



# IACF PYTHON STUDY GROUP

심화학습



네카라쿠베를 재정의하다.



NAVER



kakao



LINE



coupang



우이현행제틀





## 데이터는 내것 : 리스트

100개, 1000개 데이터가 필요

`int a1 = 1`

`int a2 = 2`

...

`int a1000 = 1000`

그럼 어떻게?

우리는 이것을 “**리스트**”로  
해결합니다.

리스트명 = `[a1, a2, ... a1000]`



## 데이터는 내것 : 리스트

대괄호안에 값을 순서대로 모아 놓은 것

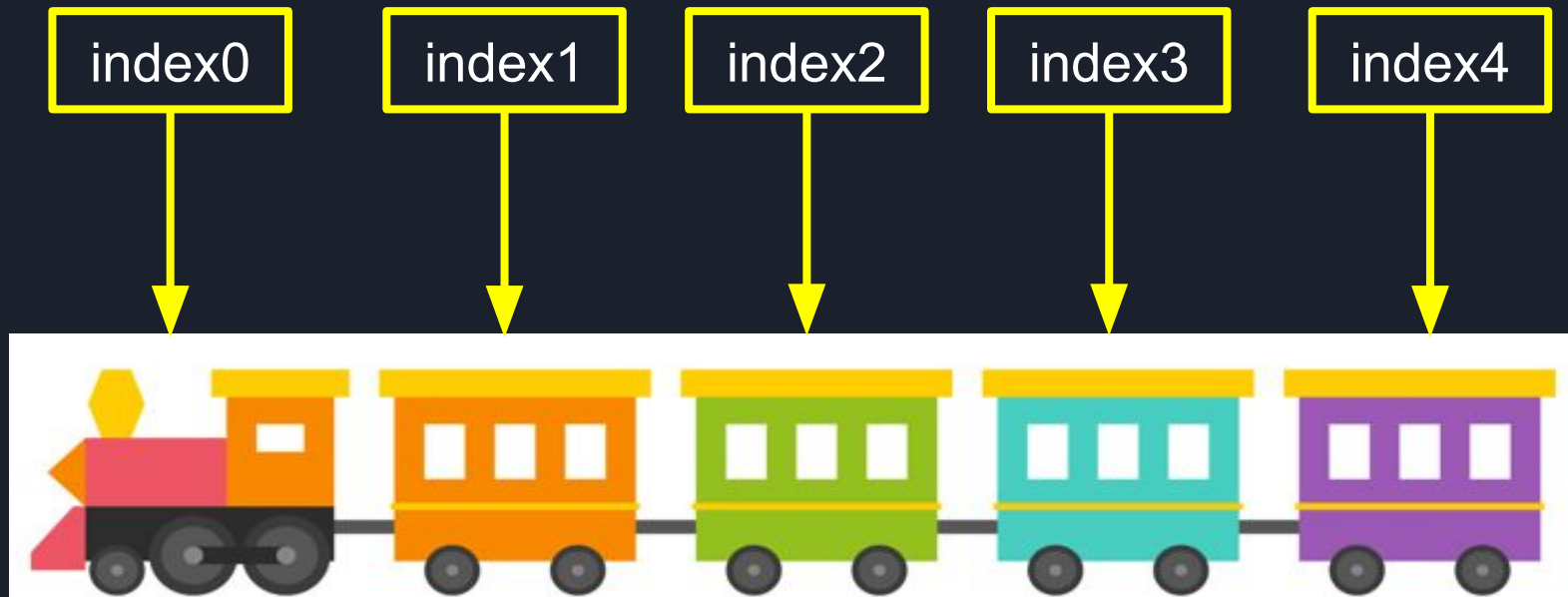
= 대괄호안에 객체를 순서대로 모아 놓은 것

[1, 2]

