

# 인공지능프로그래밍

게임콘텐츠학과 박경수

<https://github.com/ggorr/Machine-Learning/tree/master/Python>

# 10장. 자료 구조

- ▶ 10.1. 리스트
- ▶ 10.2. 객체와 클래스에 대한 간단한 소개
- ▶ 10.3. 튜플
- ▶ 10.4. 사전
- ▶ 10.5. 열거형
- ▶ 10.6. 집합
- ▶ 10.7. 참조
- ▶ 10.8. 문자열에 대한 좀 더 자세한 설명
- ▶ 10.9. 요약

# 10.1. 리스트(list)

## ▶ 목록

```
x1 = [1, 2, 3]

x2 = [1.0, 2.1]
x3 = [1, 'a', 3.1, 88]

print(x1)

s = x1[0] + x1[1] + x1[2]
print(s)

x1[1] = '-4'
print(x1)

x1 += [4, 5, 6]
print(x1)
```

```
[1, 2, 3]
6
[1, '-4', 3]
[1, '-4', 3, 4, 5, 6]
```

## 10.2. 객체와 클래스에 대한 간단한 소개

- ▶ `len()` : 요소의 개수
- ▶ `append()` : 추가
- ▶ `pop()` : 제거(删除)
- ▶ `sort()` : 정렬(分类, 排序)


```
shoplist = ['apple', 'mango', 'carrot', 'banana']  
print(len(shoplist))  
for item in shoplist:  
    print(item, end=' ')  
print()
```

```
shoplist.append('rice')  
print(shoplist)
```

```
shoplist.sort()  
print(shoplist)
```

```
shoplist.pop(1)  
print(shoplist)
```

```
4  
apple mango carrot banana  
['apple', 'mango', 'carrot', 'banana', 'rice']  
['apple', 'banana', 'carrot', 'mango', 'rice']  
['apple', 'carrot', 'mango', 'rice']
```

- 
- ▶ 연습문제 10.1. [1.0, 2.3, 4.1, 5.4, 2.7]의 평균(平均)과 표준편차(标准偏差)를 구하시오
  - ▶ 연습문제 10.2. n의 약수(除数)를 모두 구하는(return) 함수를 작성하시오.
  - ▶ 연습문제 10.3. 1부터 n까지 소수(质数)를 모두 구하는(return) 함수(函数) prime(n)을 작성하시오.

## 10.3. 튜플(tuple)

### ▶ 목록

```
zoo = ('python', 'elephant', 'penguin')
print(len(zoo))
print(zoo[1])
for animal in zoo:
    print(animal, end=' ')
print()

zoo_sim = 'python', 'elephant', 'penguin'
print(zoo == zoo_sim)

# tuple in tuple
new_zoo = 'monkey', 'camel', zoo
print(new_zoo)
print(new_zoo[2][0])
```

```
3
elephant
python elephant penguin
True
('monkey', 'camel', ('python', 'elephant', 'penguin'))
python
```

## ▶ 변경 불가

```
x = (1, 2, 3)
x[0] = -2
```

## ▶ 단일 요소

```
a = (1)    # not tuple
b = (1, )  # tuple
```

## 10.4. 사전(dictionary)

- ▶ key-value pair

```
ab = {'Swaroop': 'swaroop@swaroopch.com',  
      'Larry': 'larry@wall.org',  
      'Matsumoto': 'matz@ruby-lang.org',  
      'Spammer': 'spammer@hotmail.com'}  
  
print(len(ab))  
print(ab.keys())  
print(ab['Spammer'])  
  
# new key-value  
ab['New Person'] = 'new@new.person.pe'  
print(ab.keys())
```

```
4  
dict_keys(['Swaroop', 'Larry', 'Matsumoto', 'Spammer'])  
spammer@hotmail.com  
dict_keys(['Swaroop', 'Larry', 'Matsumoto', 'Spammer', 'New Person'])
```



