The “ repr ” constructor:

Displaying output to the user can be difficult because the user is often unknown. The often-used dictionaries and list representation by developers are not always easily readable by users. In Data Science for example, it is very common to use a table format where a table has rows and columns. The pandas library in Python is here very useful and it works with Dataframes where every Dataframe has rows and columns and it also allows to give columns a name, to select particular columns, etc. But object-oriented programming (OOP) also has a nice constructor that can be used to transform your output easily so that it becomes much more user friendly. Taking the appointment_sched class (see in “Projects”) and change it a bit (changes are incorporated in red), one can see that the output in “__repr__” is only changed in the way that every variable used a name “Date”, “Staff”, “Patient” (see the output at the end). This is only a small adaptation to the output seen before and one downside of the constructor is that when one also wants to send the data back to a database or csv file, the data has to undergo some transformation (cancelling again the names of the variables). Generally, one has to think about the trade-off between readability vs. efficiency of the code.

```
class appointment_sched:
    list1 = []
    class appointment:

        def __init__(self, type, staff, patient):
            self.type = type
            self.staff = staff
            self.patient = patient

        def add_appoint(self):
            appointment_sched.list1.append([self.type,self.staff,self.patient])(self)

        def __repr__(self):
            return f'Date: {self.type}, Staff: {self.staff}, Patient: {self.patient}'

        @classmethod
        def load_csv(cls):
            with open('appoint_testdata1.csv') as f:
                c = csv.reader(f, skipinitialspace=True, delimiter=',')
                c = list(c)
```

```
        for I in c:
            appointment_sched.appointment(i[0],i[1],(i[2],i[3],i[4]))


print(appointment_sched.appointment.load_csv())
```

```
[Date: 2022-01-03 08:00:00, Staff: 2, Patient: ('claus_mann',
'hallostrasse_14_4999_koepers', '0499999999'), Date: 2022-01-03 09:00:00, Staff: 3,
Patient: ('hans_haas', 'weger_12_4785_bilzen', '0475858585'), Date: 2022-01-03
09:30:00, Staff: 5, Patient: ('jil_joske', 'heza_69_4897_don', '0475321456'), Date:
2022-01-03 10:00:00, Staff: 8, Patient: ('cloe_cizmic', 'zert_65_4578_bert',
'1234789654'), Date: 2022-01-03 09:00:00, Staff: 7, Patient: ('bilco_bibber',
'auenland_00_4563_halm', '0488585858'), Date: 2022-01-03 09:30:00, Staff: 5,
Patient: ('karl_vierstein', 'herty_2_4700_kel', '0453214578'), Date: 2022-01-03
15:00:00, Staff: 1, Patient: ('r_r', 'r_r_rr_r', 'r')]
```