



컴퓨팅사고력 향상을 위한 문제해결과 알고리즘

1교시 : 자료구조와 문제해결

양 속 희

1. 자료 구조화의 개념 및 목적을 이해할 수 있다.
2. 선형구조의 자료를 관리할 수 있다.
3. 파이썬이 지원하는 여러 가지 편리한 자료구조에 대해 이해할 수 있다.

3.1 자료 구조의 개념 및 목적

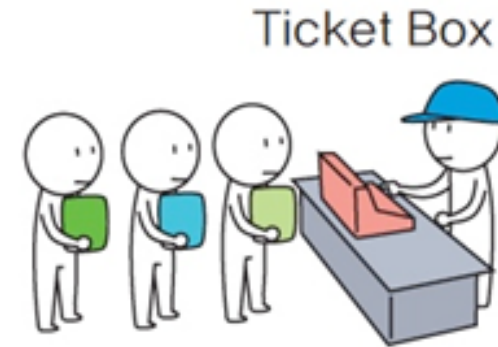
자료구조(data structure)의 개념

- 사람이 사물을 정리하는 것과 같이 프로그램에서도 **자료를 여러 가지 구조에 따라 정리**하는 것
- 자료를 효율적으로 사용하기 위해서 자료의 특성에 따라서 분류하여 구성하고 저장 및 처리하는 모든 작업을 의미
- 수 많은 자료 중 내가 직면한 문제해결에 필요한 자료를 선택하고, 저장하고, 처리하는 과정에서 문제해결이 효율적으로 이루어질 수 있도록 수집된 자료를 구조화시키는 것

