

교수의 파이썬

04_1. integer objects
small integer caching

창원대학교 정보통신공학과 교수 박동규

넌넌한 교수의 파이썬

04_1. integer objects
small integer caching

창원대학교 정보통신공학과 교수 박동규

넌넌한 교수의 고급 파이썬

04_1. integer objects
small integer caching

창원대학교 정보통신공학과 교수 박동규

넌넌한 교수의 고급 파이썬

04_1. integer objects
small integer caching

창원대학교 정보통신공학과 교수 박동규

C

변수 중심

Python

객체 중심

복습

C

변수 중심

```
int a;
```

// 정수형 변수 선언

Python

객체 중심

복습

C

변수 중심

`int a;` **// 정수형 변수 선언**

`a :`

Python

객체 중심

복습

C

변수 중심

```
int a;    // 정수형 변수 선언  
a = 100;  // 정수값 할당
```

a :

A rectangular box with a thick black border, representing a memory location for the variable 'a'.

Python

객체 중심

복습

C

변수 중심

```
int a;    // 정수형 변수 선언  
a = 100;  // 정수값 할당
```

a :

100

Python

객체 중심

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

100

Python

객체 중심

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

Python

객체 중심

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

200

Python

객체 중심

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

200

Python

객체 중심

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

200

Python

객체 중심

```
a = 100      # 객체 생성
```

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a : 200

Python

객체 중심

```
a = 100      # 객체 생성
```

100

복습

C

변수 중심

```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

a :

200

Python

객체 중심

```
a = 100      # 객체 생성  
# 객체에 대한 참조 변수 a
```

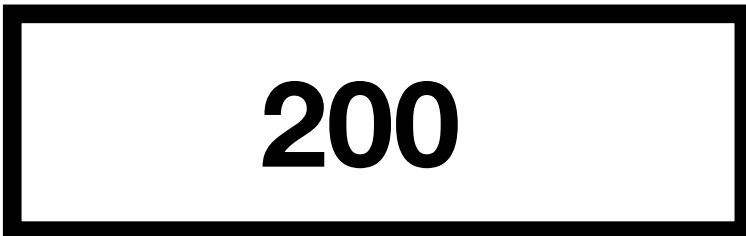
100

복습

C

변수 중심

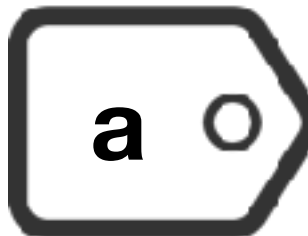
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```

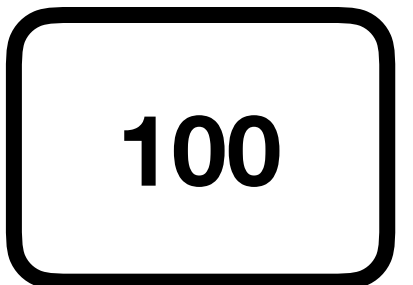
a : 

Python

객체 중심

```
a = 100      # 객체 생성  
# 객체에 대한 참조 변수 a
```

 a

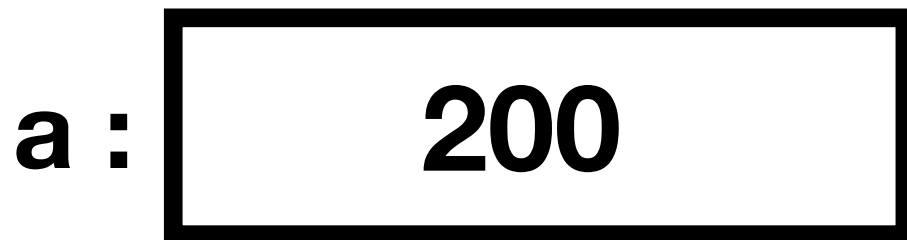
 100

복습

C

변수 중심

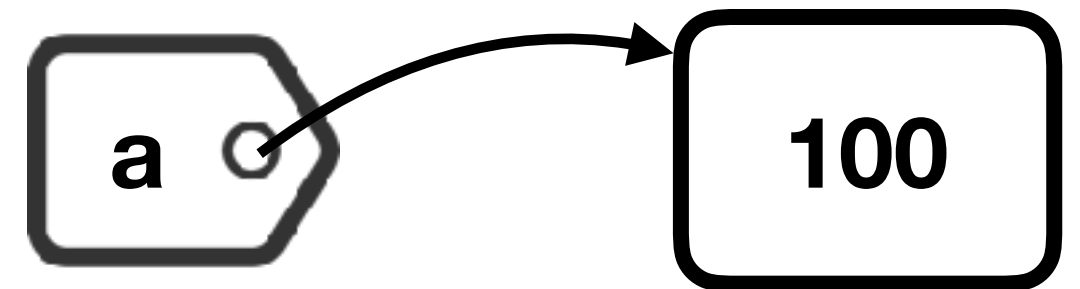
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100     # 객체 생성  
# 객체에 대한 참조 변수 a
```

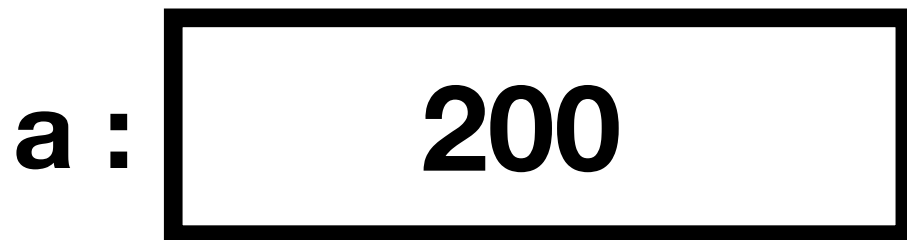


복습

C

변수 중심

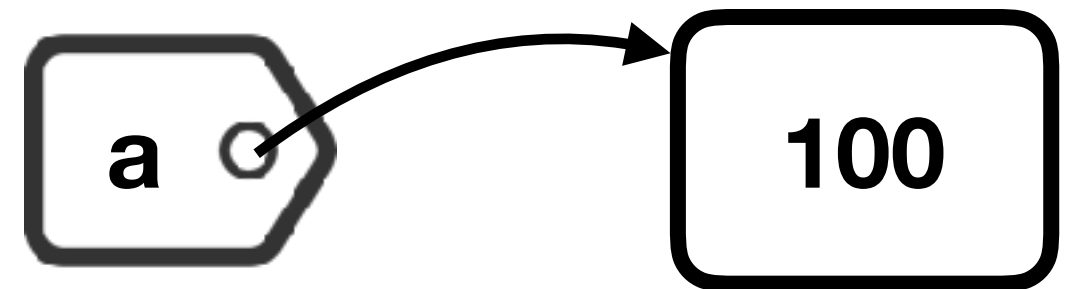
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100     # 객체 생성  
# 객체에 대한 참조 변수 a  
a = 200     # 객체 재할당
```

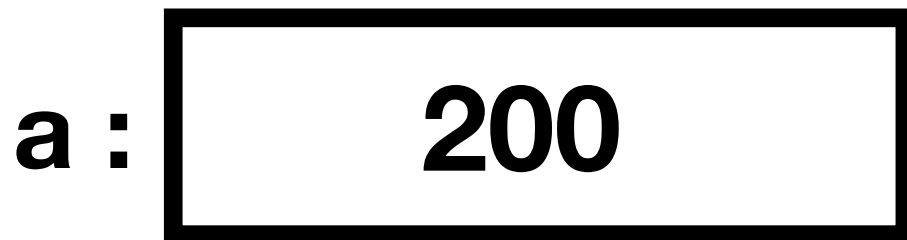


복습

C

변수 중심

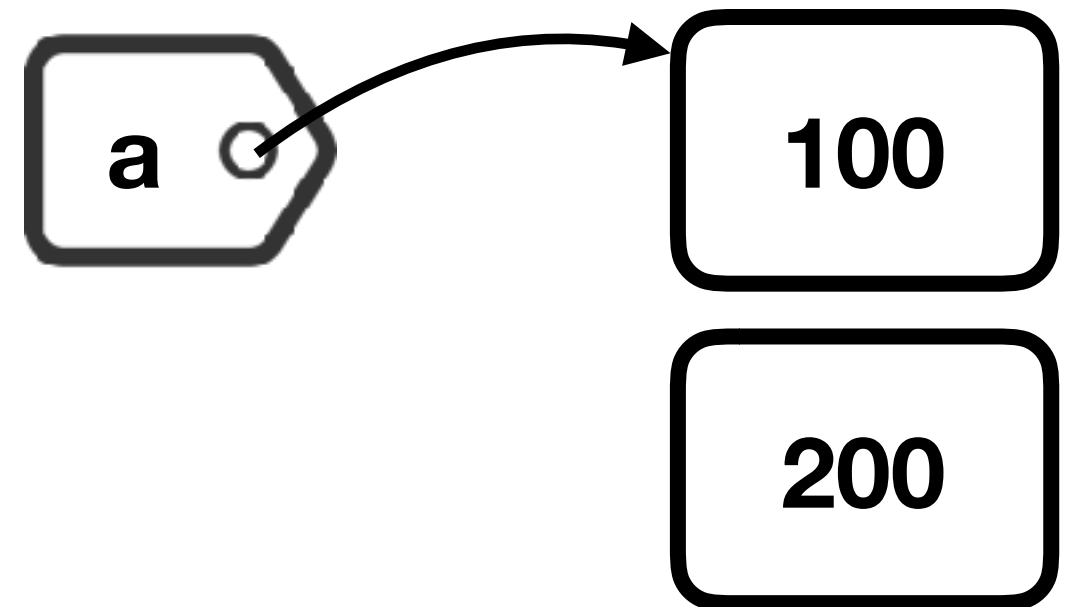
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100     # 객체 생성  
# 객체에 대한 참조 변수 a  
a = 200     # 객체 재할당
```

