### 차례





- 01. 딕셔너리의 특징을 이해합니다.
- 02. 딕셔너리를 생성하고 항목을 추가하는 방법을 학습합니다.
- 03. 딕셔너리의 항목을 탐색하고 <u>삭제하</u>는 방법을 학습합니다.
- 04. 딕셔너리를 효과적으로 사용하는 방법을 이해합니다.
- 05. 집합의 특징을 이해합니다.

### **LAB**

- 01. 가위, 바위, 보 게임
- 02. 행성까지의 여행 시간은?
- 03. 멘델의 유전 법칙 시뮬레이션
- 04. 튜링상 수상자 데이터 분석
- 05. 파이썬으로 e-mail 보내기

# 01. 딕셔너리란?

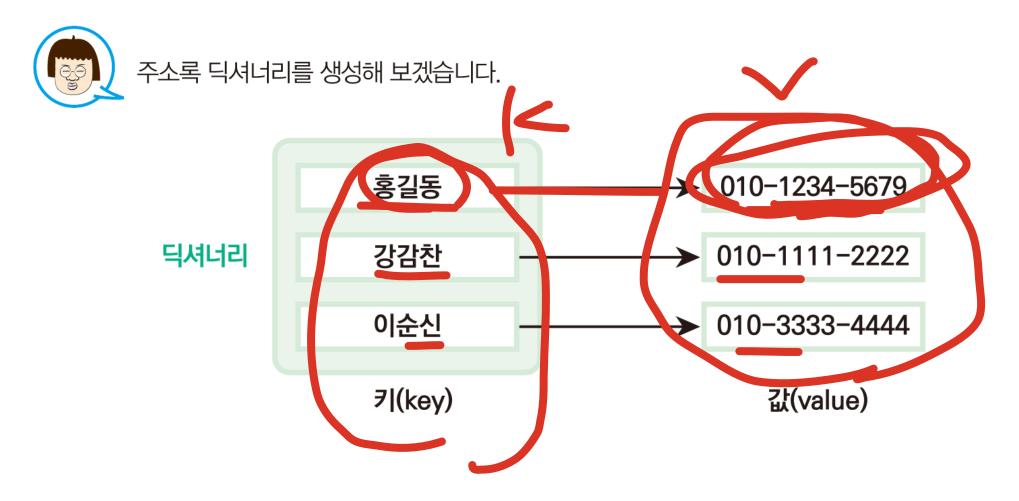


- 파이썬에서 사용되는 자료형
- 여러 개의 데이터를 한꺼번에 저장하고 처리
- 키(key)-값(value)의 짝을 이룸
- 키(key)값을 이용하여 자료에 접근



# 01. 딕셔너리란?





## 02. 딕셔너리 생성과 추가



- 파이썬의 딕셔너리는 <u>키(key)의 역할을 하는 데이터와 값(value)의</u> 역할을 하는 데이터를 하나씩 매핑 (mapping)하여 저장하는 방식
- 딕셔너리는 리스트와 마찬가지로 어떤 유형의 값도 저장할 수 있습니다.
- 정수, 문자열, 다른 리스트, 다른 딕셔너리도 항목으로 저장할 수 있습니다.



빈(empty) 딕셔너리를 생성하고 Key와 Value 쌍을 추가해 보겠습니다.

٤

실행 결과

phone\_book = { } phone\_book[<u>"홍길동</u>"] = "010-1234-5678"

print(phone\_book)

{'홍길동': '010-1234-5678'}

## 02. 딕셔너리 생성과 추가





딕셔너리를 생성하면서 동시에 초기화하는 방법도 있습니다.

#### 코드

phone\_book = {"홍길동": "010-1234-5678"} print(phone\_book)

#### 실행 결과

{'홍길동': '010-1234-5678'}



자료를 더 추가해 보겠습니다.

#### 코드

phone\_book["강감선"] = "010-1111-2222" phone\_book["이순신"] = "010-3333-4444" print(phone\_book)

### 실행 결과

('홍길동': '010-1234-5678', '강감찬': '010-1111-2222', '이순신': '010-3333-4444'}

## 02. 딕셔너리 생성과 추가





학생에 대한 정보를 딕셔너리로 저장해 보았습니다.

