

국민대학교 인공지능 연구실

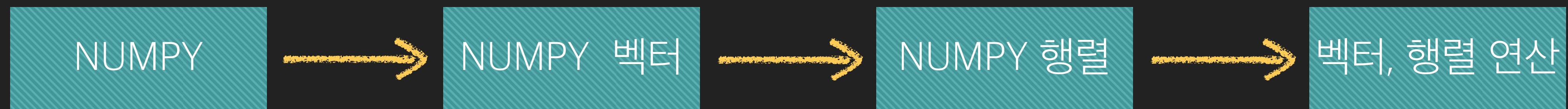
김인경

파이썬 기초 3주차

3주차 목차



다음 주에는?



WHILE 문

- ▶ Conditional loop - 특정한 조건이 만족될 때까지 반복
 - ▶ 지난 번 for문은 counting loop
 - ▶ 즉, 정해진 횟수만큼 반복하는 것
- ▶ 조건문이 참인 동안에 while 아래의 문장이 반복해서 수행

```
while <조건문>:  
    <코드>
```

NESTED LOOP

- ▶ for문을 여러개 사용할 수 있지만 2개 이상 사용하는 것을 권장하지 않음
 - ▶ 만약 2개가 넘어간다면...? → 코드를 잘못 짰을 경우가 높다!
- ▶ 작동 방법
 - ▶ for문의 경우: <시퀀스1>의 첫번째 타겟에서 <시퀀스2>가 모두 반복되면 이후 <시퀀스1>의 두번째 타겟으로 넘어가고 <시퀀스2>를 모두 반복하면서 <시퀀스1>의 타겟을 모두 반복!
 - ▶ while문의 경우: <조건문2>가 만족될 때까지 그 안에 있던 <코드1>을 반복 시행한 후 <조건문1>이 만족되면 종료
- ▶ 실습 자료 참고
- ▶ 예시
 - ▶ 구구단: 1단부터 9단까지 표현할 수 있는 방법은?
 - ▶ 삼각형 출력하기

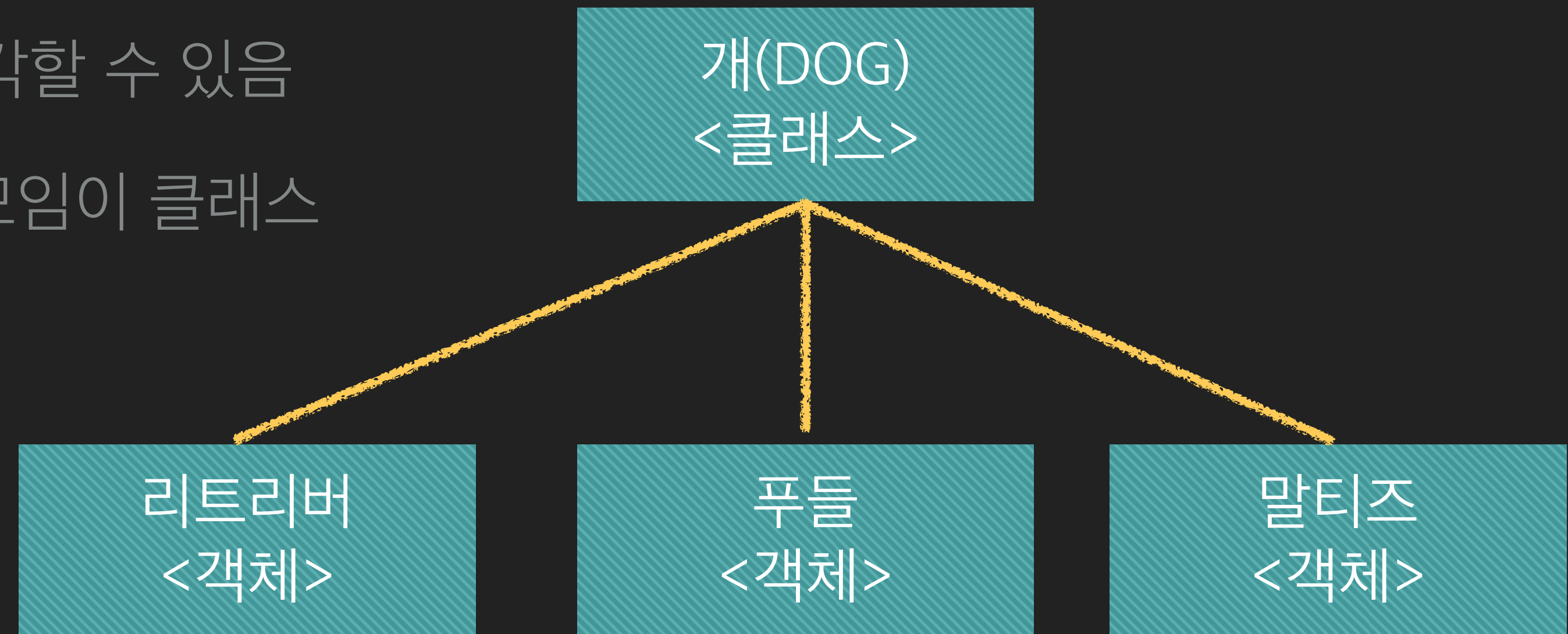
```
for <타겟1> in <시퀀스1>:  
    for <타겟2> in <시퀀스2>:  
        <코드1>  
    <코드2>
```

```
while <조건문1>:  
    while <조건문2>:  
        <코드1>  
    <코드2>
```