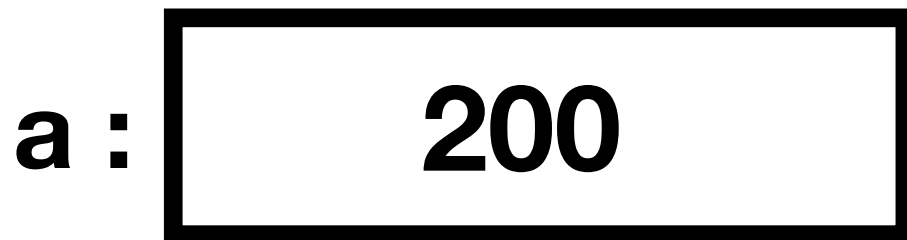


복습

C

변수 중심

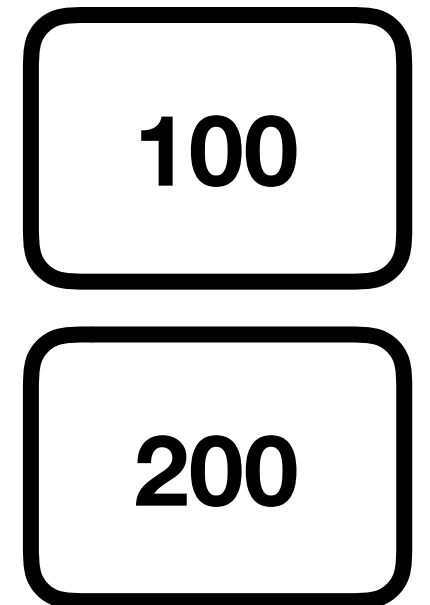
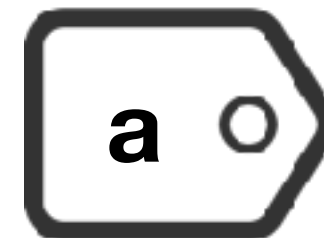
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100      # 객체 생성  
# 객체에 대한 참조 변수 a  
a = 200      # 객체 재할당
```

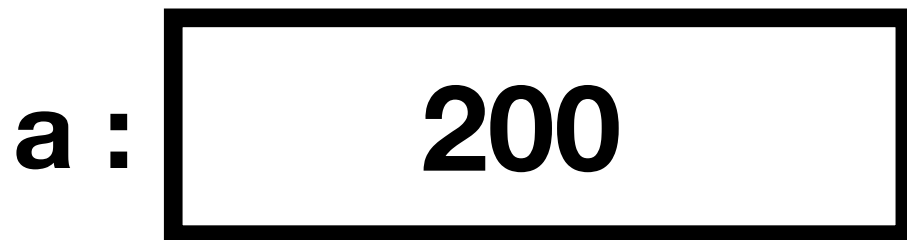


복습

C

변수 중심

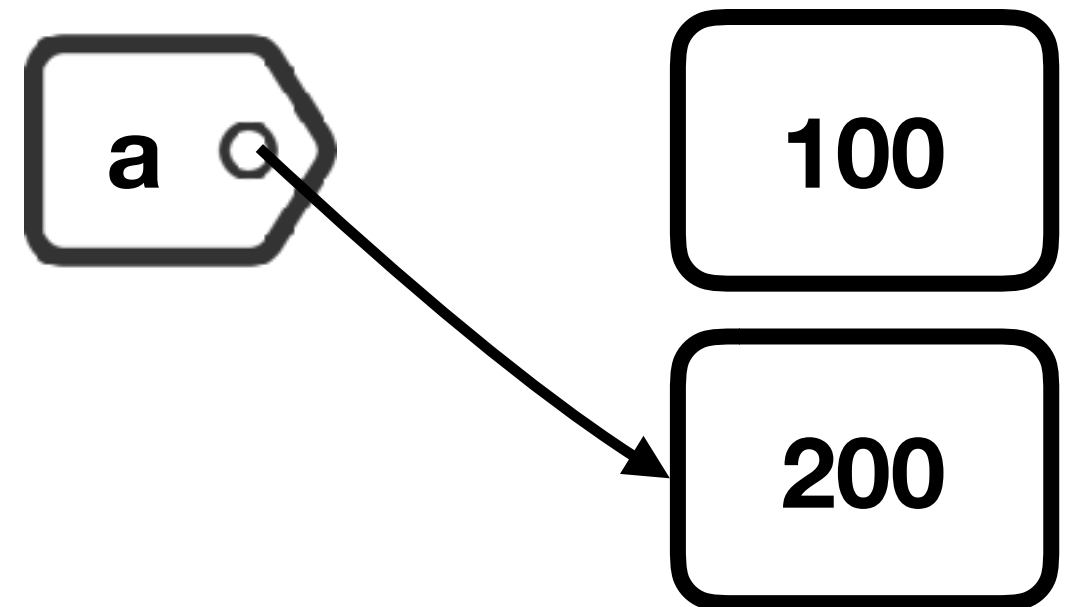
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100      # 객체 생성  
# 객체에 대한 참조 변수 a  
a = 200      # 객체 재할당
```

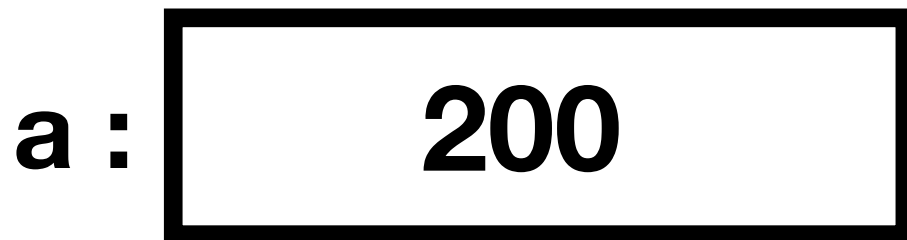


복습

C

변수 중심

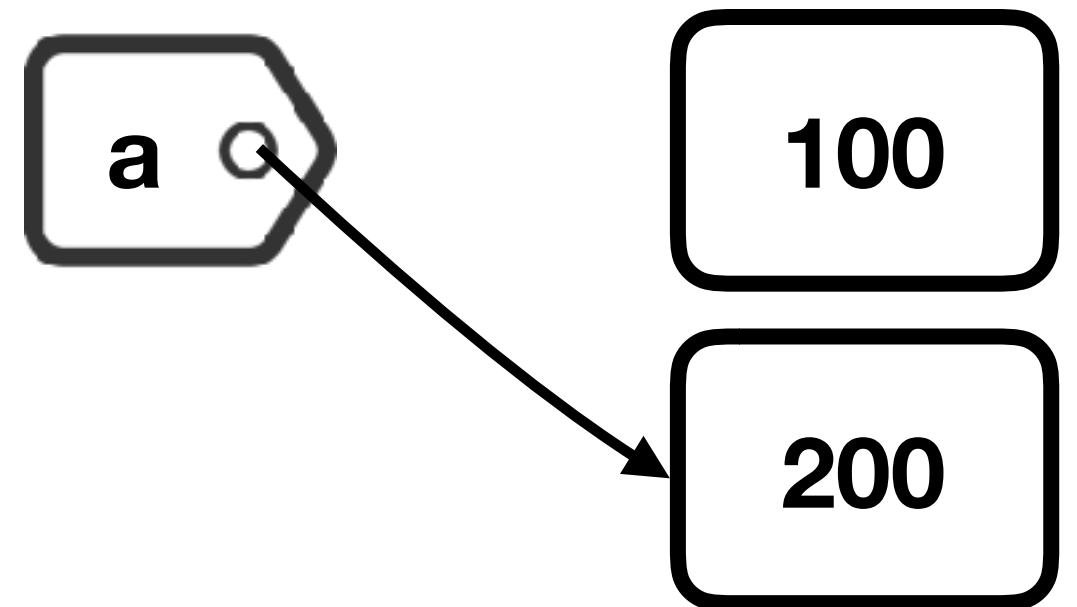
```
int a;      // 정수형 변수 선언  
a = 100;    // 정수값 할당  
a = 200;    // 정수값 재할당
```