딕셔너리에서 사용할 키는 유일하고 변경되지 않아야 합니다. 리스트와 같이 값이 변경될 수 있는 자료형은 딕셔너리에서 키로 사용할 수 없습니다.

#### 코드

dict = {'Name': '홍길동', 'Age': 7, 'Class': '초급'} print (dict['Name']) print (dict['Age'])

#### 실행 결과

홍길동





딕셔너리에서 탐색은 다음과 같은 코드로 가능합니다.

#### 코드

phone\_book = { '홍길동':'010-1234-5678',

'강감찬' '<u>010-1111-2222'.</u>

'이순신''010-3333-4444' }

print(phone\_book["강감찬"])



010-1111-2222









딕셔너리에서는 키를 모르는 경우 <mark>keys()</mark>를 사용하여 모든 키를 볼 수 있습니다.

### 코드

phone\_book : '홍길동': '010-1234-5678', '강감찬': '010-1111-2222', '이순신': '010-3333-4444'

print(phone\_book.keys())

#### 실행 결과

dict\_keys(['홍길동', '강감찬', '이순신'])







딕셔너리 출력을 정리할 수 있습니다.

### 코드

phone\_book = '홍길동':'010-1234-5678', '강감찬':'010-1111-2222', '이순신':'010-3333-4444'

for key in sorted(phone\_book.keys()):
 print(key, phone\_book[key])

#### 실행 결과

강감찬 010-1111-2222 이순신 010-3333-4444 홍길동 010-1234-5678





딕셔너리에서 사용되는 <mark>모든 값을 알고 싶다면 values()</mark>를 사용합니다.

#### 코드

phone\_book = '홍길동': '010-1234-5678',

'강감찬' '010-1111-2222',

'이순신' '010-3333-4444'

print(phone\_book.values())

#### 실행 결과

dict\_values(['010-1234-5678', '010-1111-2222', '010-3333-4444'])

# 04. 딕셔너리 수정과 삭제





del, pop()은 딕셔너리의 지정 항목을 삭제합니다.

#### 코드

phone\_book ={ '홍길동':'010-1234-567', '강감찬':'010-1111-2222', '이순신':'010-3337-4444'} del phone\_book["홍길도"] print(phone\_book)

print(phone\_book,pop('이순신'))
print(phone\_book)

#### 실행 결과

('강감찬': '010-1111-2222', '이순신': '010-3333-4444'}

010-3333-4444

{'강감찬': '010-1111-2222'}

# 04. 딕셔너리 수정과 삭제





<mark>딕셔너리의 항목을 모두 삭제하려면 clear()를 사용</mark>하여 삭제할 수 있습니다.

#### 코드

phone\_book ={ '홍길동':'010-1234-5678',

'강감찬' '010-1111-2222',

'이순신' '010-3333-4444'}

phone\_book.clear() # 모든 항목 삭제

print(phone\_book)





## 05. 딕셔너리를 좀 더 이해시켜 줄 예제





딕셔너리를 이용하여 영한사전 프로그램을 작성해 보겠습니다.

#### 코드

english\_dict = {}
english\_dict['one'] = '하나'
english\_dict['two'] = '둘'
english\_dict['three'] = 셋'

word = input("단어를 입력하시오: ")
print (english\_dict[word])

#### 실행 결과 1

단어를 입력하시오: one 하나

#### 실행 결과 2

단어를 입력하시오: two 둘

english\_dict = dict()

### 05. 딕셔너리를 좀 더 이해시켜 줄 예제





편의점 재고 관리 프로그램을 작성해 보겠습니다.

