

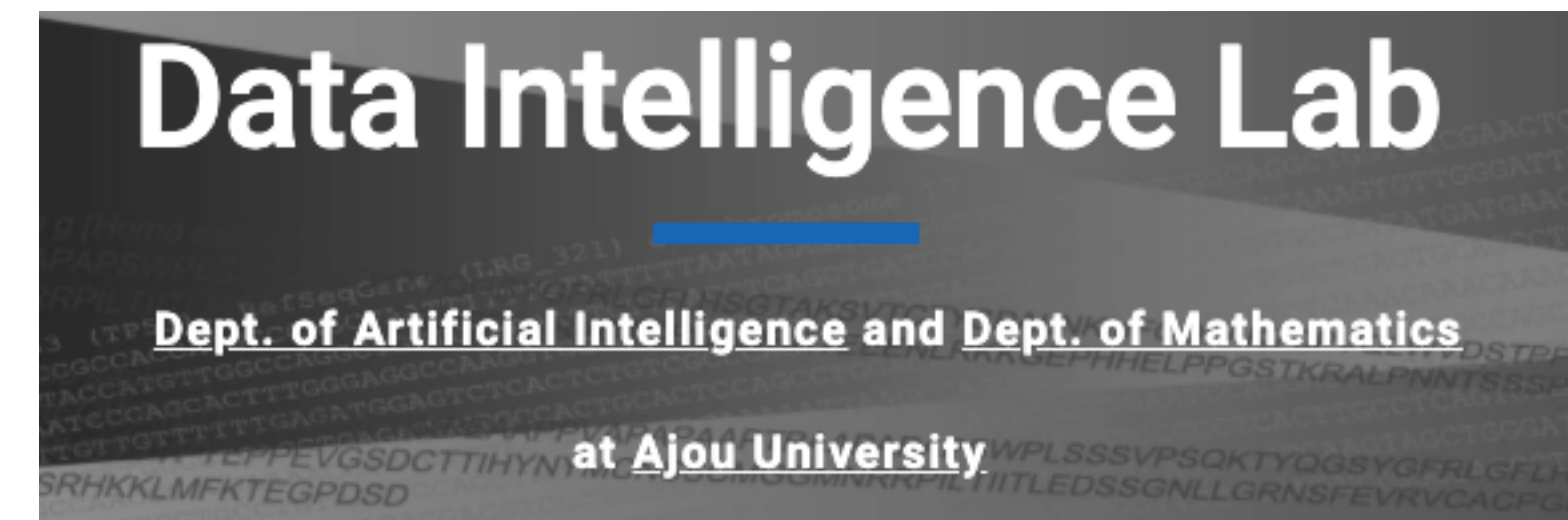
Python의 메모리 구조 및 할당

선우찬

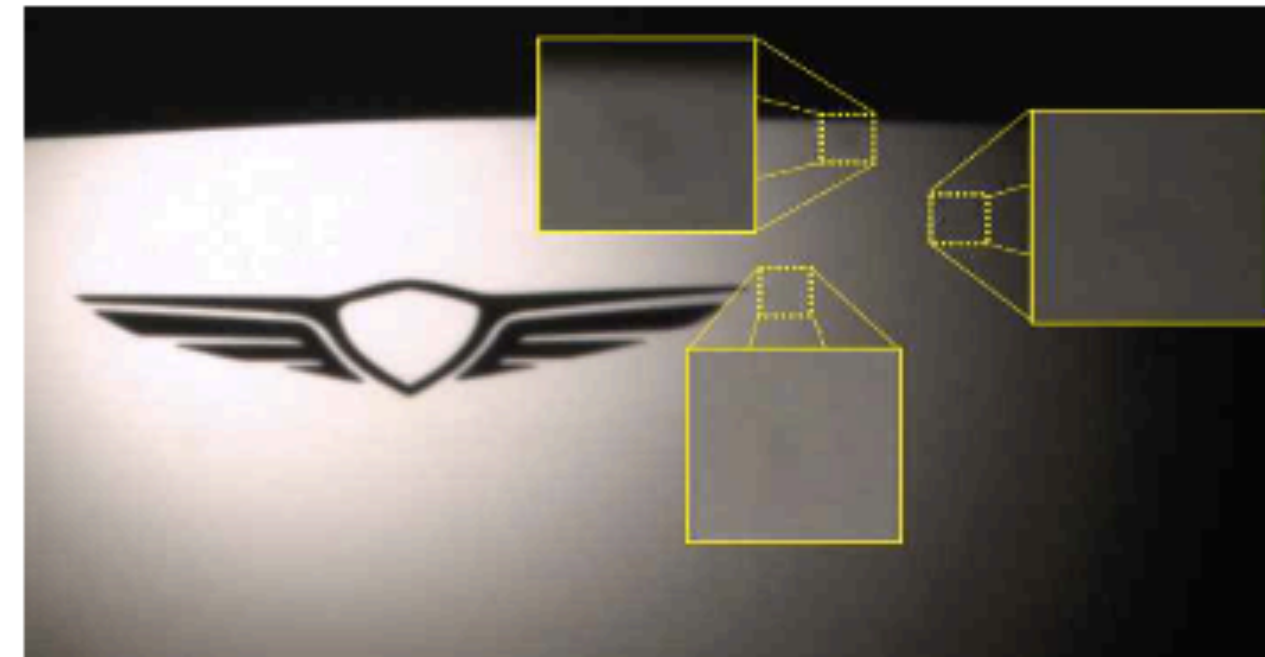