Python

객체 중심

```
a = 100      # 객체 생성  
# 객체에 대한 참조 변수 a  
a = 200      # 객체 재할당
```



복습

질문



on S · 1주 전(수정됨)

파이썬에서 a 라는 변수에 -5라는 객체를 할당해준 뒤 -6이라 객체를 재할당한다음 id값을 보면 -6과 a가 다른데 왜그런건가요?

답글

답글 3개 ^

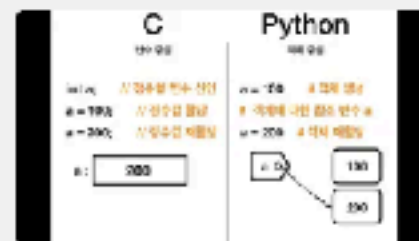


1



Donodvu Park · 3일 전

작업 메뉴



실험

```
>>> a = -5
```

```
>>> id(a)
```

```
4449958512
```

```
>>> id(-5)
```

```
4449958512
```

```
>>> # 여기까지는 자연스러워 보입니다.
```

```
>>> a = -6          # 이제 a가 -6을 참조하도록 합시다.
```

```
>>> id(a)
```

```
4500770864
```

```
>>> id(-6)
```

```
4500770896
```

```
>>>
```

```
>>> # 위의 두 객체 a와 -6의 id가 다름..???|
```

[Previous topic](#)[The `None` Object](#)[Next topic](#)[Boolean Objects](#)[This Page](#)[Report a Bug](#)[Show Source](#)

Integer Objects

All integers are implemented as “long” integer objects of arbitrary size.

On error, most `PyLong_As*` APIs return `(return_type)-1` which cannot be distinguished from a number. Use `PyErr_Occurred()` to disambiguate.

`PyLongObject`

This subtype of `PyObject` represents a Python integer object.

`PyTypeObject PyLong_Type`

This instance of `PyTypeObject` represents the Python integer type. This is the same object as `int` in the Python layer.

int `PyLong_Check(PyObject *p)`

Return true if its argument is a `PyLongObject` or a subtype of `PyLongObject`.

int `PyLong_CheckExact(PyObject *p)`

Return true if its argument is a `PyLongObject`, but not a subtype of `PyLongObject`.

`PyObject*` `PyLong_FromLong(long v)`

Return value: New reference.

Return a new `PyLongObject` object from `v`, or `NULL` on failure.

The current implementation keeps an array of integer objects for all integers between `-256` and `256`, when you create an `int` in that range you actually just get back a reference to the existing object. So it should be possible to change the value of `1`. I suspect the behaviour of Python in this case is undefined. :-)

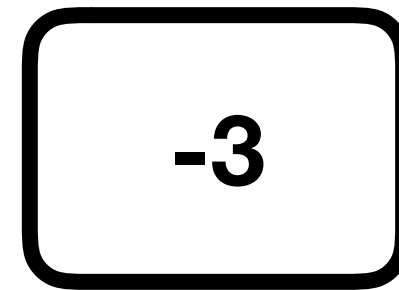
객체를 생성하는 것?

객체를 생성하는 것?

a = -3 **# 객체 생성**

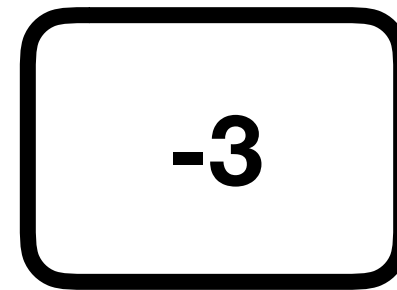
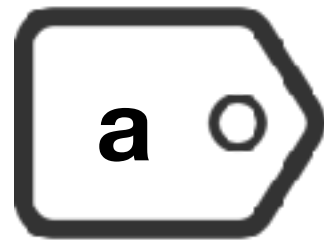
객체를 생성하는 것?

a = -3 **# 객체 생성**



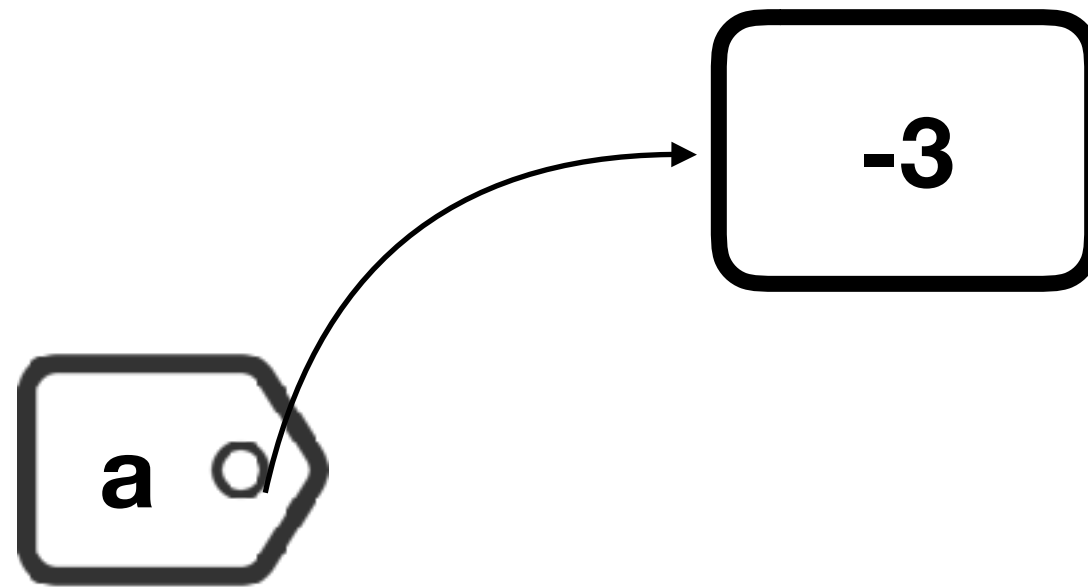
객체를 생성하는 것?

a = -3 **# 객체 생성**



객체를 생성하는 것?

