
 params = fetch_list_url()
 #print(params)
 a = 1
 for p in params:
 # 다운받을 폴더경로 입력
 urllib.request.urlretrieve(p, "C:/naverImages/"+ str(a) + ".jpg")

 a = a + 1

fetch_detail_url()

browser.quit()

```

문제439. 마이크로 소프트 회사에서 만든 검색페이지인 bing에서 이미지 검색을 할 수 있도록 하는 웹스크롤링 코드를 작성하시오.

#### #bing

```

import urllib.request # 웹 url을 파이썬이 인식할 수 있게 하는 패키지
from bs4 import BeautifulSoup # html 코드에서 원하는 지점을 빨리 찾을 수 있게 만든 모듈
from selenium import webdriver # 손으로 클릭하는 것을 컴퓨터가 대신 하게 하는 모듈
from selenium.webdriver.common.keys import Keys
import time # sleep

binary = 'C:\chromedriver/chromedriver.exe' # 크롬드라이버 위치 지정
browser = webdriver.Chrome(binary) # browser 객체 생성
browser.get("https://www.bing.com/images?FORM=Z9LH")
elem = browser.find_element_by_id("sb_form_q") # 검색창 지점을 알아내서 elem에 담는다.
#elem = browser.find_element_by_xpath("//*[@class='']") # 클래스 이름으로 찾을때 필요한 코드

검색어 입력
elem.send_keys("아이언맨") # 컴퓨터가 아이언맨을 직접 적는다.
time.sleep(5)
elem.submit() # 엔터

반복할 횟수
for i in range(1,5): # end키를 누르면서 아래로 내린다.
 browser.find_element_by_xpath("//body").send_keys(Keys.END)
 time.sleep(5) # sleep을 5초 주면서 5번 수행한다.

time.sleep(5) # 5초간 멈춘다.

html = browser.page_source # 현 페이지의 html코드를 불러온다.
soup = BeautifulSoup(html,"lxml") # beautifulsoup 을 이용할 수 있도록 파싱한다.

#print(soup)
#print(len(soup))

```

```
def fetch_list_url(): # 이미지의 상세 url 가져오는 함수
 params = []
 imgList = soup.find_all("img", class_="mimg") # img 태그의 클래스 이름 _img로 접근
 for im in imgList:
 params.append(im["src"]) # src의 값을 가져와서 params에 append한다.
 return params

def fetch_detail_url(): # 상세 이미지 url을 가져와서 이미지를 다운로드 하는 함수
 params = fetch_list_url()
 #print(params)
 a = 1
 for p in params:
 # 다운받을 폴더경로 입력
 urllib.request.urlretrieve(p, "C:/bingImages/"+ str(a) + ".jpg")

 a = a + 1

fetch_detail_url()

browser.quit()
```

# 20.12.21

2020년 12월 21일 월요일 오전 9:47

## 148. 파이썬과 oracle 연동

오라클 database : 비즈니스 데이터(정형화된 데이터)

빅데이터 기사 시험 : 데이터의 구조에 따른 종류 3가지

1. 정형화 데이터 : rdbms에 저장된 데이터
  - relational database management system (관계형 데이터 베이스 관리 시스템)
2. 반정형화 데이터 : html, 웹로그 데이터
3. 비정형화 데이터 : 텍스트, 동영상, 이미지 데이터

오라클과 파이썬을 연동하는 이유

1. 오라클 데이터 베이스에서 실시간으로 변하는 데이터를 csv파일로 매번 내리려면 자주 내려야 하므로 바로 연동한다.
2. 파이썬의 여러 장점들을 이용해서 데이터를 분석할 수 있다.  
(통계구현, 시각화, 머신러닝 구현, 업무자동화)
3. 이미지를 숫자로 변환해서 오라클 database에 저장하고 관리한다.
  - db에 저장하면 백업과 복구를 할 수 있고 효율적으로 data를 관리할 수 있다.

오라클과 파이썬 연동

1. 도스창을 열고 리스너의 상태를 확인한다.
  - 리스너 상태 확인하는 명령어 : lsnrctl status
  - 외부에서 오라클에 접속하려면 리스너가 가지고 있는 정보중 3가지를 확인해야 접속이 가능하다.
    - a. ip 주소 : localhost
    - b. 포트번호 : 1521
    - c. 서비스 이름 : orcl
2. 위의 정보들을 이용해서 오라클 database에 접속 한다.

```
sqlplus scott/tiger
exit
```

```
sqlplus system/oracle@localhost:1521/orcl
```

```
sqlplus sys/oracle as sysdba
exit
```

```
sqlplus sys/oracle@localhost:1521/orcl as sysdba
```

3. 아나콘다 프롬프트창을 열고 cx\_Oracle 모듈을 설치한다.

4. 파이썬에서 오라클과 연동하는 코드를 작성한다.

(spyder 에서 작성)

```
import cx_Oracle
import pandas as pd
```

```
dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl') # 오라클 주소를 기입
```

```
print (dsn)
```

```
db = cx_Oracle.connect('scott', 'tiger', dsn) # 오라클 접속 유저 정보
```

```
cursor = db.cursor() # 결과 데이터를 담은 메모리 이름을 cursor로 선언
```

```
cursor.execute('select * from emp ') # 쿼리문 작성결과가 cursor 메모리에 담는다.
```

```
row = cursor.fetchall() # cursor 메모리에 담긴 결과를 한번에 row 변수에 담는다.
```

```
emp = pd.DataFrame(row)
```

```
print (emp)
```

문제440. 위의 emp테이블 전체를 select 하지 않고 아래의 쿼리의 결과만 select 하시오.

```
select empno, ename, sal, deptno
 from emp;
```

```
import cx_Oracle
import pandas as pd
```

```
dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl') # 오라클 주소를 기입
```

```
print (dsn)
```

```
db = cx_Oracle.connect('scott', 'tiger', dsn) # 오라클 접속 유저 정보
```

```
cursor = db.cursor() # 결과 데이터를 담은 메모리 이름을 cursor로 선언
```

```
cursor.execute('select empno, ename, sal, deptno from emp ') # 쿼리문 작성결과가 cursor 메모리에 담는다.
```

```
row = cursor.fetchall() # cursor 메모리에 담긴 결과를 한번에 row 변수에 담는다.
```

```
emp = pd.DataFrame(row)
```

```
print (emp)
```

문제441. dept 테이블의 모든 데이터를 조회 하시오.

```
import cx_Oracle
import pandas as pd
```



```
dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute('select * from dept ')
row = cursor.fetchall()
dept = pd.DataFrame(row)

print (dept)
```

문제442. 우리반 테이블(emp12)를 조회하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute('select * from emp12 ')
row = cursor.fetchall()
emp12 = pd.DataFrame(row)

print (emp12)
```

문제443. 월급이 1200이상인 직원들의 이름과 월급을 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute('select ename, sal from emp where sal >= 1200 ')
row = cursor.fetchall()
emp = pd.DataFrame(row)

print (emp)
```

문제444. 직원 테이블의 월급을 막대 그래프로 그리시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute('select ename, sal from emp')
row = cursor.fetchall()
emp = pd.DataFrame(row)
```

```
emp.index = list(emp[0])
emp.plot(kind = 'bar')
```

문제445. 위의 그래프의 색깔을 변경하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select ename, sal from emp""")
row = cursor.fetchall()
emp = pd.DataFrame(row)
emp.index = list(emp[0])
emp.plot(kind = 'bar', color = 'gray')
```

문제446. 직업, 직업별 최대월급을 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select job, max(sal) from emp group by job """)
row = cursor.fetchall()
emp = pd.DataFrame(row)

print (emp)
```

문제447. 직업, 직업별 토탈월급을 출력하는데 업별 토탈월급이 높은 것부터 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select job, sum(sal)
 from emp
 group by job
 order by sum(sal) desc """)
row = cursor.fetchall()
emp = pd.DataFrame(row)

print (emp)
```

문제448. 이름, 월급, 순위를 출력하는데 순위가 월급이 높은 사원 순으로 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select job, sal,
 dense_rank() over(order by sal desc)
 from emp """)
row = cursor.fetchall()
emp = pd.DataFrame(row)

print (emp)
```

문제449. 부서번호, 부서번호별 토탈월급을 출력하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select deptno, sum(sal)
 from emp
 group by deptno""")
row = cursor.fetchall()
emp = pd.DataFrame(row)

print (emp)
```

문제450. 위의 결과를 막대그래프로 시각화 하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select deptno, sum(sal)
 from emp
 group by deptno""")
row = cursor.fetchall()
emp = pd.DataFrame(row)
emp.index = list(emp[0])
```

```
emp.plot(kind = 'bar', color = 'gray')
```

문제451. emp 테이블의 컬럼명이 출력되게 하시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
 from emp""")
row = cursor.fetchall()
emp = pd.DataFrame(row)
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

print (col)
```

문제452. 위의 col 리스트에 담긴 컬럼명을 위의 데이터에 매핑시키시오.

```
import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
 from emp""")
row = cursor.fetchall()
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

print (emp)
```

문제453. 위의 컬럼명을 이용해서 판다스 문법으로 이름과 월급을 출력하시오.

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
 from emp""")
row = cursor.fetchall()
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

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

```

문제454. 월급이 3000 이상인 사원들 이름과 월급을 출력하시오.