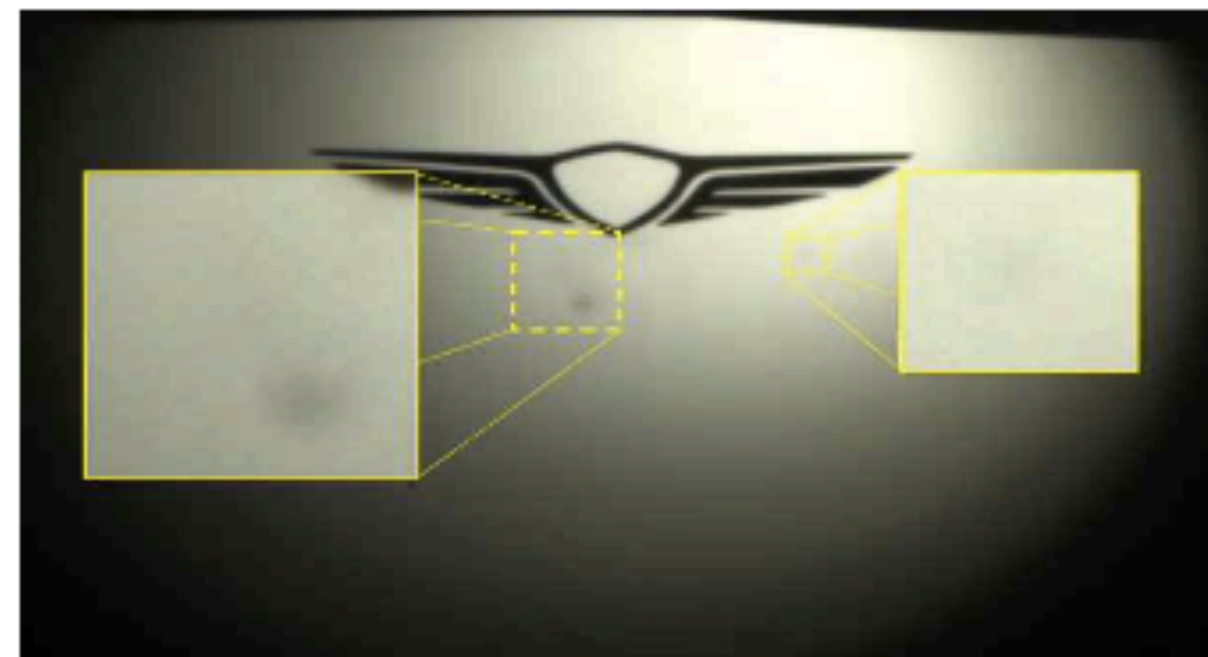
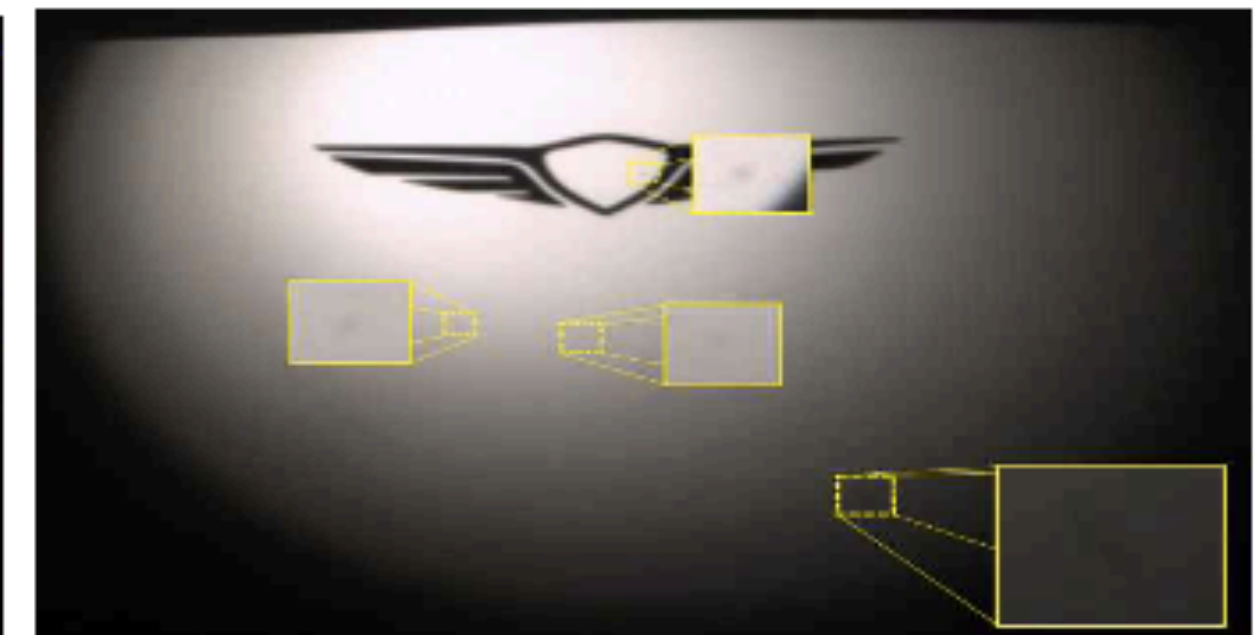
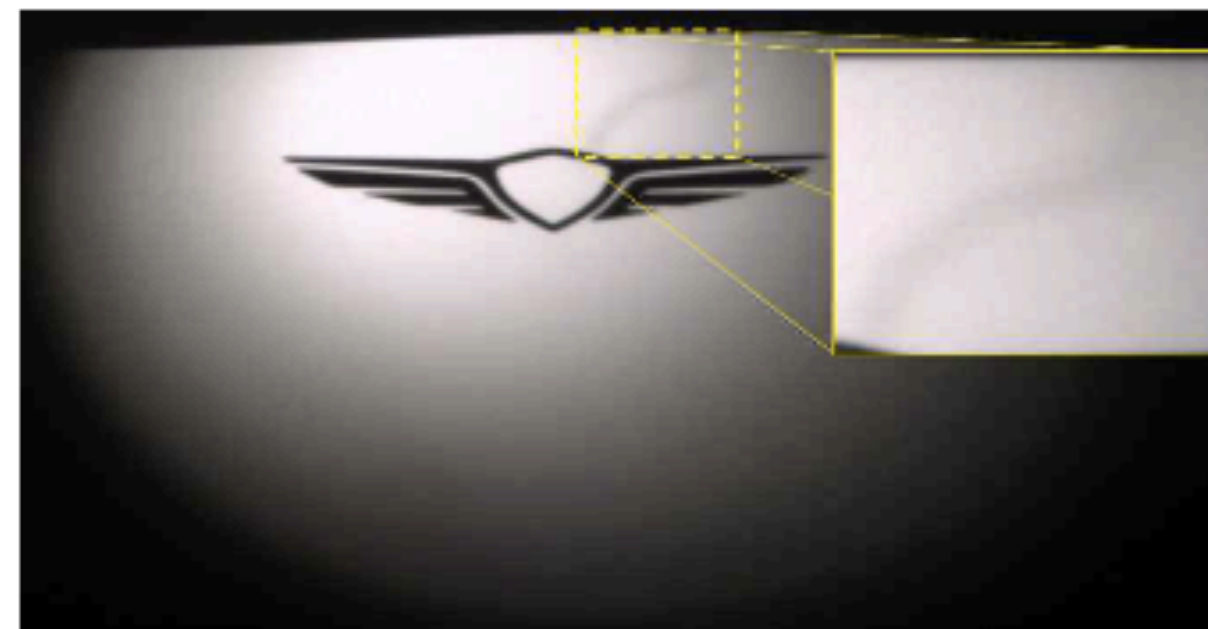
목차

