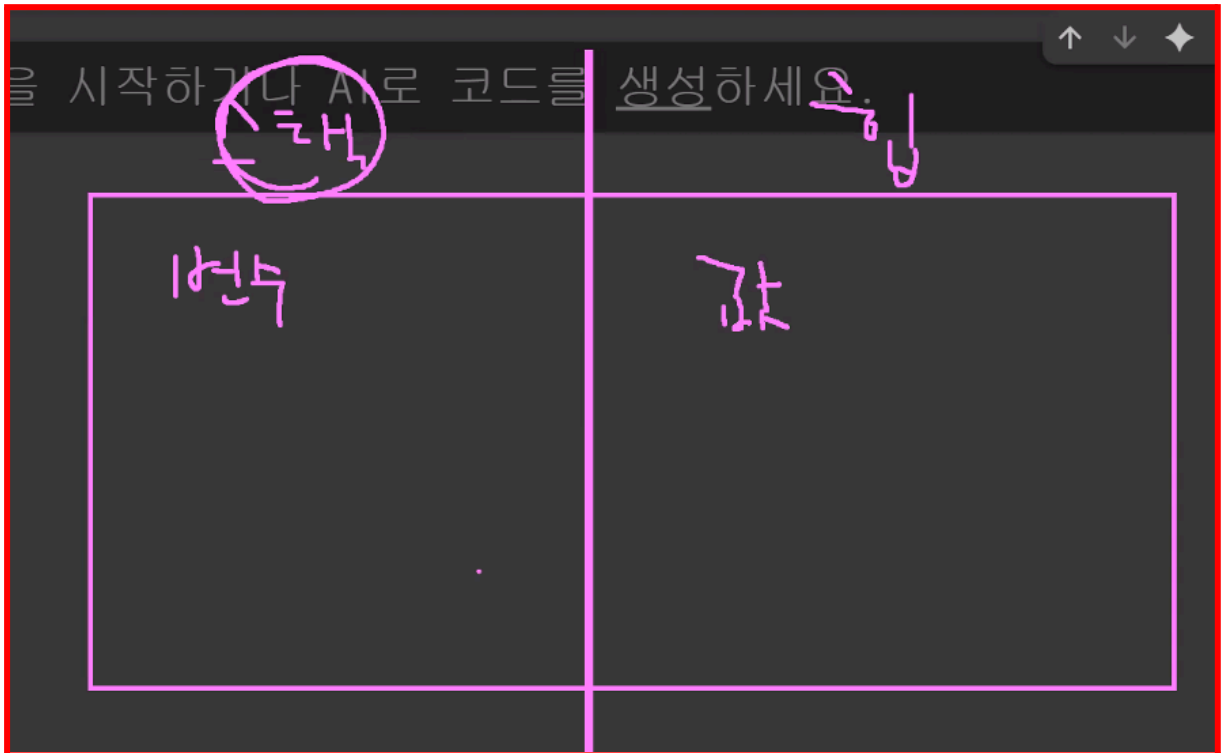


Day 2 - Variables in Python

파이썬의 변수 메모리 저장 개념:



파이썬의 메모리에는 스택과 힙이 있다.

- 스택 ⇒ 변수 를 저장
- 힙 ⇒ 값 을 저장

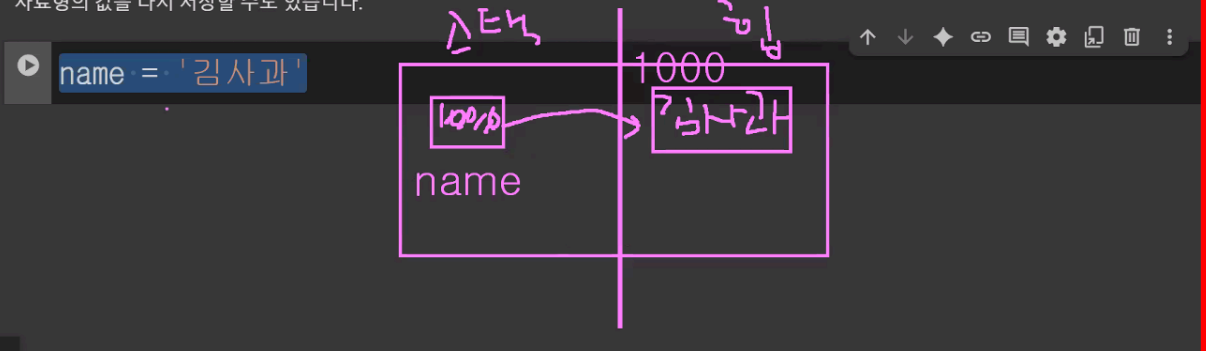
스택 안에 있는 변수는 값을 가르킨다 (변수 → 값)