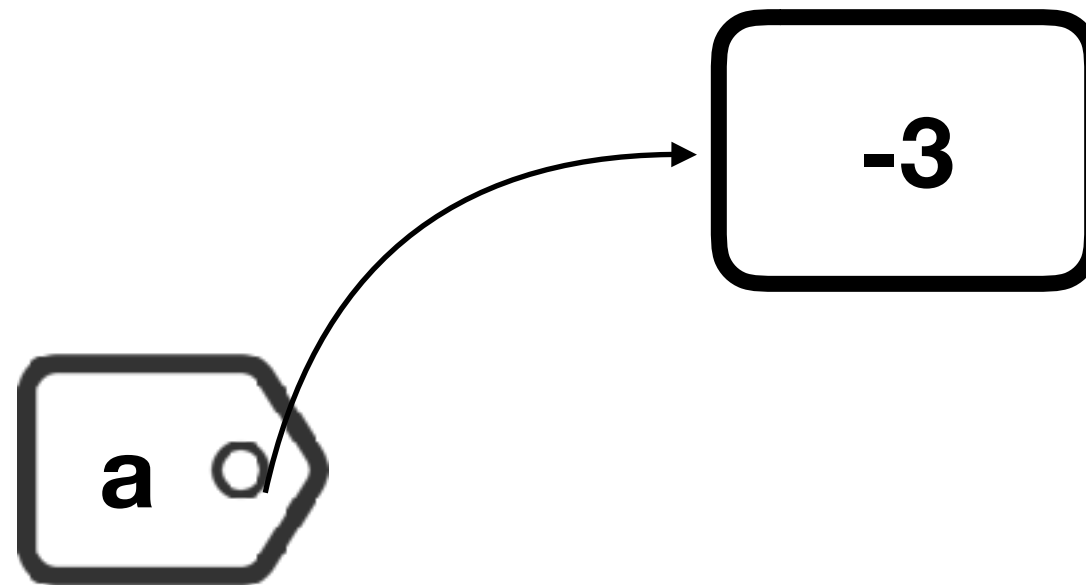
a = -3 **# 객체 생성**



객체를 생성하는 것?

a = -3 # 객체 생성

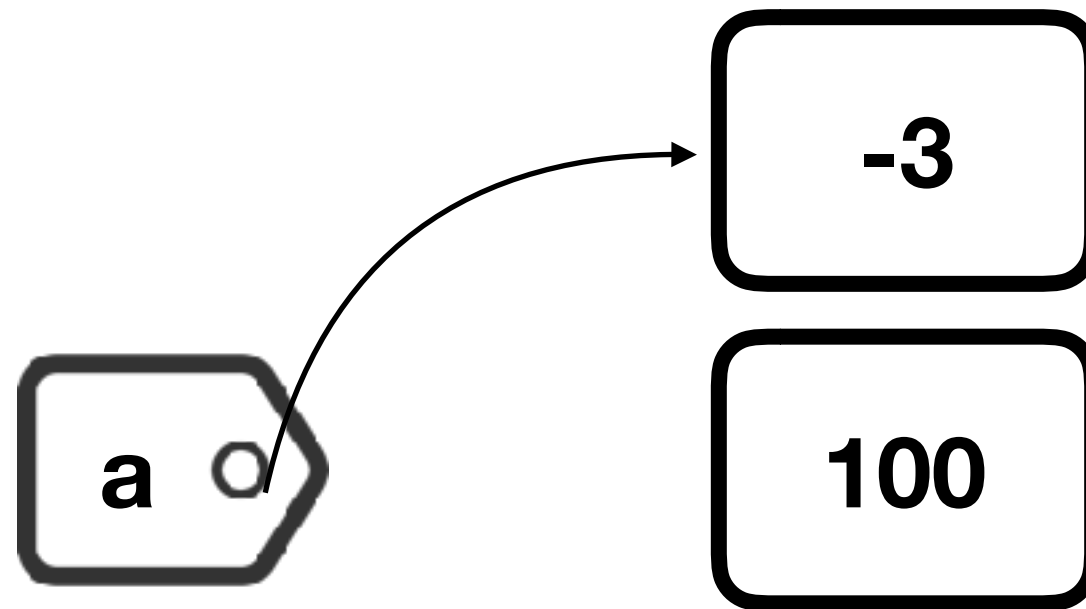
a = 100 # 객체 생성



객체를 생성하는 것?

a = -3 # 객체 생성

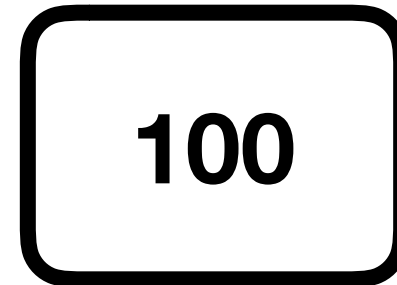
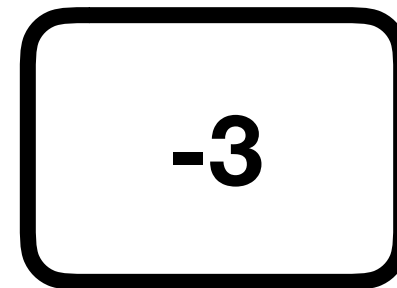
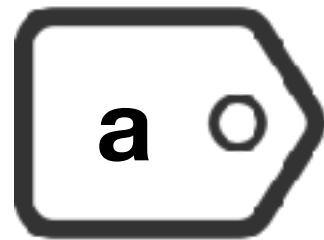
a = 100 # 객체 생성



객체를 생성하는 것?

a = -3 # 객체 생성

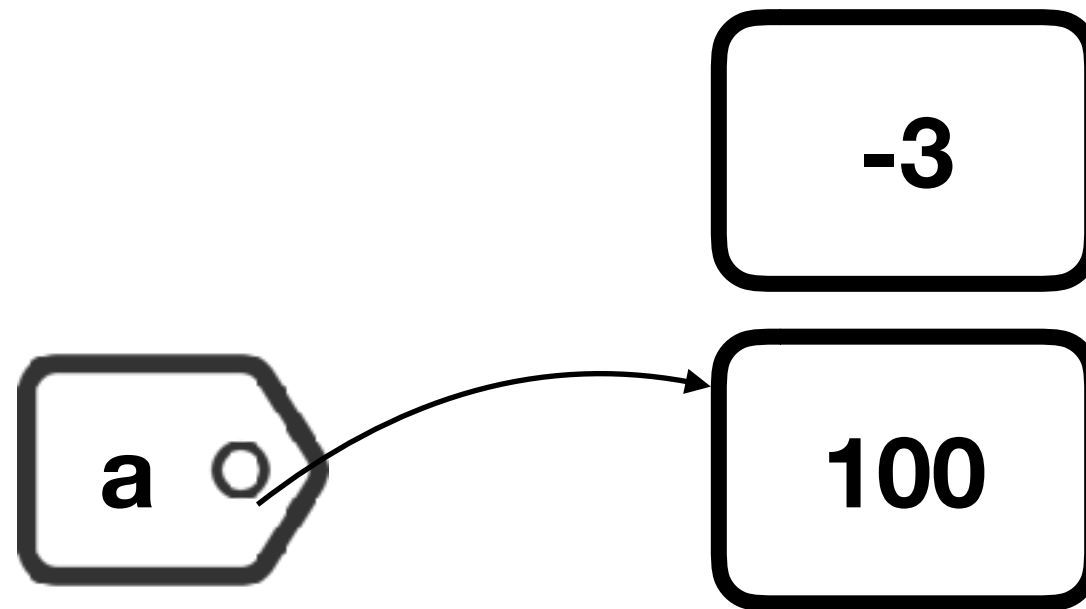
a = 100 # 객체 생성



객체를 생성하는 것?

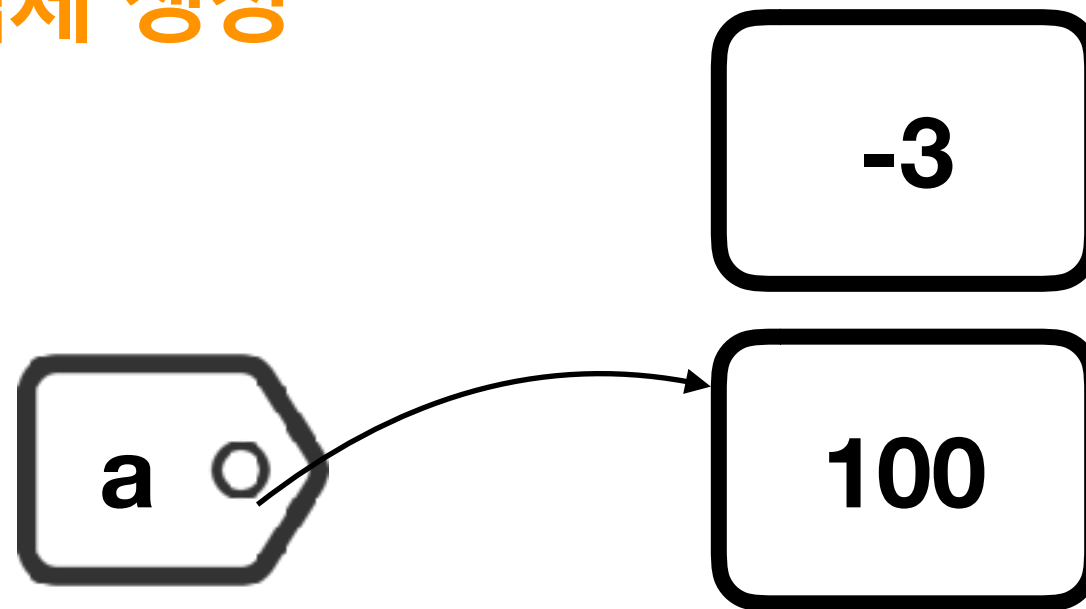
a = -3 # 객체 생성

a = 100 # 객체 생성



객체를 생성하는 것?