## 10.5. 열거형 (Enumeration, 枚举)

- ▶ 리스트, 튜플, 문자열, ...

```
apple is in shop list
False
-----
mango
banana
r
-----
['mango', 'carrot'] wa
['carrot', 'banana'] aroop
['mango', 'carrot'] waroo
['apple', 'mango', 'carrot', 'banana'] swaroop
13579
```

```
shoplist = ['apple', 'mango', 'carrot', 'banana']
name = 'swaroop'

if 'apple' in shoplist:
    print('apple is in shop list')
print('r' not in name)

# indexing
print('-----')
print(shoplist[1])
print(shoplist[-1])
print(name[-4])

# slicing
print('-----')
print(shoplist[1:3], name[1:3])
print(shoplist[2:], name[2:])
print(shoplist[1:-1], name[1:-1])
print(shoplist[:], name[:])
print('0123456789'[1::2])
```

## 10.6. 집합(set, 集合)

```
bri = {'brazil', 'russia', 'india'}  
print(bri)  
print('india' in bri)  
for item in bri:  
    print(item, end=' ')
```

```
{'india', 'brazil', 'russia'}  
True  
india brazil russia
```

## 10.7. 참조

- ▶ 메모리 참조(参考, reference)

```
x = [1, 2, 3]
y = x
z = x.copy()

y[0] = 10
print(x)
print(z)
```

```
[10, 2, 3]
[1, 2, 3]
```

## 10.8. 문자열에 대한 좀 더 자세한 설명

- ▶ startswith()
- ▶ find()
- ▶ join()

```
name = 'Swaroop'
if name.startswith('Swa'):
    print('Yes, the string starts with "Swa"')
if 'a' in name:
    print('Yes, it contains the string "a"')
if name.find('war') != -1:
    print('Yes, it contains the string "war"')
delimiter = '_*_*'
mylist = ['Brazil', 'Russia', 'India', 'China']
print(delimiter.join(mylist))
```

```
Yes, the string starts with "Swa"
Yes, it contains the string "a"
Yes, it contains the string "war"
Brazil_*_Russia_*_India_*_China
```

# 추가

```
x = range(10)
print(type(x))
```

```
y = list(x)
print(type(y))
print(y)
```

```
z = [2 * i for i in range(10)]
print(type(z))
print(z)
```

```
w = z[::2]
print(type(z))
print(z)
```

```
<class 'range'>
<class 'list'>
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
<class 'list'>
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
<class 'list'>
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

```
x = [2, 7, 3, 6, 4, 2, 5]
for i, a in enumerate(x):
    print(f'x[{i}] = {a}')
y = [1.0, 2.7, 1.5, 3.4, 7.0, 1.2, 5.4]
for i, a in enumerate(x):
    y[i] += a # y[i] += x[i]
print(y)
```

```
for a, b in zip(x, y):
    print(a, b)
```

x[0] = 2

x[1] = 7

x[2] = 3

x[3] = 6

x[4] = 4

x[5] = 2

x[6] = 5

[3.0, 9.7, 4.5, 9.4, 11.0, 3.2, 10.4]

2 3.0

7 9.7

3 4.5

6 9.4

4 11.0

2 3.2

5 10.4

- ▶ 연습문제 10.4. 다음 코드를 실행하시오.

```
d = {0: [1, 1, 1], 1: [1, 2, 3], 2: [1, 3, 5]}
key = d.keys()
print(type(key), key)
for k in d.keys():
    print(d[k])
```

d의 value 중 1의 개수를 출력하는 코드를 추가하시오.      출력 5

- ▶ 연습문제 10.5. 문자열 '010213197'에서 1의 개수를 구하시오
- ▶ 연습문제 10.6. random을 사용하여 길이 5인 두 리스트 x, y를 만드시오.  
두 리스트의 각 요소를 합하여 새로운 리스트 z를 만드시오.

```
x = [0.36528699148328003, 0.6514262718821947, 0.8037720787078022, 0.4646217034714667, 0.9505718687107783]
y = [0.8517718699569425, 0.35953025084109125, 0.9044083258226061, 0.33045203458699446, 0.9280626074122603]
z = [1.2170588614402225, 1.010956522723286, 1.7081804045304083, 0.7950737380584612, 1.8786344761230387]
```