- 근황
- Everything is object in Python
- Python의 메모리 구조
- Python의 메모리 할당 과정

근황



- DI Lab 인턴
- 비정형 객체 탐지 시스템 개발



근황

SALLY

영양성분 기반 샐러드 추천 서비스

Sally야, 나한테 맞는 샐러드 부탁해!

- Sally 논문

근황

- Kaggle 스터디 운영
- 24명, 23명 분반
- 서울대 석박, 카이스트 석사, 도쿄대 박사, 직장인



근황

- LG AI 해커톤 경진대회

농업 환경 변화에 따른 작물 병해 진단 AI 경진대회

주제 | 작물 사진과 생장 환경 데이터를 이용한
작물 병해 진단 AI 모델 개발

대상 | 대한민국 거주 중인 학생, 일반인 누구나

상금 | 총 상금 1,000만원
상위팀 입사 희망자 LG AI 인재풀 등록
상위팀 입사 희망자 LG 계열사 입사 추천
(최상위 10명 서류 전형 면제)



주최  LG AI Research

주관  DAICON

근황

- 프로그래머스 인공지능 데브코스

programmersschool (Beta)

대시보드

강의

커리큘럼

1주차 KDT 시작하기

2주차 인공지능 수학

3주차 Python 데이터 다루기

4주차 Flask 서버와 EDA 프로젝트

5주차 Django와 Monthly Proj...

6주차 인공지능과 기계학습 개요

7주차 ML_Basic

8주차 ML_Basic II

9주차 신경망 기초

10주차 CNN & RNN

11주차 SQL과 데이터분석

12주차 Spark


13주차 NLP

14주차 NLP II 와 Visual Recog...

15주차 GAN

16주차 추천시스템 구현

[3기] 프로그래머스 인공지능 데브코스 A반



선우찬 님, 오늘은 코딩으로 불태워봐요 🔥

최근 수업 이어서 하기

출석 확인

코스 시작일

2021-12-06

코스 종료일

2022-11-05

마지막 로그인

2022-01-05 15:58:03

나의 수강 현황

알림

더보기 >

단위 기간 학습 진행률

2차 단위 기간 2022-01-06 ~ 2022-02-05

66%

학습 진행률

지금까지 얼마나 학습을 완료했는지 확인해볼 수 있어요.

70%

전체 학습자 47%

충분히 잘 하고 있어요!

꾸준한 학습이 있다면 처음 목표했던 바를 반드시 이룰 수 있을 거예요!
끝까지 완료하고 성취감도 얻어가세요.

출석

2022-01-20(목요일)에 대한 출석 처리되었습니다.

2022-01-20 15:06:44

출석

2022-01-19(수요일)에 대한 출석 처리되었습니다.

2022-01-19 12:20:38

출석

2022-01-17(월요일)에 대한 출석 처리되었습니다.

2022-01-17 10:40:28

출석

2022-01-14(금요일)에 대한 출석 처리되었습니다.