a = -3 # 객체 생성
a = 100 # 객체 생성
a = 200 # 객체 생성

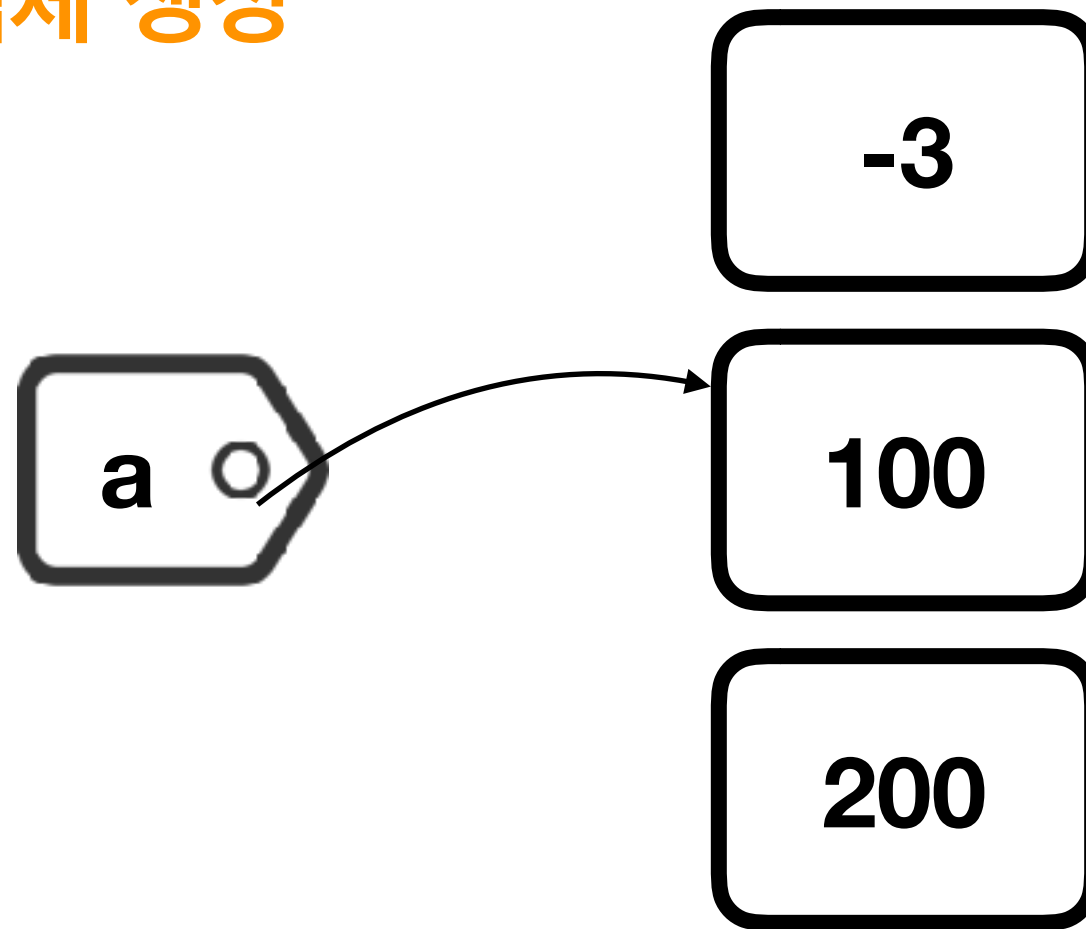


객체를 생성하는 것?

a = -3 # 객체 생성

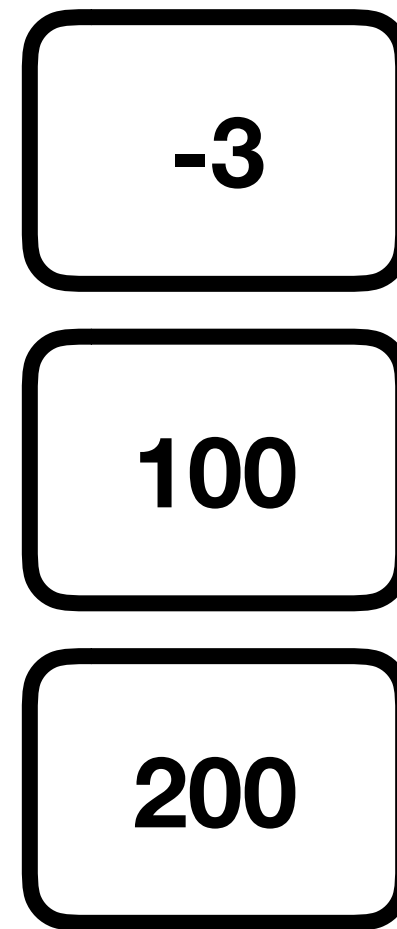
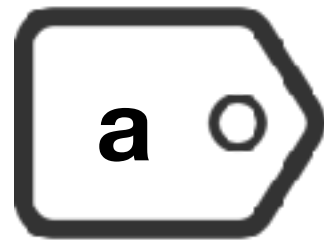
a = 100 # 객체 생성

a = 200 # 객체 생성



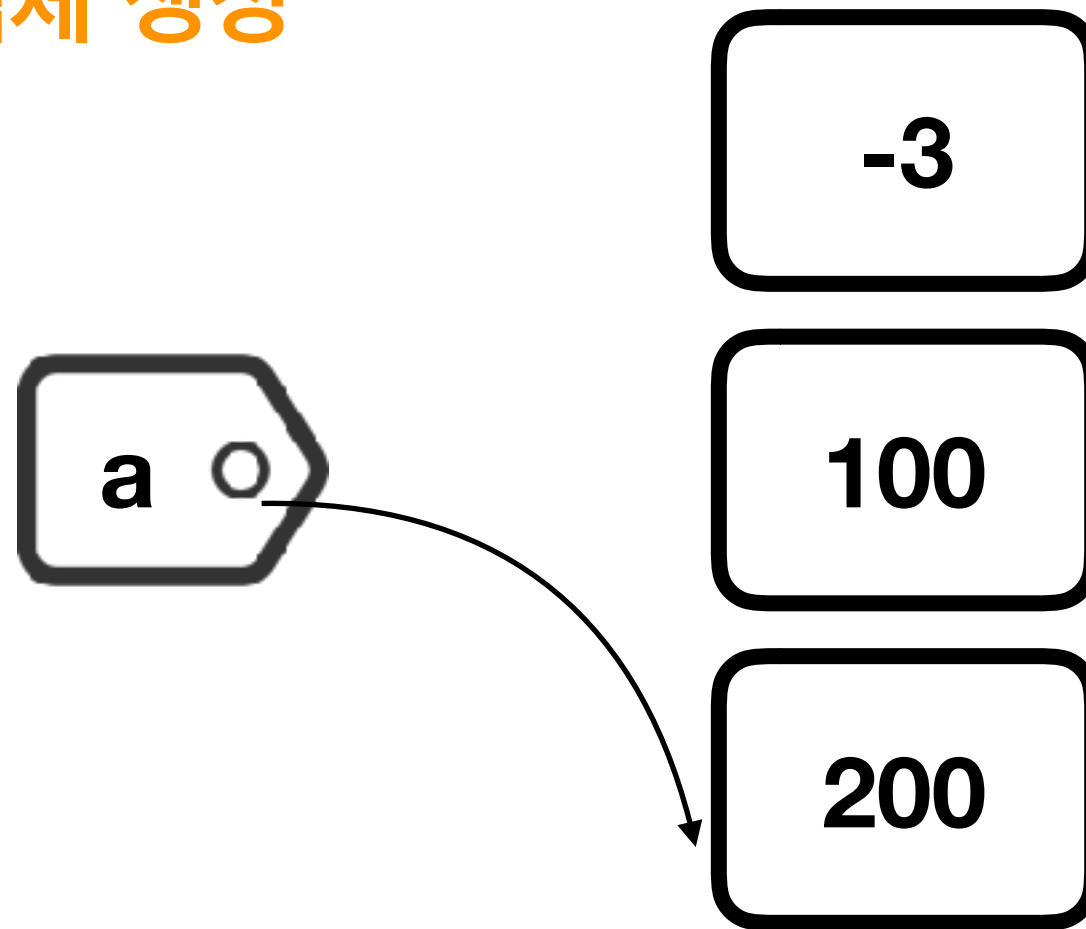
객체를 생성하는 것?

