SMP Lecture 4

데이터 구조, 파일 입출력

CSED101 프로그래밍과 문제해결

Student ID: 20240505
Name: 공현성
POVIS ID: hyunseong
Phone Number: 010-9649-3020

3월 수업 일정

03/08 (토)
03/15 (토)
03/22 (토)
03/29 (토) (TODAY)

POSTECH

Content

1. 데이터 구조
2. 파일 입출력

03/08	OT, 파이썬 설치, 개념 복습
03/15	입출력, 변수, 타입
03/22	연산자, 표현식, 흐름제어
03/29	**데이터 구조, 파일 입출력 (today)**
04/05	중간고사 대비
04/12	수업
04/19	수업
04/26	수업
05/03	수업
05/10	수업
05/17	수업
05/24	기말고사 대비

데이터 구조

C, Java 같은 언어에는 기본 자료형(primitive type)이 존재한다. 이는 가장 기본적인 데이터 타입으로 값 자체를 저장한다.

반면, 파이썬은 모든 것이 객체이다. 그러나, 객체 중에서도 다른 객체들의 모임으로 구성되는 객체가 있다. 데이터 구조(data structure)는 데이터들을 모으는 방식을 다룬다.

데이터 구조 - 리스트

리스트(list)는 여러 개의 값을 저장할 수 있는 자료형이며 대괄호 []를 사용해 생성한다.

리스트의 값에 접근하거나 변경하려면 인덱스(index)를 사용해야 한다.

리스트는 여러 기능을 포함하고 있는데, 아래는 그러한 기능들을 분류한 것이다.

① 슬라이싱(slicing): 리스트의 특정 부분의 원소를 자른다.
② 합치기(concatenate):두 리스트를 하나로 합친다.
③ 반복(repetition): 하나의 리스트를 반복적으로 합친다.
④ 삭제(deletion): 리스트의 특정 부분을 제거한다.
⑤ 언패킹(unpacking): 리스트 원소를 풀어서 각각을 다른 변수에 담는다.
⑥ 묶기(zipping): 2개 이상의 리스트를 인덱스 같은 원소끼리 묶어 새로운 리스트로 반환한다.

데이터 구조 - 리스트

아래와 같은 함수들이 존재한다.

함수 이름	설명
list.append(x)	list 뒤에 x 추가
list.pop()	맨 뒤 원소 반환하고 삭제
list.sort()	정렬
list.reverse()	순서 역순
list.index(x)	x 찾아서 반환
list.insert(loc, x)	loc에 x 삽입
list.remove(x)	x 제거. 여러 개 있으면 첫 번째만
list1.extend(list2)	list1 뒤에 list2 추가하여 확장. +와 동일한 역할
list.count(x)	list 내부에 찾을 값 x의 개수
del(list[pos])	list에서 위치 pos의 항목 삭제
len(list)	list 전체 항목 개수
zip(list1, list2)	list1과 list2를 같은 index끼리 묶기

데이터 구조 - 튜플

튜플(tuple)은 리스트와 유사하지만, 변경이 불가능하다는 특징이 있다. 소괄호 ()를 이용해 만들며, 순서가 존재한다.

함수 이름	설명
tuple.index(x)	x 찾아서 반환
tuple.count(x)	찾을 값 x의 개수
del(tuple)	tuple 삭제
len(tuple)	tuple의 전체 항목 개수
zip(tuple1, tuple2)	두 튜플 tuple1, tuple2를 같은 index끼리 묶기

데이터 구조 - 딕셔너리

딕셔너리(dictionary)는 key: value 쌍으로 구성되어 있는 자료형이다. 하나의 쌍이 원소 1개이다. 중괄호 {}를 이용해 만들 수 있으며, 내부의 요소를 자유롭게 변경할 수 있다.

튜플이나 리스트와는 다르게 딕셔너리에는 원소 간의 순서가 없다.

함수 이름	설명
dict.get(key)	key로 데이터 접근. dict[key]와 같이 간단하게 쓸 수 있음.
dict.keys()	모든 키 반환. dict_keys([1,2,'a'])와 같은 식으로 출력됨. list(dict.keys())와 같이 list로 캐스팅하여 dict_keys를 지울 수 있음.
dict.values()	모든 값 반환. dict_values([1,2,3])와 같은 식으로 출력됨.
dict.items()	key-value 쌍이 리스트로 묶인, 리스트가 나옴. dict_values([['a', 1],. . .])과 같은 식으로 출력됨.
dict in key	key 있으면 true 반환.
dict.update(dict2)	병합
pop(key)	key 제거하고 값을 반환. 없으면 None.
del(dict[key])	원소 삭제.
clear()	모든 요소 제거.