- 파이썬에서 변수는 데이터를 저장하기 위한 이름표와 같은 역할을 합니다. 변수에 값을 할당하면, 그 값은 컴퓨터 메모리에 저장되고 변수 이름은 그 값을 가리키게 됩니다. 파이썬은 동적 타입 언어이기 때문에, 변수를 선언할 때 자료형을 명시하지 않아도 되며, 한 번 정해진 변수에 다른 자료형의 값을 다시 저장할 수도 있습니다.

예시)

3. 변수

파이썬에서 변수는 데이터를 저장하기 위한 이름표와 같은 역할을 합니다. 변수에 값을 할당하면, 그 값은 컴퓨터 메모리에 저장되고 변수 이름은 그 값을 가리키게 됩니다. 파이썬은 동적 타입 언어이기 때문에, 변수를 선언할 때 자료형을 명시하지 않아도 되며, 한 번 정해진 변수에 다른 자료형의 값을 다시 저장할 수도 있습니다.

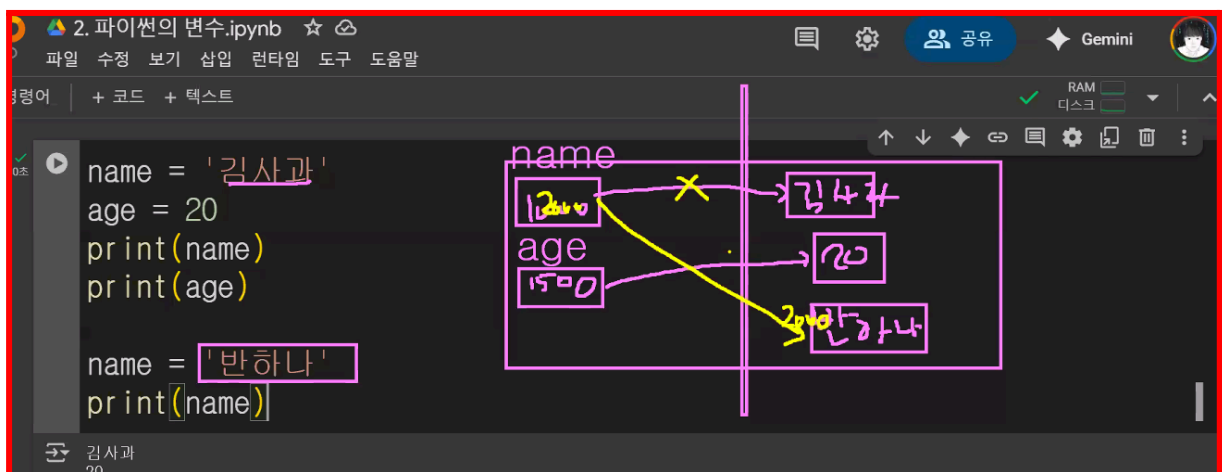


- 스택에 name이라는 변수가 저장되어있다.
- 힙에 '김사과'라는 값이 저장되어 있고 주소지는 1000번이다
- name 변수는 1000번지 주소를 가르키고 있다.

스택값이 힙을 가리키고 있을 경우, 힙에 있는 값은 절대로 지워지지 않는다.
하지만 가리키고 있지 않을 경우, 힙에 있는 값은 garbage collector에 의해 지워진다.

이름이 '변수' variable 인 이유:

- 값이 변경될 수 있다.
- 메모리에 저장되는 값을 가리키고 있지만 변할 수 있다.



- 이런식으로 변수의 값(가리키는 주소)가 변할 수 있다.