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select *
 from emp""")
row = cursor.fetchall()
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

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

```

문제455. 이름과 부서위치를 출력하시오.

(SQL로 작성)

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()

```

```

cursor.execute("""select e.ename, d.loc
 from emp e, dept d
 where e.deptno = d.deptno""")
row = cursor.fetchall()
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

print (emp)

```

문제456. 부서위치, 부서위치별 토탈월급을 출력하시오.

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
 from emp e, dept d
 where e.deptno = d.deptno
 group by d.loc""")
row = cursor.fetchall()
colname = cursor.description # emp 테이블의 컬럼명을 가져온다.
col = []

for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

print (emp)

```

문제457. 위의 결과를 막대그래프로 시각화 하시오.

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute("""select d.loc, sum(e.sal)
 from emp e, dept d
 where e.deptno = d.deptno
 group by d.loc""")
row = cursor.fetchall()
emp = pd.DataFrame(row)

```

```
emp.index = list(emp[0])
emp.plot(kind = 'bar', color = 'gray')
```

## 149. 파이썬과 mySQL 연동

현업에서는 오라클도 많이 쓰지만 mySQL도 많이 사용한다.

### 1. mySQL 설치방법

[m.blog.naver.com/bjh7007/221829548634](http://m.blog.naver.com/bjh7007/221829548634)

### 2. mySQL 실행하고 database를 생성하는 방법

비밀번호(oracle) 입력 후

```
mysql> create database orcl;
```

```
mysql> use orcl;
```

### 3. mySQL에서 emp 테이블 생성하는 방법

<http://cafe.daum.net/oracleoracle/SfZF/980>

문제458. 월급이 3000이상인 직원들의 이름과 월급을 출력하시오.

```
mysql> select ename, sal
-> from emp
-> where sal >= 3000;
```

문제459. 직업이 SALESMAN인 직원들의 이름과 월급과 직업을 출력하는데 월급이 높은 직원부터 출력하시오.

```
mysql> select ename, sal, job
-> from emp
-> where job = 'SALESMAN'
-> order by sal desc;
```

문제460. 이름과 부서위치를 출력하시오.

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

문제461. JONES 보다 더 많은 월급을 받는 직원들의 이름과 월급을 출

력하시오.

```
mysql> select ename, sal
-> from emp
-> where sal > (select sal
-> from emp
-> where ename = 'JONES');
```

문제462. 이름, 월급, 순위를 출력하시오.

```
mysql> select ename, sal, rank() over(order by sal desc) 순위
-> from emp;
```

### oracle와 mySQL 함수 비교

| oracle         | mySQL        |
|----------------|--------------|
| nvl            | ifnull       |
| sysdate        | sysdate()    |
| months_between | period_add   |
| decode         | if           |
| rollup         | with rollup  |
| listagg        | group_concat |

문제463. 이름, 커미션을 출력하는데 커미션이 null인 사람들은 0 으로 출력하시오.

```
mysql> select ename, ifnull(comm, 0)
-> from emp;
```

문제464. 오늘날짜를 출력하시오.

```
mysql> select sysdate();
```

mySQL은 dual이 없다.

문제465. 부서번호, 부서번호별 토탈월급을 출력하시오.

```
mysql> select deptno, sum(sal)
-> from emp
-> group by deptno;
```



문제466. 위의 결과를 다시 출력하는데 맨 아래에 전체 토탈월급이 출력되게 하시오.

```
mysql> select deptno,sum(sal)
-> from emp
-> group by deptno with rollup;
```

문제467. scott의 월급을 0으로 변경하시오.

```
mysql> update emp
-> set sal = 0
-> where ename = 'SCOTT';
```

```
mysql> rollback;
Query OK, 0 rows affected (0.00 sec)
```

오라클과 다르게 mySQL은 자동커밋이 활성화 되어있기 때문에 rollback 되지 않는다.

문제468. 자동커밋이 활성화 되어져 있는지 확인하시오.

```
mysql> select @@autocommit;
+-----+
| @@autocommit |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

결과가 숫자 1 이면 autocommit이 활성화 된 것이다.

문제469. 자동 커밋 기능을 끄시오.

```
mysql> set autocommit = FALSE;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select @@autocommit;
+-----+
| @@autocommit |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

결과가 숫자 0 이면 autocommit이 비활성화 된 것이다.

문제470. king의 월급을 0로 변경하시오.

```
mysql> update emp
-> set sal = 0
-> where ename = 'KING';
```

```
mysql> rollback;
```

자동커밋 기능을 비활성화 하고 나서는 rollback이 가능하다.

mySQL을 실행하고 제일 먼저 자동커밋기능이 활성화 되어있는지 확인해야 한다.

문제471. 오라클의 listagg 함수와 같은 기능인 group\_concat을 이용해서 부서번호, 부서번호별로 속한 사원들의 이름을 가로로 출력하시오.

```
mysql> select deptno, group_concat(ename)
-> from emp
-> group by deptno;
```

문제472. mySQL과 파이썬을 연동해서 mySQL의 emp 테이블을 파이썬에서 출력하시오.

```
import pymysql,pandas as pd
```

```
conn = pymysql.connect(host="localhost", user="root",password="oracle", db="orcl",charset="utf8")
```

```
curs = conn.cursor()
sql = "select * from emp"
curs.execute(sql)
rows = curs.fetchall()
colname = curs.description
col = []
for i in colname:
 col.append(i[0].lower())
emp = pd.DataFrame(list(rows),columns=col)
print(emp[['ename', 'sal']])
```

아나콘다 프롬프트 창을 열고 pymysql 을 설치한다.

```
conda install pymysql
```

문제473. 직업, 직업별 토탈월급을 출력하시오.

```
import pymysql,pandas as pd
```

```

conn = pymysql.connect(host="localhost", user="root",password="oracle", db="orcl",charset="utf8")

curs = conn.cursor()
sql = '''select job, sum(sal)
 from emp
 group by job'''
curs.execute(sql)
rows = curs.fetchall()
colname = curs.description
col = []
for i in colname:
 col.append(i[0].lower())
emp = pd.DataFrame(list(rows),columns=col)
print(emp)

```

문제474. 위의 결과를 막대그래프로 시각화 하시오.

```

import pymysql,pandas as pd

conn = pymysql.connect(host="localhost", user="root",password="oracle", db="orcl",charset="utf8")

curs = conn.cursor()
sql = ('''select job, sum(sal)
 from emp
 group by job''')
curs.execute(sql)
rows = curs.fetchall()

emp = pd.DataFrame(rows)
emp.index = list(emp[0])
emp.plot(kind = 'bar')

```

문제475. 오라클과 파이썬을 연동하여 아래의 직원들을 검색하시오.  
이름, 월급, 부서번호, 자기가 속한 부서번호의 평균월급을 출력하는데  
자신의 월급이 자기가 속한 부서번호의 평균월급보다 더 큰 직원들만  
출력하시오.

```

import cx_Oracle
import pandas as pd

dsn = cx_Oracle.makedsn('localhost', 1521, 'orcl')

db = cx_Oracle.connect('scott', 'tiger', dsn)
cursor = db.cursor()
cursor.execute('''select e.ename, e.sal, e.deptno, a.asal
 from emp e, (select deptno, avg(sal) asal
 from emp
 group by deptno) a
 where e.deptno = a.deptno
 and e.sal > a.asal''')
row = cursor.fetchall()

```

```
colname = cursor.description
col = []
for i in colname:
 col.append(i[0].lower())

emp = pd.DataFrame(list(row), columns = col)

print (emp)
```

# 20.12.22

2020년 12월 22일 화요일 오전 9:29

## 150. 이미지를 숫자로 변환하는 방법 ( 폐사진 )

딥러닝 수업을 할 때 이미지를 인공 신경망에 넣어서 학습을 시키는데 인공 신경망에 사진을 넣을 때 숫자로 변환을 해서 넣어야 한다.

ex1)

c 드라이브 밑에 images폴더에 있는 사진들의 이름을 불러오는 함수를 생성한다.

```
import os

test_image = 'c:\\images'

def image_load(path):
 file_list = os.listdir(path) # 해당 디렉토리의 파일명을 추출한다.
 return file_list

print (image_load(test_image))
```

ex2)

위의 결과에서 숫자만 나오게 하시오.

```
import os
import re # 데이터 정제 전문 모듈

test_image = 'c:\\images'

def image_load(path):
 file_list = os.listdir(path) # 해당 디렉토리의 파일명을 추출한다.
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i) # i의 값중에서 숫자가 아닌 것들은 싱글 두개를 붙인다.
 file_name.append(a)
 return file_name

print (image_load(test_image))
```

문제476. 위에서 출력되고 있는 리스트 안의 요소들은 문자인데 숫자로 변경하시오.

```
import os
import re
```

```

test_image = 'c:\\images'

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

print(image_load(test_image))

```

문제477. 위의 리스트의 요소가 ascending 하게 정렬되게 하시오.

```

import os
import re

test_image = 'c:\\images'

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

print(image_load(test_image))

```

문제478. 위의 함수의 코드를 추가하여 아래와 같이 출력되게 하시오.

```
['1.png', '2.png', '3.png', ... , '20.png']
```

```

import os
import re

test_image = 'c:\\images'

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

 file_res = []
 for k in file_name:
 file_res.append(str(k) + '.png')

 return file_res

```

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

문제479. 위의 함수의 코드를 수정해서 아래와 같이 이름 앞에 절대경로가 붙게 하시오.

```
['c:\\images\\1.png', 'c:\\images\\2.png', 'c:\\images\\3.png', ... , 'c:\\images\\20.png']
```

```
import os
import re
```

```
test_image = 'c:\\images'
```

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

```
file_res = []
for k in file_name:
 file_res.append('c:\\images\\' + str(k) + '.png')
```

```
return file_res
```

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

ex3)

폐사진 이미지를 숫자로 변환하기 위하여 필요한 파이썬 모듈을 install 하시오.

