

```
class Date: — ①
    def __init__(self, a, b, c):
        self.day = a
        self.month = b
        self.year = c
```

```
D = Date(26, 9, 2025) — ②
```

① After executing the first line, Python Interpreter will understand that the programmer has implemented a new class named Date and has implemented a constructor function as a part of class definition.

② D = Date(26, 9, 2025)

- Python recognises this to be an assignment statement.
- Execute the RHS first and object furnished by the execution of the RHS should be named by the name specified on the LHS.

- Date(26, 9, 2025)

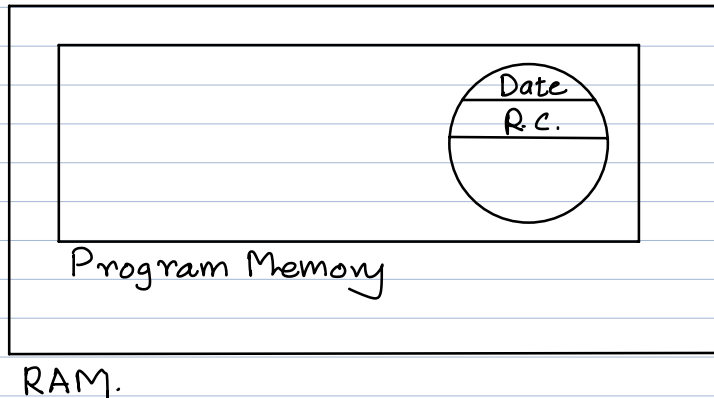
i) Programmer has put the call operator around class Date indicating that he wants to create a new object of class Date.

Date(26, 9, 2025)

ii) Programmer has supplied 3 integer objects as an initialization data.

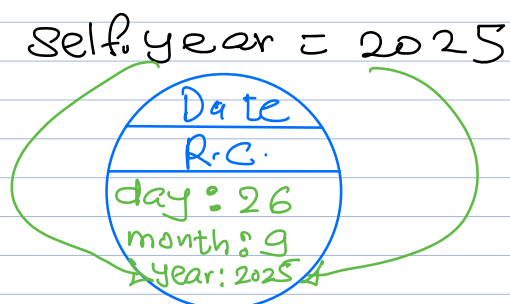
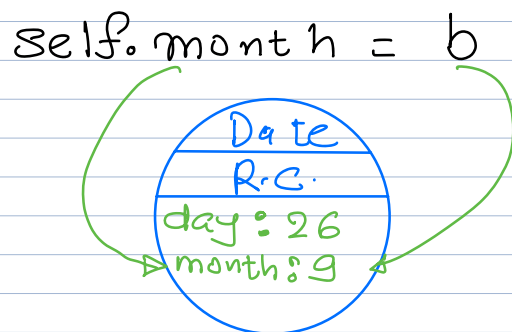
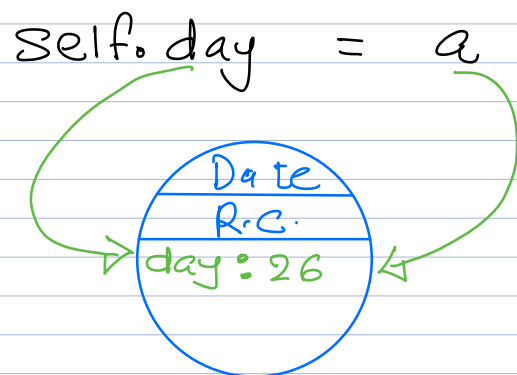
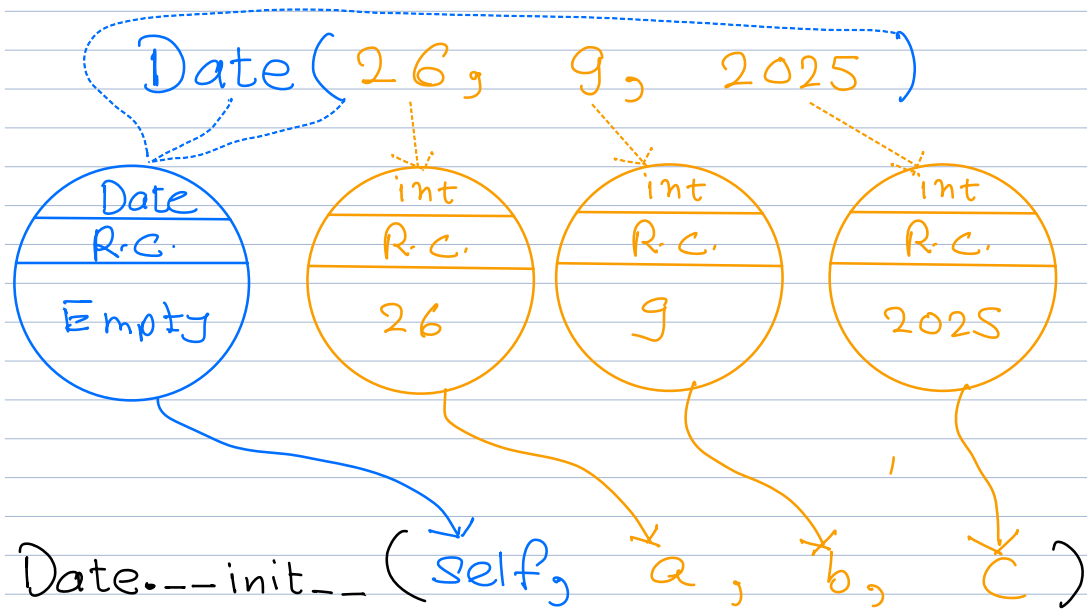
- Action:

- (i) Allocate new object in program memory.
Divide it into three compartments.
Put the class name in the first compartment.
Put the reference count in the second compartment.



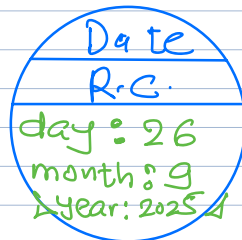
- (ii) Call constructor function `Date.__init__` and as actual parameters pass.
not only newly allocated object of class `Date`
but also three integer objects which are specified in the initialization field.

[Python thinks: For this call to succeed, `Date.__init__()` must have 4 formal parameters.
one → to hold newly allocated object of class `Date`.
three more → to hold three integers]

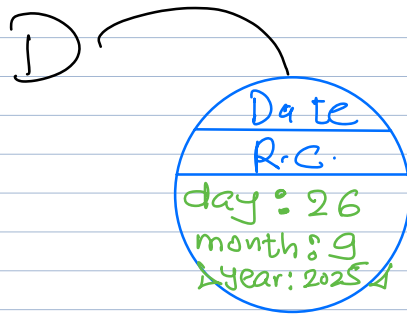


Execution of
Date(26, 9, 2025)
is over.

Object ??



Attach LHS Name to it'



$D = \text{Date}(26, 9, 2025)$

over!!
