

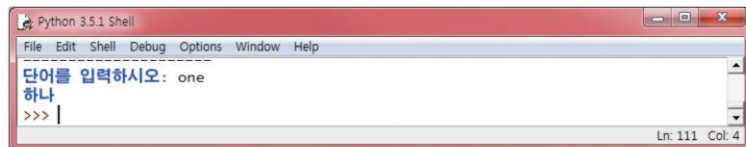
9.리스트와 딕셔너리

학습 내용

- 리스트에 대하여 학습합니다.
- 딕셔너리에 대하여 학습합니다.

이번 장에서 만들 프로그램

●영한 사전 만들기



●오률키를 그려보자. 오률키에 대한 정보를 리스트에 저장한다.



List

리스트

●어떤 경우에는 여러 개의 데이터를 하나로 묶어서 저장하는 것이 필요하다.

●리스트

● [] 기호를 사용한다.



```
heroes = [ "아이언맨", "토르", "헐크", "스칼렛 위치" ]
```

리스트

●리스트



```
numbers = [ 7, 12, 33, 777 ]
```

공백 리스트에서 추가하기

```
>>> heroes = []
>>> heroes.append("아이언맨")
>>> heroes
['아이언맨']
>>> heroes.append("닥터 스트레인지")
>>> print(heroes)
['아이언맨', '닥터 스트레인지']
```

●append() 함수

- ◆리스트에 항목을 추가한다.

점의 의미

- 파이썬에서 모든 것은 객체(object)이다.
 - 객체는 관련되는 변수와 함수를 묶은 것이다.
- 파이썬에서 리스트도 객체이다.
 - 객체 안에 있는 무엇인가를 사용하기
 - ◆객체의 이름을 쓰고 점(.)을 붙인 후에 함수의 이름을 적는다.

```
heroes.append("아이언맨")
```

2019-05-06

© Chang Seung Kim - All rights reserved

7

리스트

- 파이썬의 리스트는 어떤 유형의 데이터도 가질 수 있다.
 - 숫자, 문자열, 객체 등

```
>>> num=[1]
>>> num.append(2)
>>> num
[1, 2]
>>> list1=[1,2,3]
>>> list1.append(num)
>>> list1
[1, 2, 3, [1, 2]]
>>> list1.append("아이언맨")
>>> list1
[1, 2, 3, [1, 2], '아이언맨']
>>>
```

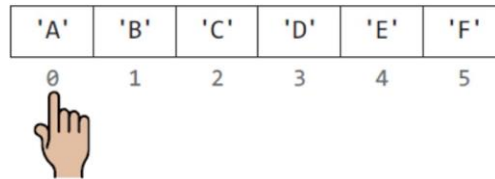
2019-05-06

© Chang Seung Kim - All rights reserved

8

리스트 항목 접근하기

```
>>> letters=['A', 'B', 'C', 'D', 'E', 'F']
>>>
>>> print(letters[0])
A
>>> print(letters[1])
B
>>> print(letters[2])
C
>>>
```



2019-05-06

© Chang Seung Kim - All rights reserved

9

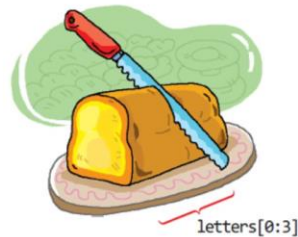
슬라이싱

● 슬라이싱(slicing)

- 리스트에서 한 번에 여러 개의 항목을 추출하는 기법이다.

```
>>> letters=['A', 'B', 'C', 'D', 'E', 'F']
>>> print(letters[0:3])
['A', 'B', 'C']
```

- ◆ [0:3] : 인덱스 0부터 시작해서 3 바로 전에 중지
- ❖ 추출항목의 갯수는 $3 - 0 = 3$ 개



2019-05-06

© Chang Seung Kim - All rights reserved

10

인덱스 생략

```
>>> print(letters[:3])
['A', 'B', 'C']
>>> print(letters[3:])
['D', 'E', 'F']
>>> print(letters[:])
['A', 'B', 'C', 'D', 'E', 'F']
```

리스트를 복사할 때
사용한다.