아나콘다 프롬프트 창을 연다.

```
conda install opencv
```

위의 명령어로 했을 때 에러가 나면 아래와 같이 입력한다.

```
pip install opencv-python
```

opencv 모듈은 이미지를 파이썬에서 숫자로 변환하고 다양한 이미지 처리를 할 수 있게 해주는 기술을 제공하는 함수

구글 지도나 카카오 지도에 street view가 있는데 사람얼굴이나 자동차 번호판을 모자이크 처리 해줘야 한다.

ex4)

위에서 설치한 opencv모듈을 이용해서 폐사진을 숫자로 변환하시오.

```
import cv2 # opencv모듈을 임포트 한다.
import os
import re

test_image = 'c:\\images'

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

 file_res = []
 for k in file_name:
 file_res.append('c:\\images\\' + str(k) + '.png')

 image = []
 for h in file_res:
 img = cv2.imread(h) # 이미지를 숫자로 변환하는 코드
 image.append(img)

 return image

print (image_load(test_image))
```

ex5)

위의 숫자로 변환한 리스트를 인공지능망에 넣기 위해서는 numpy 모듈의 array 형태로 제공해 줘야한다. 위의 리스트를 numpy array로 변환 하시오.

```
import numpy as np # 행렬 연산을 쉽고 빠르게 할 수 있게해주는 모듈
import cv2
import os
import re

test_image = 'c:\\images'

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

 file_res = []
 for k in file_name:
 file_res.append('c:\\images\\' + str(k) + '.png')
```



```

image = []
for h in file_res:
 img = cv2.imread(h)
 image.append (img)

return np.array(image, dtype = object)

print (image_load(test_image))

```

## 151. 이미지를 숫자로 변환하는 방법 (개와 고양이)

개와 고양이 사진을 분류하는 인공지능망을 만드려면 개와 고양이 사진을 각각 2000장씩 내려 받아서 숫자로 변환하는 작업을 해줘야 한다.

ex1)

머신러닝 데이터 분석의 세계대회인 캐글에서 개와 고양이 사진을 내려 받으시오.

ex2)

개 사진 30장만 c 드라이브 밑에 images2 라는 폴더에 넣으시오.

ex3)

c드라이브 밑에 images2 라는 폴더에 있는 이미지 이름을 가져오는 함수를 image\_load2라는 함수로 생성하시오.

```

import os

path = 'c:\\images2'

def image_load2(path):
 file_list = os.listdir(path)
 return file_list

print (image_load2(path))

```

ex4)

위의 개사진 이름에서 숫자만 출력하시오.

```

import os
import re

path = 'c:\\images2'

def image_load2(path):
 file_list = os.listdir(path)
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i)
 file_name.append(int(a))
 return file_name

```

```
print (image_load2(path))
```

ex5)

file\_name 리스트 안의 숫자를 정렬해서 출력되게 하시오.

```
import os
```

```
import re
```

```
path = 'c:\\images2'
```

```
def image_load2(path):
 file_list = os.listdir(path)
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i)
 file_name.append(int(a))
 file_name.sort()
 return file_name
```

```
print (image_load2(path))
```

ex6)

아래와 같이 절대 경로와 확장자가 붙어서 출력되게 하시오.

```
['c:\\images2\\dog1.jpg', 'c:\\images2\\dog2.jpg', ... , 'c:\\images2\\dog30.jpg']
```

```
import os
```

```
import re
```

```
path = 'c:\\images2'
```

```
def image_load2(path):
 file_list = os.listdir(path)
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i)
 file_name.append(int(a))
 file_name.sort()

 file_res = []
 for k in file_name:
 file_res.append('c:\\images2\\dog.' + str(k) + '.jpg')

 return file_res
```

```
print (image_load2(path))
```

ex7)

위의 개 사진 이미지들을 opencv와 numpy를 이용해서 숫자로 변환하고 numpt array로 반환되게 하시오.

```
import os
```

```

import re
import numpy as np
import cv2

path = 'c:\\images2'

def image_load2(path):
 file_list = os.listdir(path)
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i)
 file_name.append(int(a))
 file_name.sort()

 file_res = []
 for k in file_name:
 file_res.append('c:\\images2\\dog.' + str(k) + '.jpg')

 image = []
 for h in file_res:
 img = cv2.imread(h)
 image.append(img)

 return np.array(image, dtype = object)

print (image_load2(path))

```

문제480. 직접 스크롤링한 사진 20장을 images3 폴더에 넣고 숫자로 변환하는 함수를 image\_load3 로 생성하시오.

```

import os
import re
import numpy as np
import cv2

path = 'c:\\images3'

def image_load3(path):
 file_list = os.listdir(path)
 file_name = []
 for i in file_list:
 a = re.sub('[^0-9]', '', i)
 file_name.append(int(a))
 file_name.sort()

 file_res = []
 for k in file_name:
 file_res.append('c:\\images3\\' + str(k) + '.jpg')

 image = []
 for h in file_res:
 img = cv2.imread(h)
 image.append (img)

```

```
return np.array(image, dtype = object)

print (image_load3(path))
```

## 152. 필수 알고리즘1 (합성곱 연산)

프로그래머스 사이트의 알고리즘 문제 풀어보기

합성곱 연산 알고리즘 : 딥러닝의 필수 알고리즘 (퍼셉트론, 합성곱 연산 알고리즘)

- 이미지의 형상을 무시하지 않고 이미지를 그대로 인공신경망이 학습할 수 있게 해준 수학적 행렬 연산이다.

합성곱에서 원본 이미지는 학습해야할 데이터(사진)이고 필터(filter)는 원본 이미지에서 특징을 잡아내는데 사용되는 행렬이다.

특징을 잡아서 feature map을 생성하여 원본이미지의 형태를 이해하는 것을 합성곱 연산이라고 한다.

문제481. 아래의 두 행렬을 만들고 덧셈연산을 하시오.

```
[[1 2 3]
 [0 1 2]
 [3 0 1]]
```

```
[[2 0 1]
 [0 1 2]
 [1 0 2]]
```

```
import numpy as np
```

```
a = [[1,2,3],[0,1,2],[3,0,1]]
b = [[2,0,1],[0,1,2],[1,0,2]]
```

```
a2 = np.array(a)
b2 = np.array(b)
```

```
print (a2 + b2)
```

문제482. 아래의 두 행렬을 만들고 두 행렬의 원소들의 곱을 출력하십시오.

```
[[1 2 3]
 [0 1 2]
 [3 0 1]]
```

```
[[2 0 1]
 [0 1 2]]
```

[1 0 2]]

```
import numpy as np

a = [[1,2,3],[0,1,2],[3,0,1]]
b = [[2,0,1],[0,1,2],[1,0,2]]

a2 = np.array(a)
b2 = np.array(b)

print (a2 * b2)
```

문제483. 위에서 출력된 결과의 요소들을 모두 더하시오.

```
import numpy as np

a = [[1,2,3],[0,1,2],[3,0,1]]
b = [[2,0,1],[0,1,2],[1,0,2]]

a2 = np.array(a)
b2 = np.array(b)

result = a2 * b2

print (np.sum(result)) # 행렬안의 원소들의 합을 출력
```

numpy란 python 언어에서 기본적으로 지원하지 않는 배열(array) 혹은 행렬(matrix)의 계산을 쉽게 해주는 라이브러리이다. 머신러닝에서 많이 사용하는 선형대수학에 관련된 수식들을 python에서 쉽게 프로그래밍 할 수 있게 해준다.

문제484. 아래의 4 x 4 행렬을 만드시오.

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]

import numpy as np

a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]

a2 = np.array(a)

print (a2)
```

문제485. 아래의 4 x 4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

 .

풀어주세요.

```
[1 2 3 0]
[0 1 2 3]
[3 0 1 2]
[2 3 0 1]]
```

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
a2 = np.array(a)
```

```
print (a2[0:3, 0:3])
```

문제486. 아래의 4 x 4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

```
[[1 2 3 0]
[0 1 2 3]
[3 0 1 2]
[2 3 0 1]]
```

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
a2 = np.array(a)
```

```
print (a2[0:3, 1:4])
```

문제487. 아래의 4 x 4 행렬에서 빨간색으로 지정한 영역의 숫자들만 출력하시오.