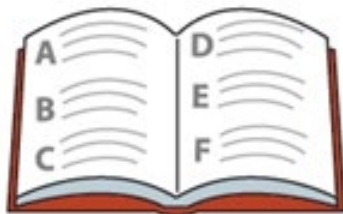
3.1 자료 구조의 개념 및 목적

일상생활에서의 정리

해야할 일 리스트

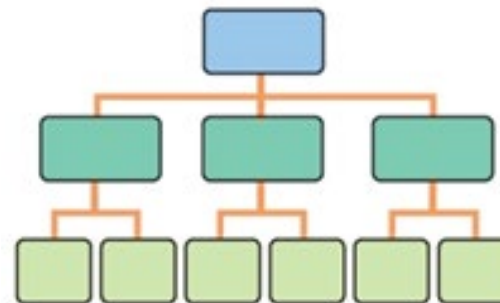


일상생활에서의 사물의조직화



사전

조직도



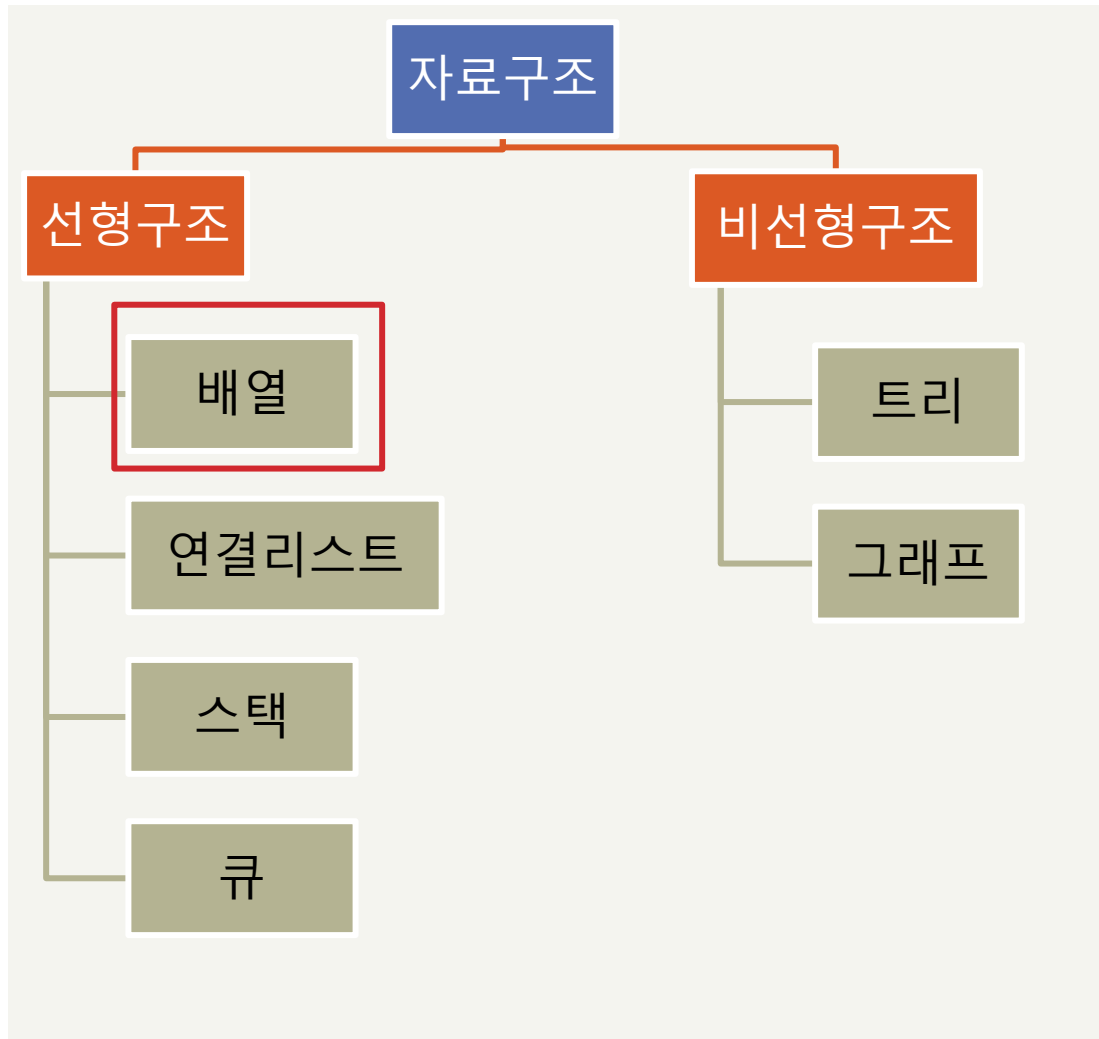
3.1 자료 구조의 개념 및 목적

자료구조의 목적

- 문제해결을 위해 내가 갖고 있는 자료를 효율적으로 저장하고 처리하고, 관리할 수 있도록 자료들 사이의 관계를 나타내는 것
- 컴퓨터의 자료를 효율적으로 사용하려면 사용자는 문제에 필요한 자료를 분석하여 구조화된 배치로 표현해야 함
- 문제해결에 대한 자료들의 관계에 해당하는 자료구조의 설계와 활용 능력이 필요함

3.1 자료 구조의 개념 및 목적

자료구조의 분류



파이썬에서 제공하는 리스트는 배열과 유사



3.1 자료 구조의 개념 및 목적

일상생활과 자료구조

일상생활에서의 예	해당하는 자료구조
물건 쌓아놓는 것	스택
학식 줄	큐
할일 리스트	리스트(배열)
영어사전	사전
지도, 길찾기	그래프
조직도	트리

선형구조의 특징

- 선형 구조는 자료 간의 앞뒤 관계가 1:1로 고정되어 있음
- 하나의 자료 뒤에 하나의 자료만이 존재하며 자료들이 직선 형태로 나열되어 있는 구조로 자료들 간에 순서를 고려함
- 선형구조의 종류
 - 배열(리스트)
 - 연결리스트
 - 스택
 - 큐

배열(array)의 개념

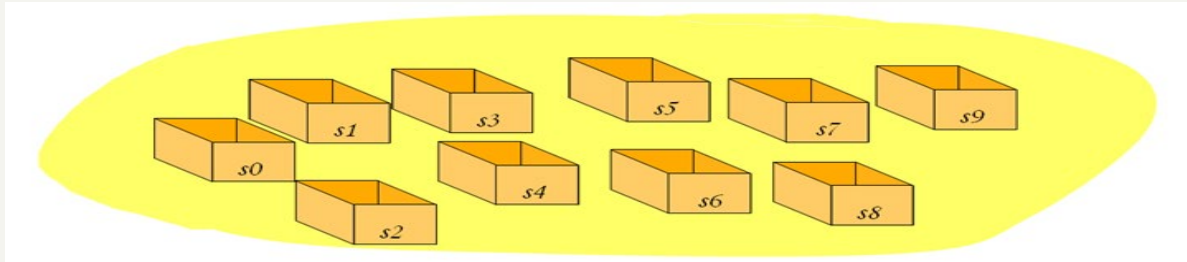
- 같은 타입의 데이터를 여러 개 저장하는 저장 장소
- 배열을 이용하면 여러 개의 값을 하나의 이름으로 지칭하여 사용
- 배열 안에 있는 각 데이터에는 정수번호(인덱스)를 이용하여 접근

