

## python 외기...

## Array vs List

- arr, numpy 모듈 필요.
- 데이터를 compact 하고 효율적으로 저장 (큰 데이터를 저장하기 좋음)
- arr 모듈 → 같은 데이터 타입 처리만 가능
- numpy 패키지 → 다른 데이터 타입 처리도 저장 가능.
- 원본 계산을 하기 좋음.

built-in  
다른 데이터 타입 저장 가능  
mutable (생성 이후 add, remove 가능)

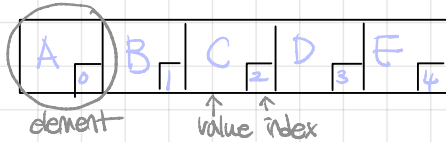
## Java 외기...

## Array

- 선언시 크기와 데이터 타입 지정 (메모리 공간에 할당할 사이즈를 정해놓음)  
→ 데이터 삽입, 삭제에 복잡함

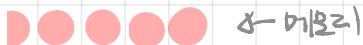
여러 데이터를 하나의 이름으로 접근하기 편리하기 위한 구조

index 존재 → 검색 편리



## Array List

- 배열을 통해 list를 구현
- 크기 지정 X
- 데이터 삽입, 삭제 시 시간 소요↑, 메모리 낭비↑  
(데이터를 꼭 밀고 당기는 방식으로 구현되기 때문)

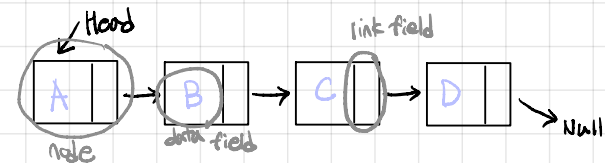


## List

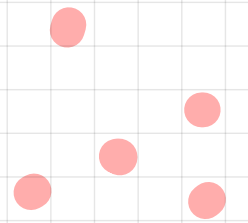
- 순서대로 저장, 중복 저장 허용  
→ index를 지정하는 것지만, 비효율적으로 저장되기 때문에 중요X.

## Linked List

- 저장되는 메모리 위치가 연속적이지 않다.
- 연결 데이터 구조이다.
- 노드 = 데이터 필드 + 다음 노드에 대한 참조
- 동적인 크기
- 삽입, 삭제 편리
- 순차적 접근만 가능하므로, 데이터 검색 시간↑  
(array를 순차접근 하더라도 linked list 보다 빠르다.)
- array 보다 많은 메모리 차지.



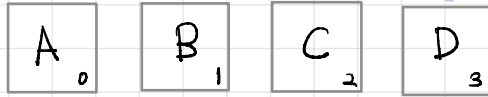
메모리



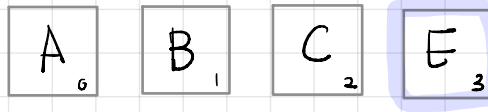
# Array vs List

# 15/11

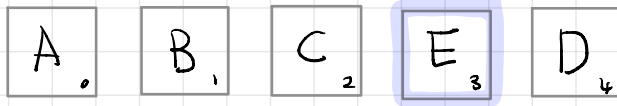
original



array

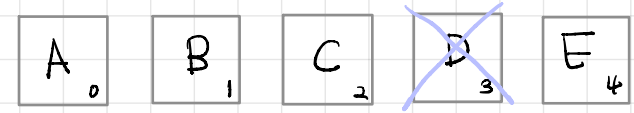


list

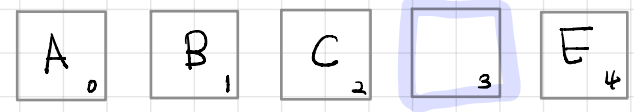


# 15/11

original



array



list