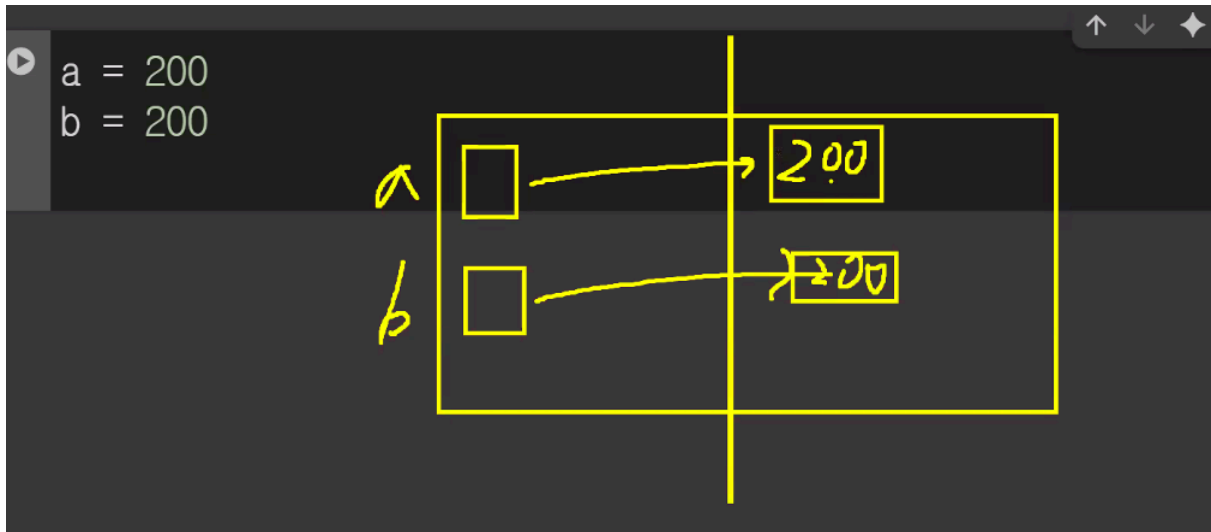
[스택] [힙]
name(메모리 주소: 1000) → 김사과
age (메모리 주소: 1500) → 20
name(메모리 주소: 2000) → 반하나

*여기서 김사과와 name의 관계는 끊기고 김사과는 stack에 의해 가르키고 있지 않기 때문에 나중에 garbage collector에 의해 김사과는 메모리에서 지워진다.

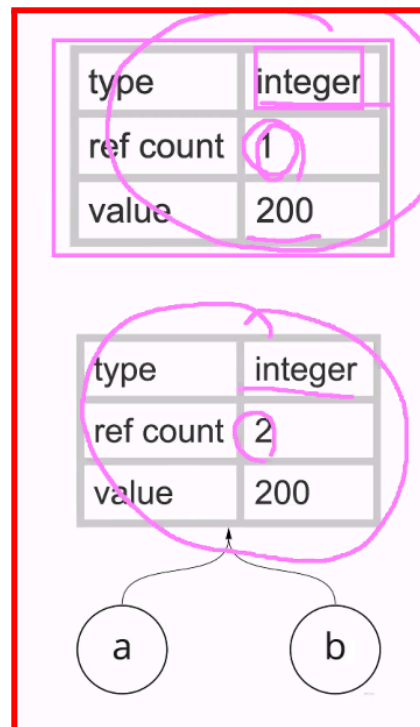
리터럴 공유(객체 캐싱)

- 리터럴 공유는 동일한 값을 가진 불변 객체가 여러 곳에서 사용될 때, 새로운 객체를 만들지 않고 기존 객체를 재사용하는 방식입니다. 이를 통해 파이썬은 메모리 사용을 최적화할 수 있습니다.

파이썬에서 같은 값을 가리킬때 보통 이렇게 생각한다:

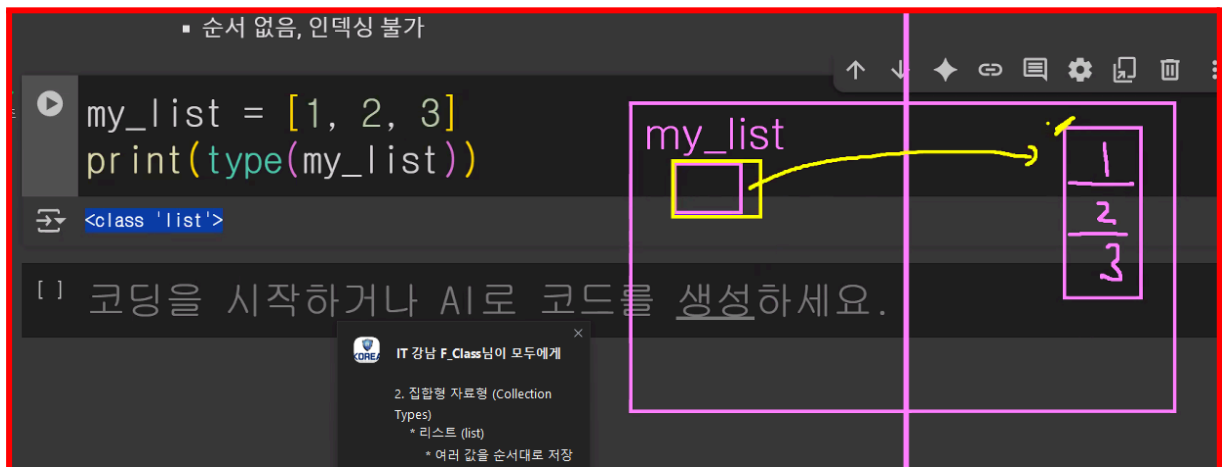


그러나 a, b는 같은 200을 가르킨다. (메모리를 사용을 최적화 하기 위해서):



a, b 변수를 만들고, 둘다 같은 200이라는 값을 메모리에 올렸을때
메모리를 아끼기 위해서 ref count를 2개로 만들고 같은 메모리 값을 가르킨다.
(공유값으로 가지고 있다)

리스트 자료형의 메모리 저장 방식:



- my_list 라는 변수를 Stack에 저장하고 값을리스트 형태로 heap에 저장한다,
- my_list는 index 0의 값을 가리키고 있다

del 함수를 사용해서 변수를 삭제할때:

The screenshot shows a Jupyter Notebook titled "2. 파이썬의 변수.ipynb". The code cell contains:

```
i = str(h)
j = 3.14
k = str(j)
print(type(i), type(k))
```

The output is `<class 'str'> <class 'str'>`. A diagram on the right illustrates memory management. A box labeled "stack" contains a variable `ref` pointing to the value `3.14`. The value `3.14` is also shown in the heap memory area.

7. 변수 삭제하기

변수를 삭제하기 위해 `del` 문을 사용할 수 있습니다. `del` 문은 지정된 변수를 제거하고 해당 메모리 공간을 해제합니다. 변수가 삭제되면 해당 이름으로 변수에 더 이상 접근할 수 없습니다.

j (stack)는 3.14(heap) 이라는 값을 가르키고 있다.

This screenshot is identical to the previous one, but the `ref` variable in the stack diagram is now crossed out with a yellow 'X', indicating its removal.

7. 변수 삭제하기

변수를 삭제하기 위해 `del` 문을 사용할 수 있습니다. `del` 문은 지정된 변수를 제거하고 해당 메모리 공간을 해제합니다. 변수가 삭제되면 해당 이름으로 변수에 더 이상 접근할 수 없습니다.

코딩을 시작하거나 AI로 코드를 생성하세요.

del을 이용해 j를 지우면 변수값 자체를 지워버린다.

This screenshot is identical to the previous one, but the value `3.14` in the heap memory area is now circled in yellow, indicating it is no longer referenced.