SHALLOW AND DEEP COPY

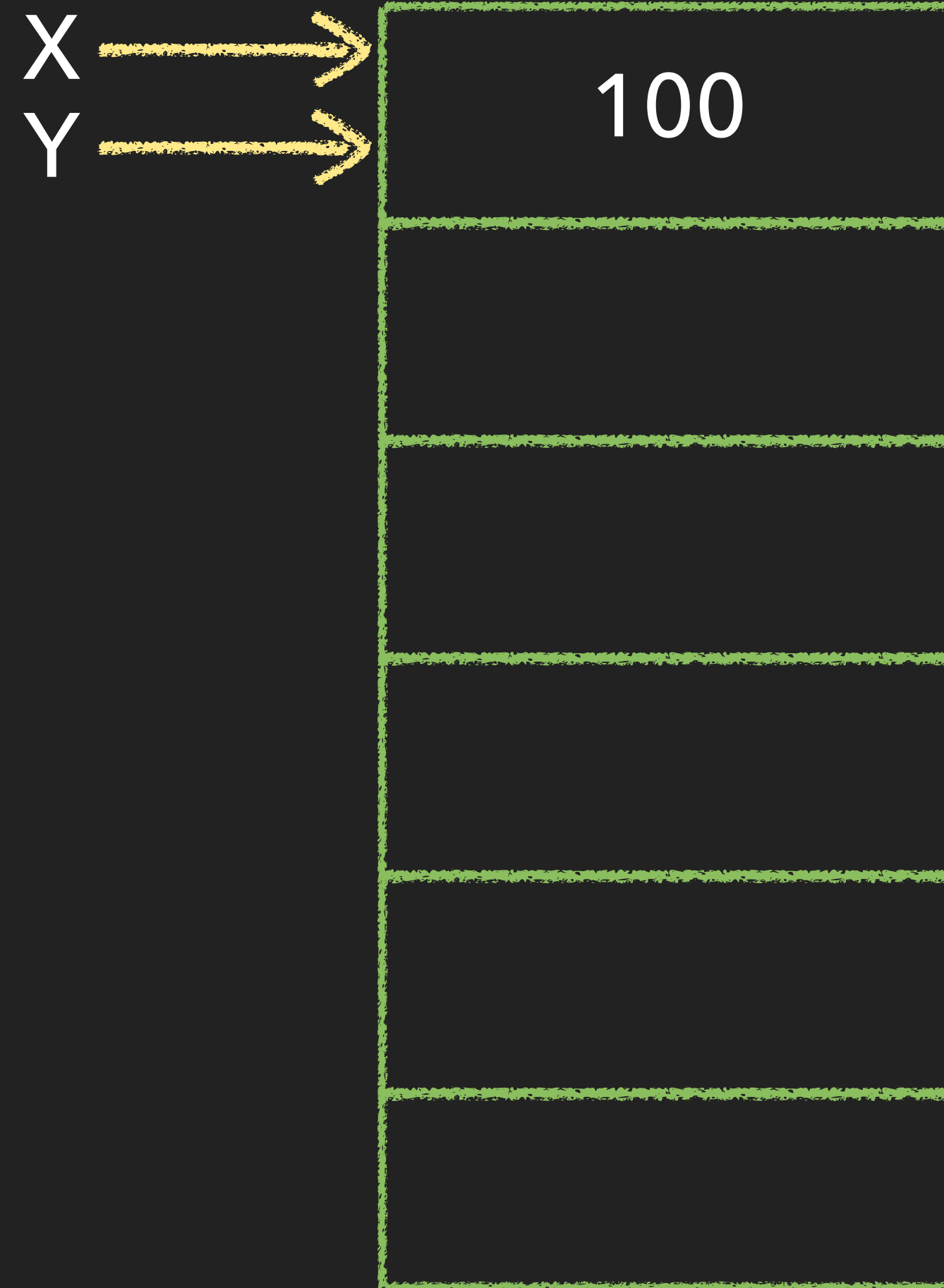
▶ 파이썬은 객체지향언어

- ▶ 컴퓨터 프로그램을 명령어의 목록이 아닌, 여러개의 독립된 단위 즉, 객체들의 모임으로 파악하는 것
- ▶ 각각의 객체는 메시지를 주고 받고 데이터를 처리할 수 있음
- ▶ 객체는 주변에 보이는 사물, 생명이라고 생각할 수 있음
- ▶ 객체의 중요한 속성을 공유하는 객체들의 모임이 클래스
- ▶ 객체에 대해서는 뒤에서 조금 더 자세히!



SHALLOW AND DEEP COPY

- ▶ 파이썬은 모든 것을 객체로 보고, 이 객체를 컴퓨터가 인식할 수 있는 identity가 존재함
 - ▶ id()함수를 사용해서 객체의 id를 확인할 수 있음
- ▶ 두 변수안에 같은 값이 있으면 데이터 자체가 복사되는 것이 아니라 주소가 복사 되는 것
- ▶ 데이터 자체를 복사해서 새로운 주소에 넣고 싶다면 깊은 복사를 해야함



SHALLOW AND DEEP COPY

- ▶ 깊은 복사를 하기 위해서는 copy 모듈을 사용
 - ▶ import copy
 - ▶ 깊은 복사를 하게 되면 주소를 공유하지 않고 새로운 주소에 같은 데이터를 새롭게 저장함
 - ▶ 때문에 id()를 통해서 출력한 주소값이 다름
- ▶ 리스트 같은 경우는 [:]를 사용해서 전체 복사를 해서 사용할 수 있음

