Class Method vs Static Method:

Understanding the difference between a class method and static method requires to see the difference between a class, an instance and variables which are outside the scope of a class. Starting with the former, in a class method (see the example below where load_csv uses a classmethod as decorator) it is important to note that the argument used only differentiates in its name (compared to self for example) to emphasize that a class method is used. But the main difference is that when" appointment_sched.list1" is called outside of the class it is done without giving it one specific instance (in green) and also inside of load_csv() an appointment instance is created without giving it any specific instance (yellow). And the reason for this is clear, we won't change one specific instance but rather change the structure of the entire class in that sense that several instances are changed or created (as seen in this case). This Is different compared to for example a method which uses an instance of an object as argument (such as add_appoint seen below). But it is also different to a method which doesn't use an instance as argument but one that would use variables which are outside of the scope of the class (or even no argument at all). An example could be here, that we would like to check whether a variable is a datetime format or not (see the example of "check_date" below). We could build this function easily outside of the scope of the class because we don't use any specific instance for this neither do we need to work with the entire class. That is called a static method. This method will only be used inside of a class (using a decorator 'staticmethod') to put it in context to the class but in general it could be easily used outside of it (Bader, N.D.).

```
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])

        @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])).add_appoint()

    @staticmethod
    def check_date():
        if type(self.type) is datetime.date:
            return True
```

```
print(appointment_sched.list1))
```

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

Reference List:

Bader, D. (N.D.) OOP Method Types in Python: @classmethod vs @staticmethod vs Instance Methods. Available from: https://realpython.com/courses/python-method-types/ [Accessed 09 May 2022]