2022-01-14 15:50:31

근황

- CNN 공부

! 쉽게 Model을 생성할 수 있는 Keras의
처음부터 Sequential Model을 사용해
안 뒤에 작고 간편하게

CNN - Functional API

AI/CNN · 2022. 1. 21. 11:59

목차 Sequential Model Sequential Model vs Functional API Functional API Functional API의 개요 연속적으로 이어지는 Function의 예 Functional API의 필요성 Keras Functiona...

Read More

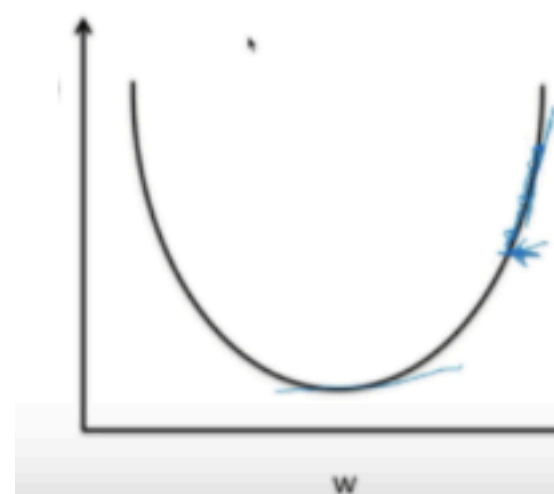


CNN - Image Array의 이해

AI/CNN · 2022. 1. 20. 17:58

Array(Tensor)의 차원 ■ 1차원: iris 샘플 하나 ■ 2차원: iris 샘플 여러 개, 명암 영상 한 장 ■ 3차원: 명암 영상 여러 장, 컬러 영상 한 장 ■ 4차원: 컬러 영상 여러 장, 컬러 동영상 하나 ■ 5...

Read More



CNN - Training Epoch, Batch Size, Learning Rate

AI/CNN · 2022. 1. 20. 16:58

머신러닝 모델이 적합한 결과를 내기 위해서는 여러번의 최적화 과정을 거친다. 다루어야 할 데이터가 많기도 하고, 메모리가 부족하기도 하기 때문에 한번의 계산으로 최적화된 값을 찾는 ...

Read More

Everything is object in Python

- class : 추상적인 것 / object : 구체적인 것



속성(attribute)
이름 국적 나이...

행동(method)
먹다 자다 일어나다..

- '인간' 이라는 타입(class)



홍길동, 임꺽정 등 실제로 존재하
는 객체(object)

Everything is object in Python

- class : 추상적인 것 / object : 구체적인 것
- 파이썬에서 존재하는 타입은 class로 정의된다.
- class의 예 : list, tuple 등
- 파이썬에서 우리가 list를 만들 때, 사실 class를 이용해서 object를 만드는 것이다.

Everything is object in Python

- 파이썬에서 존재하는 타입은 class로 정의 된다.
- 파이썬으로 우리가 list를 만들 때는, 사실 class를 사용해서 object를 만드는 것

List

속성(attribute)
Items

행동(method)
append extend ...

• 'List' 이라는 타입(class)

[1, 2, 3]

['Hello', 'World']

실제로 존재하는 객체(object)

Everything is object in Python

- C에서 `x = 10` 으로 변수를 할당하면, 메모리에 바로 해당 값이 저장
- Python에서 `x = 10` 으로 변수를 할당하면, integer object가 만들어져서
- 변수 `x`가 int object인 10을 가리킨다.

```
x = 10
print(type(x))

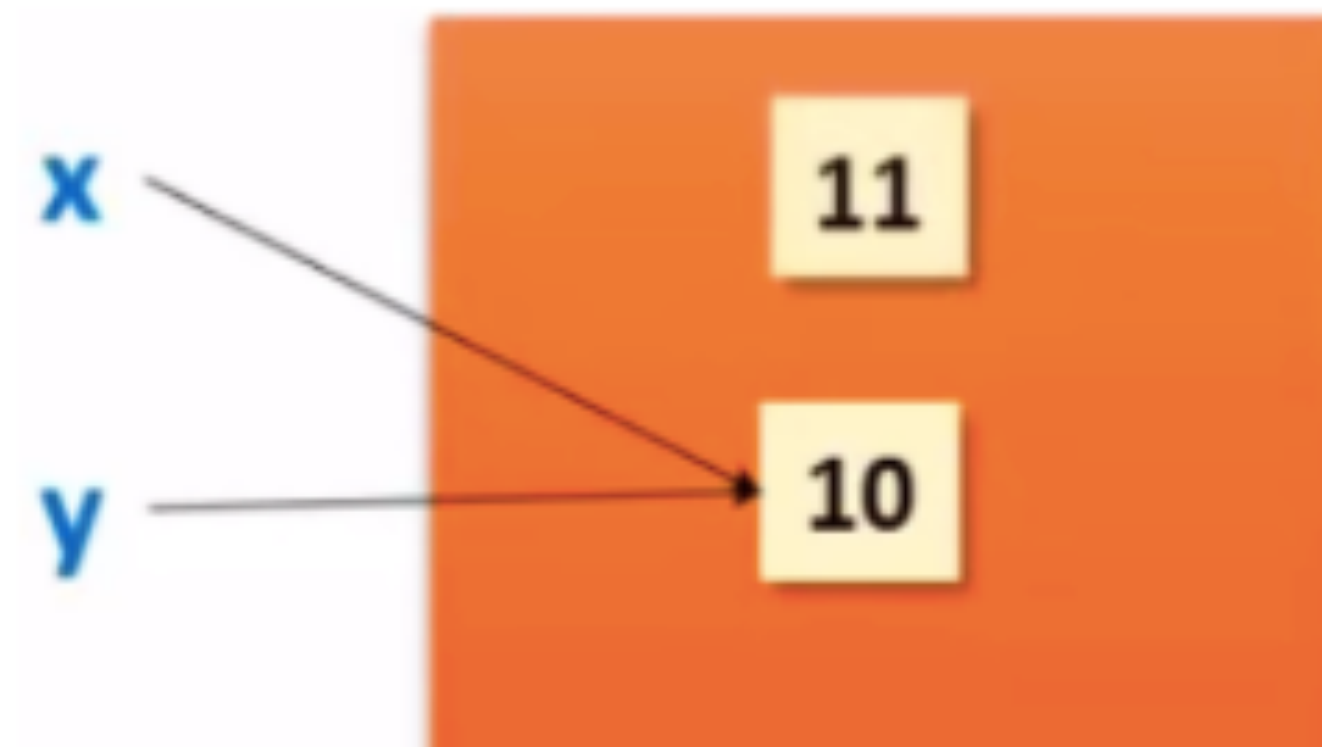
>> <class 'int'>
```