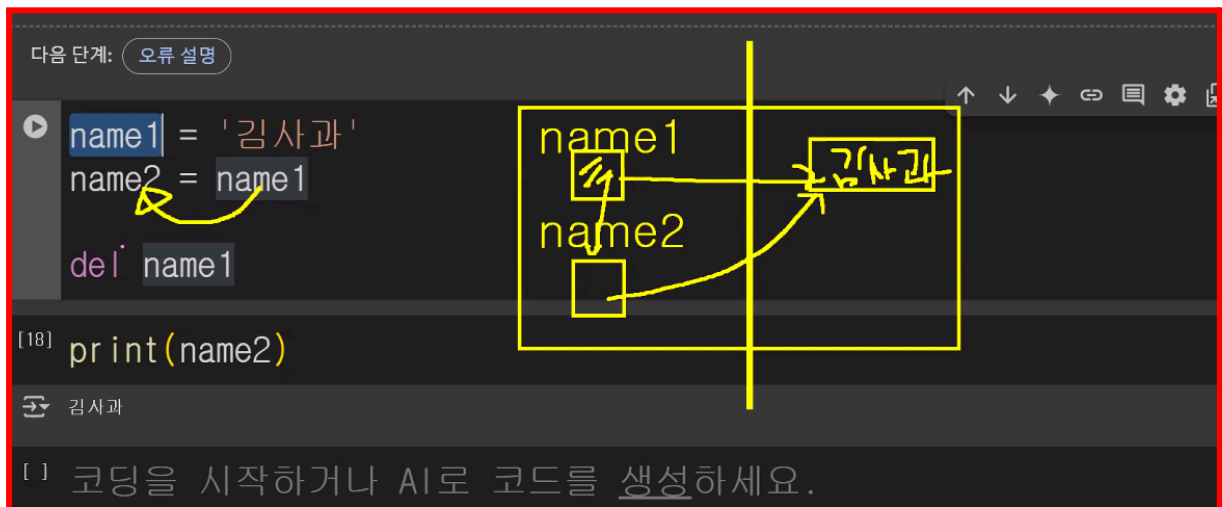
7. 변수 삭제하기

변수를 삭제하기 위해 `del` 문을 사용할 수 있습니다. `del` 문은 지정된 변수를 제거하고 해당 메모리 공간을 해제합니다. 변수가 삭제되면 해당 이름으로 변수에 더 이상 접근할 수 없습니다.

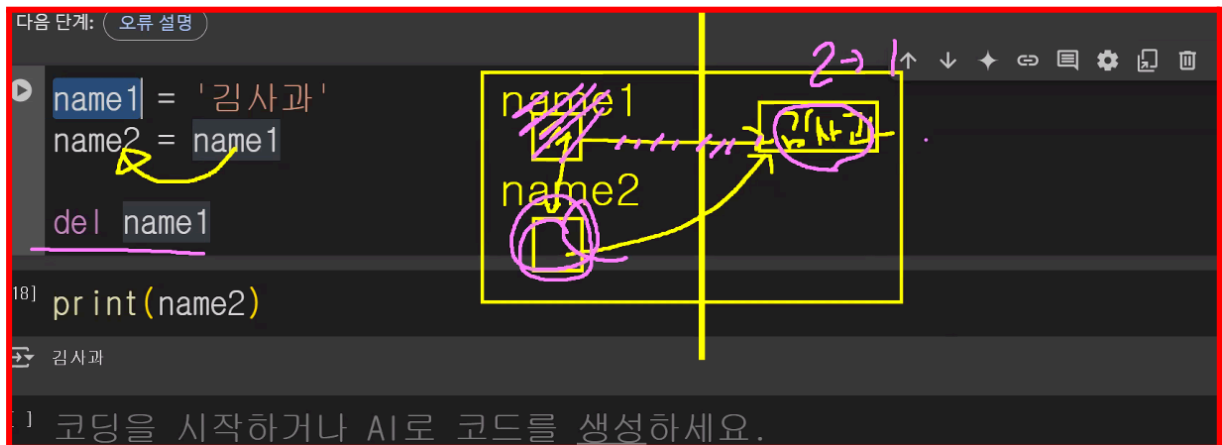
코딩을 시작하거나 AI로 코드를 생성하세요.

따라서 heap에 있는 3.14라는 값은 stack에 있는 변수에 의해 refer 당하고 있지 않으므로 garbage collector가 지워준다.

다른 변수가 같은 값을 가르킬때:

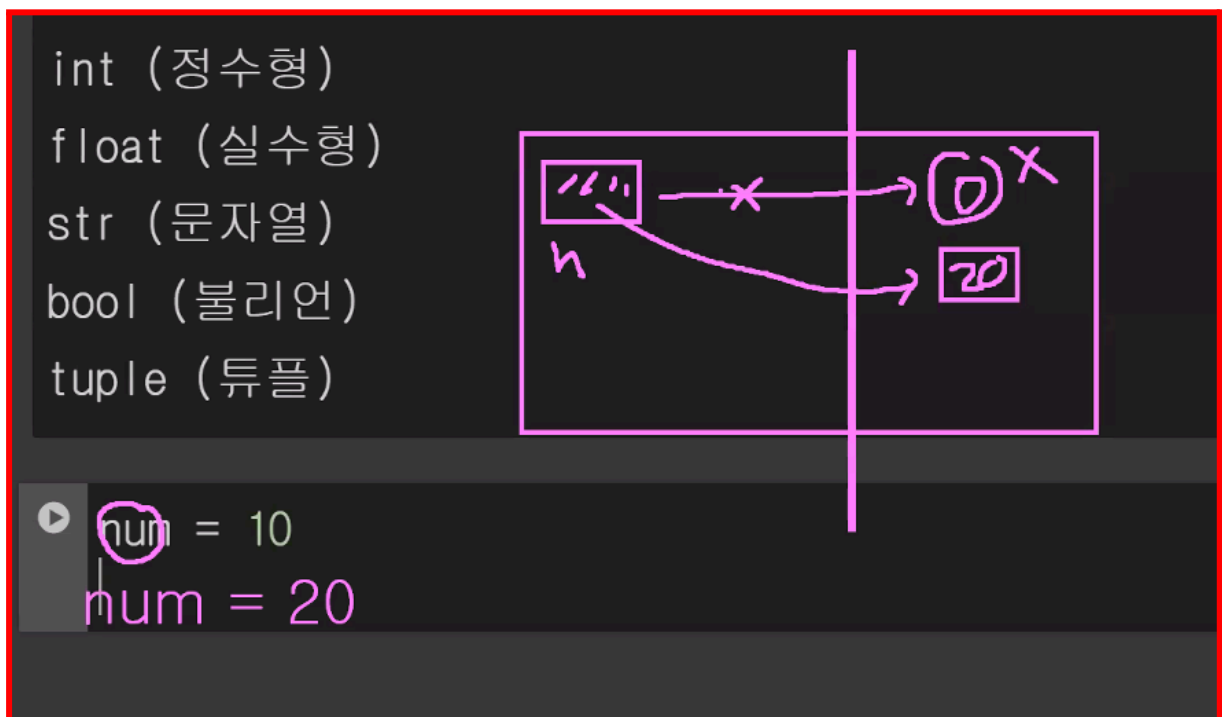


- 이런식으로 같은 heap 메모리에 있는 값('김사과')을 가리키고 있는 name1, name2 변수가 있다.



- 김사과의 포인터가 2에서 1로 지워진다.
- name1이 지워져도 name2는 김사과를 계속 가리키고 있기때문에 사용가능하다.

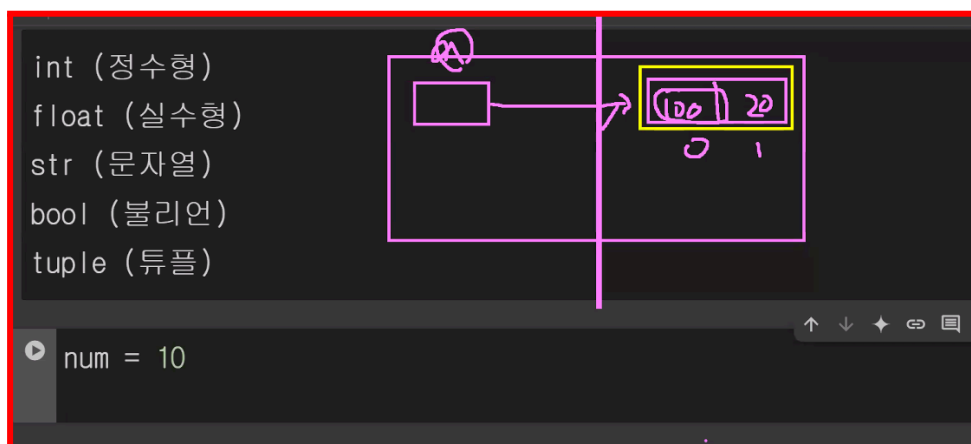
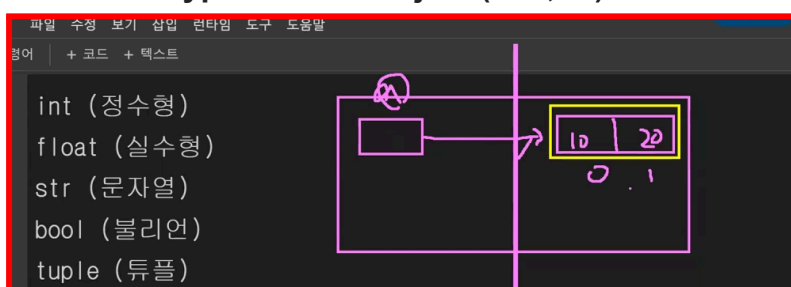
Immutable object (int, float, str, bool, tuple):



값이 변하지 않고 참조가 변한다.

