15.10.2. Exercise

In the previous chapter, a series of exercises asked you to write a Date class and several functions that work with Date objects. Now let's practice rewriting those functions as methods.

1. Write a definition for a Date class that represents a date – that is, a year, month, and day of the month.
2. Write an **init** method that takes year, month, and day as parameters and assigns the parameters to attributes. Create an object that represents June 22, 1933.
3. Write **str** method that uses an f-string to format the attributes and returns the result. If you test it with the Date you created, the result should be 1933-06-22.
4. Write a method called is_after that takes two Date objects and returns True if the first comes after the second. Create a second object that represents September 17, 1933, and check whether it comes after the first object.

Hint: You might find it useful write a method called to_tuple that returns a tuple that contains the attributes of a Date object in year-month-day order.

In [1]:
```
from date_15_10_2 import Date
```

Part 1 Completed in date_15_10_2 file

Part 2

In [2]:
```
june_22_1933 = Date(1933, 6, 22)
```

Part 3

In [3]:
```
print(june_22_1933)
```

```
1933-June-22
```

Part 4

In [4]:
```
sept_17_1933 = Date(1933, 9, 17)
print(sept_17_1933.is_after(june_22_1933))
```

```
True
```