# 리스트 튜플 세트 딕셔너리

# 연습문제1

- **친구들의 시험 점수를 저장한 다음 검색하는 기능을 구현해봅시다.**

-입력하려는 학생의 숫자를 입력받습니다.
-입력받은 숫자만큼 학생의 이름과 시험점수를 입력받습니다. 이때 학생 이름과 시험 점수는 값이 변경되지 않으니  반드시 튜플타입으로 저장합니다. 그리고 전체 학생의 정보는 리스트 타입에 저장합니다.
-입력이 모두 끝나면 검색하고자 하는 학생의 이름을 입력합니다.
-만약, 저장한 데이터에 존재하지 않는 학생이면 다시 시도합니다.
-존재하는 학생이면 점수를 출력한 뒤 프로그램을 종료합니다.
-예상 출력 결과는 다음과 같습니다.

# 연습문제1

```
입력하는 학생수가 총 몇명인가요? : 3
학생의 이름과 시험 점수를 차례대로 입력하세요!
1 번째 학생 =====
* 이름: 안나
* 점수: 95
2 번째 학생 =====
* 이름: 신후
* 점수: 92
3 번째 학생 =====
* 이름: 지희
* 점수: 88
어떤 학생의 점수가 궁금하신가요? 이름을 입력하세요. : 인석
찾는 학생( 인석 )의 점수가 존재하지 않습니다.
어떤 학생의 점수가 궁금하신가요? 이름을 입력하세요. : 안나
안나 학생의 점수 :  95
프로그램을 종료합니다.
```

# 연습문제 2

- **선배들이 신입생들의 취미 생활 종류와 빈도수를 파악하여 교내 동아리 설명회를 열고자 합니다.**

-입력하려는 학생의 숫자를 입력받는다.
-입력받은 숫자만큼 학생의 이름과 취미 생활을 입력받는다.
-이때 학생 이름과 취미 생활은 값이 변경되지 않으니 반드시 튜플타입으로 저장한다.
-그리고 전체 학생을 위한 정보는 리스트 타입에 저장한다.
-전체 학생들의 리스트를 출력한다.
-전체학생들의 취미항목을 중복없이 출력한다.
-입력 및 출력은 다음을 참고한다.

# 연습문제2

```
입력하는 학생수가 총 몇명인가요? : 6
학생의 이름과 취미를 차례대로 입력하세요!
1 번째 학생 =====
* 이름 : 안나
* 취미 : 우쿨렐레
2 번째 학생 =====
* 이름 : 건아
* 취미 : 피아노
3 번째 학생 =====
* 이름 : 혜안
* 취미 : 우클렐레
4 번째 학생 =====
* 이름 : 신후
* 취미 : 바둑
5 번째 학생 =====
* 이름 : 이안
* 취미 : 피아노
6 번째 학생 =====
* 이름 : 예쭌
* 취미 : 우쿨렐레

 == 전체 학생 리스트 정보 :  ['안나', '건아', '혜안', '신후', '이안', '예쭌']

 == 전체 취미 세트    정보 :  {'우클렐레', '바둑', '피아노', '우쿨렐레'}
```

# 연습문제 3

- **추석날 온 가족이 모여 1:1 씨름을 하기로 했습니다. 아이들은 총 4명이고 이름은 '안나', '신후', '유나', '원준' 입니다. 4명의 아이들이 각각 1:1 로 경기를 한다면 총 12번의 시합이 이루어지게 됩니다. (한 쌍이 자리를 바꾸어 각각 두번의 경기를 합니다.) 이 12명의 대진표를 출력하는 프로그램을 작성하시오.**

-아이들을 위한 리스트를 생성한다
-대진표를 위한 빈 리스트를 생성하고 반복문과 조건문을 중첩으로 작성하시오.
-위에서 생성한 리스트의 값을 출력하시오.

-위 소스코드에서 사용한 중첩블럭문 대신 리스트 생성시 반복문과 조건문을 동시에 작성하여 한줄로 된 새로운 리스트를 만드시오.
-새로운 리스트를 출력하시오
-다음을 참고하시오.

# 연습문제3

```
{('안나', '신후'), ('신후', '안나'), ('신후', '원준'), ('안나', '원준'), ('원준', '안나'), ('유나', '안나'),
('신후', '유나'), ('원준', '유나'), ('안나', '유나'), ('유나', '원준'), ('유나', '신후'), ('원준', '신후')}
[('안나', '신후'), ('안나', '유나'), ('안나', '원준'), ('신후', '안나'), ('신후', '유나'), ('신후', '원준'),
('유나', '안나'), ('유나', '신후'), ('유나', '원준'), ('원준', '안나'), ('원준', '신후'), ('원준', '유나')]
```

# 연습문제 4

- **달리기 시합에 출전하는 선수들이 시합 결과를 출력하는 프로그램을 작성합니다.**

-전체 선수 숫자를 입력받습니다.
-시합결과를 저장하기 위한 리스트를 생성한 후, 선수가 결승선을 통과할 때마다 선수의 이름을 입력받아 리스트에 추가합니다.
-모든 선수가 결승선을 통과하면 시합결과를 등수와 함께 출력합니다.
-출력 결과는 다음과 같습니다.