데이터 구조 - 문자열

문자열(string)은 문자들의 나열로 구성된 데이터 타입이다. 글자(character)의 리스트라고 생각할 수 있다.

len(str) 함수를 이용해서 문자열의 길이를 알 수 있다.

함수 이름	설명
str.upper()	모든 문자를 대문자로 만듦.
str.lower()	모든 문자를 소문자로 만듦.
str.swapcase()	모든 문자의 대/소문자를 뒤바꿈.
str.title()	모든 단어의 첫글자만 대문자로 만듦.
ord(ch)	ch를 대응하는 고유한 숫자로 변환. 한글도 변환 가능하다.
chr(ch)	ord와 반대로 ch에 해당하는 문자 반환.
ch.islower()	소문자인지 판별.
ch.isupper()	대문자인지 판별.

데이터 구조 - 집합

집합(set)은 중복을 허용하지 않는 리스트이다. 순서가 없으며 중괄호 {}를 이용해 생성한다.

딕셔너리와는 다르게 데이터가 한 쌍으로 구성된 것이 아니라 단일 데이터로 구성되어 있기 때문에 구별 가능하다.

수학에서의 집합처럼 합집합(|), 교집합(&), 차집합(-), 대칭차집합(^)을 계산하는 연산자가 따로 있다.

데이터 구조 - 집합

아래와 같은 함수들이 존재한다.

함수 이름	설명
set.add(x)	set에 단일 요소 x를 추가.
set.update(i)	다른 i(리스트, 집합, 튜플 등)의 여러 요소를 추가.
set.remove(x)	set에서 x 제거하고 없으면 KeyError 발생.
set.discard(x)	set에서 x 있으면 제거. 없어도 에러가 발생하지 않음.
set.pop()	임의의 요소 제거하고 반환.
set.clear()	set의 모든 요소 제거.
set1.union(set2)	두 집합 set1, set2의 합집합.
set1.intersection(set2)	두 집합 set1, set2의 교집합.
set1.difference(set2)	두 집합 set1, set2의 차집합.
set1.symmetric_diff erence(set2)	두 집합 set1, set2의 대칭 차집합(서로 공통되지 않은 요소).

데이터 구조 - 자료형 변환

각 자료형은 서로의 자료형으로 변환이 가능하다.

함수 `list()`, `set()`, `dict()`, `tuple()`을 이용한다.

특별하게, 딕셔너리로 변환할 때는 모든 원소가 2개 원소를 갖는 튜플로 구성되어 있어야 한다. 혹은, 각 요소를 key로 하고 기본값을 따로 설정할 수 있다.

데이터 구조 - in, not in 연산자

데이터 구조에 특정 원소가 포함되어있는지를 검사하는 연산자이다.

딕셔너리는 key를 검사한다. 딕셔너리 함수를 사용하여 원소를 추출하는 방식으로 value나 key: value 쌍이 존재하는지도 확인할 수 있다.

파일 입출력

파일 입출력(file IO)는 입출력이 파일에 적용되었을 때의 경우에 대해 다룬다.

앞서 배웠던 input(), print() 함수는 입출력이 콘솔에서 이루어졌다. 여기서 콘솔(console)이란 물리적인 장치로 키보드, 모니터가 모두 포함된다.

반면, 파일 입출력에서는 파일의 값을 입력값으로 쓰고, 파일에 새로운 출력값을 쓴다.
-> open() 함수를 이용하는데, 아래의 읽기 형식에 따라 입력을 할 것인지 출력을 할 것인지가 구분된다.

읽기 형식	설명
r (reading)	읽기, 기본값
w (writing)	덮어쓰기, 파일 없으면 생성
a (append)	새로운 텍스트를 추가
b (binary)	바이너리를 읽고 씀

파일 입출력

파일 조작이 끝난 후에는 close() 함수를 이용해 파일을 닫아주어야 한다. 파일을 닫지 않으면, 파일이 손상되는 등의 문제가 발생할 수도 있다.

이때, with 구문을 이용하면 자동으로 열고 닫기를 수행할 수 있다.

파일 입출력

아래는 파일 입출력 관련 함수이다.

함수 이름	설명
read()	모든 파일의 데이터를 한 번에 읽음.
readline()	파일을 한 줄씩 읽음.
readlines()	파일을 전부 읽어서 리스트를 반환함. 이때, 파일이 줄바꿈으로 구분되어 있었다면 리스트에 \n도 추가되어 있음.
write(x)	파일에 문자열 데이터를 씀.
writelines(list)	파일에 문자열 리스트를 한번에 씀. 자동 줄 바꿈이 없기 때문에, 문자열 리스트 각 원소의 끝 부분마다 수동으로 \n 넣어주어야 줄 바꿈이 수행됨.
close()	파일을 닫음.

Lecture4 끝
Q/A?