

## C에서 변수가 저장되는 방식: 변수 = 메모리

C에서 특정한 값이 변수에 저장되는 방식은 간단하다. 예를 들어, 아래와 같은 C코드가 작성되어 있다고 하자.

```
// C 코드
int a = 5;
int b = a;
```

코드를 실행하는 세부 과정은 다음과 같다.

1. int 4바이트 만큼의 메모리를 할당하고 a라는 이름을 붙인다 (컴퓨터는 a를 방금 할당한 메모리의 주소로 기억한다.)
2. 변수 a 자리에 5를 저장한다. (0x5의 형태로 4바이트만큼 사용할 것이다.)
3. int 4바이트 만큼의 메모리를 다시 할당하고 b라는 이름을 붙인다.
4. 변수 a에 있는 값을 꺼내서 (당연히 5일 것이다) 변수 b에 저장한다.

이렇듯 C코드에서의 변수는 메모리를 직접적으로 나타내고, 변수의 값은 할당한 메모리에 저장되어 있는 실제 값을 나타낸다. 그런데 파이썬에서는 변수를 다른 방식으로 관리한다고 한다!

## 파이썬에서 변수가 저장되는 방식: Everything is Object

파이썬에서는 모든 것 (부울, 정수, 실수, 문자열, 데이터 구조, 함수, 프로그램)이 객체 (Object)로 구현되어 있다. ... 파이썬 변수의 핵심을 살펴보자. 변수는 단지 이름일 뿐이다. 할당한다는 의미는 값을 복사하는 것이 아니다. 데이터가 담긴 객체에 그냥 이름을 붙이는 것이다. 그 이름은 객체 자신에 포함되는 것이라기보다는 객체의 참조다. 이름을 포스트잇처럼 생각하자. (Introducing Python p.42-43)

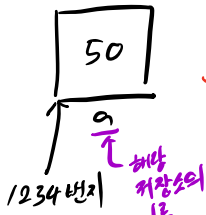
간단히 말해 5라고 하는 값이 사실은 객체였다는 것이다. 정수는 정수 클래스의 객체로, 문자열은 문자열 클래스의 객체로, 모든 변수와 값들이 사실은 객체로 다루어지고 있다는 것이다. C에서는 변수 a에 5라는 값을 저장하고, 변수 b에 변수 a의 값을 대입하면, 5라는 값이 "복사"되었다. 그런데 파이썬에서는 변수가 자신만을 위한 메모리를 가지는 것이 아니라 5라는 값을 가진 객체를 가리키도록 되어있다는 것이다.

```
# 파이썬 코드
a = 5
b = a
```

같은 로직의 위 코드를 실행하면 5라는 값을 가지는 정수 객체가 생기고, 변수 a와 변수 b는 단지 정수 5 객체의 주소를 참조하게 되는 것이다. 이 과정에 대해서 더 자세히 설명하고 있는 블로그의 설명을 들여다보자.

< C >

① int a = 50 ;

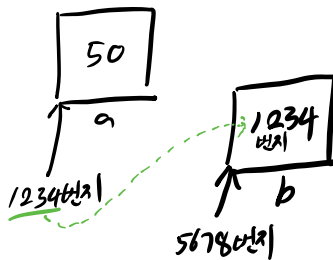


~~X~~ a = 50

~~X~~ a = 1234번지  
b = 1234번지

~~X~~ b = 5678번지

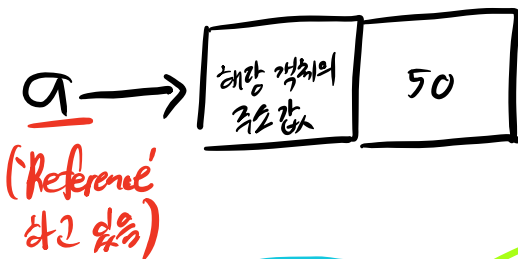
② int \*b = &a  
↑ 포인터!!



< X C에선, 변수에 선언한 타입과  
데이터의 타입이 일치해야  
한다. >

< 파이썬 >

a = 50



이 때문에,  
파이썬의 변수는  
C의 '포인터'에 해당하고,  
파이썬에서 변수를 선언할  
때 자료형을 선언할  
필요가 없다.

~~X~~ 파이썬으로 어떤 프로그램을

실행시키면, 파이썬이 주름되는  
것이 아니라 내부에 존재하는

'C'가 주름되는 것이다.

그러므로 'a = 50' 이라고 입력된  
것을 실행시키면, 내부에 존재하는

'C'에서 '50'이 어떤 변수에  
저장되고 (이때 변수명은 알지 못함)

이라는 변수 (파이썬이 존재하는 변수)

에는 '50'이 생성되어진

RAM상의 주소값이 저장된다.