alphabet

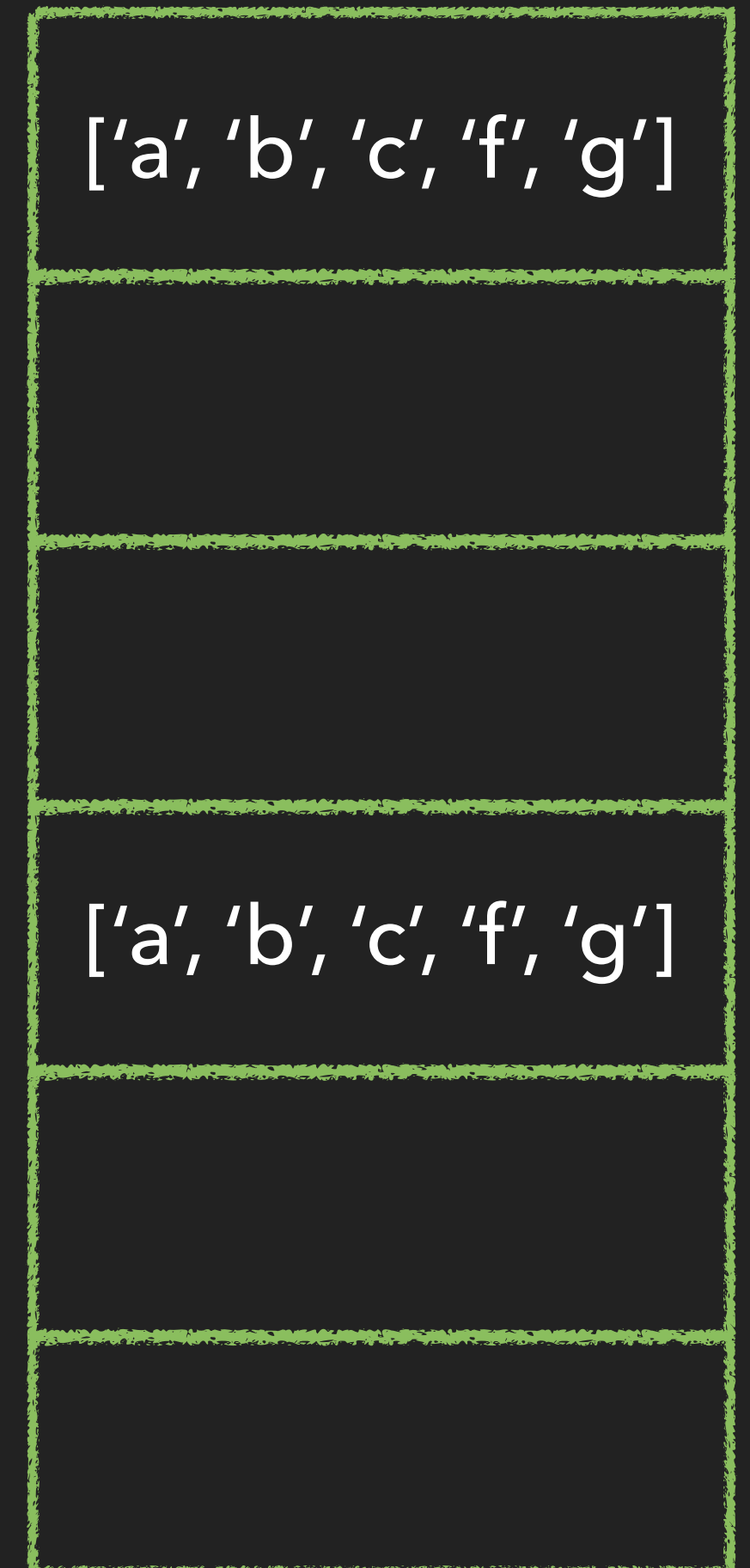


['a', 'b', 'c', 'f', 'g']

alphabet_copy



['a', 'b', 'c', 'f', 'g']



함수

- ▶ <매개 변수>는 없을 수 있음
- ▶ 리턴 값은 없을 수 있음
- ▶ 가변 매개변수를 사용한 함수

```
def <함수이름> (<매개변수1>, <매개변수2>, ...):  
    <수행할 문장1>  
    <수행할 문장2>
```

- ▶ 파이썬에서는 매개변수를 원하는만큼 받을 수 있는 함수를 만들 수 있음
- ▶ 가변 매개변수는 하나만 사용할 수 있으며 가변 매개변수 뒤에는 일반 매개변수가 올 수 없음
 - ▶ 가변 매개변수 함수를 정의할 때는 *를 가변 매개변수 이름 앞에 붙임
- ▶ 파이썬에서는 데이터 타입을 지정해주지 않아도 되기 때문에 매개변수를 5개까지 사용할 것을 권장 —> 너무 많으면 가독성 저하

함수

```
def <함수이름> (<매개변수1>, <매개변수2>, ...):  
    <수행할 문장1>  
    <수행할 문장2>
```

```
def <함수이름> (<매개변수1>, <매개변수2>, ..., *<가변 매개변수>):  
    <수행할 문장1>  
    <수행할 문장2>
```

함수

▶ 기본 매개변수

- ▶ 인자가 전달되지 않을 경우, 값을 정의하는 매개변수
- ▶ 기본 매개변수 뒤에는 일반 매개변수가 올 수 없음

▶ 함수 호출(call)

- ▶ 함수는 호출될 때, 독자적 메모리 공간이 할당됨
- ▶ 함수가 종료될 때, 할당되었던 메모리 공간은 소멸됨 → 메모리 공간을 사용하고 있던 모든 변수 및, 인자도 동시에 소멸됨
- ▶ 따라서 함수가 끝나고 나서도 해당 값을 유지하고 싶다면 리턴해주어야 함

객체(OBJECT)와 클래스(CLASS)

▶ 객체란?