리스트 복사하기

```
>>> letters=['a', 'b', 'c', 'd', 'e', 'f']
>>> copy1 = letters
>>> copy1
['a', 'b', 'c', 'd', 'e', 'f']
>>> letters.append('g')
>>> letters
['a', 'b', 'c', 'd', 'e', 'f', 'g']
>>> copy1
['a', 'b', 'c', 'd', 'e', 'f', 'g']
>>> copy2 = letters[:]
>>> copy2
['a', 'b', 'c', 'd', 'e', 'f', 'g']
>>> letters.append('h')
>>> letters
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
>>> copy2
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

copy1 은 letters 와 같은 리스트이다.

copy2 은 letters 를 복사한 다른
리스트이다.

리스트 항목 변경하기

```
>>> heroes = ["아이언맨", "토르", "헐크", "스칼렛 위치"]
>>> heroes[1] = "닥터 스트레인지"
>>> print(heroes)
['아이언맨', '닥터 스트레인지', '헐크', '스칼렛 위치']
```

인덱스를 이용한다.

함수를 이용하여 추가하기

```
>>> heroes.append("스파이더맨")
>>> print(heroes)
['아이언맨', '닥터 스트레인지', '헐크', '스칼렛 위치', '스파이더맨']
>>>
>>>
>>> heroes.insert(1, "배트맨")
>>> print(heroes)
['아이언맨', '배트맨', '닥터 스트레인지', '헐크', '스칼렛 위치', '스파이더맨']
```

항목 삭제하기

```
>>> print(heroes)
['아이언맨', '배트맨', '닥터 스트레인지', '헐크', '스칼렛 위치', '스파이더맨']
>>>
>>> heroes.remove("헐크")
>>> print(heroes)
['아이언맨', '배트맨', '닥터 스트레인지', '스칼렛 위치', '스파이더맨']
```

2019-05-06

© Chang Seung Kim - All rights reserved

15

항목이 리스트 안에 있는지 체크

```
>>> print(heroes)
['아이언맨', '배트맨', '닥터 스트레인지', '스칼렛 위치', '스파이더맨']
>>>
>>> if "배트맨" in heroes:
    heroes.remove("배트맨")

>>> print(heroes)
['아이언맨', '닥터 스트레인지', '스칼렛 위치', '스파이더맨']
```

2019-05-06

© Chang Seung Kim - All rights reserved

16

del

●del

- 인덱스를 사용하여 항목을 삭제한다.

```
>>> print(heroes)
['아이언맨', '닥터 스트레인지', '스칼렛 위치', '스파이더맨']
>>> del heroes[0]
>>> print(heroes)
['닥터 스트레인지', '스칼렛 위치', '스파이더맨']
```

2019-05-06

© Chang Seung Kim - All rights reserved

17

pop()

●pop()

- 리스트에서 마지막 항목을 반환하면서 삭제한다.

```
>>> print(heroes)
['닥터 스트레인지', '스칼렛 위치', '스파이더맨']
>>>
>>> last_hero = heroes.pop()
>>> print(last_hero)
스파이더맨
>>> print(heroes)
['닥터 스트레인지', '스칼렛 위치']
```

2019-05-06

© Chang Seung Kim - All rights reserved

18

리스트 탐색하기

●index() 사용

```
>>> heroes = ["아이언맨", "토르", "헐크", "스칼렛 위치"]  
>>> print(heroes.index("헐크"))  
2
```

2019-05-06

© Chang Seung Kim - All rights reserved

19

리스트 방문하기

```
>>> print(heroes)  
['아이언맨', '토르', '헐크', '스칼렛 위치']  
>>> for hero in heroes:  
    print(hero)
```

```
아이언맨  
토르  
헐크  
스칼렛 위치
```

2019-05-06

© Chang Seung Kim - All rights reserved

20

리스트 정렬하기

```
>>> print(heroes)
['아이언맨', '토르', '헐크', '스칼렛 위치']
>>> heroes.sort()
>>> print(heroes)
['스칼렛 위치', '아이언맨', '토르', '헐크']
```



2019-05-06

© Chang Seung Kim - All rights reserved

21

문장을 단어 리스트로 만들기

```
>>> m = '파이썬은 정말 쉬운 언어다'
>>> mlist = m.split()
>>> mlist
['파이썬은', '정말', '쉬운', '언어다']
>>> mlist.sort(key=len)
>>> mlist
['정말', '쉬운', '언어다', '파이썬은']
```

● `sort(*, key=None, reverse=False)`

2019-05-06

© Chang Seung Kim - All rights reserved

22

알아두면 편리한 함수

●리스트의 속성과 메소드 정보확인

●dir(객체) 함수

◆객체가 가지고 있는 속성과 메소드의 정보를 리스트로 보여준다.

