

27장

객체 사이의 맵인 사전

27장 객체 사이의 맵인 사전

27.1 사전, 키, 값 만들기

27.2 사전에 키-값 쌍 추가하기

27.3 사전에서 키-값 쌍 제거하기

27.4 사전의 키와 값을 모두 가져오기

27.5 사전을 사용해야 하는 경우는?

27.6 요약



27 객체 사이의 맵인 사전

- » 일상생활에서 데이터의 쌍을 처리하는 경우 – 단어와 단어 뜻, 사람과 전화번호, 영화와 평점, 노래와 가수 등
- » 식료품 예제를 사전에 적용한 예.
- » 리스트는 식료품 항목을 0, 1, 2, ... 등의 숫자와 순서대로 연결(mapping, 매핑)한다고 생각할 수 있다. 사전은 각 품목과 수량을 서로 매핑한다.

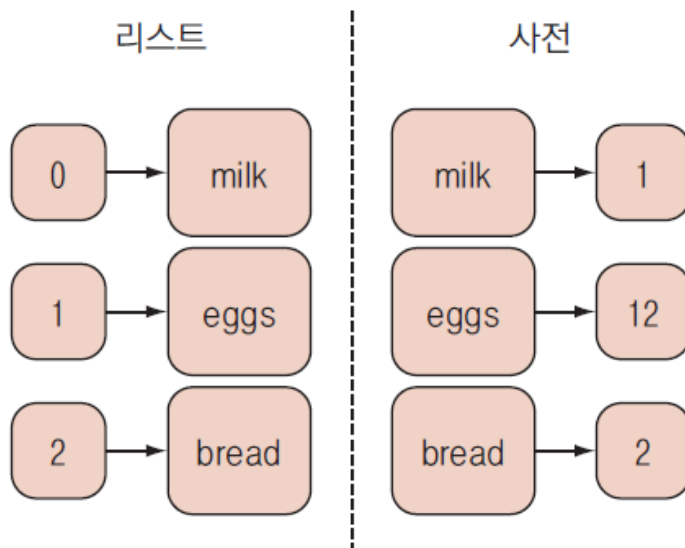


그림 27-1
리스트는 첫 번째 원소를 0, 두 번째 원소를 1, 세 번째 원소를 2와 같이 순서대로 연관시킨다. 사전은 위치가 없다. 대신 한 객체를 다른 객체와 연관시킨다. 여기서는 식료품 품목을 그 수량과 연관시켰다

27.1 사전, 키, 값 만들기



27.1 사전, 키, 값 만들기

» 많은 프로그래밍 언어가 객체를 다른 객체와 연관시키고, 한 객체를 통해 그와 연관된 객체를 검색하는 방법을 제공한다. 파이썬에서는 그런 객체를 사전(dictionary)이라고 부르며 dict이라는 객체 타입이 사전을 표현한다.

» 사전을 사용하면 한 객체를 사용해 다른 객체를 쉽게 찾을 수 있다.

- 프로그래밍에서는 검색에 사용하는 객체(영한 사전의 경우 영어 단어)를 키(key)라고 부르고, 검색 결과가 반환하는 값(영한 사전의 경우 우리말로 풀어 쓴 단어 의미)을 값(value)이라 부른다.
- 프로그래밍에서는 사전에 들어 있는 데이터를 원소라 부르며, 각 원소는 키-값 쌍이다. 한 객체(키)를 사용하면 다른 객체(값)를 찾을 수 있다.

```
grocery = {}
```

```
grocery = { "milk" : 1, "eggs" : 12, "bread" : 2 }
```

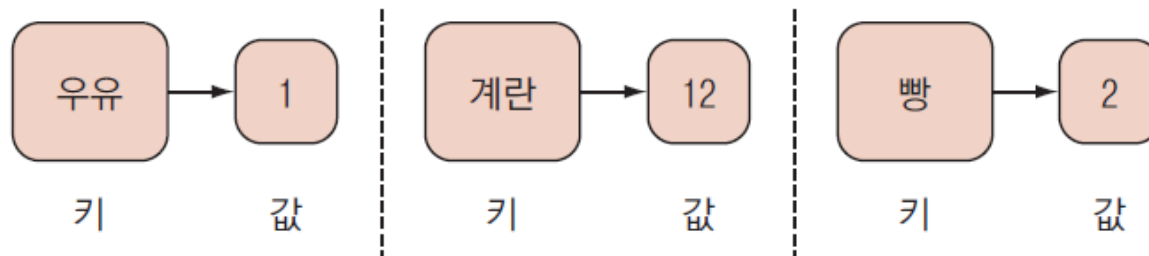


그림 26-3

세 항목으로 초기화된 사전. 각 항목은 콤마로 구분되며, 각 항목에서 키는 콜론의 왼쪽, 값은 콜론의 오른쪽에 있다



27.1 사전, 키, 값 만들기

» 빈 파이썬 사전을 만들 때는 중괄호를 쓴다.

