

문제해결과 알고리즘

선형 자료구조

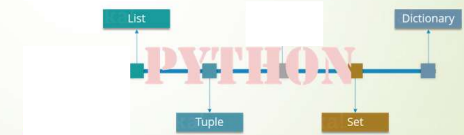
JEE, Jung Eun
rosaliejee@skku.edu

2

Contents

I. 컬렉션 자료구조

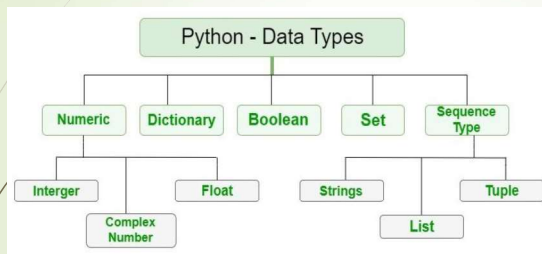
II. 실습 예제



3

1. 컬렉션 자료구조 (1/4)

Python Data Types



시퀀스(sequence)

- 순서를 가진 요소들의 집합으로 구성된 자료형
- Ex. 문자열, 바이트 시퀀스, 바이트 배열, 리스트, 튜플, range 객체

4

1. 컬렉션 자료구조 (2/4)

컬렉션 자료형

1) 리스트(list)

- 순서가 있는 데이터의 집합
- Ex) `aname = ['성균', 30, True]`

2) 튜플(tuple)

- 수정할 수 없는 데이터의 집합
- Ex) `bname = ('a', 'b', 'c')`

3) 딕셔너리(dictionary)

- 키(key)와 값(value)으로 이루어진 데이터 집합
- Ex) `dname = {1:'성균이', 2:'명륜이'}`

4) 집합(set)

- 중복을 허용하지 않는 데이터 집합
- Ex) `sname = {10, 20, 30, 40}`

1. 컬렉션 자료구조 (3/4)

리스트 자료형과 관련된 메서드

함수	설명	사용 예
<code>append()</code>	리스트 끝에 요소 추가	<code>mylist.append(value)</code>
<code>insert()</code>	리스트의 원하는 위치에 요소 삽입	<code>mylist.insert(index, value)</code>
<code>extend()</code>	리스트 확장	<code>mylist.extend(value)</code>
<code>remove()</code>	리스트 요소 제거	<code>mylist.remove(value)</code>
<code>count()</code>	리스트의 요소 개수 세기	<code>mylist.count()</code>
<code>sort()</code>	리스트 요소 정렬(오름차순)	<code>mylist.sort()</code>
<code>reverse()</code>	리스트 요소 역순으로 출력	<code>mylist.reverse()</code>
<code>index()</code>	리스트 요소의 위치 반환	<code>mylist.index(value)</code>
<code>pop()</code>	리스트에서 맨 마지막 요소 삭제	<code>mylist.pop()</code>
<code>len()</code>	리스트 요소 개수	<code>mylist.len(list명)</code>

1. 컬렉션 자료구조 (4/4)