```
>>> heroes = [ "아이언맨", "토르", "헐크", "스칼렛 위치" ]
>>> dir(heroes)
['_add_', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

2019-05-06

© Chang Seung Kim - All rights reserved

23

알아두면 편리한 함수

●객체의 정보 확인하기

●help(객체)

◆객체에 대한 자세한 정보를 출력

```
>>> help(heroes)
Squeezed text (127 lines).
```

더블클릭

```
>>> help(heroes)
Help on list object:

class list(object)
| list(iterable=(), /)
|
| Built-in mutable sequence.
|
| If no argument is given, the constructor creates a new empty list.
| The argument must be an iterable if specified.
|
| Methods defined here:
|
| __add__(self, value, /)
|     Return self+value.
|
| __contains__(self, key, /)
|     Return key in self.
|
| __delitem__(self, key, /)
|     Delete self[key].
|
| __eq__(self, value, /)
|     Return self==value.
|
| __ge__(self, value, /)
|     Return self>=value.
```

2019-05-06

© Chang Seung Kim - All rights reserved

24

실습 - P01

- P6부터 P24까지 나와 있는 모든 예제를 IDLE에서 연습하고 화면을 캡처하여 제출한다.
 - 직접 입력할 경우 입력 오류 발생할 수 있다.
 - ◆ 편집기를 사용하여 입력할 내용을 미리 작성한다.
 - ◆ 한줄씩 복사하여 IDLE에 넣는다.
 - 더 추가적인 데이터를 다루는 것을 추천
 - 각 페이지의 기능을 반드시 연습하여 확인하여야 한다.

2019-05-06

© Chang Seung Kim - All rights reserved

25

실습 - P02

- P19의 `index()` 함수는 리스트의 내용을 검색하여 인덱스를 반환한다.
- `index()` 함수와 같은 기능을 하는 프로그램을 `for` 문장을 사용하여 구현하시오.
 - 함수 이름은 `myIndex()` 이다.
 - 매개변수로 리스트(`list1`)와 찾고자하는 값(`item`)을 전달한다.
 - 만약, 리스트에 찾고자하는 데이터가 없을 경우에는 `-1` 을 출력한다.
 - 데이터를 찾았을 경우에는 데이터가 있는 위치의 인덱스를 출력한다.
- `for()` 문장을 사용할 때 리스트의 항목수를 알기위하여 `len()` 함수를 사용한다.
 - `len(리스트변수이름)`

2019-05-06

© Chang Seung Kim - All rights reserved

26

실습 - P02

```

heroes = ['아이언맨', '토르', '헐크', '스칼렛 위치']

def myIndex(sourceList, item):
    

    return -1

print(myIndex(heroes, '헐크'))