파이썬에서 제공하는 리스트는 여러 타입의 데이터도 저장이 가능

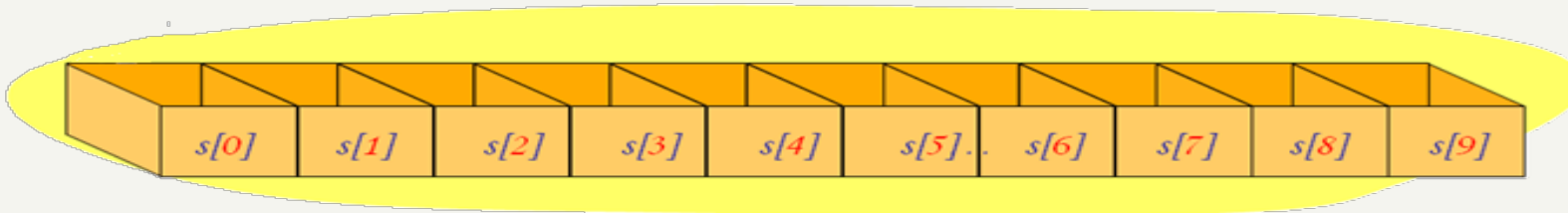
3.2 선형 구조 자료

- 배열의 필요성

- 별도의 이름은 가진 경우 관리가 어려움



- 배열은 하나의 이름을 공유하여 조작이 편리



3.2 선형 구조 자료

배열의 활용 예

- 원소의 위치=인덱스
- 인덱스는 0부터 시작

표 3-1 배열명: 대학 이름

인덱스	값
0	가천대학교
1	경북대학교
2	고려대학교
3	서강대학교
4	성균관대학교
5	세종대학교
6	아주대학교
7	충남대학교

스택(stack)의 개념

- 스택은 "나중에 들어온 자료가 먼저 나오는 (Last In First Out : LIFO)" 특성을 가지는 자료 구조
- 사용자가 위치를 명시하지 않음
- 스택에 자료를 넣을 때(push)에는 넣고자 하는 자료가 항상 가장 위쪽에 놓이게 됨
- 자료를 꺼낼 때(pop)에도 항상 가장 위쪽에 위치하고 있는 자료가 꺼내지게 됨