```
[[1 2 3 0]
[0 1 2 3]
[3 0 1 2]
[2 3 0 1]]
```

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
a2 = np.array(a)
```

```
print (a2[1:4, 0:3])
```

문제488. 아래의 4 x 4 행렬에서 빨간색으로 지정한 영역의 숫자들만

출력하시오.

```
[[1 2 3 0]
 [0 1 2 3]
 [3 0 1 2]
 [2 3 0 1]]
```

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
a2 = np.array(a)
```

```
print (a2[1:4, 1:4])
```

문제489. 아래의 4 x 4 행렬에서 위에서 빨간색으로 지정된 4개의 영역의 숫자들을 for loop문을 이용해서 모두 출력하시오.

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
a2 = np.array(a)
```

```
for i in range(2): # 0, 1
 for k in range(2): # 0, 1
 print (a2[i:i+3, k:k+3])
 # i 가 0 일 때 j 가 0, 1
 # 0:3, 0:3
 # 0:3, 1:4
 # i 가 1 일 때 j 가 0, 1
 # 1:4, 0:3
 # 1:4, 1:4
```

문제490. 위에서 선택한 4개의 행렬(3 x 3)과 아래의 filter 행렬(3 x 3)의 원소의 곱을 출력하시오.

```
import numpy as np
```

```
a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
```

```
f = [[2,3,4],[1,2,3],[2,0,1]]
```

```
a2 = np.array(a)
```

```
f2 = np.array(f)
```

```
for i in range(2):
 for k in range(2):
 print (a2[i:i+3, k:k+3] * f)
```

문제491. 위에서 출력된 행렬 4개에 대한 원소들의 합이 각각 출력되게 하시오.

```
import numpy as np

a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
f = [[2,3,4],[1,2,3],[2,0,1]]

a2 = np.array(a)
f2 = np.array(f)

for i in range(2):
 for k in range(2):
 print (np.sum(a2[i:i+3, k:k+3] * f2))
```

문제492. 위에서 출력된 4개의 숫자로 아래의 행렬 (2 x 2)을 만드시오.

```
35 29
21 35
```

```
import numpy as np

a = [[1,2,3,0],[0,1,2,3],[3,0,1,2],[2,3,0,1]]
f = [[2,3,4],[1,2,3],[2,0,1]]

a2 = np.array(a)
f2 = np.array(f)

result = []

for i in range(2):
 for k in range(2):
 result.append (np.sum(a2[i:i+3, k:k+3] * f2))

result2 = np.array(result).reshape(2,2)
print (result2)
```

a2 라는 원본 이미지(개사진)에 filter (랜덤으로 생선한 이미지)를 가지고 원본 이미지를 스트라이드(양옆 위 아래로 스캔)하면서 특징을 잡아내어 특징 이미지를 추출 (result2) 하는 것을 합성곱이라고 한다.

문제493. 아래의 원본 이미지 행렬(5 x 5)에서 필터행렬(4 x 4)로 스트라이딩해서 합성곱해서 특징을 추출하시오.

```
[[2 3 1 4 7]
 [3 1 6 4 3]
 [2 1 5 3 1]
 [6 2 4 1 2]
```



```
[7 3 1 2 3]]
```

```
[[3 1 4 1]
[2 3 3 4]
[5 1 2 1]
[6 1 3 4]]
```

```
import numpy as np
```

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

```
a2 = np.array(a)
f2 = np.array(f)
```

```
result = []
```

```
for i in range(2):
 for k in range(2):
 result.append (np.sum(a2[i:i+4, k:k+4] * f2))
```

```
result2 = np.array(result).reshape(2,2)
print (result2)
```

## 153. 필수 알고리즘2 (이진 탐색)

문제494. 아래의 리스트에서 숫자 3이 있는지 순차 탐색으로 구현하고 있으면 숫자 3이 있습니다. 라는 메시지가 출력되게 하시오.

```
a = [15, 11, 1, 3, 8]
```

```
a = [15, 11, 1, 3, 8]
```

```
for i in a:
 if i == 3:
 print ('숫자 3이 있습니다.')
 break
else:
 print ('숫자 3이 없습니다.')
```

순차탐색 : 주어진 데이터를 처음부터 차례대로 비교하면서 찾는 방법

문제495. 위의 코드를 수정해서 숫자를 물어보게 하고 숫자를 입력하면 해당 숫자가 존재하는 존재하지 않는지 출력되게 하시오.

```
num = int(input('숫자를 입력하시오.'))
```

```
a = [15, 11, 1, 3, 8]
```

```
for i in a:
 if i == num:
 print('숫자', num, '이 있습니다.')
 break
 else:
 print('숫자', num, '이 없습니다.')
```

이진탐색 : 정렬된 데이터를 좌우 둘로 나눠서 원하는 값의 탐색범위를 좁혀가며 찾는 방법

문제496. 아래의 a 리스트에서 중앙값을 찾으시오.

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
```

```
import numpy as np
```

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
a_n = np.array(a)
```

```
print(np.median(a_n))
```

문제497. a 리스트에서 첫번째 숫자부터 중앙값에 해당하는 숫자까지만 검색하시오.

```
import numpy as np
```

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
a_n = np.array(a)
a_m = np.median(a_n)
```

```
print(a[0:a.index(a_m) + 1])
```

문제498. 위의 a 리스트에서 문제497번에서 선택된 숫자들을 중앙값까지 포함해서 모두 지우고 아래의 결과만 출력되게 하시오.

```
[51, 64, 67, 77, 139, 672, 871]
```

```
import numpy as np
```

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
a_n = np.array(a)
a_m = np.median(a_n)
```

```
del(a[0:a.index(a_m) + 1])
print(a)
```

문제499. 위의 결과로 출력된 아래의 리스트에서 중앙값을 출력하시오.

[51, 64, 67, 77, 139, 672, 871]

```
import numpy as np

a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
a_n = np.array(a)
a_m = np.median(a_n)

del (a[0:a.index(a_m) + 1])

a_m2 = np.median(a)
print (a_m2)
```

문제500. 위에서 출력한 중앙값 77 이 내가 검색하고자 하는 67 보다 크다면 아래의 결과 리스트만 출력되게 하시오.

[51, 64, 67]

```
import numpy as np

a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
a_n = np.array(a)
a_m = np.median(a_n)

del (a[0:a.index(a_m) + 1])

a_m2 = np.median(a)
if a_m2 > 67:
 del(a[a.index(a_m2):])
else:
 del(a[:a.index(a_m2)])

print (a)
```

문제501. ebs에 나온 영상대로 이진탐색을 구현하시오.

a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]

a 리스트에서 검색할 숫자를 입력하시오. 67

67은 이진탐색 3번만에 검색되었습니다.

a 리스트에서 검색할 숫자를 입력하시오. 68

68은 a 리스트에 없습니다.

```
import numpy as np
```

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
```

```
num = int(input('a 리스트에서 검색할 숫자를 입력하시오.'))
```

```
cnt = 0
```

```
while True:
```

```
 cnt += 1
```

```
 a_m = np.median(a)
```

```
 if a_m > num:
```

```
 del (a[a.index(a_m):])
```

```
 elif a_m < num:
```

```
 del (a[a.index(a_m)+1])
```

```
 elif a_m == num:
```

```
 print ('%d은 이진탐색 %d 번만에 검색되었습니다.'%(num,cnt))
```

```
 break
```

```
 else:
```

```
 print ('%d은 a 리스트에 없습니다.'%(num))
```

```
 break
```

# 20.12.23

2020년 12월 23일 수요일    오전 9:43

## 154. 필수 알고리즘3 (버블정렬)

인접한 두 수를 비교해서 순서에 맞지 않는 요소를 인접한 요소와 서로 교환하여 정렬하는 정렬 방법

문제502. 아래의 리스트를 만들고 첫번째 요소와 두번째 요소의 순서를 변경하시오.

```
a = [10, 5, 20, 9, 8]
```

```
a = [10, 5, 20, 9, 8]
```

```
temp = a[1] # 1번째 요소를 temp변수에 임시저장
```

```
a[1] = a[0] # 1번째 요소를 0번째 요소로 변경
```

```
a[0] = temp # 0번째 요소를 1번째 요소로 변경
```

```
print (a)
```

문제503. 아래의 a리스트의 첫번째 요소와 두번째 요소의 크기를 비교해서 첫번째 요소의 숫자가 두번째 요소의 숫자보다 크다면 서로의 위치를 바꾸시오.

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
if a[0] > a[1]:
```

```
 temp = a[1]
```

```
 a[1] = a[0]
```

```
 a[0] = temp
```

```
print (a)
```

문제504. 위의 코드에 for loop문을 넣어서 버블정렬하시오.