```

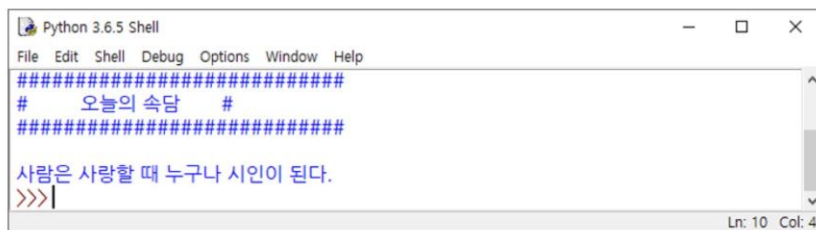
2019-05-06

© Chang Seung Kim - All rights reserved

27

실습 - P03 : 오늘의 속담

- 리스트에 여러 개의 속담을 저장한 후에 속담 중에서 하나를 랜덤하게 골라서 오늘의 속담으로 제공한다



2019-05-06

© Chang Seung Kim - All rights reserved

28

실습 - P03 : 오늘의 속담

```
import random

quotes = []

quotes.append("꿈을 지녀라. 그러면 어려운 현실을 이길 수 있다.")
quotes.append("분노는 바보들의 가슴속에서만 살아간다..")
quotes.append("고생 없이 얻을 수 있는 진실로 귀중한 것은 하나도 없다.")
quotes.append("사람은 사랑할 때 누구나 시인이 된다.")
quotes.append("시작이 반이다.")

dailyQuote = random.choice(quotes)
print("#####")
print("#    오늘의 속담    #")
print("#####")
print("")
print(dailyQuote)
```

● random.choice(리스트)

◆ 리스트의 항목 중에서 임의로 하나를 선택하여 반환

2019-05-06

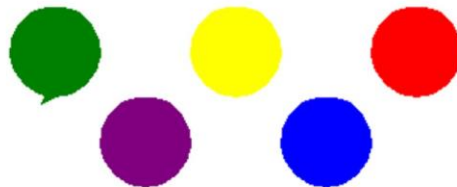
© Chang Seung Kim - All rights reserved

29

실습 - P04 : 오륜기 그리

● 반복 구조를 사용하여 화면에 오륜기를 그려보자.

● 오륜기의 색상과 위치를 리스트에 저장해보자.



2019-05-06

© Chang Seung Kim - All rights reserved

30

실습 - P04 : 오류기 그리

```

>>> positions = [[0, 0, "blue"], [-120, 0, "purple"], [60, 60, "red"],
[-60, 60, "yellow"], [-180, 60, "green"]]
>>> x, y, c = positions[0]
>>> x
0
>>> y
0
>>> c
'blue'
>>> for pos in positions:
    print(pos)

[0, 0, 'blue']
[-120, 0, 'purple']
[60, 60, 'red']
[-60, 60, 'yellow']
[-180, 60, 'green']
>>>

```

2019-05-06

© Chang Seung Kim - All rights reserved

31

실습 - P04 : 오류기 그리

```

import turtle

def draw_olympic_symbol():
    positions = [[0, 0, "blue"], [-120, 0, "purple"], [60, 60, "red"],
[-60, 60, "yellow"], [-180, 60, "green"]]
    for x, y, c in positions:
        t.penup()
        t.goto(x, y)
        t.pendown()
        t.color(c, c)
        t.begin_fill()
        t.circle(30)
        t.end_fill()

t = turtle.Turtle()
draw_olympic_symbol()

```

2019-05-06

© Chang Seung Kim - All rights reserved

32

실습 - P05

● 함수의 실습-P05의 square 함수와 같이 circle 함수를 만들어, (x, y)좌표로 이동하고 원을 그리는 함수를 만들어 적용하시오.

● 함수명 : circle

● 매개변수 : (x,y) 좌표, 반지름, 색상


2019-05-06

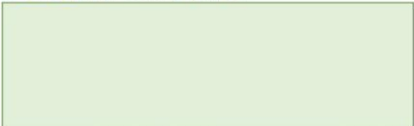
© Chang Seung Kim - All rights reserved


33

실습 - P05

```
import turtle

def move(x, y):
    

def circle(x, y, color, radius):
    

def draw_olympic_symbol():
    positions = [[0, 0, "blue"], [-120, 0, "purple"], [60, 60, "red"],
                 [-60, 60, "yellow"], [-180, 60, "green"]]
    

t = turtle.Turtle()
draw_olympic_symbol()
```

2019-05-06

© Chang Seung Kim - All rights reserved

34

실습 - P06

- 리스트는 문자열을 저장할 수 있다. 강아지를 많이 키우는 사람을 가정하자. 강아지들의 이름을 저장하였다가 출력하는 프로그램을 작성해보자.

```
강아지의 이름을 입력하시오(종료시에는 엔터키) 미나
강아지의 이름을 입력하시오(종료시에는 엔터키) 초롱이
강아지의 이름을 입력하시오(종료시에는 엔터키) 써니
강아지의 이름을 입력하시오(종료시에는 엔터키) 텅커벨
강아지의 이름을 입력하시오(종료시에는 엔터키)
강아지들의 이름:
미나, 초롱이, 써니, 텅커벨,
```