Everything is object in Python

- `y = x` 라고 하면,
- `x`가 가리키는 int object 10을,
- `y`도 그냥 가리키기만 한다.
- 따라서 `x`와 `y`는 10이라는 int object를 가리키는 것이다.

```
y = x
if (id(x)==id(y)):
    print("x 와 y 는 같은 객체를 가리킨다")

>> x와 y는 같은 객체를 가리킨다
```



Everything is object in Python

- `x = x + 1`은
- 11 (10+1) 이라는 새로운 int object를 생성한다.
- 그럼 이제 x는 새로 만들어진 int object를 가리킨다.

```
x = x + 1
if (id(x) != id(y)):
    print("x와 y는 다른 객체를 가리킨다")

>> x와 y는 다른 객체를 가리킨다.
```

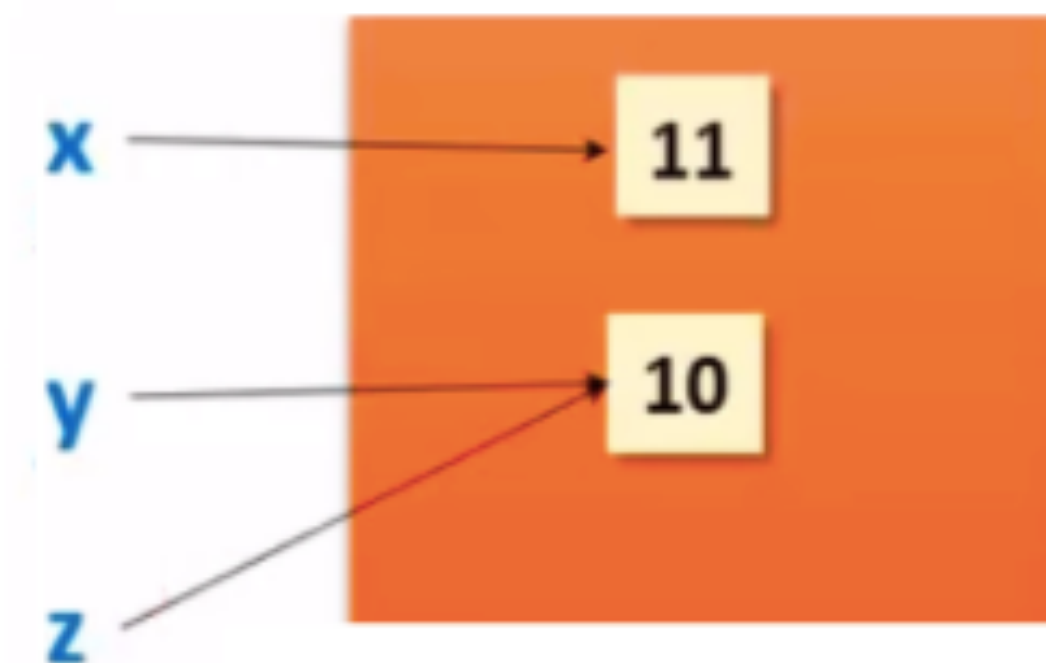

Everything is object in Python

- `z = 10` 은
- 10 이라는 int object를 가리키는데
- int object 10은 이미 생성 되어 있으니 object를 만들 필요는 없고
- 그냥 int object 10이 `z`를 가리키기만 한다.

```
z = 10

if (id(y)==id(z)):
    print("y와 z는 같은 memory를 가리킨다")
else:
    print("y와 z는 다른 object를 가리킨다")

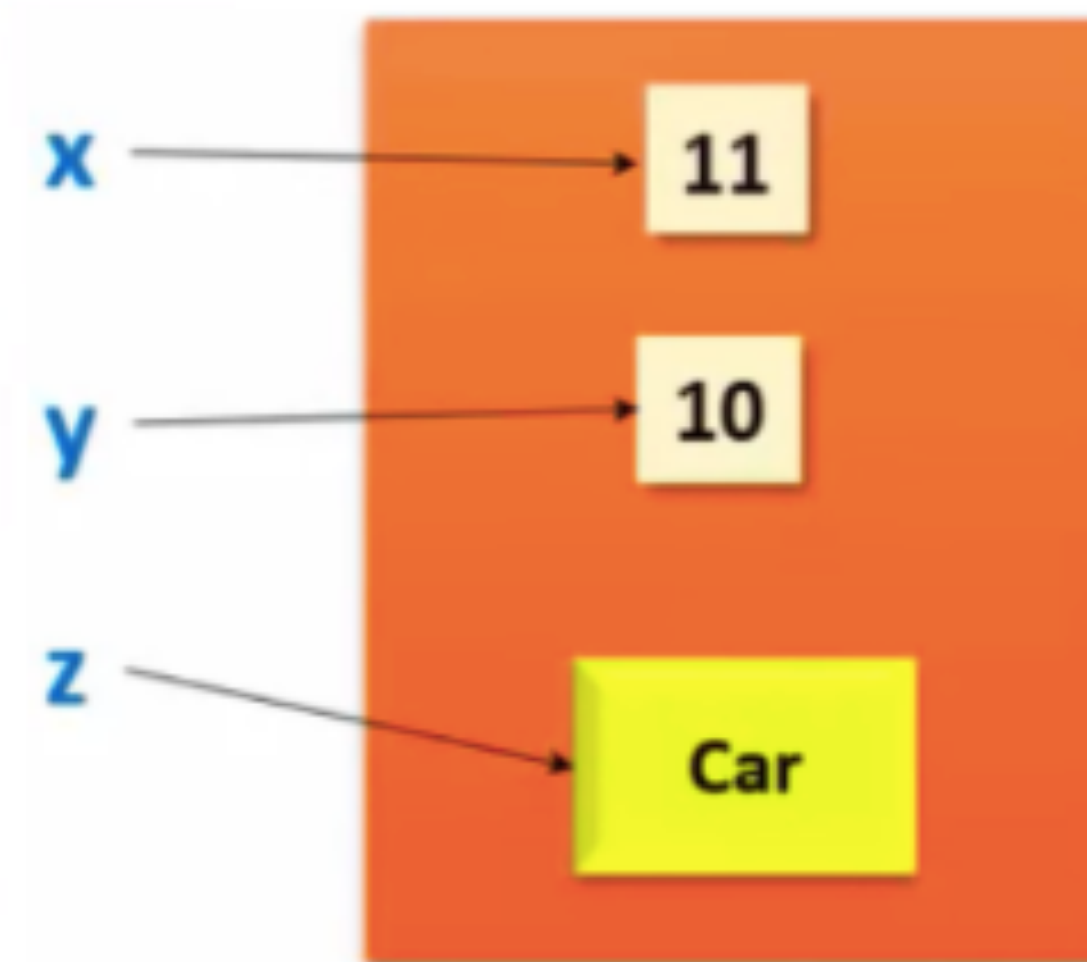
>> y와 z는 같은 메모리를 가리킨다
```



Everything is object in Python

- `z = Car()` 에서
- `Car` 라는 object가 생성되고
- `z`는 `Car` object를 가리킨다.
- 따라서 `z`의 `type`은 `Car` 이다

```
z = Car()  
print(type(z))  
  
>> <class '__main__.Car'>
```



Python의 메모리 구조

- Text 영역(= 코드 영역) : 실행할 프로그램의 코드가 저장
- Data 영역 : 전역변수, 정적변수 저장
프로그램이 시작과 함께 할당, 프로그램이 종료되면 소멸
- Stack 영역 : 지역변수, 매개변수 저장
함수 호출과 함께 할당, 함수 호출이 완료되면 소멸
- Heap 영역 : 사용자의 동적 할당으로 생성되는 공간