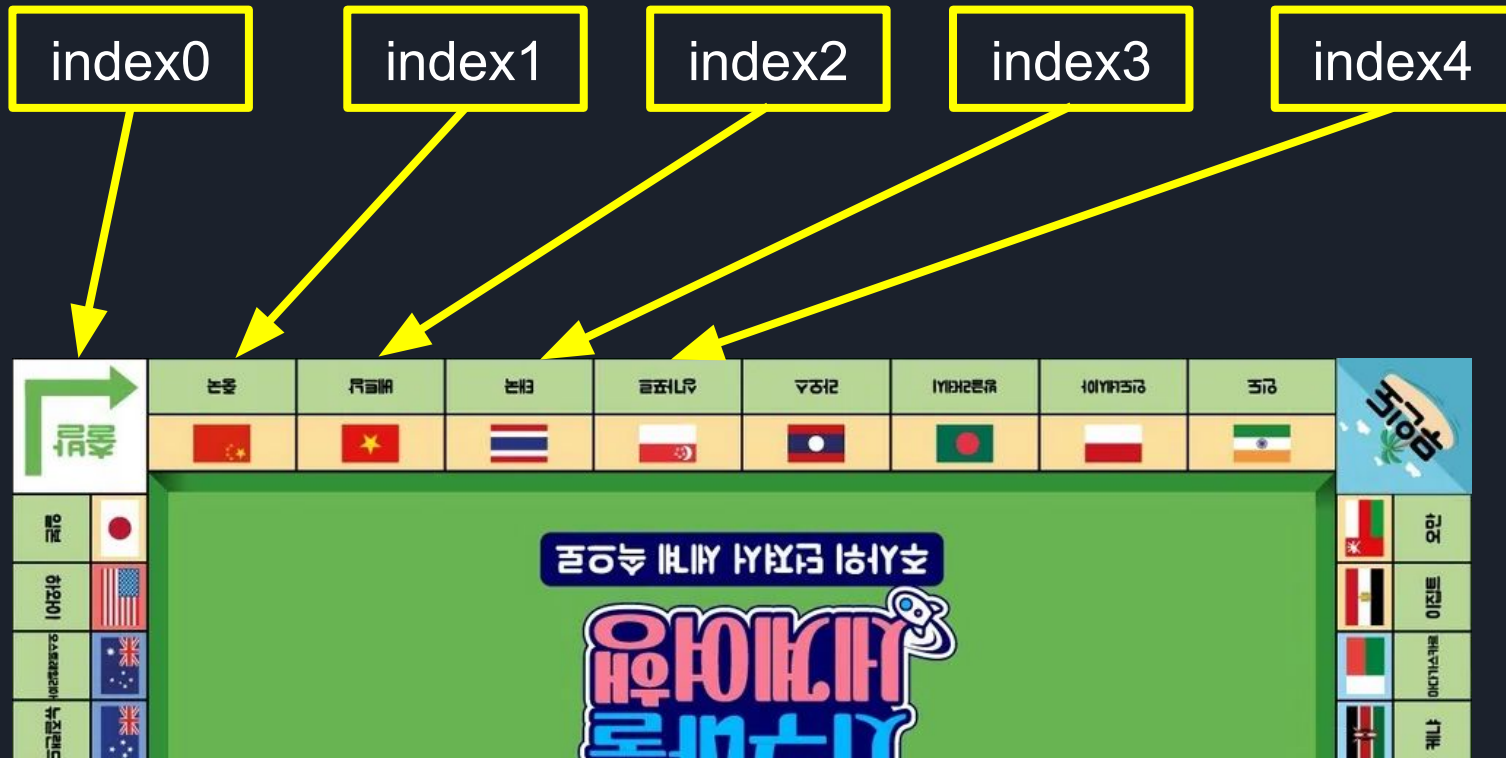
[1, "A", 3, [ 1, 2], 5]

[1, "A", 3, [ 1, 2, 3, ["B","C"],6], 5]