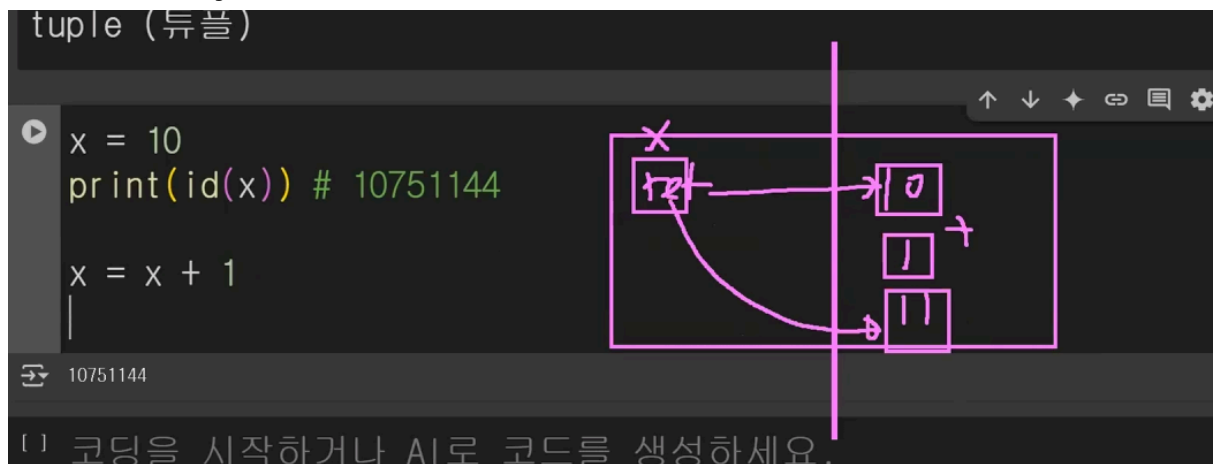
원래 num안에 10이 들어 있었다면 20이 힙에 추가된후 num은 20을 향하고 10의 연결을 끊어버린다.

Collection Type Mutable object (List, 등)



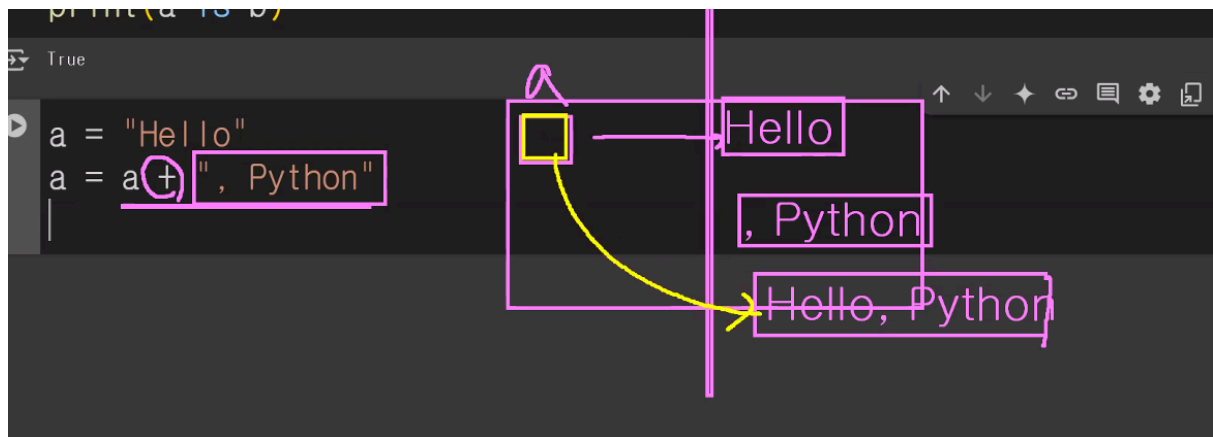
이런식으로 index 0에 있는 값이 10이었다면 100으로 바꿀 수 있다.

Immutable Object 데이터 저장 방식:



1. 이 경우, x가 10을 가리키고 있다.
2. 1을 추가로 힙 메모리에 올린다.
3. 10 + 1을 계산해서 11을 힙 메모리에 추가로 올린다.
4. x가 11을 가리키는걸로 바꾼다.
5. 10, 1은 garbage collector에 의해 날라간다 (참조를 더이상 안함).

문자열(Immutable) 메모리 저장 방식:



a = "Hello"

1. a(stack) → Hello(heap) 를 가리키고 있다

a = a + ", Python"

2. , Python 을 생성한다.
3. Hello, Python이라는걸 만들어서 a가 가르킨다.