2019-05-06

© Chang Seung Kim - All rights reserved

35

실습 - P06

```
dogNames = []
while True:
    name = input('강아지의 이름을 입력하시오(종료시에는 엔터키) ')
    if name == '':
        break
    dogNames.append(name)

print('강아지들의 이름: ')
for name in dogNames:
    print(name, end=" ", " ")
```

2019-05-06

© Chang Seung Kim - All rights reserved

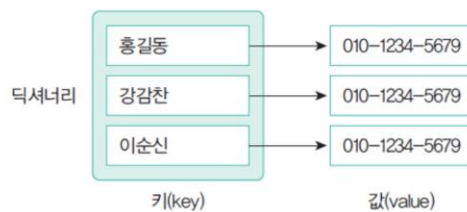
36

Dictionary

딕셔너리

●딕셔너리(dictionary)

- 리스트와 같이 값을 저장하는 방법이다.
- 딕셔너리에는 값(value)과 관련된 키(key)가 있다.
- 딕셔너리는 { } 를 사용한다.



실습 - P07 : 딕셔너리

● 딕셔너리에 키와 값 삽입하기

- 딕셔너리이름 = { 키:값 }
- 딕셔너리이름[키] = 값

```
>>> phone_book = {}
>>> phone_book = { "홍길동": "010-1111-2222" }
>>> print(phone_book)
{'홍길동': '010-1111-2222'}
```

```
>>> phone_book["강감찬"] = "010-1234-5679"
>>> phone_book["이순신"] = "010-1234-5680"
>>> print(phone_book)
{'홍길동': '010-1111-2222', '강감찬': '010-1234-5679', '이순신': '010-1234-5680'}
```

실습 - P07 : 딕셔너리에서 탐색

● 키를 가지고 값을 찾는다.

```
>>> print(phone_book["강감찬"])
010-1234-5679
```

실습 - P07 : 딕셔너리 모든 키와 값 출력하기

- 키 보기 : 딕셔너리.keys()
- 값 보기 : 딕셔너리.values()

```
>>> print(phone_book.keys())
dict_keys(['홍길동', '강감찬', '이순신'])
>>> print(phone_book.values())
dict_values(['010-1111-2222', '010-1234-5679', '010-1234-5680'])
```

- 키값을 출력하기

```
>>> for key in phone_book.keys():
    print(key)
```

```
홍길동
강감찬
이순신
```

실습 - P07 : 딕셔너리 항목 방문

```
>>> for key in sorted(phone_book.keys()):
    print(key, phone_book[key])
```

```
강감찬 010-1234-5679
이순신 010-1234-5680
홍길동 010-1111-2222
```

- sorted(리스트)
 - ◆ 리스트의 내용을 오름차순으로 정렬하여 반환
- sorted(리스트, reverse=True)
 - ◆ 리스트의 내용을 내림차순으로 정렬하여 반환

실습 - P07 : 딕셔너리 항목 방문

● 항목 삭제

```
>>> del phone_book["홍길동"]

>>> print(phone_book)
{'강감찬': '010-1234-5679', '이순신': '010-1234-5680'}
```

● 딕셔너리의 모든 항목 삭제

```
>>> phone_book.clear()
>>> print(phone_book)
{}
0
```

● 실습-P07 을 모두 하나의 파일로 만든다.