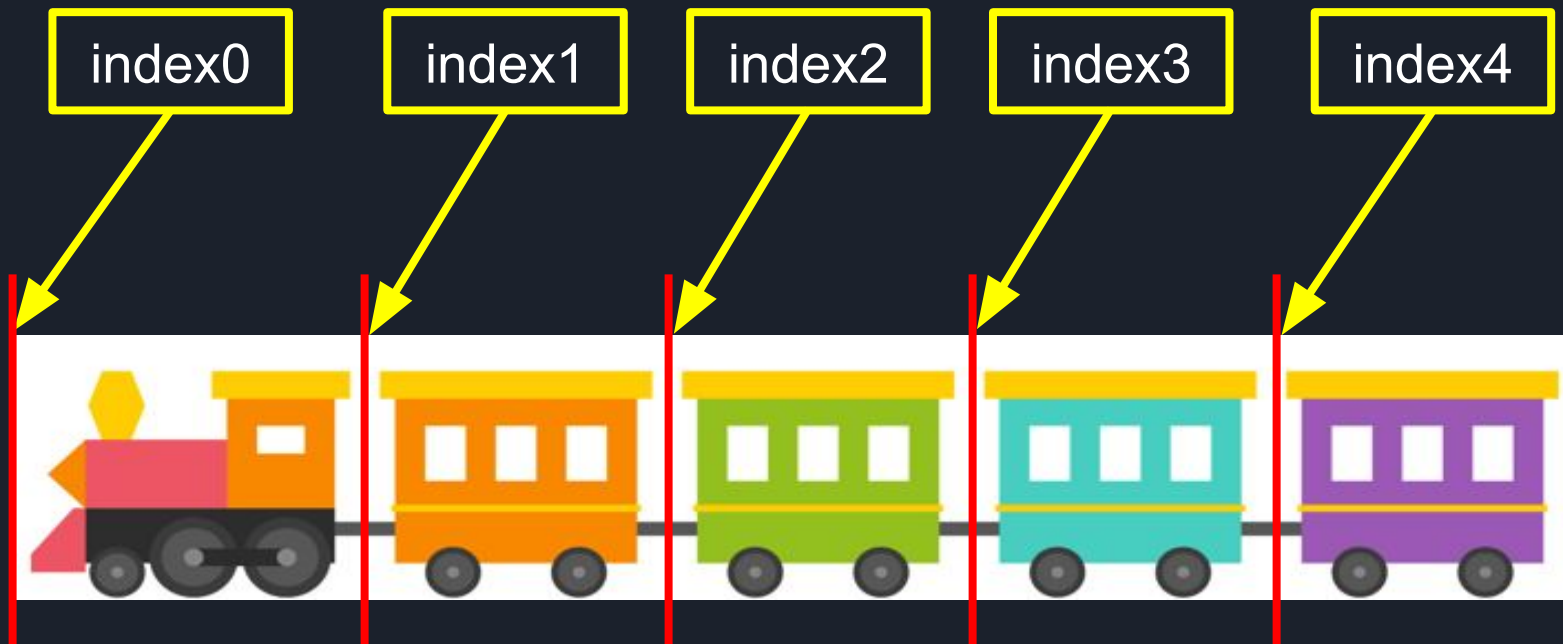
## 데이터는 내것 : 리스트



# 데이터는 내것 : 리스트



## 데이터는 내것 : 리스트





# 데이터는 내것 : 리스트

1. `append()`
  - 리스트 맨 끝에 항목 추가하기
2. `insert()`
  - 리스트 중간에, 원하는 위치(index) 에, 값을 추가하기
3. `remove()`
  - 저장되어 있는 값으로 삭제하기
4. `del`
  - 인덱스를 이용해서, 삭제하기 원하는 위치의 값을 삭제하기

<https://vision-ai.tistory.com/entry/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%ED%95%AD%EB%AA%A9-%EC%B6%94%EA%B0%80-%EC%82%AD%EC%A0%9C>





## 데이터는 내것 : 튜플(Tuple)

정의:

파이썬 딕셔너리는 변경 불가능(**immutable**)한 데이터 구조로, 여러 값을 저장할 수 있는 순서가 있는 컬렉션입니다. 리스트와 비슷하지만, 튜플은 한 번 생성하면 값을 변경할 수 없음

특징:

변경 불가능: 튜플의 요소는 생성 후 수정할 수 없음

순서 유지: 튜플은 요소의 순서를 유지하며, 인덱스를 통해 접근할 수 있음



## 데이터는 내것 : 튜플(Tuple)

```
fruits = (  
    "apple",  
    "banana",  
    "cherry"  
)
```

1. 접근은 인덱스로 : `fruits[0]`
2. 리스트에 비해서 폐쇄적
  - 데이터의 안정성, 상수값 등
3. 효율성
  - 대량의 데이터를 처리에 유리



## 데이터는 내것 : 딕셔너리

정의:

파이썬 딕셔너리는 변경 가능(**mutable**)한 데이터 타입으로, 키-값 쌍을 사용하여 데이터를 저장합니다.

특징:

키는 유일해야 함, 동일한 키를 두 번 사용할 수 없음

값은 변경 가능: 모든 자료형(리스트, 튜플, 다른 딕셔너리 등)



## 데이터는 내것 : 딕셔너리

```
person = {  
    "name": "Alice",  
    "age": 30,  
    "city": "Seoul"  
}
```

1. 값의 의미를 파악할 수 있음
2. 순서와 상관없이 원하는 데이터를 제어 가능함



## 데이터는 내것 : 예시

# List 예시

```
numbers = [10, 20, 30, 40]
```

```
numbers.append(50)
```

```
print(numbers) # 출력: [10, 20, 30, 40, 50]
```

# Tuple 예시

```
coordinates = (100, 200)
```

```
print(coordinates[0]) # 출력: 100
```

# Dictionary 예시

```
student = {'name': 'Alice', 'age': 30, 'grade': 'A'}
```

```
print(student['name']) # 출력: Alice
```

## 데이터는 내것 : 비교

특성	Dictionary	List	Tuple
정의	키-값 쌍으로 이루어진 변경 가능한 자료형	순서가 있는 변경 가능한 데이터의 목록	순서가 있는 변경 불가능한 데이터의 목록
변경가능 여부	변경 가능 (mutable)		변경 불가능 (immutable)
데이터저장	키-값 쌍 (key-value pair)	값의 목록 (elements)	
접근방법	키 (key)로 접근	인덱스 (index)로 접근 (0부터 시작)	
중복된 항목 허용 여부	키는 중복 불가, 값은 중복 허용	중복된 값 허용	

## 데이터는 내것 : 비교

특성	Dictionary	List	Tuple
선언 방법	<code>{key: value, ...}</code>	<code>[value1, value2, ...]</code>	<code>(value1, value2, ...)</code>
사용 예시	<code>{'name': 'Alice', 'age': 30}</code>	<code>[10, 20, 30, 40]</code>	<code>(1, 2, 3, 4)</code>
데이터 컨트롤	가능 ( <code>dict[key] = value</code> , <code>del dict[key]</code> )	가능 ( <code>list.append()</code> , <code>list.remove()</code> )	불가능 (새로운 튜플 생성 필요)
반복문 사용	가능 ( <code>for key in dict:</code> )	가능 ( <code>for item in list:</code> )	가능 ( <code>for item in tuple:</code> )
주요 사용 목적	키-값 매핑이 필요할 때	순서가 있는 데이터의 수정이 필요할 때	변경되지 않는 순서 있는 데이터를 저장할 때