# 연습문제 4

```
지금부터 달리기 시합을 시작하겠습니다.

달리기 선수가 몇 명인가요? 3

지금 들어온 선수 이름을 입력하세요 : 안나

지금 들어온 선수 이름을 입력하세요 : 혜안

지금 들어온 선수 이름을 입력하세요 : 건아

달리기 시합 결과!!

1 등 안나
2 등 혜안
3 등 건아

수고 많았습니다, 여러분!!
```

# 연습문제 5

- 다양한 데이터 구조 활용에 초점을 맞추어 작성.
- 아래와 같은 읽은 책 목록을 책마다 책 제목, 출판 연도, 출판사, 페이지수, 추천 유무에 대한 정보를 저장한다.

| 제목 | 출판 연도 | 출판사 | 페이지 수 | 추천 유무 |
|---|---|---|---|---|
| 개발자의 코드 | 2013.02.28 | A | 200 | X |
| 클린코드 | 2010.03.04 | B | 584 | O |
| 빅데이터 마케팅 | 2014.07.02 | A | 296 | O |
| 구글드 | 2010.02.10 | B | 526 | X |
| 강의력 | 2013.12.12 | A | 248 | O |

# 연습문제 5

1. 데이터 담기
-여러 책을 담을 수 있는 리스트인 변수 books를 선언 및 초기화 합니다.
-books의 각 항목에 한권의 책을 담기 위한 딕셔너리 데이터를 추가합니다.
-딕셔너리 데이터는 변수 선언 없이 리스트에 바로 추가합니다.
-딕셔너리 데이터에는 4개의 키와 값의 쌍이 존재합니다. 키는 표의 컬럼명이고 값은 각 책에 해당하는 컬럼의 값이 됩니다.
-제목, 출판 연도, 출판사의 값은 문자열, 페이지수의 값은 숫자 타입, 추천 유무는 논리 타입으로 표기 합니다.

2. 여러 데이터 만들어보기
*아래조건에 맞는 변수를 선업합니다.
-내가 읽은 책중 250 쪽이 넘는 책의 제목으로 이루어진 리스트 타입 변수 many_page를 만듭니다.
-내가 읽은 책 중 추천하고 싶은 책의 제목으로 이루어진 리스트 타입 변수 recommands를 만듭니다.
-내가 읽은 책의 전체 페이지수를 담는 숫자 타입 변수 all_pages를 만듭니다.
-내가 읽은 책의 출판사를 위한 딕셔너리 타입 변수 pub_company를 만듭니다.
*책을 한권씩 꺼내기 위한 for문을 선언합니다.
*위 조건에서 선언한 변수의 값을 채우기 위한 구문들을 for 문의 블록문에 모두 기술합니다.

# 연습문제 5

3. 소스 코드 단순화 하기
앞에서 작성한 소스코드를 한줄로 조합하여 표현하는 연습을 해봅니다. 아래 조건에 만족하는 리스트를 만들어보세요. 변수 선언, if문을 모두 한 줄로 표현해보기 바랍니다.
-내가 읽은 책 중 250 페이지가 넘는 책의 제목으로 이루어진 리스트 타입 변수 many_page를 만듭니다.

```
전체 책 목록
{'제목': '개발자의 코드', '출판연도': '2013.02.28', '출판사': 'A', '쪽수': 200, '추천유무': False}
{'제목': '클린 코드', '출판연도': '2013.03.04', '출판사': 'B', '쪽수': 584, '추천유무': True}
{'제목': '빅데이터 마케팅', '출판연도': '2014.07.02', '출판사': 'A', '쪽수': 296, '추천유무': True}
{'제목': '구글드', '출판연도': '2010.02.10', '출판사': 'B', '쪽수': 526, '추천유무': False}
{'제목': '강의력', '출판연도': '2013.12.12', '출판사': 'A', '쪽수': 248, '추천유무': True}


쪽수가 250 쪽 넘는 책 리스트: ['클린 코드', '빅데이터 마케팅', '구글드']
내가 추천하는 책 리스트: ['클린 코드', '빅데이터 마케팅', '강의력']
내가 읽은 책 전체 쪽수: 1854
내가 읽은 책의 출판사 목록: {'A', 'B'}


한 줄로 만든 쪽수가 250 쪽 넘는 책 리스트: ['클린 코드', '빅데이터 마케팅', '구글드']
```