- 각각의 줄을 복사하여 IDLE 에 복사하고 실행한다.
- 실행한 결과들을 캡처하여 제출

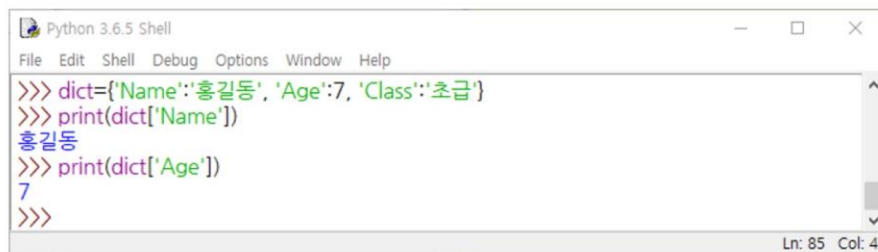
2019-05-06

© Chang Seung Kim - All rights reserved

43

예제

● 한 학생에 대한 정보를 딕셔너리로 저장하기



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help

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

Ln: 85 Col: 4

2019-05-06

© Chang Seung Kim - All rights reserved

44

실습 - P08 : 도전문제

- 리스트(list) 에는 각 학생의 정보가 딕셔너리로 들어 있다.
- 딕셔너리의 학생정보의 키
 - Name, Tel, Dept
- 4명의 학생 정보를 입력하고, 다음과 같이 출력하시오
 - 리스트의 모든 내용을 출력한다.
 - 학생정보의 이름만 출력한다.
 - 다음과 같이 출력한다.
 - 컴퓨터공학과 학생의 이름을 모두 출력하시오.

2019-05-06

© Chang Seung Kim - All rights reserved

45

도전문제 (실습 - P0906)

●출력

```
리스트 데이터
[{'Name': '홍길동', 'Tel': '010-1111-0001', 'Dept': '그린화학과'}, {'Name': '홍길서', 'Tel': '010-1111-0002', 'Dept': '컴퓨터공학과'}, {'Name': '홍길남', 'Tel': '010-1111-0003', 'Dept': '전자공학과'}, {'Name': '홍길북', 'Tel': '010-1111-0004', 'Dept': '바이오생명공학과'}]
```

```
전체 학생 이름
홍길동 홍길서 홍길남 홍길북
```

```
전체 학생 명단
홍길동 [전화번호] 010-1111-0001 [학과] 그린화학과
홍길서 [전화번호] 010-1111-0002 [학과] 컴퓨터공학과
홍길남 [전화번호] 010-1111-0003 [학과] 전자공학과
홍길북 [전화번호] 010-1111-0004 [학과] 바이오생명공학과
```

```
컴퓨터 공학과 학생 명단
홍길서
```

2019-05-06

© Chang Seung Kim - All rights reserved

46

도전문제 (실습 - P0906)

```
lt = []
```

```
print("리스트 데이터")
print(lt)
print()
```

```
print("전체 학생 이름")
```

```
print()
print()
```

2019-05-06

© Chang Seung Kim - All rights reserved

47

도전문제 (실습 - P0906)

```
print("전체 학생 명단")
```

```
print()
print("컴퓨터 공학과 학생 명단")
```

2019-05-06

© Chang Seung Kim - All rights reserved

48

실습 - P09 : 편의점 재고 관리

- 편의점에서 재고 관리를 수행하는 프로그램을 작성해보자. 편의점에서 판매하는 물건의 재고를 딕셔너리에 저장한다.

물건의 이름을 입력하시오: 콜라
4



2019-05-06

© Chang Seung Kim - All rights reserved

49

실습 - P09 : 편의점 재고 관리

```
items = { "커피음료": 7, "펜": 3, "종이컵": 2, "우유": 1, "콜라": 4, "책": 5 }

item = input("물건의 이름을 입력하시오: ");
print (items[item])
```

2019-05-06

© Chang Seung Kim - All rights reserved

50

실습 - P10 : 영한사전

- 영한사전을 구현해보자. 영어 단어를 키로 하고 설명을 값으로 하여 저장하여 구현한다.

단어를 입력하시오: one
하나
단어를 입력하시오: two
둘



2019-05-06

© Chang Seung Kim - All rights reserved

51

실습 - P10 : 영한사전

```
english_dict = dict()

english_dict['one'] = '하나'
english_dict['two'] = '둘'
english_dict['three'] = '셋'

word = input("단어를 입력하시오: ");
print (english_dict[word])
```

2019-05-06

© Chang Seung Kim - All rights reserved

52

실습- P11 : 이메일 보내기

```
import smtplib
from email.mime.text import MIMEText

me = 'abc@server.kr' # 보내는 사람 메일 주소
you = 'def@server.com' # 받는 사람 메일 주소
contents = '12월 20일에 동창회가 있으니 참석해주시기 바랍니다'

msg = MIMEText(contents, _charset='euc-kr')
msg['Subject'] = '동창회 모임'
msg['From'] = me
msg['To'] = you

server = smtplib.SMTP('smtp.gmail.com', 587)
server.ehlo()
server.starttls()
server.ehlo()

server.login("자신의 아이디", "패스워드")

server.sendmail(me, you, msg.as_string())
server.quit()
```