3.2 선형 구조 자료

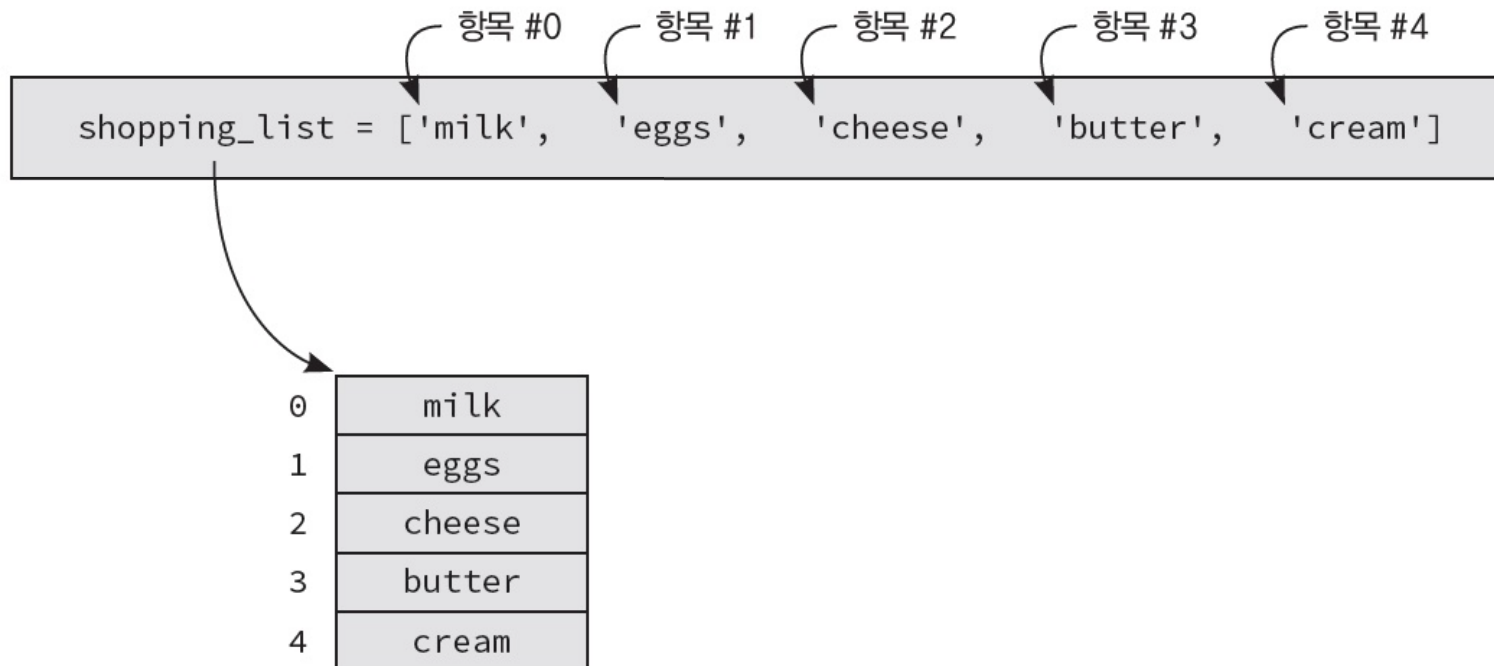
은행 번호표

큐(queue)의 개념

- “먼저 들어온 자료가 먼저 나가는(First In First Out : FIFO)” 특성을 가지는 자료 구조
- 큐에 자료를 넣을 때에는 넣는 자료가 가장 뒤쪽에 놓이게 됨
- 자료를 꺼낼 때에는 가장 앞쪽에 위치하고 있는 자료가 꺼내지게 됨



- ✓ 리스트 (list)는 여러 가지 자료를 저장할 수 있는 자료형
- ✓ 요소(element)는 리스트 내부에 있는 각각의 내용을 의미
- ✓ 인덱스(index)는 리스트 내부에서 값의 위치를 의미



파이썬의 리스트가 필요한 이유

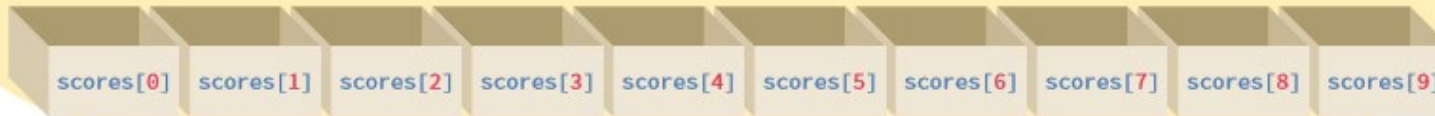
리스트는 하나의 이름을
공유해 자료의 조작이
편리해요^^



별도의 이름을 가지니
조작하기가 어려워!



리스트는 하나의 이름을 공유한다.



과제명

- 문자열 처리 프로그램

요구사항

학교 건물 이름을 입력하여 리스트에 저장한 후 리스트에 저장된 건물 수와 전체 항목을 출력해보시오.