리스트 함수

Ex.

```
>>> mylist=[1,4,2,2,2,6,3,3,3,7,3,3]
>>> #리스트 끝에 8 요소 추가
>>> mylist.append(8)
>>> mylist
[1, 4, 2, 2, 2, 6, 3, 3, 3, 7, 3, 3, 8]
>>> #리스트 3 인덱스에 5 요소 삽입
>>> mylist.insert(3,5)
>>> mylist
[1, 4, 2, 5, 2, 2, 6, 3, 3, 3, 7, 3, 3, 8]
>>> #리스트 확장으로 [9] 추가
>>> mylist.extend(9)
Traceback (most recent call last):
  File "<pyshell#48>", line 1, in <module>
    mylist.extend(9)
TypeError: 'int' object is not iterable
>>> mylist.extend([9])
>>> mylist
[1, 4, 2, 5, 2, 2, 6, 3, 3, 3, 7, 3, 3, 8, 9]
>>> #리스트 2 요소 제거
>>> mylist.remove(2)
>>> mylist
[1, 4, 5, 2, 2, 6, 3, 3, 3, 7, 3, 3, 8, 9]
>>> #리스트의 3 요소 개수
>>> mylist.count(3)
5
>>>
```

```
>>> #리스트 요소 정렬 : 오름차순
>>> mylist.sort()
>>> mylist
[1, 2, 2, 3, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9]
>>> #리스트 요소 정렬 : 내림차순
>>> mylist.sort(reverse=True)
>>> mylist
[9, 8, 7, 6, 5, 4, 3, 3, 3, 3, 3, 2, 2, 1]
>>> #리스트 요소 역순으로 출력
>>> mylist.reverse()
>>> mylist
[1, 2, 2, 3, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9]
>>> #리스트 1 요소 위치 반환
>>> mylist.index(1)
0
>>> #리스트에서 맨 마지막 요소 삭제
>>> mylist.pop()
9
>>> mylist
[1, 2, 2, 3, 3, 3, 3, 3, 4, 5, 6, 7, 8]
>>> mylist.pop(3)
3
>>> mylist
[1, 2, 2, 3, 3, 3, 3, 3, 4, 5, 6, 7, 8]
>>> #리스트의 요소 개수
>>> len(mylist)
12
>>>
```

2. 실습 예제 (1/6)

[w4_Ex1] 친구 목록 관리 프로그램

요구사항

- 리스트 내장 함수를 사용하여 친구 목록을 관리하는 프로그램을 작성하시오.
- 친구 목록 관리 프로그램 기능
 - 메뉴 출력, 친구 이름 추가/삭제/수정, 친구 리스트 출력
- 함수로 기능 구현
 - 메뉴 출력, 친구 이름 추가/삭제/수정

문제해결 및 알고리즘

- 친구 이름 저장 : 리스트(list)
- 친구 이름 추가 → `append()` 사용
- 친구 이름 삭제 → `remove()` 사용
- 친구 이름 수정 → `index()` 사용

[실행 예시]

```
[친구 목록 관리 프로그램]
1. 친구 추가
2. 친구 삭제
3. 친구 수정
4. 리스트 출력
0. 종료
메뉴 입력 : 4
>친구 리스트=> []
>추가할 이름 입력 : 지정은
>삭제할 이름 입력 : 지정은
>수정할 이름 입력 : 지정은
>새로운 이름 입력 : 정준이
>>>
```

2. 실습 예제 (2/6)

[w4_Ex1] 실행 예시

```
[친구 목록 관리 프로그램]
1. 친구 추가
2. 친구 삭제
3. 친구 수정
4. 리스트 출력
0. 종료
메뉴 입력 : 4
>친구 리스트=> []
>추가할 이름 입력 : 정준이
>삭제할 이름 입력 : 정준이
>수정할 이름 입력 : 지정은
>새로운 이름 입력 : 정준이
>>>
```

```
[친구 목록 관리 프로그램]
1. 친구 추가
2. 친구 삭제
3. 친구 수정
4. 리스트 출력
0. 종료
메뉴 입력 : 1
>추가할 이름 입력 : 정준이
>삭제할 이름 입력 : 정준이
>수정할 이름 입력 : 지정은
>새로운 이름 입력 : 정준이
>>>
```

```
[친구 목록 관리 프로그램]
1. 친구 추가
2. 친구 삭제
3. 친구 수정
4. 리스트 출력
0. 종료
메뉴 입력 : 4
>친구 리스트=> ['명준이', '성균이']
>추가할 이름 입력 : 정준이
>삭제할 이름 입력 : 명준이
>수정할 이름 입력 : 지정은
>새로운 이름 입력 : 정준이
>>>
```

```
[친구 목록 관리 프로그램]
1. 친구 추가
2. 친구 삭제
3. 친구 수정
4. 리스트 출력
0. 종료
메뉴 입력 : 0
프로그램을 종료합니다.
>>>
```

➡ [w4_Ex2] 스택 자료구조 삽입(push)/삭제(pop)

- 요구사항
 - 스택 자료구조로 삽입(push)/삭제(pop)을 수행하는 프로그램을 구현 하시오.
 - 스택 자료구조 : 리스트(list) 사용
 - 스택에 자료 삽입(push)
 - 문자('A', 'B', 'C...') 자료를 삽입하여 스택에 저장하고, '0'을 입력하면 저장을 끝내시오.