- 이 명령은 내부에 연관된 항목(entry)이 들어 있지 않은 빈 사전을 만들고, 그 사전 객체를 grocery라는 변수에 대입한다.

» 항목이 들어 있는 사전을 만들 수도 있다. 식료품 목록을 예로 들어 식료품 품목과 수량을 연결하면 다음과 같다.

- grocery 사전의 키는 식료품 품목을 표현하는 문자열이고, grocery 사전의 값은 수량을 표현하는 정수다.



27.1 사전, 키, 값 만들기

- » 키와 값은 모두 하나의 객체로 이뤄진다.
- » 사전의 한 항목 안에는 키가 둘 이상 있을 수 없고, 값도 둘 이상 있을 수 없다.
- » 값으로 둘 이상의 객체를 저장하고 싶다면 모든 객체를 튜플 등의 컬렉션으로 저장하면 된다. 튜플 자체도 하나의 객체이기 때문에 사전에 넣을 수 있다.
- » 셀프 체크 27.1
- » 셀프 체크 27.2

27.2 사전에 키-값 쌍 추가하기



27.2 사전에 키-값 쌍 추가하기

$$d[k] = v$$

» k라는 키와 v라는 값을 연관시켜 d 사전의 항목으로 추가한다.

```
legs = {} ---- 빈 사전
legs["사람"] = 2 ---- "사람"과 2 연관시키기
legs["고양이"] = 4 ---- "고양이"와 4 연관시키기
legs["뱀"] = 0 ---- "뱀"과 0 연관시키기
print(len(legs)) ---- 사전에 세 가지 항목이 있으므로 3 출력
legs["고양이"] = 3 ---- "고양이"와 연관된 값을 3으로 바꾸기
print(len(legs)) ---- "고양이"와 연관된 값만 변경했으므로 3을 출력
print(legs) ---- {'사람': 2, '고양이': 3, '뱀': 0} 출력
```

코드 27-1

키-값 쌍 사전에 추가하기



27.2.1 키 값 제약에 대한 부연 설명

» 사전에는 키가 같은 항목이 여러 개 들어 있을 수 없다.

- 만약 여러 개가 있다면 어떤 키에 해당하는 값을 출력해야 할지 파이썬이 분간할 수 없을 것이다.
- 이런 상황을 방지하기 위해 파이썬에서는 사전의 모든 키가 유일성을 만족하도록 보장한다.

» 파이썬은 키가 주어지면 그 키를 저장할 위치를 특별한 공식을 사용하여 결정한다.

- 이때 사용하는 공식을 해시 함수(hash function)(긴데이터를 더 짧은 데이터로 줄여주는 함수)라고 한다.
- 사전은 키를 찾을 때 검색하려는 키를 모든 키와 비교하는 대신, 검색하려는 키의 해시 값을 사용해 빠르게 해당 항목이 들어 있는 위치를 찾아낸다.
- 항목을 제대로 찾으려면 키 객체에 대해 이 해시 함수를 적용한 결과가 항상 같아야 한다.
- 그렇지 않으면 매번 해시 함수를 호출할 때마다 해시 값이 달라져서 항목을 엉뚱한 위치에서 찾게 되고, 결국 원하는 값을 찾지 못한다.

» 파이썬이 제공하는 기본 불변 객체는 해시 함수를 제공한다.

- 반면 변경 가능한 컬렉션(리스트, 사전 등) 객체는 해시 함수를 제공하지 않는다.
- 따라서 (현재 여러분 수준에서는) 사전의 키로는 불변 객체를 사용해야 한다고 생각하는 것이 편하다. 참고로 나중에 클래스에 대해 배우고 나면 사전의 키로 사용할 수 있는 다른 타입의 값을 만들어낼 수 있다.

