

Python Programming

For Beginner

List

For Python 3

리스트 개요

1. 순서대로 늘어 놓고 원하는 방식으로 액세스할 수 있는 자료 구조
2. 리스트 요소 액세스
 - 순서대로 반복
 - 임의의 위치에서 요소 뽑기 → 일종의 카드 더미
3. 서수(Ordinal) → 순서(Order)
 - 첫 번째부터 시작
4. 기수(Cardinal) → 색인
 - 0에서 시작

리스트에서 함수 호출 시 동작

```
mystuff.append('hello')
```

1. 변수 목록에서 'mystuff'를 찾는다.
2. '.' 연산자를 지나 mystuff의 멤버를 찾는다.
3. 코드의 append와 일치하는 mystuff의 멤버 append가 함수임을 인지.
4. 코드의 append를 함수로 호출하면서 매개변수를 하나 더 덧붙임

→ append(mystuff, 'hello')

```
class Something(object):  
    def test(msg):  
        print(msg)
```

```
>>>a = Something()  
>>>a.test("hello")
```

Traceback (most recent call last):

File "<stdin>", Line 1, in <module>

TypeError:test() takes 1 positional argument but 2 were given

자주 사용하는 리스트 인스턴스의 멤버 함수

내장 함수	설명
append(value)	리스트 맨 뒤에 값 추가
insert(index, value)	인덱스 위치에 값 추가
extend(iterable)	반복 가능한 인자를 전달해 리스트에 추가
copy()	리스트 복사
remove(value)	전달한 값 삭제 (중복 값의 경우 처음 나오는 값)
pop(index)	해당 인덱스의 값을 반환하며 리스트에서는 삭제, 인자 없으면 맨 뒤 값 반환
clear()	리스트의 값 모두 삭제 (빈 리스트 만들기)
count(value)	인자로 전달한 값의 개수 확인
len(list)	리스트의 길이(크기) 확인
index(value)	해당 값의 인덱스 확인
reverse()	리스트에 있는 값을 역순으로 변경 저장
sort()	리스트 내용 오름차순 정렬 (내림 차순: reverse=true)
join(list)	특정 구분자를 포함해 문자열로 변환하는 함수
split(separator)	문자열을 특정 구분자를 기준으로 리스트로 변환하는 함수

리스트를 써야 할 때

1. 순서 관리

- 목록의 순번 (정렬된 순서 아님)

2. 기수를 이용해 임의의 위치에서 접근해야 할 때.

3. 요소를 순서대로 처리해야 할 필요가 있을 때

- For 순환문에 활용

Dictionary

For Python 3

사전 개요

1. 자료 저장 및 체계화에 사용

- 유사 데이터베이스

2. 리스트 vs. 사전

- 리스트는 숫자로 색인
- 리스트에서 요소 조회는 숫자로만 가능
- 리스트의 요소는 mutable
- 사전은 다양한 형식으로 값 조회 가능
- 사전은 한 항목(key)을 다른 항목(value)과 연관시킬 수 있다(associate)
- 사전에서 key는 immutable, value는 mutable

사전 관련 주요 내장 함수

내장 함수	설명
get(key, default_value)	키의 값을 반환한다.
pop(key)	해당 키의 값을 반환하고 항목 삭제
popitem()	무작위로 키와 값을 반환하고 삭제
clear()	사전의 모든 요소 삭제
copy()	해당 사전을 복사해 반환
update({key:value,...})	기존 사전에 인자로 받은 사전 데이터 추가
setdefault(key, default_value)	키가 있는 경우 값 반환, 키 없으면 기본 값으로 새로운 요소 추가
fromkeys([key1,...],default_value)	Iterable 키를 받아서 사전을 만든다. 기본 값 제공하거나 없으면 None
keys()	사전의 키 반환(dict_keys)
values()	사전의 값 반환(dict_values)
items()	사전의 키와 값 쌍을 모두 반환

사전을 사용하는 시기

1. 이름, 주소 등 키가 될 만한 식별자를 이용해 값을 꺼내야 할 때
2. 순서를 지킬 필요가 없을 때.
 - 순서가 필요하면 리스트
3. 항목과 키를 함께 추가하거나 조회할 때

수고했습니다!