a = -3 # 객체 생성
a = 100 # 객체 생성
a = 200 # 객체 생성



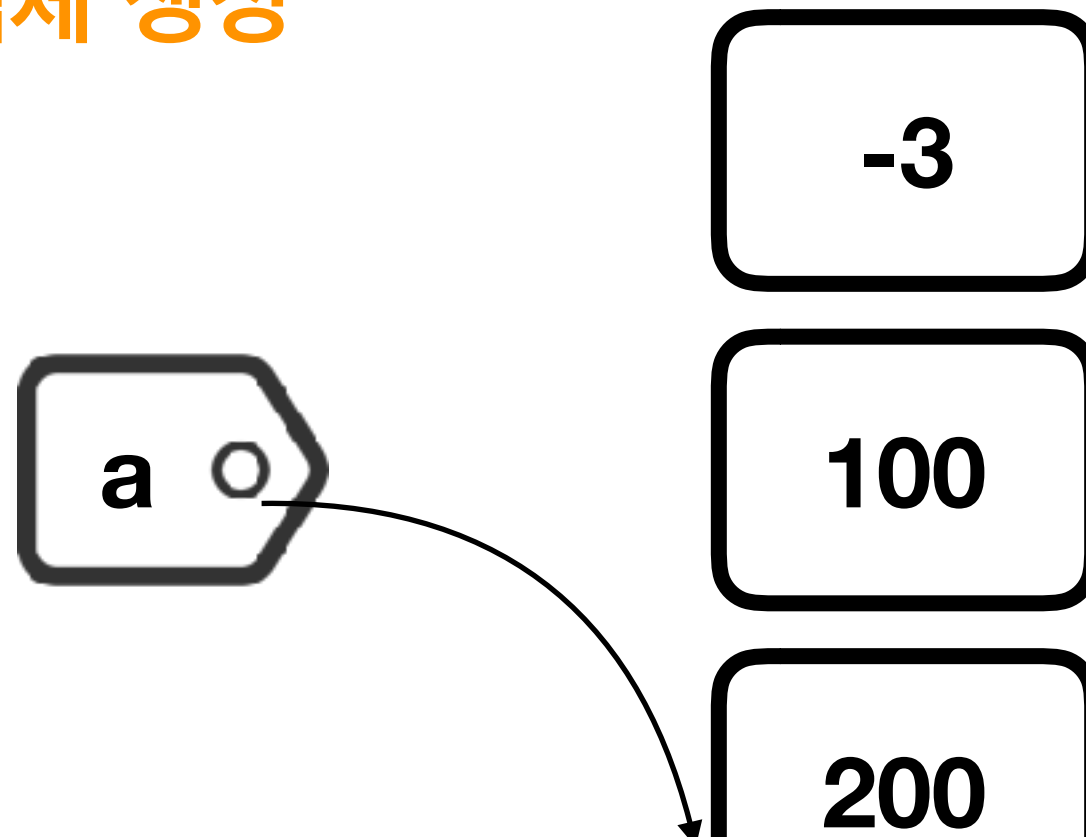
객체를 생성하는 것?

a = -3 # 객체 생성
a = 100 # 객체 생성
a = 200 # 객체 생성



객체를 생성하는 것?

a = -3 # 객체 생성
a = 100 # 객체 생성
a = 200 # 객체 생성



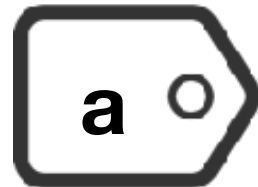
잘은 객체 생성과 참조 변경은 시스템 수행 속도를 떨어뜨린다.

미리 생성해 둔 객체

- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.
- 따라서 $a = -5$ 일 경우 $\text{id}(a)$ 와 $\text{id}(-5)$ 는 항상 동일하다.

미리 생성해 둔 객체

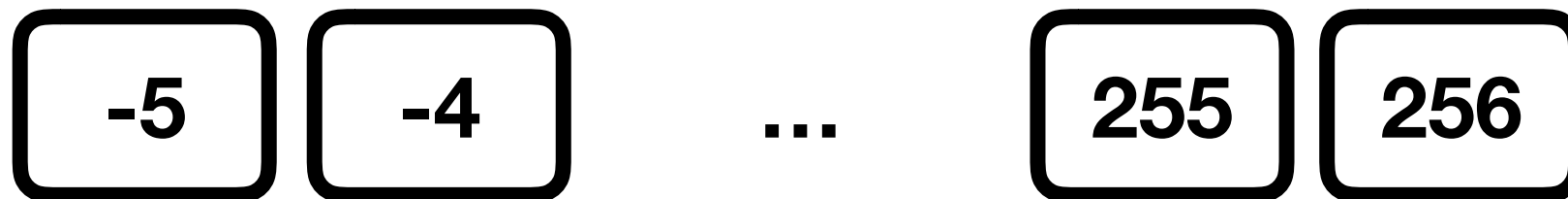
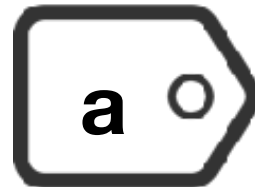
- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.



- 따라서 $a = -5$ 일 경우 $\text{id}(a)$ 와 $\text{id}(-5)$ 는 항상 동일하다.

미리 생성해 둔 객체

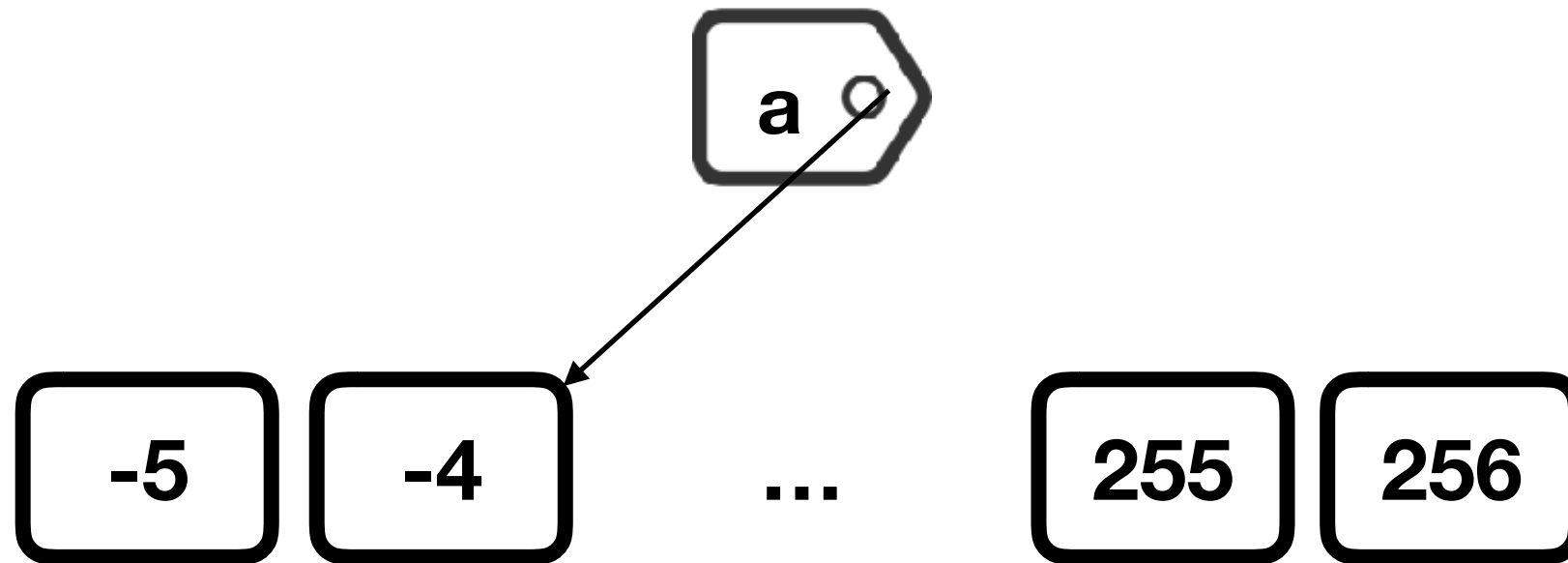
- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.