27.3 사전에서 키-값 쌍 제거하기



27.3 사전에서 키-값 쌍 제거하기

» 리스트와 마찬가지로 pop() 연산을 사용해 사전에서 항목을 제거할 수 있다.

- d.pop(k)는 k를 키로하는 키-값 항목을 사전 d에서 없앤다.
- 이 연산도 (리스트의 pop과 마찬가지로) 사전에서 제거한(k와 연관되어 있던) 값을 돌려준다.
- 다음 코드에서 pop 연산을 수행한 뒤에는 household 사전에 {"사람": 4, "고양이": 2, "개": 1}이 남아있을 것이다.

```
household = { "사람": 4, "고양이": 2, "개": 1, "금붕어": 2 } ---- 사전을 채움
removed = household.pop("금붕어") ----- 키가 "금붕어"인 항목을 찾아서 제거하되,
print(removed) ----- 제거한 항목의 값을 출력                      "금붕어"라는 키와 연관됐던 값을 removed에 저장함
```

코드 27-2

사전에서 키-값 쌍 제거하기

» 셀프 체크 27.4

27.4 사전의 키와 값을 모두 가져오기



27.4 사전의 키와 값을 모두 가져오기

- » 노래와 평점이 songs라는 사전에 저장되어 있을 때
- » `songs.keys()`를 하면 그 안의 모든 키를 얻을 수 있다.

```
songs = {"believe": 3, "roar": 5, "let it be": 4}
print(songs.keys())
```

- » 얻은 키에 대해 for 루프를 돌 수도 있다.

```
for one_song in songs.keys():
```

- » 반환 받은 키를 리스트로 변환할 수도 있다.

```
all_songs = list(songs.keys())
```

- » `songs.values()`는 songs 사전에 들어 있는 모든 값을 돌려준다.



27.4 사전의 키와 값을 모두 가져오기

Name	Quiz 1 Grade	Quiz 2 Grade
Chris	100	70
Angela	90	100
Bruce	80	40
Stacey	70	70

코드 27-3

사전을 사용해 학생 성적 처리하기

```
grades = {}
grades["프랭크"] = [100, 70]
grades["토비"] = [90, 100]
grades["도남"] = [80, 40]
grades["제이스"] = [70, 70]

for student in grades.keys():
    print(student)

for quizzes in grades.values():
    print(sum(quizzes)/2)

for student in grades.keys():
    scores = grades[student]
    grades[student].append(sum(scores)/2)

print(grades)
```

문자열과 두 퀴즈 성적을 연관 짓는 사전을 설정함

키에 대해 루프를 돌며 출력

값에 대해 루프를 돌며 평균을 출력

모든 키에 대해 루프를 돌

다음 줄에서 평균을 계산하기 위해 각 학생의 성적을 가져와서 scores 변수에 대입

원소들의 평균을 계산하고, 계산한 평균값을 리스트 뒤에 덧붙임

{'프랭크': [100, 70, 85.0], '토비': [90, 100, 95.0], '도남': [80, 40, 60.0], '제이스': [70, 70, 70.0]}을 출력

27.4 사전의 키와 값을 모두 가져오기



» 셀프 체크 27.5



27.4.1 사전에 저장된 키-값 쌍의 순서

- » 파이썬 3.5까지(그리고 파이썬 3.6 일부 구현)는 키나 값을 물어볼 때 각각이 어떤 순서로 반환될지를 보장하지 않았다.
- » 파이썬 3.7부터는 정확히 사전에 추가한 순서대로 키를 돌려주는 것을 보장하므로 이 성질을 활용해도 좋다.

27.5 사전을 사용해야 하는 경우는?



27.5.1 빈도수를 사전으로 유지하기

» 사전의 가장 일반적인 용례 중 하나는 객체들의 빈도를 세는 것이다.

- 스크래블(Scrabble) 단어 게임에서 사전을 사용해 알파벳 글자들의 개수를 관리할 수 있다.