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
for i in range(0, 6):
```

```

if a[i] > a[i+1]:
 temp = a[i+1]
 a[i+1] = a[i]
 a[i] = temp

```

```
print (a)
```

문제505. 위의 코드를이용해서 버블정렬을 하는 함수를 아래와 같이 생성하시오.

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
print (bubble_sort(a))
```

```
a = [1, 2, 3, 4, 5, 7, 8, 10]
```

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```

def bubble_sort(a):
 for j in range(len(a)):
 for i in range(len(a)-1):
 if a[i] > a[i+1]:
 temp = a[i+1]
 a[i+1] = a[i]
 a[i] = temp
 return a

```

```
print (bubble_sort(a))
```

- 참고

<http://cafe.daum.net/oracleoracle/SfZF/1125>

<http://cafe.daum.net/oracleoracle/SfZF/1111>

## 155. 필수 알고리즘4 (탐욕알고리즘)

머신러닝 배울때 의사결정트리를 구현할 때 사용하는 알고리즘

탐욕알고리즘이란 매 순간마다 최선의 선택을 하는것이다.

선택할 때마다 가장 좋다고 생각되는 것을 선택해 나가며 최종적인 해답을 구하는 알고리즘이다. 이 알고리즘을 설계할 때 주의할 점은 전체를 고려하는게 아니라 문제를 부분적으로 나누어서 나누어진 문제에 대한 최적의 해답을 구하게끔 해야한다.

ex)

14원의 잔돈을 줘야하는데 잔돈의 종류가 10원, 7원, 1원이 있으면 잔돈을 가장 빨리 줄 수 있는 방법

7원 짜리 2개를 준다.

탐욕알고리즘은 10원 1개, 7원 0개, 1원 4개로 주는게 탐욕 알고리즘이다.

문제506. 14를 10으로 나눈 몫을 출력하시오.

```
print (14/10) # 1.4
print (int(14/10)) # 1
print (14//10) # 1
```

문제507. 14를 10으로 나눈 나머지값을 출력하시오.

```
print (14%10) # 4
```

문제508. 숫자를 물어보게 하고 숫자를 10으로 나눈 몫과 나머지 값을 출력하시오.

```
num = int(input('숫자를 입력하시오.'))

print ('몫은 :', num//10)
print ('나머지값은 :', num%10)
```

문제509. 아래의 잔돈리스트를 만들고 잔돈 리스트의 첫번째 요소로 나눈 몫과 나머지값이 출력되게 하시오.

```
coin = [10, 7, 1]

coin = [10, 7, 1]

num = int(input('숫자를 입력하시오.'))

print ('몫은 :', num//coin[0])
print ('나머지값은 :', num%coin[0])
```

문제510. 위의 코드를 함수로 만들어서 실행되게 하시오.

```
def greedy():
 coin = [10, 7, 1]
 num = int(input('잔돈을 입력하시오.'))

 print ('몫은 :', num//coin[0])
```

```
print('나머지값은 :', num%coin[0])
```

```
greedy()
```

문제511. 탐욕 알고리즘을 파이썬으로 구현하시오.

잔돈을 입력하시오. 14

10원 동전 1개, 7원 동전 0개, 1원 동전 4개를 줍니다.

```
def greedy():
```

```
 coin = [10, 7, 1]
```

```
 coin = sorted(coin, reverse = True)
```

```
 num = int(input('잔돈을 입력하시오.'))
```

```
 print('10원 동전', num//coin[0], '개,', '7원 동전', (num%coin[0])//coin[1], '개,',
```

```
 '1원 동전', (num%coin[0])%coin[1], '개로 줍니다.')
```

```
greedy()
```

## 156. 필수 알고리즘5 (재귀알고리즘)

재귀 알고리즘을 이용하면 loop문을 최소화 하면서 코드를 간단하게 작성 수 있는 알고리즘이다.

1. 재귀함수 : 반복문 + 스택구조가 결합된 함수
  - 스택구조 : 후입선출 (나중에 들어간게 먼저 나오는 구조)
2. 재귀함수의 특징 : 재귀함수는 함수 내에서 다시 자기 자신을 호출한 후 그 함수가 끝날때 까지 함수 호출 이후의 명령문을 수행하지 않는다.
3. 함수내에서 다른 함수를 호출하는 예제

```
def hap(a,b):
 return (a+b)
```

```
def gob(a,b):
 return (a*b)
```

```
def hap_gob(a,b):
 k = hap(a,b)
 m = gob(a,b)
 return k + m
```



```
print (hap_gob(2,3))
```

4. 숫자를 입력하면 1부터 해당 숫자까지의 합을 출력하는 함수를 생성하시오.

```
def add_func(a):
 cnt = 0
 for i in range(1,a+1):
 cnt += i
 return cnt
```

```
print (add_func(5))
```

5. 위의 add\_func 함수를 재귀함수로 구현하시오. (재귀함수를 사용하면 loop문을 사용하지 않아도 된다.)

```
def add_func(a):
 if a == 0:
 return 0
 else:
 return a + add_func(a-1)
```

```
print (add_func(5))
```

문제512. factorial 함수를 만드시오.

```
def factorial(a):
 if a == 1:
 return 1
 else:
 return a * factorial(a-1)
```

```
print (factorial(5))
```

문제513. 위의 factorial 함수를 재귀를 이용하지 않고 for loop문으로 구현하시오.

```
def factorial(a):
 cnt = 1
 for i in range(1, a+1):
 cnt *= i
 return cnt
```

```
print (factorial(5))
```

재귀를 이용하면 2가지 장점

1. loop문을 복잡하게 이용하지 않아도 된다.
2. 코드가 더 간결해 진다.

문제514. 오라클의 power함수를 파이썬으로 구현하시오.

```
select power(2,3)
 from dual; # 8
```

```
def power(a,b):
 cnt = 1
 for i in range(1,b+1):
 cnt *= a
 return cnt
```

```
print (power(2,3))
```

문제515. 위의 power함수를 재귀함수로 구현하시오.

```
def power(a,b):
 if b == 0:
 return 1
 else:
 return a * power(a,b-1)
```

```
print (power(2,3))
```

문제516. 구구단 2단을 아래와 같이 출력하는 함수를 생성하시오.

```
print (multi_table2dan(9))
```

```
2 x 1 = 2
2 x 2 = 4
:
2 x 9 = 18
```

```
def multi_table_2dan(a):
 for i in range (1, a+1):
 print ('2 x' , i, '=', 2*i)
```

```
multi_table_2dan(9)
```

문제517. 두 숫자를 입력해서 함수를 실행하면 두 숫자의 최대공약수가 출력되는 함수를 생성하시오.

```
def gcd(a, b):
 for i in range (min(a,b), 0, -1):
 if a%i == 0 and b%i == 0:
 return i
```

```
print (gcd(16,24))
```

문제518. 최대공약수를 출력하는 함수를 재귀함수로 구현하시오.

```
def gcd(n1,n2):
 if n2 == 0:
 return n1
 else:
 return gcd(n2, n1%n2)

print(gcd(16,24))
```

문제519. 구구단 2단을 재귀함수로 출력하시오.

```
def multi_table_2(n):
 if n==0:
 return 0
 else:
 multi_table_2(n-1)
 print('2 x', n, '=', 2*n)

multi_table_2(9)
```

# 20.12.24

2020년 12월 24일 목요일 오전 9:32

## 157. 필수 알고리즘6 (LRU 알고리즘)

Least Recent Used

LRU 알고리즘이란 Oracle DATABASE의 메모리 관리를 효율적으로 하기 위해 고안된 대표적인 알고리즘으로 최신 데이터를 메모리에 유지시키고 오래된 데이터는 메모리에서 내보내게 하는 알고리즘

메모리에 데이터가 있어서 메모리에서 조회 : cache hit

메모리에 데이터가 없어서 디스크에서 조회 : cache miss

한번 디스크에서 읽은 데이터를 메모리에 올려놓고 메모리에서 빠르게 데이터를 조회 할 수 있도록 LRU알고리즘을 구현해서 만든 소프트웨어이다. 그런데 이 메모리공간이 한정된 공간이다 보니 무한히 데이터를 올릴수 없어서 오래된 데이터는 메모리에서 빠져나가고 최신 데이터가 그 빠져나간 자리에 올라가게 된다.

스택 : 후입선출

큐 : 선입선출

알고리즘 문제를 해결하는 순서

1. 문제를 정확하게 이해해야 한다.
2. 번호 순서대로 해결방법을 기술한다.
3. 번호 순서대로 기술한 해결방법을 코딩으로 작성한다.

큰 문제를 작은 문제들로 쪼개서 작은 문제들을 하나씩 해결해 나가는 방법으로 코딩

<http://cafe.daum.net/oracleoracle/SfZF/1180>

LRU알고리즘 문제를 파이썬으로 구현하는 순서

1. 문제를 정확하게 이해해야 한다.
  - cities = ['Jeu', 'Pangyo', 'New York', 'new york']  
print (cacheProcess(cities, 4))  
# 수행시간
    - LRU 알고리즘에 대한 이해가 있어야한다.
    - 함수를 생성해야하는데 입력값이 2개 이다.

(도시이름, 캐시사이즈)

- 함수의 출력값은 총실행시간(초) 이다.

```
a = ['jeju', 'pangyo', 'new york', 'mew york']
```

```
ptinr (함수이름(도시이름,4))
```

```
총 수행시간 (5 + 5 + 5 +1) = 16
```

```
a = ['jeju', 'jeju', 'jeju', 'jeju']
```

```
ptinr (함수이름(도시이름,4))
```

```
총 수행시간 (5 + 1 + 1 +1) = 8
```

```
a = ['jeju', 'jeju', 'jeju', 'jeju']
```

```
ptinr (함수이름(도시이름,0))
```

```
총 수행시간 (5 + 5 + 5 +5) = 20
```

## 2. 번호 순서대로 해결방법을 기술한다.

- 도시이름은 대소문자를 구분하지 않는다.

- ex) 아래의 리스트를 받아서 아래의 결과를 출력하시오.