- 따라서 $a = -5$ 일 경우 $\text{id}(a)$ 와 $\text{id}(-5)$ 는 항상 동일하다.

미리 생성해 둔 객체

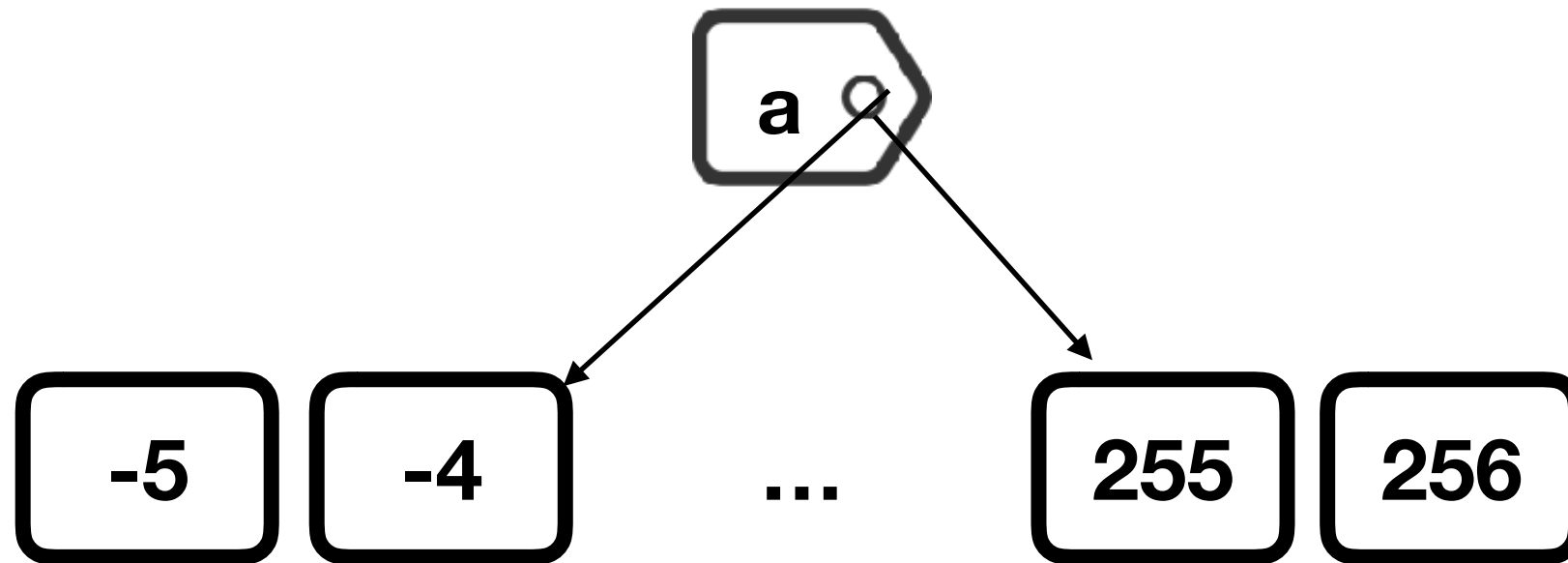
- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.



- 따라서 $a = -5$ 일 경우 $\text{id}(a)$ 와 $\text{id}(-5)$ 는 항상 동일하다.

미리 생성해 둔 객체

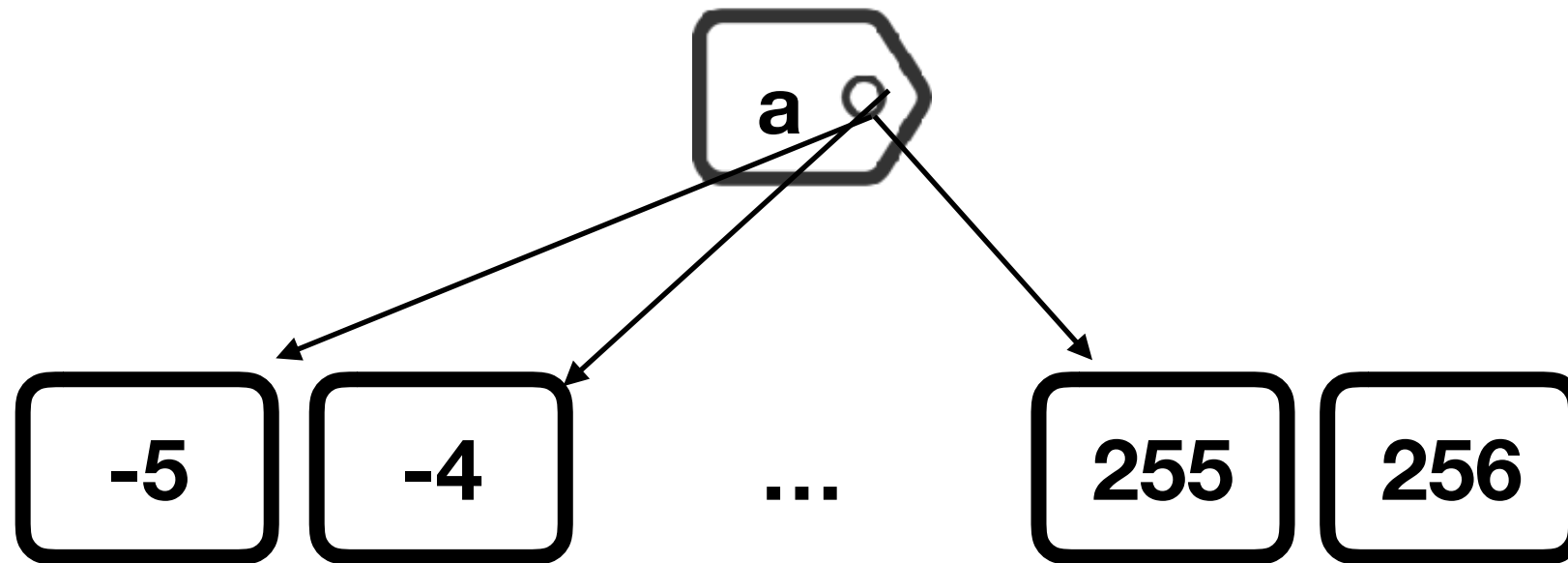
- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.



- 따라서 `a = -5`일 경우 `id(a)`와 `id(-5)`는 항상 동일하다.

미리 생성해 둔 객체

- -5부터 256까지의 정수객체는 미리 생성해 놓고 사용한다.



- 따라서 `a = -5`일 경우 `id(a)`와 `id(-5)`는 항상 동일하다.

새로 생성하는 객체

- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6          # 이제 a가 -6을 참조하도록 합시다.  
>>> id(a)  
4500770864  
>>> id(-6)  
4500770896
```

새로 생성하는 객체

- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6          # 이제 a가 -6을 참조하도록 합시다.  
>>> id(a)  
4500770864  
>>> id(-6)  
4500770896
```

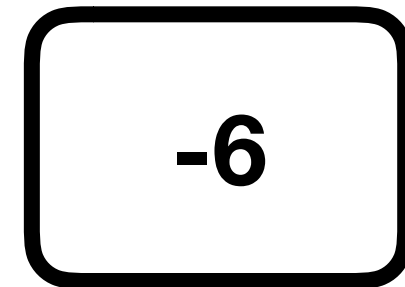

새로 생성하는 객체

- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6 # 이제 a가 -6을 참조하도록 합시다.
```

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



새로 생성하는 객체

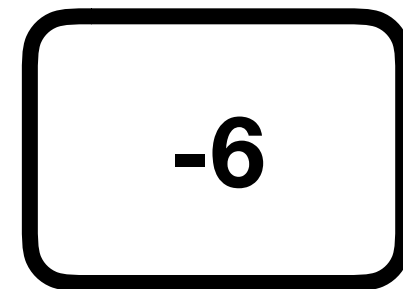
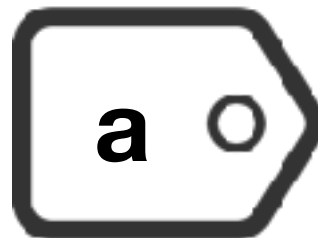
- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6
```

이제 a가 -6을 참조하도록 합시다.