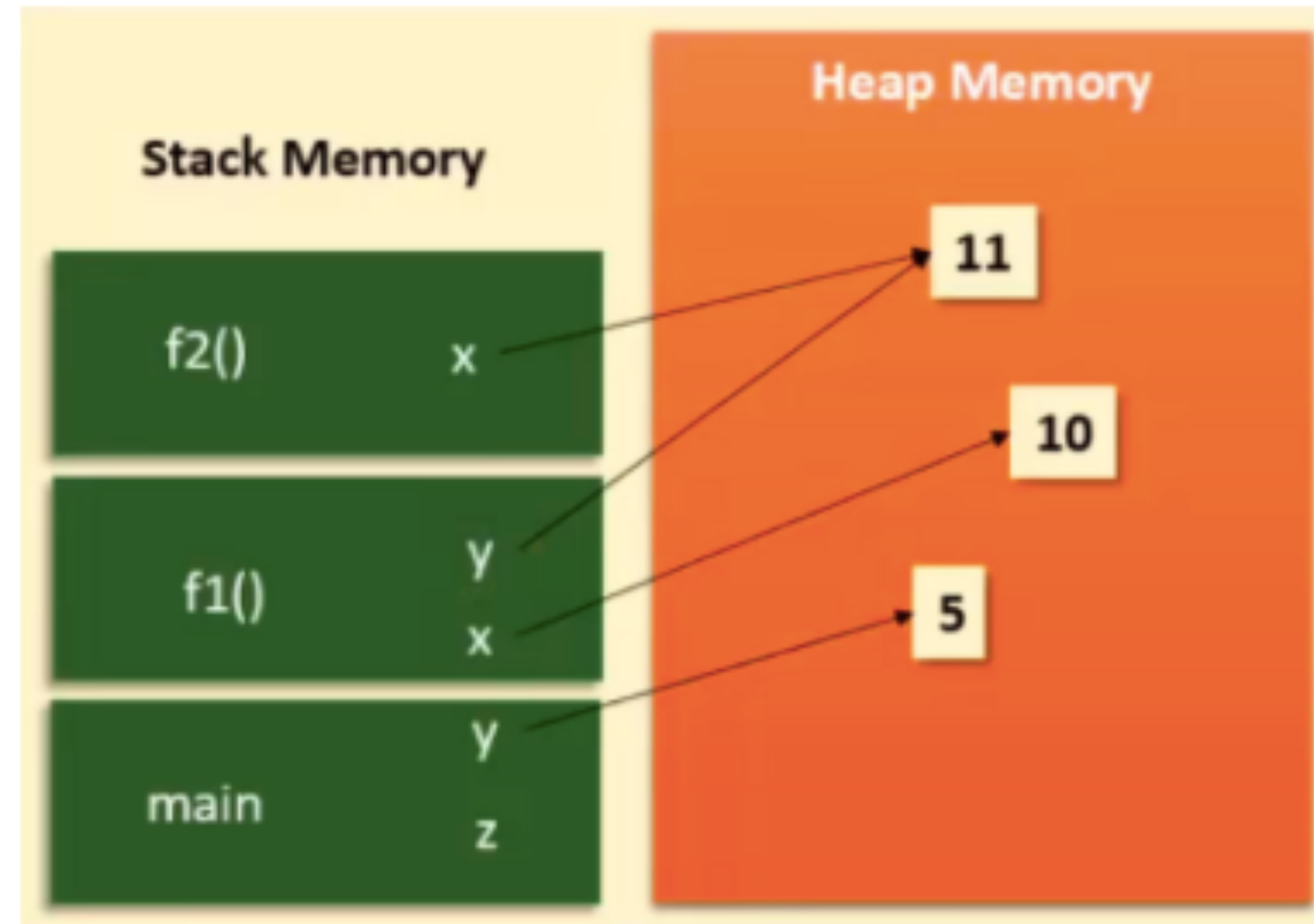


Python에서의 메모리 할당 과정

```
def f2(x):  
    x = x + 1  
    return x
```

```
def f1(x):  
    x = x * 2  
    y = f2(x)  
    return y
```

```
#main  
y = 5  
z = f1(y)
```