2019-05-06

© Chang Seung Kim - All rights reserved

53

과제 : 도전문제(StockDict.py)

- 편의점 재고관리 문제는 물품을 키(key)로하여 재고만 가지고 있다.
- 각 물품에 대하여 다음의 정보를 가지도록 추가한다. (물품의 정보는 리스트 사용)
 - 각 물품의 정보 : 상품의 가격, 판매수량, 재고수량
- 다음과 같이 동작하도록 프로그램을 작성한다.

```
명령을 입력하시오(중지,출력):출력
커피      가격=3000      판매량= 5      재고= 10
펜        가격=1000     판매량= 10     재고= 20
종이컵    가격=20       판매량= 1050   재고= 550
우유      가격=500      판매량= 10     재고= 5
콜라      가격=1050     판매량= 22     재고= 10
명령을 입력하시오(중지,출력):중지
재고관리 프로그램을 중지합니다.
```

- 무한 루프를 돌면서, 명령을 물어본다.
 - 명령에 "중지"를 입력하면 "재고관리 프로그램을 중지합니다." 를 출력하고 프로그램을 중지한다.
 - 명령에 "출력"을 입력하면 현황을 출력한다.
 - 이외의 명령을 입력하면 "적합하지 않은 명령입니다." 를 출력한다.

2019-05-06

© Chang Seung Kim - All rights reserved

54

과제 : 도전문제(StockDict.py)

● 각 물품에 대하여 다음의 정보를 가지도록 추가한다. (상품의 정보는 리스트 사용)

- 물품의 가격, 판매수량, 재고수량
- ◆ 딕셔너리의 키는 물품명
- ◆ 물품의 가격, 판매수량, 재고수량

```
items = {"커피": [3000, 5, 10],
         "펜": [1000, 10, 20],
         "종이컵": [20, 1050, 550],
         "우유": [500, 10, 5],
         "콜라": [1050, 22, 10]}
```

2019-05-06

© Chang Seung Kim - All rights reserved

55

과제 : 도전문제(StockDict.py)

● 다음과 같이 현황을 출력하도록 프로그램을 작성한다.

커피	가격= 3000	판매량= 5	재고= 10
펜	가격= 1000	판매량= 10	재고= 20
종이컵	가격= 20	판매량= 1050	재고= 550
우유	가격= 500	판매량= 10	재고= 5
콜라	가격= 1050	판매량= 22	재고= 10

● 출력하는 부분이 여러곳이므로 함수를 사용하여 정의한다.

◆ `statusPrint()` - 모든 물품의 키를 조회하여 각 물품을 출력하는 함수를 호출한다.

◆ `printItem(key)` - 물품이 있으면 키를 키로 조회하여 물품의 정보를 출력한

```
def printItem(key):
```

```
def statusPrint():
```

2019-05-06

© Chang Seung Kim - All rights reserved

56

과제 : 도전문제(StockDict.py)

●무한 루프를 돌면서, 명령을 물어본다.

- 명령에 "중지"를 입력하면 "재고관리 프로그램을 중지합니다." 를 출력하고 프로그램을 중지한다.
- 명령에 "출력"을 입력하면 현황을 출력한다.
- 이외의 명령을 입력하면 "적합하지 않은 명령입니다." 를 출력한다.

```
while True:
    cmd = input("명령을 입력하시오(중지, 출력): ")
```

2019-05-06

© Chang Seung Kim - All rights reserved

57

과제 : 도전문제(StockDict.py)

```
items = {
    "커피": [3000, 5, 10],
    "권": [1000, 10, 20],
    "종이컵": [20, 1050, 550],
    "우유": [500, 10, 5],
    "콜라": [1050, 22, 10]}

```

```
def printItem(key):
```

```
def statusPrint():
```

```
while True:
    cmd = input("명령을 입력하시오(중지, 출력): ")
```

2019-05-06

© Chang Seung Kim - All rights reserved

58

과제 : 도전문제(StockDict.py)

●실행

```

명령을 입력하시오(중지,출력):출력
커피      가격=3000      판매량= 5      재고= 10
펜        가격=1000      판매량= 10     재고= 20
종이컵     가격=20       판매량= 1050   재고= 550
우유       가격=500      판매량= 10     재고= 5
콜라       가격=1050     판매량= 22     재고= 10
명령을 입력하시오(중지,출력):판매
적합하지 않은 명령입니다.
명령을 입력하시오(중지,출력):중지
재고관리 프로그램을 중지합니다.
  
