

# CHAPTER 03

## 파이썬 기초 실습

김 찬, 윤 영 선

[ckim.esw@gmail.com](mailto:ckim.esw@gmail.com), [ysyun@hnu.kr](mailto:ysyun@hnu.kr)

정보통신공학과



# 목차

## ■ 3.1 파이썬의 자료 구조

- ◆ 3.1.1 상수(Constant)와 리터럴(literal)

- ◆ 3.1.2 변수(Variable)

- ◆ 3.1.3 컬렉션 리터럴(List, Tuple, Dictionary, Set)

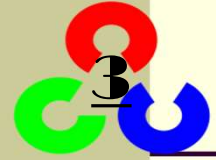
- ◆ 3.1.4 슬라이스 연산자

## ■ 3.2 if문

## ■ 3.3 for문

- ◆ range

- ◆ for each



## 3.1 파이썬의 자료구조

### ■ 3.1.1 상수(Constant)

#### ◆ 상수(Constant)와 리터럴(literal)

- 상수는 항상 똑같은 값이 저장된 곳이다.
  - C언어의 `const` 또는 `#define`과 같은 키워드를 제공하지 않음
  - 그래서 원칙적으로 상수를 사용하지 못함
- 리터럴은 숫자나, 문자열을 말한다.

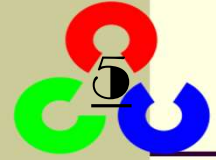


## 3.1 파이썬의 자료구조

### ■ 3.1.1 상수(Constant)

#### ◆ 리터럴의 종류

- 숫자 리터럴
  - 정수, 실수, 복소수
    - 4, 3.141592, 3+5j
- 문자 리터럴
  - 따옴표로 묶인 문자
    - “This is Python”, ‘This is Python’
- 논리값 리터럴
  - True, False



## 3.1 파이썬의 자료구조

### ■ 3.1.1 상수(Constant)

#### ◆ 리터럴의 종류

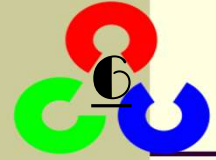
- 특수 리터럴

- None

- 다른 언어에서는 null을 많이 사용함.

- 컬렉션 리터럴

- List, Tuple, Set, Dict



## 3.1 파이썬의 자료구조

### ■ 3.1.2 변수(Variable)

◆ 변수 = 변하는 수 (추상적 개념)

```
1 # 변수
2 X = 30
3 print(X, type(X))
4
5 X = "30"
6 print(X, type(X))
```

문제 출력 터미널 GITLENS JUPYTER

Microsoft Windows [Version 10.0.17134.1] (c) Microsoft Corporation. All rights reserved.

D:\github\OpenCV-Python> cmd /c python

30 <class 'int'>  
30 <class 'str'>

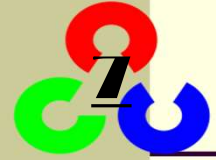
```
1 # 변수
2 X = 30 + 30
3 print(X, type(X))
4
5 X = "30" + "30"
6 print(X, type(X))
```

문제 출력 터미널 GITLENS JUPYTER

Microsoft Windows [Version 10.0.17134.1] (c) Microsoft Corporation. All rights reserved.

D:\github\OpenCV-Python> cmd /c python

60 <class 'int'>  
3030 <class 'str'>



## 3.1 파이썬의 자료구조

### ■ 3.1.3 컬렉션 리터럴

◆ List = []

- 일반적인 배열(변경 가능)

◆ Tuple = ()

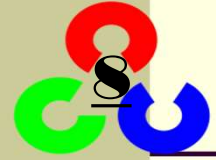
- 변경 불가능한 배열

◆ Set = {}

- 중복 불가능한 배열, 순서를 보장받지 못함

◆ Dict = {key:value}

- key와 value쌍으로 이루어진 자료형
- Server ↔ Client 데이터 전송에 가장 많이 사용되는 Json 형식



## 3.1 파이썬의 자료구조

### ■ 3.1.3 컬렉션 리터럴

#### ◆ List = []

- 값 추가: `List.append(value)`
- 특정 위치에 값 추가: `List.insert(index, value)`
- 값 삭제: `List.remove(value)`
- 값 삭제: `List.pop(index)`
- 값 삭제: `del List[index]`

#### ◆ Tuple = ()

- 추가, 삭제 등 값이 변화되는 행위 불가능





## 3.1 파이썬의 자료구조

### ■ 3.1.3 컬렉션 리터럴

#### ◆ Set = {}

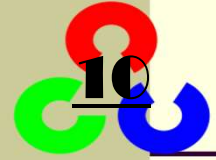
- 정수형에 한해서 자동 정렬됨
- 중복 불가능한 배열(집합)

#### ◆ Dict = {key: value}

- Id: kimchan
- Pw: chan

```
data = {'Id': 'kimchan', 'Pw': 'chan'}  
print(data, type(data))  
print(data['Id'])  
print(data['Pw'])
```

```
{'Id': 'kimchan', 'Pw': 'chan'} <class 'dict'>  
kimchan  
chan
```



## 3.1 파이썬의 자료구조

### ■ 3.1.4 슬라이스 연산자

◆ List = []

- 가장 많이 쓰는 배열
- [시작 인덱스 : 종료 인덱스 : 증가폭]

```
36 # List 슬라이스 연산
37 list_data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
38 print(list_data)
39 print(list_data[:])
40 print(list_data[:2])
41 print(list_data[::2])
42 print(list_data[::-1])
```

문제 출력 터미널 GITLENS JUPYTER AZURE 디

```
4.0\pythonFiles\lib\python\debugpy\adapter/.
on\2022-09-13\main.py "
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1]
[0, 2, 4, 6, 8]
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

# 11 3.1 파이썬의 자료구조

## ■ 3.2 if

### ◆ 조건문

- if == x
- if ~ in ~