#### 코드

items = {"커피":7, "펜":3, "종이컵":10, "우유":5, "콜라":4, "라면":11}

print("판매 전 재고", items)

sell = input("판매한 물건을 입력하세요:

if sell in items:
 items[sell] == 1
else:

print("판매 제품이 아닙니다.")

print("판매 후 재고", items)

#### 실행 결과 1

현재 재고 {'커피': 7, '펜': 3, '종이컵': 10, '우유': 5, '콜

라': 4. '라면': 11)

판매한 물건을 입력하세요: 펜

('커피': 7, '펜': 2, '종이컵': 10, '우유': 5, '콜라': 4, '라면': 11}

#### 실행 결과 2

판매 전 재고 {'커피': 7, '펜': 3, '종이컵': 10, '우유': 5, '콜라': 4, '라면': 11}

판매한 물건을 입력하세요: 햄

판매 제품이 아닙니다.

판매 후 재고 {'커피': 7, '펜': 3, '종이컵': 10, '우유': 5, '콜

라': 4, '라면': 11}

## 06. 딕셔너리와 반복문의 궁합



• 딕셔너리의 항목을 처리할 때 비슷한 작업을 반복문으로 처리하면 효과적으로 수행



반복문과 딕셔너리의 keys(), values()를 이용하여 key 또는 value를 출력해 봅시다.

```
phone_book = {'홍길동':'010-1234-5678',
'강감찬':'010-1111-2222',
'이순신':'010-3333-4444'}

for i in phone_book.keys():
  print(i)

for i in phone_book.values():
  print(i)
```

#### 실행 결과

홍길동 강감찬 이순신 010-1234-5678 010-1111-2222 010-3333-4444

# 06. 딕셔너리와 반복문의 궁합





반복문과 딕셔너리의 items()를 이용하여 key와 vlaue를 동시에 출력해 봅시다.

### 코드

phone\_book = {'홍길동':'010-1234-5678', '강감찬':'010-1111-2222',

'이순신':'010-3333-4444'}

for k, v in phone\_book.items():
print('{ }의 전화번호는 { }입니다.'.format(k, v))

#### 실행 결과

홍길동의 전화번호는 010-1234-5678입니다. 강감찬의 전화번호는 010-1111-2222입니다. 이순신의 전화번호는 010-3333-4444입니다.



# 07. 집합



- 자료의 묶음
- 집합은 중복된 데이터를 가질 수 없으며 순서가 없음

파이썬의 집합은 수학의 집합이론을 그대로 흉내 내었습니다.



집합을 생성해 봅시다.

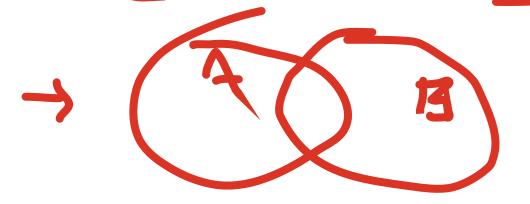
#### 코드 1

>>> s = set()

#### 코드 2

>>>  $s = \{1, 2, 5\}$ 

# 집합 생성







# 07. 집합





집합에 <mark>원소를 추가</mark>하려면 <mark>add( )</mark>를 사용합니다.

### 코드

>>> s.add(10)

>>> s.add(3)

>>> print(s)

### 실행 결과

{1, 2, 3, 5, 10}

# 07. 집합



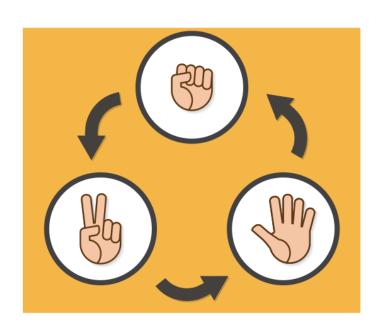


집합의 <mark>항목을 삭제하려면 discard(), remove(), clear() 함수를 사용합니다.</mark>

>>> s.discard(5) >>> print(s)	실행 결과 {1, 2, 3, 10}
<pre>&gt;&gt;&gt; s.remove(2) &gt;&gt;&gt; print(s)</pre>	{1, 3, 10}
>>> s.remove(7)	<에러 메시지 생략> s.remove(7) KeyError: 7
<pre>&gt;&gt;&gt; s.clear() &gt;&gt;&gt; print(s)</pre>	set()

# Lab. 가위, 바위, 보 게임





### 실행 결과 1

가위, 바위, 보 입력: 가위

비겼습니다.