문제해결 및 알고리즘

- 스택 자료구조 : 리스트(list)
- 자료 삽입(push) → append() 사용
- 자료 삭제(pop) → pop() 사용

[실행 예시]

삽입 (PUSH)	A
삭제 (POP)	A
삽입된 자료	['A']
삭제된 자료	B
삽입된 자료	['A', 'B']
삭제된 자료	C
삽입된 자료	['A', 'B', 'C']
삭제된 자료	D
삽입된 자료	['A', 'B', 'C', 'D']
삭제된 자료	0
삽입된 자료	['A', 'B', 'C']
삭제된 자료	['A', 'B']
삽입된 자료	B
삭제된 자료	['A']
삽입된 자료	0
삭제된 자료	['A']



J.F.F., Jung-Eun


▶ [w4_Ex3] 아이스크림 맛 고르기/먹기

- 요구사항
 - 아이스크림 맛 고르기/먹기를 수행하는 프로그램을 구현 하시오.
 - 함수를 이용한 스택 자료구조 삽입(push)/삭제(pop)
 - 아이스크림 맛을 골라 쌓는 사용자 정의 함수(stack_puch())를 작성하시오.
 - 스택에 자료 삽입(push)
 - 쌓인 아이스크림 맛을 먹는 사용자 정의 함수(stack_pop())를 작성하시오.
 - 스택에 저장된 자료 삭제(pop)



[실행 예시]

1. 그  맛: ['민들']
2. 그  맛: ['민들', '바쁘라']
3. 그  맛: ['민들', '바쁘라', '베리']

1. 모  맛: 베리
3. 모  맛: 바쁘라

●

▶ [W4_Ex4] 중국 코스 요리 순서대로 먹기

- ## 요구사항
- 큐 자료구조 삽입(enqueue)/삭제(dequeue)로 중국 코스 요리를 순서대로 먹기를 수행하는 프로그램을 구현 하시오.
 - 큐 자료구조 : 리스트(list) 사용
 - 중국 코스 요리가 나오는 순서를 입력하고, '끝'을 입력하여 요리 입력을 끝낸다.
 - 큐에 자료 삽입하는 사용자 정의 함수(enqueue()) 정의
 - 요리 먹는 순서를 입력하여 순차적으로 식사를 하고, 순서를 잘못 입력하였으면 다시 입력하도록 오류 메시지를 출력한다. 식사가 완료되었으면 프로그램을 종료한다.
 - 큐에 자료 삭제하는 사용자 정의 함수(dequeue()) 정의

문제해결 및 알고리즘

- 큐 자료구조 : 리스트(list)
- 큐에 자료 삽입(enqueue) → append() 사용
- 큐에 자료 삭제(dequeue) → pop(0) 사용



JEE, Jang-Eun

▶ [w4 Ex4] 실행 예시

[illegible][illegible]

●

[W4] 도전과제 (1/4)

➤ [w4_hw1] 문자열의 알파벳, 숫자, 스페이스의 개수를 출력하는 프로그램을 작성하시오.