- ▶ 어떤 대상에 대해 설명 가능한 속성과 그 대상을 가지고 할 수 있는 행동을 모아놓은 것
- ▶ 객체 = 속성 + 메소드
 - ▶ 속성 = 대상에 대한 특징 혹은 대상에 대해 알고 있는 정보
 - ▶ 메소드 = 대상의 행동 또는 대상을 가지고 할 수 있는 일

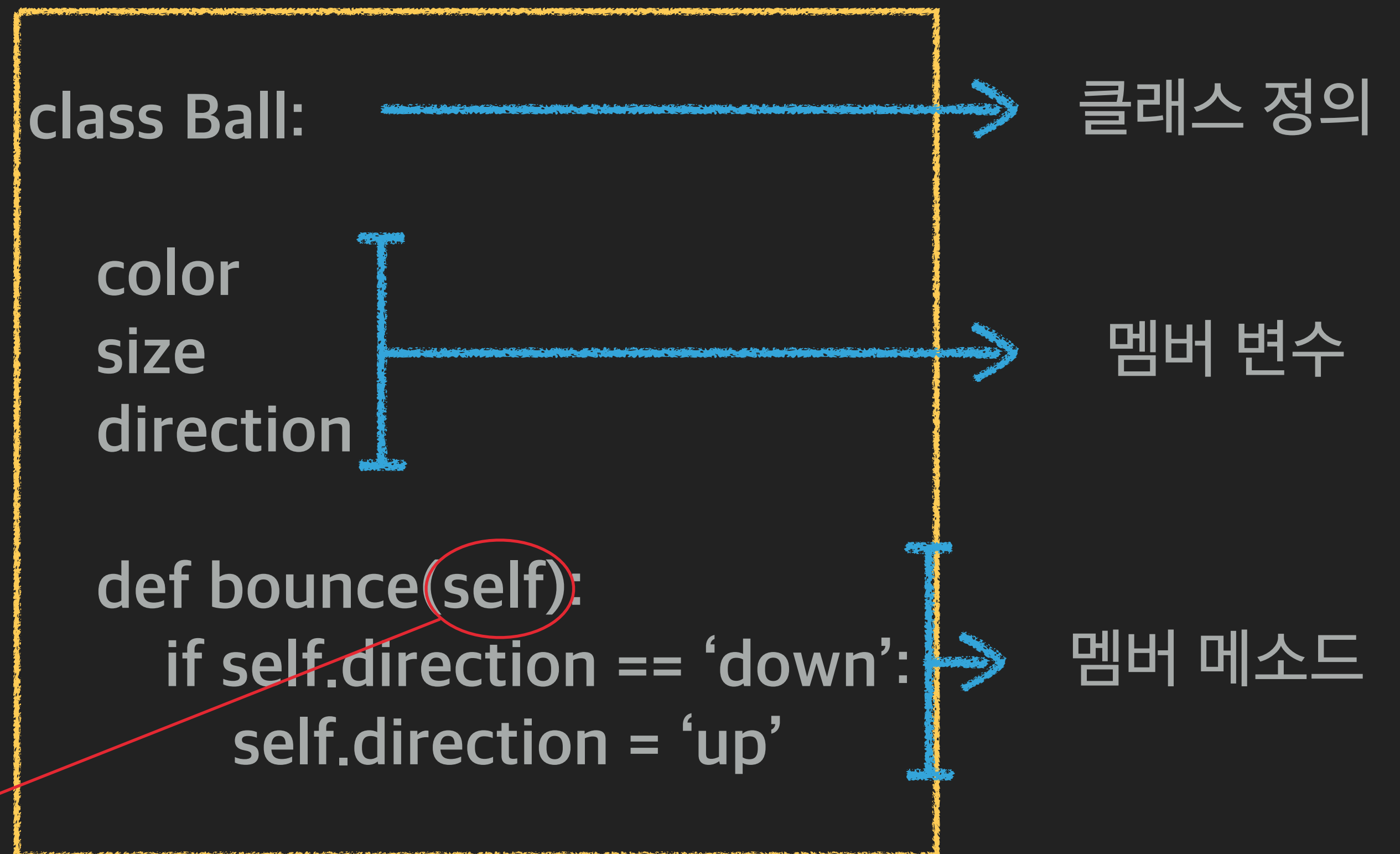
▶ 클래스란?