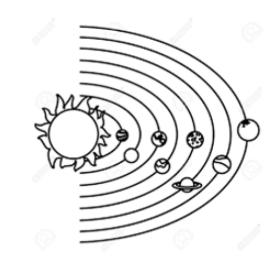
#### 실행 결과 2

가위, 바위, 보 입력: 바위 이겼습니다.

# Lab. 행성까지의 여행 시간은?



행성 이름	지구에서 행성까지의 거리
수성	91,700,000km
금성	41,400,000km
화성	78,400,000km
목성	628,700,000km
토성	1,277,400,000km
천왕성	2,750,400,000km
해왕성	4,347,400,000km



#### 실행 결과 1

행성 이름: 화성

이동 속도(km/h): 300

이동 시간: 약 261333.333333333334 시간 이동 시간: 약 29 년 10 월 3 일 21 시간

#### 실행 결과 2

행성 이름: 해왕성

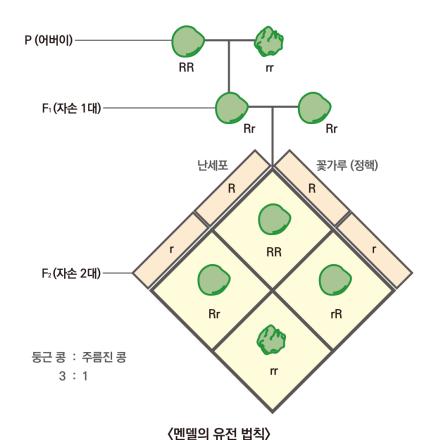
이동 속도(km/h): 700

이동 시간: 약 6210571.428571428 시간 이동 시간: 약 708 년 11 월 23 일 19 시간

## Lab. 멘델의 유전 법칙 시뮬레이션



• 부모가 둥근 완두콩(Rr)일 때 둥근 완두콩(RR, Rr, rR): 주름진 완두콩(rr)의 비율 = 3:1



#### 실행 결과

{'rR': 24, 'RR': 28, 'Rr': 24, 'rr': 24}

3.166666666666665:1



- 컴퓨터 과학의 노벨상으로 튜링상
- 역대 튜링상 수상자들의 명단과 수상년도, 국적 정리 프로그램

이름	수상년도	국적	대표업적
팀 버너스리	2016	영국	월드 와이드 웹의 하이퍼텍스트 시스템을 고안하여 개발
리처드 해밍	1968	미국	오류 검출 부호 및 오류 정정 부호
에츠허르 데이크스트라	1972	네덜란드	프로그래밍 언어 연구, 데이크스트라 알고리즘
더글러스 엥겔바트	1997	미국	마우스의 발명, 대화형 컴퓨팅
데니스 리치	1983	미국	유닉스 운영 체제 개발, C언어 개발



• 수상자 명단

### 코드

for award in awards: print(award['0|름'])

### 실행 결과

팀 버너스리 리처드 해밍 에츠허르 데이크스트라 더글러스 엥겔바트 데니스 리치



• 1990년 이전의 수상자와 연도를 출력

### 코드

for award in awards: award['수상년도'] <= 1990 : print(award['이름'], award['수상년도'])

### 실행 결과

리처드 해밍 1968 에츠허르 데이크스트라 1972 더글러스 엥겔바트 1977 데니스 리치 1983



• 수상자들의 국적 출력

### 코드

nationality = set()

for award in awards:
 nationality.add(award['국적'])
print(nationality)

### 실행 결과

{'영국', '미국', '네덜란드'}



# Lab. 파이썬으로 e-mail 보내기



- SMTP(Simple Mail Transfer Protocol) : 인터넷에서 이메일을 보내기 위해 이용되는 프로토콜
- MIME(Multipurpose Internet Mail Extensions) : 전자 우편을 위한 인터넷 표준 포맷
- euc-kr : 인코딩 표준