```
cities = ['Jeu', 'Pangyo', 'New York', 'new york']
```

```
['jeju', 'pangyo', 'new york', 'mew york']
```

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']
```

```
city = []
```

```
for i in cities:
```

```
 city.append(i.lower())
```

```
print (city)
```

```
위의 코드를 comprehension을 사용하여 간단하게 수행하시오.
```

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']
```

```
city = [i.lower() for i in cities]
```

```
print (city)
```

```
comprehension 코드로 위의 5줄 코드를 3줄로 간소화한 코드
```

```
[출력 표현식 for 요소 in 입력시퀀스 if 조건식]
```

- ex) 1부터 20까지의 홀수를 담는 리스트를 만드시오.

```
a = [i for i in range(1, 21) if i%2 == 1]
```

```
print (a)
```

- 메모리(cache)를 만들어야 한다. (입력 숫자만큼 cache 사이즈가 구성)

- ex) 아래의 리스트를 생성하시오.

```
[None, None, None, None]
```

```
cache = [None for i in range(4)]
```

```
print (cache)
```

- 메모리에 데이터를 올린다. (cache miss (5초))

- ex) 아래와 같이 cache 에 data를 올리시오.

```
[None, None, None, 'jeju']
```

```
cache = [None for i in range(4)]
```

```
cache.append('jeju')
```

```
del cache[0]
```

```
print (cache)
```

- ex) 아래와 같이 cache 에 data를 올리시오.

```
[None, None, 'jeju', 'pangyo']
```

```
cache = [None for i in range(4)]
```

```
cache.append('jeju')
```

```
del cache[0]
```

```
cache.append('pangyo')
```

```
del cache[0]
```

```
print (cache)
```

- 메모리에서 데이터를 찾는다. (cache hit (1초))

- ex) 위에서 만든 코드를 가지고 함수를 생성하시오.

```
cities = ['Jeju', 'Pangyo', 'New York', 'new york']
```

```
def cacheProcess (cities, cachesize):
```

```
 city = [i.lower() for i in cities] # 도시이름을 소문자로 변경
```

```
 cache = [None for i in range(1,cachesize+1)] # cache 구성
```

```
 for i in range(0,len(city)):
```

```
 cache.append(city[i])
```

```
 del cache[0]
```

```
 return cache
```

```
print (cacheProcess(cities,4))
```

- ex) 위의 함수의 결과가 cache의 결과가 아니라 수행시간이 출력되게 하시오.

```

cities = ['Jeju', 'Pangyo', 'New York', 'new york']

def cacheProcess (cities, cachesize):
 city = [i.lower() for i in cities]
 cache = [None for i in range(1,cachesize+1)]
 cnt = 0

 for i in range(0,len(city)):
 if city[i] in cache:
 cnt += 1
 cache.append(city[i])
 del cache[0]
 elif city[i] not in cache:
 cnt += 5
 cache.append(city[i])
 del cache[0]
 return cnt

print (cacheProcess(cities,4))

```

문제520. 위에서 만든 cacheProcess 함수를 검증하시오.

```

def cacheProcess (cities, cachesize):
 city = [i.lower() for i in cities]
 cache = [None for i in range(1,cachesize+1)]
 cnt = 0

 for i in range(0,len(city)):
 if city[i] in cache:
 cnt += 1
 cache.append(city[i])
 del cache[0]
 elif city[i] not in cache:
 cnt += 5
 cache.append(city[i])
 del cache[0]
 return cnt

check = []
a = cacheProcess(["Jeju", "Pangyo", "Seoul", "NewYork", "LA", "Jeju", "Pangyo", "Seoul", "NewYork",
"LA"],3)
check.append(a)
a = cacheProcess(["Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul", "Jeju", "Pangyo", "Seoul"],3)
check.append(a)
a =
cacheProcess(["Jeju","Pangyo","Seoul","NewYork","LA","SanFrancisco","Seoul","Rome","Paris","Jeju","Ne
wYork","Rome"],2)
check.append(a)
a =
cacheProcess(["Jeju","Pangyo","Seoul","NewYork","LA","SanFrancisco","Seoul","Rome","Paris","Jeju","Ne
wYork","Rome"],5)
check.append(a)
a = cacheProcess(["Jeju","Pangyo",'NewYork','newyork'],2)
check.append(a)

```

```

a = cacheProcess(["Jeju", "Pangyo", "Seoul", "NewYork", "LA"], 0)
check.append(a)
a = cacheProcess(['Jeju', 'Jeju', 'Jeju'], 3)
check.append(a)
print(check)

```

```

correct = [50, 21, 60, 52, 16, 25, 7]

```

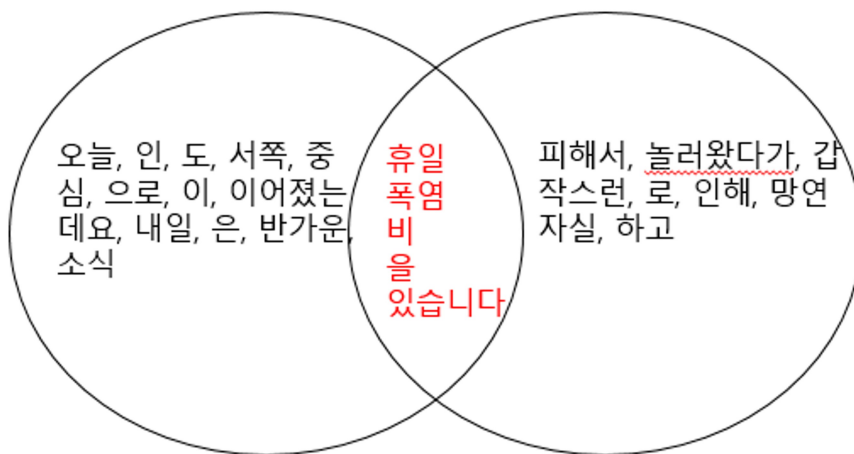
```

for i in range(len(check)) :
 if check[i] != correct[i] :
 print("%i번째 경우가 틀립니다." % i)

```

## 158. 필수 알고리즘7 (자카드 유사도 알고리즘)

자카드 유사도는 두 문장을 각각의 집합으로 만든 뒤 두 집합을 통해 유사도를 측정하는 알고리즘이다.



두 문장의 교집합이 6개, 합집합이 24개 이므로 자카드 유사도는  $6/24 = 0.25$  가 된다.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

ex1) 다음 A 집합과 B 집합의 자카드 유사도를 구하시오.

A = {1, 2, 3}

B = {2, 3, 4}

$A \cap B = \{2, 3\}$

$A \cup B = \{1, 2, 3, 4\}$



$$J(A, B) = 2/4 = 0.5$$

ex2) FRANCE, FRENCH가 주어졌을 때 자카드 유사도를 구하시오.

A = {FR, RA, AN, NC, CE}

B = {FR, RE, EN, NC, CH}

$A \cap B = \{FR, NC\}$

$A \cup B = \{FR, RA, AN, NC, CE, RE, EN, CH\}$

$$J(A, B) = 2/8 = 0.25$$

## 파이썬으로 합집합과 교집합 구현하기

a = {1, 2, 3, 4}

b = {2, 4, 5}

print (type(a)) # <class 'set'> 집합 자료구조

### 1. a 와 b 의 합집합 구하기

a = {1, 2, 3, 4}

b = {2, 4, 5}

result1 = a.union(b)

print (result1) # {1, 2, 3, 4, 5}

### 2. a 와 b 의 교집합 구하기

a = {1, 2, 3, 4}

b = {2, 4, 5}

result2 = a.intersection(b)

print (result2) # {2, 4}

### 3. a 와 b 의 자카드 유사도 구하기

a = {1, 2, 3, 4}

b = {2, 4, 5}

result1 = a.union(b)

print (result1)

result2 = a.intersection(b)

print (result2)

result3 = len(result2)/len(result1)

print (result3) # 0.4

## 리스트로 합집합과 교집합 구현하기

### 1. a 와 b 의 합집합 구하기

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

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

```
result1 = set(a+b)
```

```
print (result1) # {1, 2, 3, 4, 5}
```

### 2. a 와 b 의 교집합 구하기

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

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

```
result2 = []
```

```
for i in a: # 개수가 많은 것을 for 문에 넣는다.
```

```
 if i in b:
```

```
 result2.append(i)
```

```
print (result2) # [2, 4]
```

## 자카드 유사도 알고리즘 문제를 파이썬으로 구현

### 1. 문제를 정확하게 이해해야 한다.

- FRANCE, FRENCH 두 단어의 자카드 유사도를 확인하시오.

### 2. 번호 순서대로 해결방법을 기술한다.

- 아래와 같이 두 문장을 받아서 두개의 철자로 분리하여 리스트에 저장한다.

문자로 된 글자 쌍만 유효하다. (함수이름 str\_split로 생성)

```
a = {FR, RA, AN, NC, CE}
```

```
b = {FR, RE, EN, NC, CH}
```

```
▪ str1 = input('문자열을 입력하시오.').upper()
```

```
str2 = input('문자열을 입력하시오.').upper()
```

```
def str_split(string):
```

```
 res = []
```

```
 for i in range(len(string)-1):
```

```
 if string[i].isalpha() and string[i+1].isalpha(): # 문자로 된 글자 쌍인지 확인
```

```
 res.append (string[i:i+2])
```

```
 return res
```

```
print (str_split(str1))
```

```
print (str_split(str2))
```

- 두 리스트의 a 와 b 두개의 리스트의 교집합을 구한다.