```

2019-05-06

© Chang Seung Kim - All rights reserved

59

과제 : 도전문제(SalesDict.py)

●추가 기능

●명령에 "판매" 을 입력

- ◆전체 판매하는 물품 리스트를 출력한다.
- ◆물품이름을 물어본다. ("판매한 물품은?")
- ◆입력한 물품이 딕셔너리에 있으면, 현재 상태를 출력한다.
 - ❖없으면, "판매하지 않는 물품입니다." 를 출력한다.
- ◆"판매한 수량은?" 을 물어보고 수량을 입력받는다.
- ◆판매량을 입력하면 판매량이 적당한지 확인한다.
 - ❖판매량 > 재고 : 에러메시지 → "판매한 수량이 재고보다 많습니다. 입력할 수 없습니다."
 - ❖판매량 <= 재고 : 판매량을 입력한 수만큼 증가시키고 재고를 감소시킨다.
- ◆수정된 물품 현황을 출력한다.

2019-05-06

© Chang Seung Kim - All rights reserved

60

과제 : 도전문제(SalesDict.py)

●물품리스트 출력하는 함수

```
def printKey():
```

2019-05-06

© Chang Seung Kim - All rights reserved

61

과제 : 도전문제(SalesDict.py)

●추가 기능

●명령에 "판매" 을 입력

- ◆전체 판매하는 물품 리스트를 출력한다.
- ◆물품이름을 물어본다. ("판매한 물품은?")
- ◆입력한 물품이 딕셔너리에 있으면, 현재 상태를 출력한다.
- ❖없으면, "판매하지 않는 물품입니다." 를 출력한다.

```
def saleFunc():
    printKey()
```

2019-05-06

© Chang Seung Kim - All rights reserved

62

과제 : 도전문제(SalesDict.py)

●추가 기능

●명령에 "판매" 을 입력

- ◆"판매한 수량은?" 을 물어보고 수량을 입력받는다.
- ◆판매량을 입력하면 판매량이 적당한지 확인한다.
 - ❖판매량 > 재고 : 에러메시지 → "판매한 수량이 재고보다 많습니다. 입력할 수 없습니다."
 - ❖판매량 <= 재고 : 판매량을 입력한 수만큼 증가시키고 재고를 감소시킨다.
- ◆수정된 물품 현황을 출력한다.

2019-05-06

© Chang Seung Kim - All rights reserved

63

과제 : 도전문제(SalesDict.py)

●추가 기능

●명령에 "판매" 을 입력

```
def saleFunc():
    printKey()
    key = input("판매한 물품은?")
    if key in items.keys():
        
    else:
        print("판매하지 않는 물품입니다.")
```

2019-05-06

© Chang Seung Kim - All rights reserved

64

과제 : 도전문제(SalesDict.py)

```
items = {"커피": [3000, 5, 10],
         "판": [1000, 10, 20],
         "종이컵": [20, 1050, 550],
         "우유": [500, 10, 5],
         "콜라": [1050, 22, 10]}
```

```
def printItem(key):
```

```
def statusPrint():
```

```
def printKey():
```

```
def saleFunc():
    printKey()
```

```
while True:
```

```
    cmd = input("명령을 입력하십시오(종지, 출력, 판매):")
```

2019-05-06

© Chang Seung Kim - All rights reserved

65

이번 장에서 배운 것

- 리스트는 항목들을 모아둔 곳이다.
- 리스트의 항목은 어떤 것이든 가능하다.
- 공백 리스트를 만들고 append()를 호출하여서 코드로 항목을 추가할 수 있다.
- 딕셔너리는 키와 값으로 이루어진다.
- 딕셔너리에 키를 제시하면 값을 반환한다.

2019-05-06

© Chang Seung Kim - All rights reserved

66