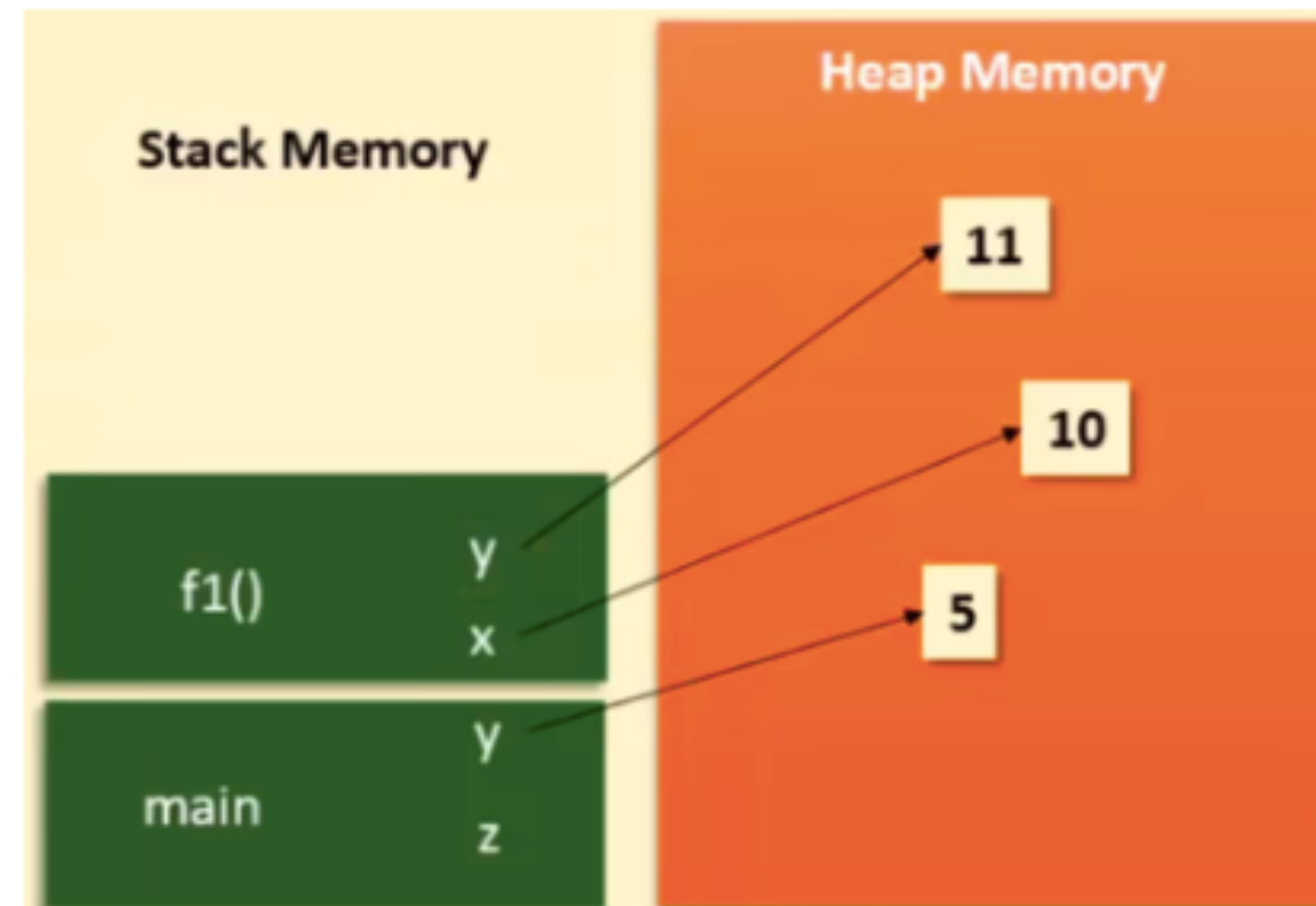


메모리 할당 해제

```
def f2(x):  
    x = x + 1  
    return x
```

```
def f1(x):  
    x = x * 2  
    y = f2(x)  
    return y
```

```
#main  
y = 5  
z = f1(y)
```

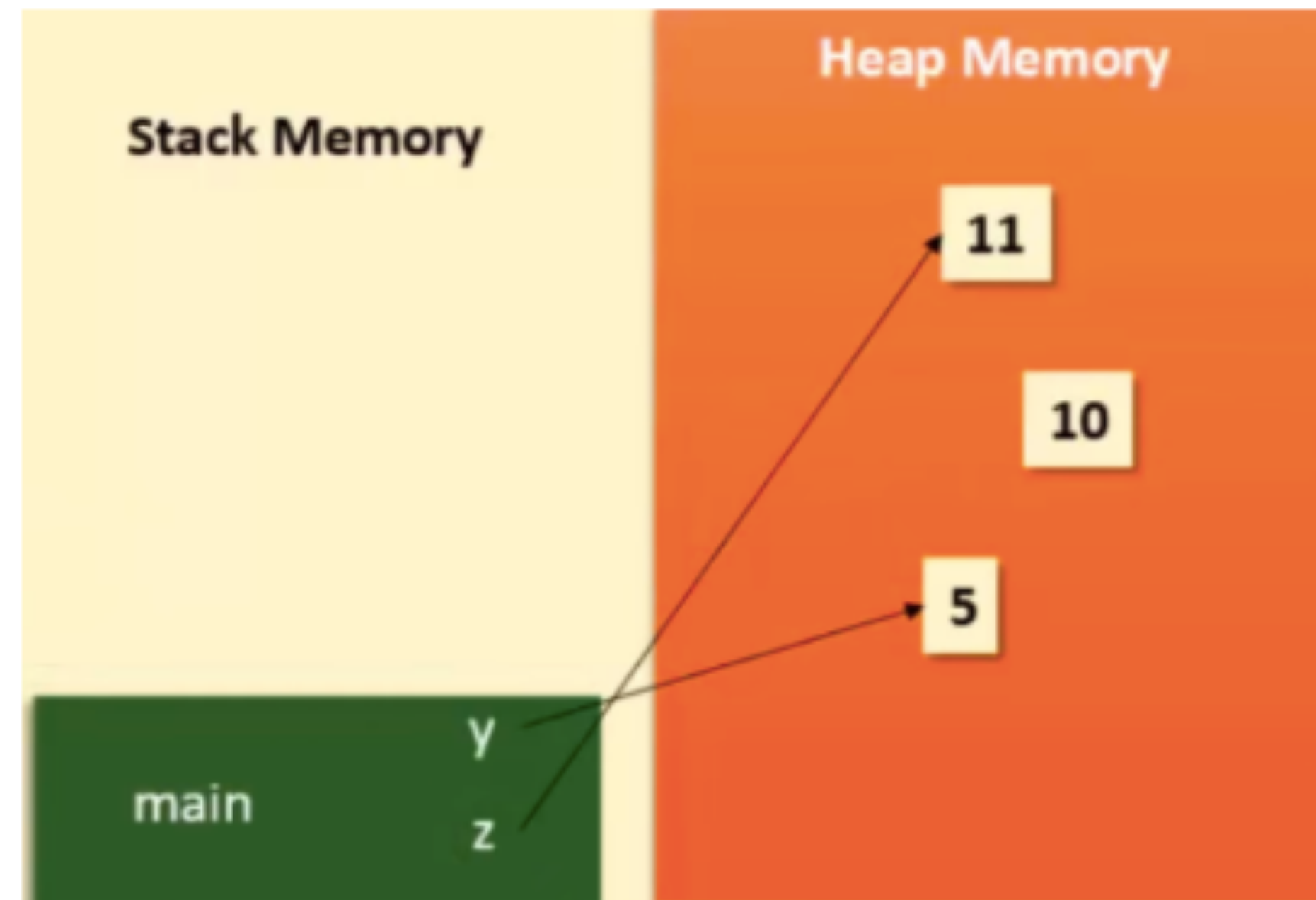


메모리 할당 해제

```
def f2(x):  
    x = x + 1  
    return x
```

```
def f1(x):  
    x = x * 2  
    y = f2(x)  
    return y
```

```
#main  
y = 5  
z = f1(y)
```



Reference Counting

Reference

<https://www.youtube.com/watch?v=arxWaw-E8QQ>



Memory Allocation and Management in Python - simplified tutorial f...



나중에 시청...



공유




Reference Counting



Memory Management in Python

- Anuj Jajoo -

다음에서 보기:  YouTube

감사합니다.