■ 요구사항

1. **딕셔너리 자료형**을 사용하시오.
 - 키(key): 알파벳, 숫자, 스페이스 항목(문자열)
 - 값(value): 각 항목의 개수
2. **사용자 정의 함수**(매개변수X, 반환값X)를 정의하여 각 항목의 **개수**를 카운트하고 실행 예시를 참고하여 딕셔너리 자료형을 사용하여 값을 출력하시오.
 - 문자열의 알파벳, 숫자, 스페이스의 개수는 **내장 함수**를 사용하시오.
 - isalpha(), isdigit(), isspace()
 - ※reference: [w3_Ex3]
3. 임의의 변수를 사용하고, 코드에 간단한 주석(설명)을 작성하시오.
4. **구름edu**에서 테스트케이스 일치 여부를 확인한 후, 제출하시오.
 - 수업자료의 실행예시와 같이 출력되도록, 구름edu에서 프로그래밍 하시오.

[W4] 도전과제 (2/4)

[w4_hw1 실행 예시]

```
문자열 입력 : BTS Dynamite Tropical Remix-192K.mp3
{'알파벳': 27, '숫자': 4, '스페이스': 3}
알파벳 개수 : 27
숫자 개수 : 4
스페이스 개수 : 3
```

```
문자열 입력 : BTS Billboard Hot 100 number 1hit:Butter-256k.mp3
{'알파벳': 33, '숫자': 8, '스페이스': 5}
알파벳 개수 : 33
숫자 개수 : 8
스페이스 개수 : 5
```

❖ 본인 스스로 요구사항에 맞는 알고리즘을 구상해보고, 입력 데이터에 따른 올바른 결과가 출력되도록 코드를 작성하세요.

[W4] 도전과제 (3/4)

➤ [w4_hw2] 스택(stack) 자료구조를 이용하여 초밥점시를 계산하는 프로그램을 작성하시오.

■ 요구사항

- 회전초밥음식점에서는 접시 색에 따라 가격이 다르다. 성균이는 먹고 싶은 초밥점시를 골라 위로 쌓으면서 초밥을 먹고, 맨 위에 쌓인 접시부터 가격 계산을 하려고 한다.
1. 무한히(무한 반복) 초밥 접시가격을 입력 받고, 0을 입력하면 접시 쌓기를 멈춘다.
 - 접시가격의 범위 : 2000~15000 (천원 단위)
 - stack_push(스택변수,접시가격) 함수를 정의하여 초밥점시를 위로 쌓는다.
2. 초밥점시 쌓기를 멈추면 맨 위에 쌓인접시부터 가격을 계산하고 남은 접시가 '없음'이면 '계산 완료'를 하고, 계산해야 할 총 가격을 출력한다.
 - stack_pop(스택변수) 함수를 정의하여 쌓인 초밥 점시를 계산한다.
3. 임의의 변수를 사용하고, 코드에 간단한 주석(설명)을 작성하시오.
4. **구름edu**에서 테스트케이스 일치 여부를 확인한 후, 제출하시오.
 - 수업자료의 실행예시와 같이 출력되도록, 구름edu에서 프로그래밍 하시오.



[W4] 도전과제 (4/4)

[w4_hw2 실행 예시]

```
[초밥 점시 계산하기]
접시가격:10000
접시가격:15000
접시가격:8000
접시가격:0
쌓인접시: [10000, 15000, 8000]
계산접시: 8000
잔여접시: [10000, 15000]
계산접시: 15000
잔여접시: [10000]
계산접시: 10000
잔여접시: []
계산완료
총 가격: 33000
```

```
[초밥 점시 계산하기]
접시가격:5000
접시가격:8000
접시가격:3000
접시가격:5000
접시가격:2000
접시가격:0
쌓인접시: [5000, 8000, 3000, 5000, 2000]
계산접시: 2000
잔여접시: [5000, 8000, 3000, 5000]
계산접시: 5000
잔여접시: [5000, 8000, 3000]
계산접시: 3000
잔여접시: [5000, 8000]
계산접시: 8000
잔여접시: [5000]
계산접시: 5000
잔여접시: []
계산완료
총 가격: 23000
```

❖ 본인 스스로 요구사항에 맞는 알고리즘을 구상해보고, 입력 데이터에 따른 올바른 결과가 출력되도록 코드를 작성하세요.