- ```
a = ['FR', 'RA', 'AN', 'NC', 'CE']  
b = ['FR', 'RE', 'EN', 'NC', 'CH']
```

```
intersection = []  
for i in a:  
    if i in b:  
        intersection.append(i)  
  
print (intersection)
```

- 두 리스트의 a 와 b 두개의 리스트의 합집합을 구한다.

- ```
a = ['FR', 'RA', 'AN', 'NC', 'CE']
b = ['FR', 'RE', 'EN', 'NC', 'CH']
```

```
union = set(a + b)

print (union)
```

- 두 리스트의 자카도 유사도를 구한다. (결과에 65536을 곱해서 출력)

- ```
a = ['FR', 'RA', 'AN', 'NC', 'CE']  
b = ['FR', 'RE', 'EN', 'NC', 'CH']
```

```
intersection = []  
for i in a:  
    if i in b:  
        intersection.append(i)  
  
union = set(a + b)  
  
len_i = len(intersection)  
len_u = len (union)  
  
print (int(len_i / len_u * 65536))
```

문제521. 지금까지의 코드를 함수로 구현해서 아래와 같이 실행되게 하시오.

```
print (Jaccard())
```

실행결과:

문자열을 입력해주세요. : FRANCE

문자열을 입력해주세요. : french

16384

```

def Jaccard():

    str1 = input('문자열을 입력하시오.').upper()
    str2 = input('문자열을 입력하시오.').upper()

    def str_split(string):
        res = []
        for i in range(len(string)-1):
            if string[i].isalpha() and string[i+1].isalpha(): # 문자로 된 글자 쌍인지 확인
                res.append (string[i:i+2])
        return res

    a = (str_split(str1))
    b = (str_split(str2))

    intersection = []
    for i in a:
        if i in b:
            intersection.append(i)

    union = set(a + b)
    len_i = len(intersection)
    len_u = len (union)

    return (int(len_i / len_u * 65536))

print (Jaccard())

```

3. 번호 순서대로 기술한 해결방법을 코딩으로 작성한다.

문제522. 지금까지 만든 Jaccard 함수에 아래 내용의 코드를 추가하시오. <http://cafe.daum.net/oracleoracle/SfZF/1208>

(집합 A와 B 가 모두 공집합일 경우 1로 정의한다.)

```

def Jaccard():

    str1 = input('문자열을 입력하시오.').upper()
    str2 = input('문자열을 입력하시오.').upper()

    def str_split(string):
        res = []
        for i in range(len(string)-1):
            if string[i].isalpha() and string[i+1].isalpha(): # 문자로 된 글자 쌍인지 확인
                res.append (string[i:i+2])
        return res

    a = (str_split(str1))
    b = (str_split(str2))

```

```

intersection = []
for i in a:
    if i in b:
        intersection.append(i)

union = set(a + b)
len_i = len(intersection)
len_u = len(union)

if len_i == 0 and len_u == 0:
    return 1 * 65536
else:
    return (int(len_i / len_u * 65536))

print (Jaccard())

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

result = set(a + b)

print (result) # {1, 2, 3, 4, 5}

```

위의 합집합은 중복을 제거한다.

문제523. set을 이용하지 않고 아래의 a 와 b 두개의 리스트 합집합이 {1, 1, 1, 2, 2, 3, 4, 5} 와 같이 출력되게 하시오.

```

import collections
a = [1,1,1,2,2,3]
b = [1,1,2,2,4,5]

result = collections.Counter(a) & collections.Counter(b) # 교집합
intersection = list(result.elements()) # 요소만 리스트로 빼내오기

result2 = collections.Counter(a) | collections.Counter(b) # 합집합
union = list(result2.elements()) # 요소만 리스트로 빼내오기

print(intersection)
print(union)

```

문제524. collections을 이용하여 자카드 유사도를 구현하시오.

```

import collections

def Jaccard():

    str1 = input('문자열을 입력하시오.').upper()
    str2 = input('문자열을 입력하시오.').upper()

```

```

def str_split(string):
    res = []
    for i in range(len(string)-1):
        if string[i].isalpha() and string[i+1].isalpha(): # 문자로 된 글자 쌍인지 확인
            res.append (string[i:i+2])
    return res

a = (str_split(str1))
b = (str_split(str2))
result = collections.Counter(a) & collections.Counter(b)
intersection = list(result.elements())
result2 = collections.Counter(a) | collections.Counter(b)
union = list(result2.elements())
len_i = len(intersection)
len_u = len (union)

if len_i == 0 and len_u == 0:
    return 1 * 65536
else:
    return (int(len_i / len_u * 65536))

print (Jaccard())

```

20.12.28

2020년 12월 28일 월요일 오전 9:45

159. 필수 알고리즘8 (비밀지도)

<http://cafe.daum.net/oracleoracle/SfZF/1238>

알고리즘 문제를 해결하는 순서

1. 문제를 정확하게 이해해야 한다.
2. 번호 순서대로 해결방법을 기술한다.
 - 십진수를 이진수로 변환한다.

문제525. 숫자 9 를 이진수로 변환 하시오.

```
print (bin(9))
```

문제526. 위의 결과에서 숫자만 출력되게 하시오.

```
print (bin(9)[2:])
```

문제527. 숫자 30의 이진수를 아래와 같이 출력하시오.

결과 : 11110

```
print (bin(30)[2:])
```

문제528. 아래와 같이 출력되는 change 함수를 생성하시오.

```
print (change(30))
```

```
# 11110
```

```
def change(num):  
    return bin(num)[2:]
```

```
print (change(30))
```

문제529. 위에서 만든 change 함수에 숫자 9를 넣고 출력하시오.

```
def change(num):  
    return bin(num)[2:]
```

```
print (change(9))
```

문제530. change 함수를 실행할 때 아래와 같이 입력 매개변수를 하나 더 만들어서 입력매개변수의 길이 만큼 숫자 0이 채워져서 출력되게 하시오.

```
print (change(9, 5))
```

```
# 01001
```

```
def change(num, n):  
    return bin(num)[2:].rjust(n, '0')
```

```
print (change(9, 5))
```

rjust(5, '0')은 출력되는 자리수를 전체 5자리로 잡고 나머지 왼쪽에 '0'을 채워넣는다.

문제531. 카카오 문제의 예제를 가지고 아래의 결과를 출력하시오.

```
n = 5
```

```
arr1 = [9, 20, 28, 18, 11]
```

```
arr2 = [30, 1, 21, 17, 28]
```

```
# 01001 11110
```

```
# 10100 00001
```

```
# 11100 10101
```

```
# 10010 10001
```

```
# 01011 11100
```

```
def change(num, n):  
    return bin(num)[2:].rjust(n, '0')
```

```
n = 5
```

```
arr1 = [9, 20, 28, 18, 11]
```

```
arr2 = [30, 1, 21, 17, 28]
```

```
for i in range(len(arr1)):  
    print (change(arr1[i], n), change(arr2[i], n))
```

○ 두개의 이진수로 #과 공백을 나타낸다.

문제532. 위에서 출력된 결과를 이용해서 두 지도의 숫자가 둘다 0 이면 공백을 출력하고 아니면 #을 출력하시오.

```
def change(num, n):  
    return bin(num)[2:].rjust(n, '0')
```

```
n = 5
```



```

arr1 = [9, 20, 28, 18, 11]
arr2 = [30, 1, 21, 17, 28]
result = []

for i in range(len(arr1)):
    a1 = change(arr1[i], n)
    a2 = change(arr2[i], n)
    f = ""
    for k in range(n):
        if a1[k] == '0' and a2[k] == '0':
            fi = ' '
            f = f + fi
        else:
            fi = '#'
            f = f + fi
    result.append(f)

print (result)

```

문제533. 위의 코드를 함수로 생성해서 아래와 같이 수행되게 하시오.

```

n = 6
arr1 = [46, 33, 33, 22, 31, 50]
arr2 = [27, 56, 19, 14, 14, 10]

secretmap(arr1, arr2, n)

# ['#####', '### #', '## ##', '####', '#####', '### #']

arr1 = [46, 33, 33, 22, 31, 50]
arr2 = [27, 56, 19, 14, 14, 10]
n = 6
result = []

def secretmap(arr1, arr2, n):
    def change(num, n):
        return bin(num)[2:].rjust(n, '0')

    for i in range(len(arr1)):
        a1 = change(arr1[i], n)
        a2 = change(arr2[i], n)
        f = ""
        for k in range(n):
            if a1[k] == '0' and a2[k] == '0':
                fi = ' '
                f = f + fi
            else:
                fi = '#'
                f = f + fi
        result.append(f)

    return result

print (secretmap(arr1, arr2, n))

```

3. 번호 순서대로 기술한 해결방법을 코딩으로 작성한다.

160. 통계 마지막 문제 (도박사 이야기)

<http://cafe.daum.net/oracleoracle/SfZF/1280>

문제534. 동전 하나를 10000 번 던져서 뒷면이 나오는 횟수를 출력하시오.