» 코드 27-4는 단어와 그 단어가 노래에 나온 횟수를 연결해주는 빈도 사전을 보여준다

```
lyrics = "Happy birthday to you Happy birthday to you Happy birthday dear Frank Happy
    birthday to you" ---- 노래 가사 문자열
counts = {} ---- 비어있는 빈도 사전
words = lyrics.split(" ") ---- 문자열을 공백을 기준으로 나눠서 단어 리스트를 만들
for w in words: ---- 앞 줄에서 만든 리스트의 모든 단어를 이터레이션함
    w = w.lower() ---- 소문자로 바꿈
    if w not in counts:
        counts[w] = 1 ---- 사전에 단어가 없으므로 단어를 사전에 키로 추가하면서 값을 1로 지정
    else:
        counts[w] += 1 ---- 사전에 이미 단어가 있으므로, 빈도수를 1 증가시킴

print(counts) ---- {'happy': 4, 'birthday': 4, 'to': 3, 'you': 3, 'dear': 1, 'frank': 1}를 출력
```



27.5.2 일반적이지 않은 사전 만들기

» 사전을 사용하면 한 객체의 값을 검색해 다른 객체를 찾을 수 있다.

- 예 - 일반적인 이름을 함수와 연결하는 것이다. 문자열과 함수 자체를 연결시키는 사전을 만들 수 있다.

```
def square(x):
    return x*x
```

정사각형 넓이를 계산하는 함수

```
def circle(r):
    return 3.14*r*r
```

원 넓이를 계산하는 함수

```
def equilateraltriangle(s):
    return (s*s)*(3**0.5)/4
```

정삼각형 넓이를 계산하는 함수

```
areas = {"sq": square, "ci": circle, "eqtri": equilateraltriangle}
```

문자열과 함수를 매핑하는 사전

```
n = 2
print(areas["sq"](n))
print(areas["ci"](n))
print(areas["eqtri"](n))
```

사전에서 "sq"라는 키가 가리키는 함수에 n을 넘겨 호출. n은 2
 사전에서 "ci"라는 키가 가리키는 함수에 n을 넘겨 호출. n은 2
 사전에서 "eqtri"라는 키가 가리키는 함수에 n을 넘겨 호출. n은 2

27.6 요약



27.6 요약

- » 사전은 변경 가능하다.
- » 사전의 키는 해시 계산을 지원하는 객체여야 한다. 변경할 수 없는 객체는 모두 해시를 제공 하므로 키로 사용할 수 있다.
- » 변경 가능하든 가능하지 않든 관계 없이 어떤 객체든 사전의 값이 될 수 있다.
- » 파이썬 3.7부터는 키가 추가된 순서대로 키 순서가 정해진다



27.6 요약

» (Q27.1) 사전을 사용해 다음 과제를 수행하는 프로그램을 만들라. 노래 이름(문자열)과 평점(정수)을 연관시켜 주는 사전이 주어졌다. 평점이 5인 모든 노래 제목을 출력하라.

-> 네이버 영화에서 영화제목과 평점으로. 평점이 사용자로 입력받은 x 값 이상인 영화제목들을 출력하라.



27.6 요약

» (Q27.2) `replace`라는 이름의 함수를 만들라. 이 함수는 사전 `d`와 두 값 `v`, `e`를 매개변수로 받는다. 이 함수는 값을 반환하지 않는다. 단지 `d`에 있는 값 중 `v`를 모두 `e`로 바꾼다. 예를 들면 다음과 같다.

- `replace({1:2, 3:4, 4:2}, 2, 7)`는 `d`의 상태를 `{1: 7, 3: 4, 4: 7}`로 바꾼다.
- `replace({1:2, 3:1, 4:2}, 1, 2)`는 `d`의 상태를 `{1: 2, 3: 2, 4: 2}`로 바꾼다.



27.6 요약

» (Q27.3) invert라는 함수를 작성하라. invert는 사전 d를 매개변수로 받아서 새로운 사전(편의상 d_inv라고 하자)을 반환한다. d_inv 사전에는 사전 d의 값들이 (중복 없이) 키로 들어가며, d_inv의 키에 대응하는 값은 리스트다. 이 리스트에는 d_inv의 키에 대응했던 d의 키가 들어간다. 예를 들면 다음과 같다.

- invert({1:2, 3:4, 5:6})는 {2: [1], 4: [3], 6: [5]}를 반환한다.
- invert({1:2, 2:1, 3:3})는 {1: [2], 2: [1], 3: [3]}를 반환한다.
- invert({1:1, 3:1, 5:1})는 {1: [1, 3, 5]}를 반환한다(원본 사전에서 1이 세 번 중복됐으므로, 반환하는 사전에서는 1에 대응하는 리스트의 원소가 세 개여야 한다)