<실행결과>

```
building name : 수선관  
building name : 호암관  
building name : 경제관  
building name : 인문관  
building name :  
건물수: 4  
수선관/호암관/경제관/인문관/
```

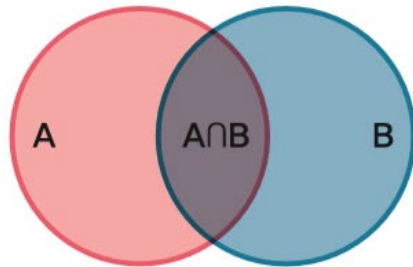


```
bname=[ ]  
  
while True:  
    name=input('building name : ')  
    if [redacted]:break  
    bname.append(name)  
  
print('건물수:', len(bname))  
  
for n in bname:  
    print(n, end='/')
```

리스트 조작 함수		
함수	설명	사용법
append()	리스트에 요소를 마지막 위치에 새로 추가	리스트.append(값)
pop()	리스트 제일 뒤의 항목을 빼내고, 빼낸 항목은 삭제 제거할 위치에 있는 요소를 제거	리스트.pop() 리스트.pop(위치)
insert()	리스트의 해당 위치에 요소를 새로 삽입	리스트.insert(위치, 값)
count()	해당 요소의 개수를 반환	리스트.count(찾을 값)
remove()	해당 요소를 찾아 삭제	리스트.remove(삭제할 값)
sort()	오름차순 정렬	리스트.sort()
reverse()	현재의 리스트를 그대로 거꾸로 뒤집는다.	리스트.reverse()
len()	리스트 요소 개수	len(리스트)

파이썬의 세트(SET)

- ✓ 수학의 집합과 같다.
- ✓ 중복되지 않은 항목들로 구성됨
- ✓ 세트간의 순서가 없다.
- ✓ 세트생성 : { }



전체적인 구조



세트 = { 항목1 , 항목2 , ... , 항목n }

```
>>> A = {1, 2, 3}
```

```
>>> B = {3, 4, 5}
```

```
>>> A | B
```

```
{1, 2, 3, 4, 5}
```

```
>>> A & B
```

```
{3}
```

```
>>> A - B
```

```
{1, 2}
```

합집합 | 연산자 또는 union()

교집합 & 연산자 또는 intersection()

차집합 - 연산자 또는 difference()

동아리 A, B에 아래와 같이 학생들이 가입이 되어있다.

clubA={'yang', 'kim', 'park'}

clubB={'lee', 'kim', 'choi'}

1. 동아리에 가입한 학생들의 모든 명단을 출력하시오.
2. A,B 동아리에 둘 다 가입한 학생의 명단을 출력하시오.



- ✓ 사전과 같다 : 단어와 단어의 설명이 저장
- ✓ 파이썬의 딕셔너리는 키(key)와 값(value)의 쌍을 저장한다.
- ✓ 키를 이용하여 값을 검색한다.

전체적인 구조



딕셔너리 = { 키1 : 값1 , 키2 : 값2 , ... }

키(key)	값(value)
"Kim"	"01012345678"
"Park"	"01012345679"
"Lee"	"01012345680"

딕셔너리 조작 함수		
함수	설명	사용법
get()	항목접근하기	딕셔너리.get(key)
pop()	항목삭제하기	딕셔너리.pop(key)
del ()	항목 꺼내고 삭제하기	del 딕셔너리[key]
items()	딕셔너리에 저장된 항목	딕셔너리.items()
keys()	딕셔너리에 저장된 키	딕셔너리.keys()
values()	딕셔너리에 저장된 값	딕셔너리.values()

```
>>> dept={1:'상경대',2:'사범대학',3:'인문대'}
>>> dept.get(1,'not Found')
'상경대'
>>> dept.get(4,'not Found')
'not Found'
>>> dept.items()
dict_items([(1, '상경대'), (2, '사범대학'), (3, '인문대')])
>>> dept.keys()
dict_keys([1, 2, 3])
>>> dept.values()
dict_values(['상경대', '사범대학', '인문대'])
>>> for i in dept:
        print(i,dept[i])
```

```
1 상경대
2 사범대학
3 인문대
```


공백 딕셔너리를 만들고 아래와 같이 신조어를 키로 하고 설명을 값으로 하여 저장하여 신조어를 검색해보자.

키(key)	값(value)
스라벨	Study and Life Balance
순삭	순간 삭제
오늘아눔	오우~놀 줄 아는 나눔가?

```
dic={}

dic['스라벨']='Study and Life Balance'
dic['순삭']='순간 삭제'
dic['오늘아눔']='오우~놀 줄 아는 농인가?'

for i in [ ]:
    print('%s:%s' % ( [ ] ))
    # [ ]

# 단어 검색
while True:
    w=input('search word :')
    if [ ]:break
    [ ]

# [ ]
```

수고하셨습니다.