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



새로 생성하는 객체

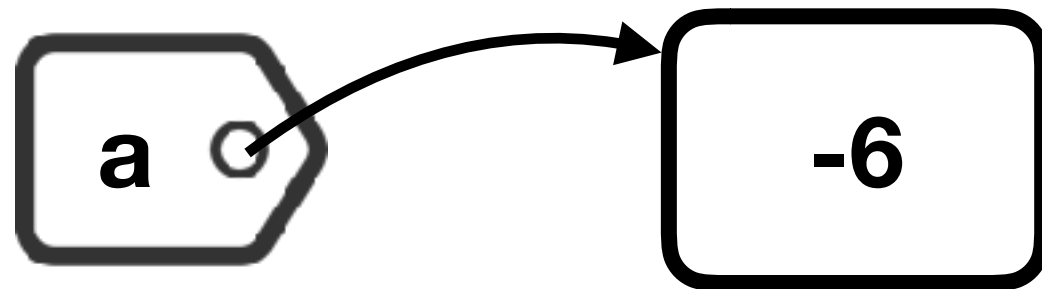
- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6
```

이제 a가 -6을 참조하도록 합시다.

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



새로 생성하는 객체

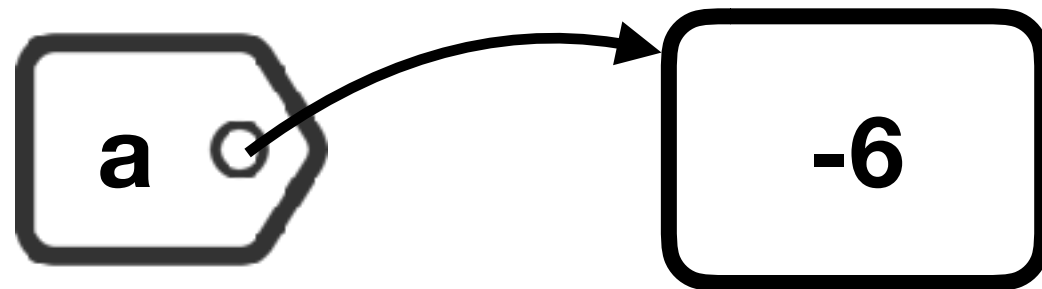
- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6
```

이제 a가 -6을 참조하도록 합시다.

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



새로 생성하는 객체

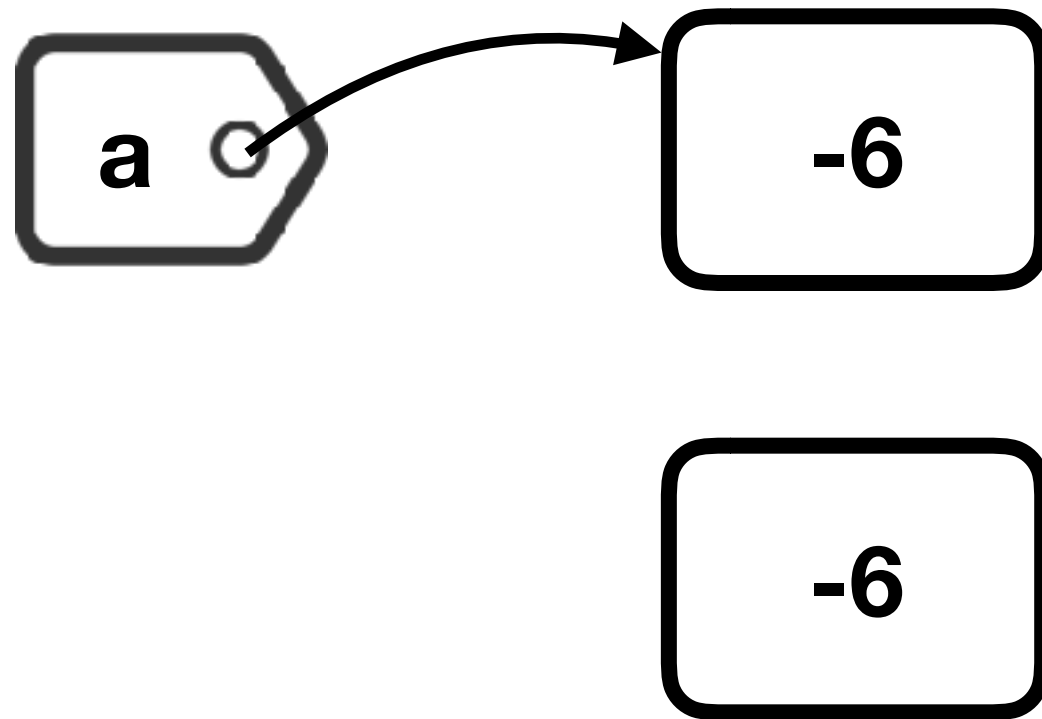
- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6
```

이제 a가 -6을 참조하도록 합시다.

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



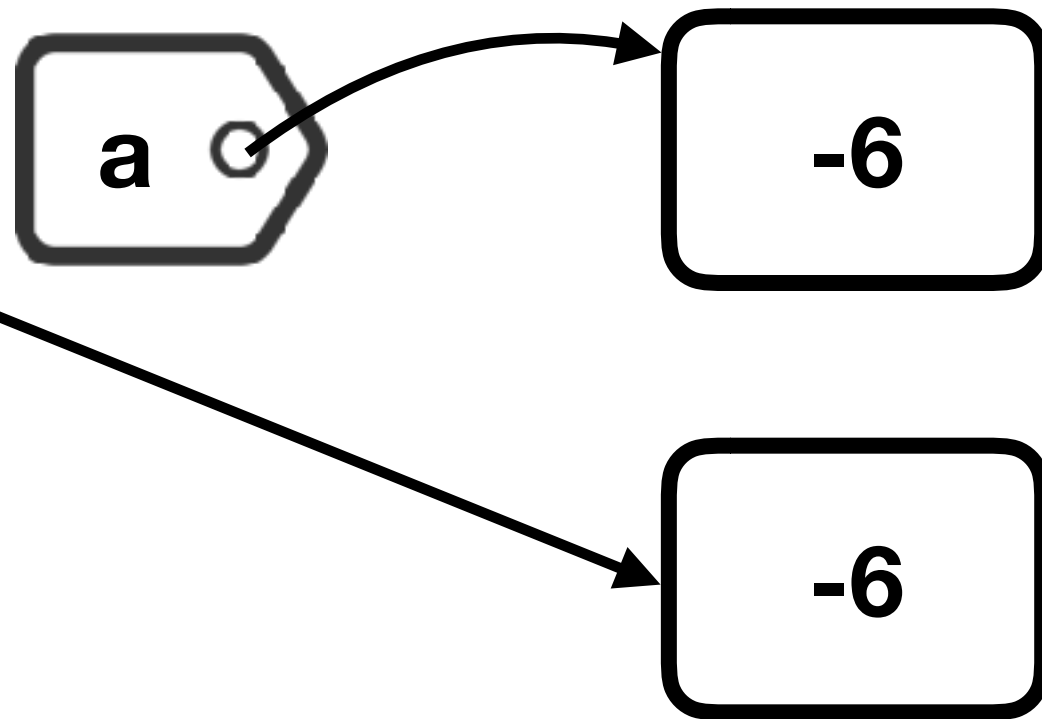
새로 생성하는 객체

- -6과 같은 객체는 필요할 때마다 새롭게 객체를 생성하고 참조 변수가 이를 참조한다

```
>>> a = -6 # 이제 a가 -6을 참조하도록 합시다.
```

```
>>> id(a)  
4500770864
```

```
>>> id(-6)  
4500770896
```



Test

Test

```
>>> a = 5  
>>> id(5)  
4449958832  
>>> id(5)  
4449958832  
>>> id(5)  
4449958832
```


Test

```
>>> a = 5
>>> id(5)
4449958832
>>> id(5)
4449958832
>>> id(5)
4449958832
```

```
>>> a = -9
>>> id(a)
4500771024
>>> id(-9)
4500770960
>>> id(-9)
4500771088
>>> id(-9)
4500771120
```

Test

```
>>> a = 5
>>> id(5)
4449958832
>>> id(5)
4449958832
>>> id(5)
4449958832
```

```
>>> a = -9
>>> id(a)
4500771024
>>> id(-9)
4500770960
>>> id(-9)
4500771088
>>> id(-9)
4500771120
```

-5에서 256사이의 값을 제외한 변수는 매번 새로 생성되므로
id() 값이 달라집니다

Lab

감사합니다

감사합니다

많은 질문 부탁드립니다