```
import random

cnt = 0

for i in range(10000):
    result = random.randint(0,1) # 숫자 0(뒷면)과 1(앞면)중 하나를 랜덤으로 출력
    if result == 0:
        cnt += 1

print (cnt)
```

문제535. 동전을 8번 던져서 뒷면이 나오는 횟수를 출력하시오.

```
import random

cnt = 0

for i in range(8):
    result = random.randint(0,1)
    if result == 0:
        cnt += 1

print (cnt)
```

문제536. 동전하나를 8번 던졌을때 그 중 뒷면이 5번 나오는 횟수를 출력하시오.

(동전 하나를 8번 던지는 실험을 10000번 하시오.)

```
import random

count = 0

for j in range(10000):
    cnt = 0
    for i in range(8):
        result = random.randint(0,1)
```

```
    if result == 0:
        cnt += 1
    if cnt == 5:
        count += 1

print (count)
```

문제537. 아래의 결과를 출력하시오.

```
print (a)
print (b)

[0, 1, 2, 3, 4, 5, 6, 7, 8]
[0, 0, 0, 0, 0, 0, 0, 0, 0]

a = list(range(9))
b = [0] * 9

print (a)
print (b)
```

문제538. 동전을 8번 던졌을 때 뒷면이 0번에서 8 번 나오는 횟수가 아래와 같이 출력되게 하시오.

```
[43, 318, 1044, 2263, 2796, 2141, 1092, 280, 33]
```

```
import random

b = []

for k in range(9):
    count = 0
    for j in range(10000):
        cnt = 0
        for i in range(8):
            result = random.randint (0,1)
            if result == 0:
                cnt += 1
        if cnt == k:
            count += 1
    b.append(count)

print (b)
```

문제539. 위의 횟수를 가지고 아래와 같이 확률이 출력되게 하시오.

```
import random

b = []

for k in range(9):
```

```

count = 0
for j in range(10000):
    cnt = 0
    for i in range(8):
        result = random.randint (0,1)
        if result == 0:
            cnt += 1
    if cnt == k:
        count += 1
b.append(count/10000)

print (b)

```

문제540. coin_cnt를 x 축으로 하고 확률을 y 축으로 해서 막대그래프를 그리시오.

```
coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
```

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

```
coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
b = []
```

```

for k in range(9):
    count = 0
    for j in range(10000):
        cnt = 0
        for i in range(8):
            result = random.randint (0,1)
            if result == 0:
                cnt += 1
        if cnt == k:
            count += 1
    b.append(count/10000)

```

```

plt.bar(coin_cnt, b, tick_label = coin_cnt, align = 'center', color = 'red')
plt.show()

```

문제541. coin_cnt 를 입력값으로 해서 coin_cnt의 요소들의 평균값과 표준편차를 출력하시오.

```
import numpy as np
```

```

coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)

```

```

print (c_m)
print (c_s)

```

문제542. 위에서 구한 평균값과 표준편차를 이용해서 coin_cnt의 요소 값에 대한 확률밀도함수 값을 출력하시오.

(scipy.stats 의 norm 사용)

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

coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)
result = norm.pdf(coin_cnt, c_m, c_s)

print(result)
```

문제543. coin_cnt를 x 축으로 두고 위의 확률밀도 함수값을 y 축으로 두어서 확률 밀도 함수 그래프를 그리시오.

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)
result = norm.pdf(coin_cnt, c_m, c_s)

plt.plot(coin_cnt, result, color = 'black')
plt.show()
```

문제544. 위의 그래프에서 신뢰구간 68%에 해당하는 부분만 녹색으로 색칠하시오.

신뢰구간 범위

1. 68% : 평균 ± 1 * 표준편차
2. 95% : 평균 ± 1.96 * 표준편차
3. 99% : 평균 ± 2.58 * 표준편차

```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

coin_cnt = [0, 1, 2, 3, 4, 5, 6, 7, 8]
c_m = np.mean(coin_cnt)
c_s = np.std(coin_cnt)
result = norm.pdf(coin_cnt, c_m, c_s)

plt.plot(coin_cnt, result, color = 'black')
```

```
plt.fill_between(coin_cnt, result, where = (coin_cnt < c_m + c_s) & (coin_cnt >= c_m - c_s), color = 'green')
plt.show()
```

문제545. 위의 코드들을 이용하여 아래와 같이 출력되게 하시오.

동전을 8번 던졌을 때 동전이 뒷면이 나오는 횟수가 신뢰구간 68% 안에 드는지 아닌지를 출력하는 함수를 생성하시오

```
print ( coin_hypo(동전 뒷면의 횟수) )
```

```
print ( coin_hypo(4) )
```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 4번이 나올 확률은 신뢰구간 68% 안에 있습니다.

```
print ( coin_hypo(8) )
```

동전을 8번 던졌을 때 뒷면이 나오는 횟수가 8번이 나올 확률은 신뢰구간 68% 안에 없습니다.

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

```
def coin_avg_std(num): # 동전을 던진 횟수에 따른 평균과 표준편차를 생성
```

```
    result=[]
    for i in range(num):
        cnt = 0
        for t in range(8):
            cnt += (random.randint(0,1))
        result.append(cnt)
    c_m= np.mean(result)
    c_s = np.std(result)
```

```
    return c_m,c_s
```

```
def coin_hypo(num):
```

```
    c_m,c_s = coin_avg_std(10000) # 동전을 던지는 횟수를 선택
```

```
    if c_m-c_s<=num<=c_m+c_s:
```

```
        return f'동전을 8번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간 68%안에 있습니다.'
```

```
    else:
```

```
        return f'동전을 8번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간 68%안에 없습니다.'
```

```
print(coin_hypo(4))
```

```
print(coin_hypo(8))
```

파이썬 시험 유형

2020년 12월 29일 화요일 오후 4:22

파이썬 시험 유형 (5문제)

1. 두 숫자를 입력받아 사각형을 출력하시오.

가로의 길이를 입력하시오. 5

세로의 길이를 입력하시오. 4

```
★★★★★  
★★★★★  
★★★★★  
★★★★★
```

```
num1 = int(input('가로의 길이를 입력하시오.'))  
num2 = int(input('세로의 길이를 입력하시오.'))
```

```
for i in range(1, num2 + 1):  
    print (num1 * '★')
```

2. 두 숫자를 입력받아 최대 공약수를 출력하시오.

첫번째 숫자를 입력하시오. 16

두번째 숫자를 입력하시오. 24

8

```
n1 = int(input('첫번째 숫자를 입력하시오.'))  
n2 = int(input('두번째 숫자를 입력하시오.'))
```

```
for i in range(max(n1, n2), 1, -1):  
    if n1%i == 0 and n2%i == 0:  
        break
```

```
print (i)
```

함수

```
def gcd(a, b):  
    for i in range (max(a,b), 1, -1):  
        if a%i == 0 and b%i == 0:  
            return i
```

```
print (gcd(12,24))
```

3. 버블정렬

```
a = [5, 4, 3, 2, 1, 8, 7, 10]
```

```
def bubble_sort(a):  
    for j in range(len(a)):  
        cnt = 0  
        for i in range(len(a)-1):  
            if a[i] > a[i+1]:  
                temp = a[i+1]  
                a[i+1] = a[i]  
                a[i] = temp  
                cnt = 1  
        if cnt == 0:  
            break  
    return a
```

```
print (bubble_sort(a))
```

4. 이진탐색 문제

```
import numpy as np
```

```
a = [1, 7, 11, 12, 14, 23, 33, 47, 51, 64, 67, 77, 139, 672, 871]
```

```
num = int(input('a 리스트에서 검색할 숫자를 입력하시오.'))
```

```
cnt = 0
```

```
while True:  
    cnt += 1  
    a_m = np.median(a)  
    if a_m > num:  
        del (a[a.index(a_m):])  
    elif a_m < num:  
        del (a[:a.index(a_m)+1])  
    elif a_m == num:  
        print ('%d은 이진탐색 %d 번만에 검색되었습니다.'%(num,cnt))  
        break  
    else:  
        print ('%d은 a 리스트에 없습니다.'%(num))  
        break
```

5. 신뢰구간 (동전 문제를 응용)

<http://cafe.daum.net/oracleoracle/SfZF/1324>

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



```

def coin_avg_std(num):
    result=[]
    for i in range(num):
        cnt = 0
        for t in range(10):
            cnt += (random.randint(0,1))
        result.append(cnt)
    c_m = np.mean(result)
    c_s = np.std(result)
    return c_m, c_s

def coin_hypo(num):
    c_m,c_s = coin_avg_std(10000)
    if c_m-c_s*1.96<=num<=c_m+c_s*1.96:
        return f'동전을 10번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간
95%안에 있습니다.'
    else:
        return f'동전을 10번 던졌을 때 뒷면이 나오는 횟수가 {num}번이 나올 확률은 신뢰구간
95%안에 없습니다.'

print(coin_hypo(4))
print(coin_hypo(10))

```