- ▶ 추상적이지만 어떤 요소들에 대한 공통적이고 일반적인 '정의'를 할 수 있는 개념
- ▶ 객체의 속성을 저장하기 위한 변수들과 객체의 행동을 수행하기 위한 메소드들이 같은 이름 공간에 묶인 집합체

객체(OBJECT)와 클래스(CLASS)

▶ 클래스를 비유하자면

- ▶ 과자 틀 —> 클래스
- ▶ 과자 틀에 의해 만들어진 과자 —> 객체
- ▶ 클래스는 크게 클래스 정의, 멤버 변수, 멤버 메소드 3가지 부분으로 구성됨
- ▶ 멤버 변수와 메소드는 반드시 필요한 것은 아님



self는 파이썬만의 독특한 변수로 클래스 내에서 정의되는 함수는 무조건 첫 번째 인자로 self를 사용해야 함

객체(OBJECT)와 클래스(CLASS)

- ▶ 인스턴스(Instance)
 - ▶ 클래스를 사용하여 만든 실제 객체
 - ▶ 클래스는 인스턴스를 만드는 하나의 틀
 - ▶ 인스턴스는 생성이 완료된 직후 원본 클래스와 동일한 데이터와 함수를 가짐
- ▶ 인스턴스를 생성한 후 인스턴스의 데이터를 변경하고, 메소드를 호출하고, 멤버 변수를 추가하거나 삭제할 수 있음
- ▶ 인스턴스를 통해서 자신을 생성한 클래스를 참조할 수 있음
 - ▶ 내장 속성 ‘`__class__`’를 사용

객체(OBJECT)와 클래스(CLASS)

▶ self

- ▶ 현재의 인스턴스 객체를 가르키는 기능을 하는 지시어
 - ▶ 즉, 메소드가 어떤 인스턴스에서 호출되었는지 알려주는 인스턴스 참조자

▶ 생성자 메소드

- ▶ 인스턴스가 생성될 때 자동으로 호출
- ▶ 인스턴스가 생성되었을 때 초기화 작업을 하기 위함
- ▶ 특수 메소드 ‘__init__()’으로 정의
- ▶ 객체를 생성하면서 속성값을 설정하기 위한 방법
- ▶ 생성자를 통해 인스턴스 생성 시 초기화할 멤버 변수 값을 인자로 전달

객체(OBJECT)와 클래스(CLASS)

▶ 인스턴스 초기화

- ▶ 객체를 생성하면서 속성값을 설정하기 위한 방법
- ▶ 생성자를 통해 인스턴스 생성 시, 초기화 할 멤버 변수 값을 인자로 전달

▶ 소멸자 메소드

- ▶ 인스턴스의 레퍼런스 카운트가 0이 될때 자동으로 호출
- ▶ 특수 메소드 ‘__del__()’로 정의
- ▶ 파이썬의 경우 자동으로 실행되기 때문에 필요할 때만 사용

▶ 객체 출력 메소드

- ▶ 객체를 출력할 때 무엇을 보여줄지 알려줌
- ▶ 특수 메소드 ‘__str__()’로 정의

객체(OBJECT)와 클래스(CLASS)

▶ 다형성(Polymorphism)

- ▶ 이름이 같은 메소드를 서로 다른 클래스마다 가질 수 있는 것
- ▶ 동일한 메소드에 대하여 구체적인 인스턴스마다 다른 동작을 수행하는 특징

▶ 상속(Inheritance)

- ▶ 클래스를 정의할 때, 다른 클래스에 있는 공통된 속성 및 메소드를 매번 처음부터 만들지 않고 사용하는 것
- ▶ 기존 클래스가 라이브러리 형태로 제공되거나 수정이 허용되지 않는 경우도 있기 때문에 상속을 사용함

참고 자료

- ▶ <https://wikidocs.net/28>
- ▶ <http://pythonstudy.xyz/python/article/16-%ED%95%A8%EC%88%98>