

파이썬 워크샵

실습: 기본 구문 및 수학메소드

- $y=ax^2+bx+c$ 으로 표현되는 2차방정식이 있을 때 $y=0$ 이 되는 x 를 구하는 근의 공식은 다음과 같다.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- 근의 공식을 이용하여 파이썬으로 $2x^2 + x - 3 = 0$ 의 해를 구하라.

실습: 시퀀스 슬라이싱

- 아래와 같은 홍길동씨의 주민번호를 대쉬 기호 전/후로 쪼개보자.

```
In [24]: # 홍길동씨 주민번호  
pin = "881120-1068234"
```

실습: 리스트 정렬

- 아래의 a 를 $[5, 4, 3, 2, 1]$ 과 같이 출력하라.
- 힌트: a 객체에서 사용 가능한 함수를 사용할 것.

```
In [55]: a = [1, 3, 5, 4, 2]
         a
         [1, 3, 5, 4, 2]
```

실습: 거리계산

- 리스트를 이용하여 점 사이의 거리를 계산 해 보자.
- 직교 좌표 위에서 A는 (1, 1), B는 (3, 2), C는 (5,7)일 때 X(2, 3)와 A/B/C와의 거리를 각각 구하여라.
- 위 코드블럭을 이용하여 distMeasure 함수를 정의하라.

실습: 사전

- 다음의 주어진 문자열 `s`에서 알파벳 별 빈도를 사전형인 `d`에 등록하는 파이썬 코드블럭을 작성하시오.

```
s = 'life is short, so python is easy.'
```

```
punct = ',.'
```

```
d = {}
```

결과 print d (순서는 상관 없음)

```
{'a': 1, 'e': 2, 'f': 1, 'i': 3, 'h': 2, 'l': 1, 'o': 3, 'n': 1, 'p': 1, 's': 5, 'r': 1, 't': 2, 'y': 2}
```

- 위의 코드블럭을 응용하여 다수결을 구현한 `majorityCnt` 함수를 작성하시오.

실습: 텍스트 파일을 열어 WordCount 결과 반환

- WordCount 함수에서
ihaveadream.txt 파일을 열어 전체 내용 중
단어를 키로, 빈도를 값으로 매핑한 사전형 자료를 반환하라.
- majorityCnt 함수에서
사전형 자료를 인수로 받아 값을 기준으로 내림차순으로 정렬하여 값이 가장 큰 키를 반환하라.
- WordCount 함수의 결과를 result.txt로 저장하라.
- 함수들을 myUtil.py 파일 안에 넣어 모듈의 형태로 재사용할 수 있도록 하라.

실습: 리스트와 반복문

- 다음의 주어진 리스트 l1의 요소 중에서 l2의 요소와 값이 같은 경우 삭제하는 파이썬 코드블럭을 작성하시오.

```
l1 = ['a', 'b', 'c', 'd', 'a', 'b', 'a', 'b']
```

```
l2 = ['b', 'a']
```

결과 l1

```
['c', 'd']
```

실습: 파일 다루기

- 'in.txt' 파일에 있는 각 컬럼의 데이터를 읽어들이어서 다른 형식으로 저장하고 각 컬럼 데이터 별로 따로 저장하라. (아래는 in.txt의 내용)

```
some comment line
1.5
    tmp-measurements    tmp-model1    tmp-model2
        0.0             0.1             1.0
        0.1             0.1             0.188
        0.2             0.2             0.2
```

이상형 월드컵

- idealperson.py를 수정하여 아래의 과제를 수행하라.
- 8명의 후보에 대하여 프로필 정보를 사전형 자료에 준비한다.
- 8명의 후보 목록을 이용하여 토너먼트 방식으로 이상형 선발전을 수행하여 최종 이상형을 정한다.

실습: ndarray 생성 및 matplotlib plot 연습

- numpy와 matplotlib을 이용하여 $y=3\sin(x)$ 의 산점도 그래프를 그려보자.

- python_score.csv 파일을 읽어들이어서 numpy ndarray로 만들고 np.dot() 메소드를 이용해 개인별 점수를 구하라.

```
import numpy as np
score = np.loadtxt('python_score.csv', delimiter=',', dtype='float64')
```

- 데이터 설명:
 - 0~9 열이 개인별 객관식 답안 및 주관식 점수 데이터셋임.
 - 10 열이 정답
 - 0~19 행이 객관식 답안
 - 20~21 행이 주관식 점수
- 과제:
 - 정답 열을 색인하여 answer를 생성한다.
 - 객관식 답안 열을 색인하여 dset을 생성한다.
 - dset 중 정답은 True 오답인 경우는 False로 표시한 bool_answer를 생성한다.
 - 객관식 답안은 정답과 비교하여 정답이면 1, 오답이면 0으로 표시한다. 주관식 답안은 점수가 포함된 scoreset을 생성한다.
 - 개인 별 객관식 배점은 4점이고 주관식은 점수가 기입되어 있다. 그러므로 문제별 배점을 points에 입력 후 np.dot()을 이용하여 개인별 점수를 계산한다.

실습: ndarray 이용한 점수계산

- 아래와 같이 영미, 두현, 준이의 국어, 수학, 음악 성적이 dset으로 주어지고 과목별 국어:수학:음악 과목의 가중치 비율은 1.0 : 1.2 : 2.0일 때 총점에 따른 개인별 순위를 계산하는 코드를 작성하라.

```
-----  
dset={  
'영미': {'국어': 80, '수학': 90, '음악': 100},  
'두현': {'국어': 100, '수학': 98, '음악': 90},  
'준이': {'국어': 95, '수학': 93, '음악': 92}  
}  
-----
```

실습: kNN

- 머신러닝 알고리즘인 kNN을 파이썬으로 구현해보자.
- kNN (k Nearest Neighborhood): 데이터의 속성 간 거리를 계산하여 거리가 가까운 데이터의 분류값을 기준으로 미지의 데이터를 분류하는 방법
- 구현 절차
 1. 분류할 데이터 inX와 분류 라벨이 있는 데이터셋 dSet 사이의 거리를 계산한다.
 2. 1에서 계산한 거리를 기준으로 가장 가까운 k개에 대하여 라벨을 구한다.
 3. 2에서 선별된 kNN 중 빈도가 가장 큰 분류항목을 inX의 분류 라벨로 반환한다.

감사합니다.