```
37 # pythonic한 if문
38 list_data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
39 for i in range(len(list_data)):
40     if list_data[i] is 4:
41         print(list_data[i])
42
43     if list_data[i] == 4:
44         print(list_data[i])
45
```

## ■ 3.3 for

### ◆ 반복문

- range
- for each
- enumerate

```
37 # pythonic한 if문
38 list_data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
39 for i in range(len(list_data)):
40     if list_data[i] is not 4:
41         print(list_data[i])
42
43     if list_data[i] != 4:
44         print(list_data[i])
45
```

## 3.1 파이썬의 자료구조

### ■ 3.2 if

#### ◆ 조건문

- if == x
- if ~ in ~

```
# range
list_data = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
for i in range(7):
    print(list_data[i])
```

a  
b  
c  
d  
e  
f  
g

### ■ 3.3 for

#### ◆ 반복문

- range
- for each

```
43 list_data = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
44 ✓ for i in list_data:
45     print(i)
46
```

a  
b  
c  
d  
e  
f  
g

# 삽입 정렬

## ■ 정의

- ◆ 자신의 크기에 맞는 위치에 원소를 저장

## ■ 입력: tuple (수정 불가)

## ■ 출력: list

### ◆ 값 추가

- `.append(value)` # 마지막에 원소 추가
- `.insert(pos, value)` # 위치에 값을 추가

## 3.1 파이썬의 자료구조

### ■ 3.1.5 실습문제1

#### ◆ sort

### ■ 기본 소스 코드1

```
tp = (8, 41, 21, 3, 35, 5, 25, 54)
li = list(tp)
print(li)
```

```
[8, 41, 21, 3, 35, 5, 25, 54]
```

```
li.sort()
print(li)
```

```
[3, 5, 8, 21, 25, 35, 41, 54]
```

### ■ 기본 소스 코드2

```
tv = (8, 41, 21, 3, 35, 5, 25, 54)
lv = []
for i in range(len(tv)):
    pos = -1
    for j in range(len(lv)):
        # if tv[i] > lv[j]: # descending order, 내림차순
        if tv[i] <= lv[j]: # ascending order, 오름차순
            pos = j
            break
    if pos == -1: # len(lv) == 0, for loop j에서 pos을 못 찾았는 경우
        lv.append(tv[i])
    else:
        lv.insert(pos, tv[i])
print(tv)
```

```
[8, 41, 21, 3, 35, 5, 25, 54]
```

```
print(lv)
```

```
[3, 5, 8, 21, 25, 35, 41, 54]
```



## 3.1 파이썬의 자료구조

### ■ 3.1.5 실습문제2

#### ◆ reverse

### ■ 기본 소스 코드1

```
tv = (8, 41, 21, 3, 35, 5, 25, 54)
lv = list(tv)
lv.reverse()
print(lv)
```

```
[54, 25, 5, 35, 3, 21, 41, 8]
```

### ■ 기본 소스 코드2

```
for i in range(n):
    lv[i] = tv[n-1-i]
print(lv)
```

```
[54, 25, 5, 35, 3, 21, 41, 8]
```

### ■ 기본 소스 코드3

```
lv = []
for i in range(n):
    lv.append(tv[n-1-i])
print(lv)
```

```
[54, 25, 5, 35, 3, 21, 41, 8]
```

## 3.1 파이썬의 자료구조

### ■ 3.1.5 실습문제2

#### ◆ reverse

### ■ 기본 소스 코드1

```
list(reversed(range(8)))
```

```
[7, 6, 5, 4, 3, 2, 1, 0]
```

### ■ 기본 소스 코드2

```
list(range(7, -1, -1))
```

```
[7, 6, 5, 4, 3, 2, 1, 0]
```

### ■ 기본 소스 코드3

```
lv = []  
for i in reversed(range(8)):  
    lv.append(tv[i])  
print(lv)
```

```
[54, 25, 5, 35, 3, 21, 41, 8]
```

### ■ 기본 소스 코드4

```
lv = tv[::-1]  
print(lv)
```

```
[54, 25, 5, 35, 3, 21, 41, 8]
```



## 3.1 파이썬의 자료구조

### ■ 3.1.5 실습문제3

◆ string to list

◆ 입력:

- s="I am a student. You're a teacher!!"

### ■ 기본 소스 코드1

```
lv=list(s)
print(lv)
```

```
['I', ' ', 'a', 'n', ' ', 'a', ' ', 's', 't', 'u', 'd', 'e', 'n', 't', ' ', ' ', 'Y', 'o', 'u', ' ', ' ', 'r', 'e', ' ', ' ', 'a', ' ', ' ', 't', 'e', 'a', 'c', 'h', 'e', 'r', '!', '!']
```

### ■ 기본 소스 코드2

```
lv.remove('a')
print(lv)
```

```
['I', ' ', 'n', ' ', ' ', 'a', ' ', 's', 't', 'u', 'd', 'e', 'n', 't', ' ', ' ', 'Y', 'o', 'u', ' ', ' ', 'r', 'e', ' ', ' ', ' ', ' ', 't', 'e', 'a', 'c', 'h', 'e', 'r', '!', '!']
```

### ■ 기본 소스 코드3

```
nc = 0
for i in range(len(lv)):
    np = i - nc
    if lv[np] == 'a':
        lv.pop(np)
        nc += 1
print(lv)
```

```
['I', ' ', 'n', ' ', ' ', 's', 't', 'u', 'd', 'e', 'n', 't', ' ', ' ', 'Y', 'o', 'u', ' ', ' ', 'r', 'e', ' ', ' ', ' ', 't', 'e', 'a', 'c', 'h', 'e', 'r', '!', '!']
```