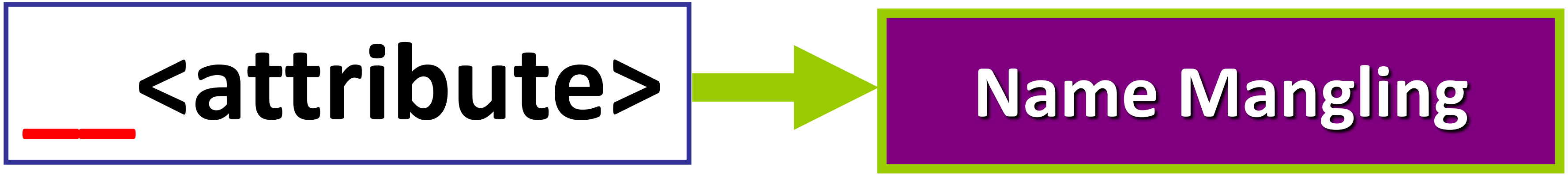
Lecture

Name Mangling

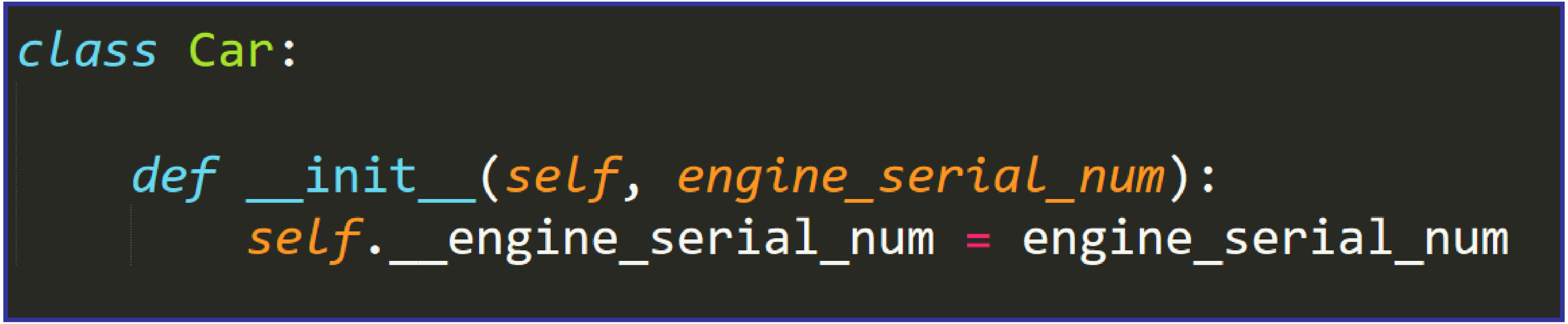
Name Mangling



__<attribute>

Name Mangling

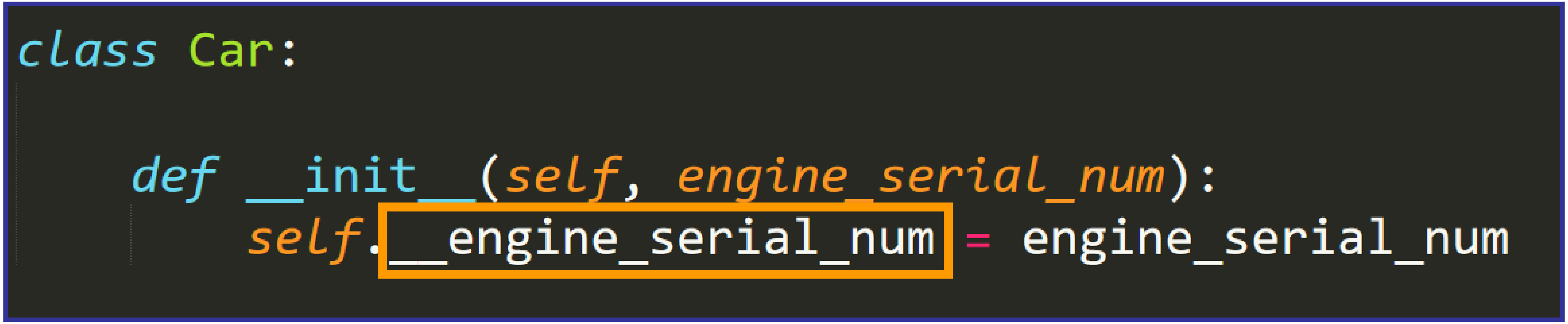
Name Mangling



```python
class Car:

    def __init__(self, engine_serial_num):
        self.__engine_serial_num = engine_serial_num
```

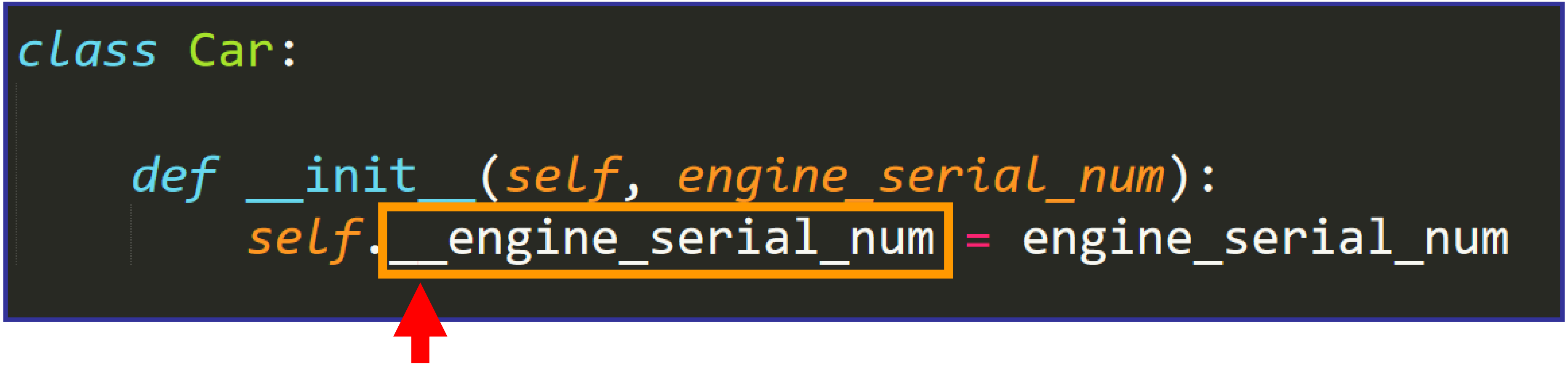
Name Mangling



```
class Car:

    def __init__(self, engine_serial_num):
        self.__engine_serial_num = engine_serial_num
```

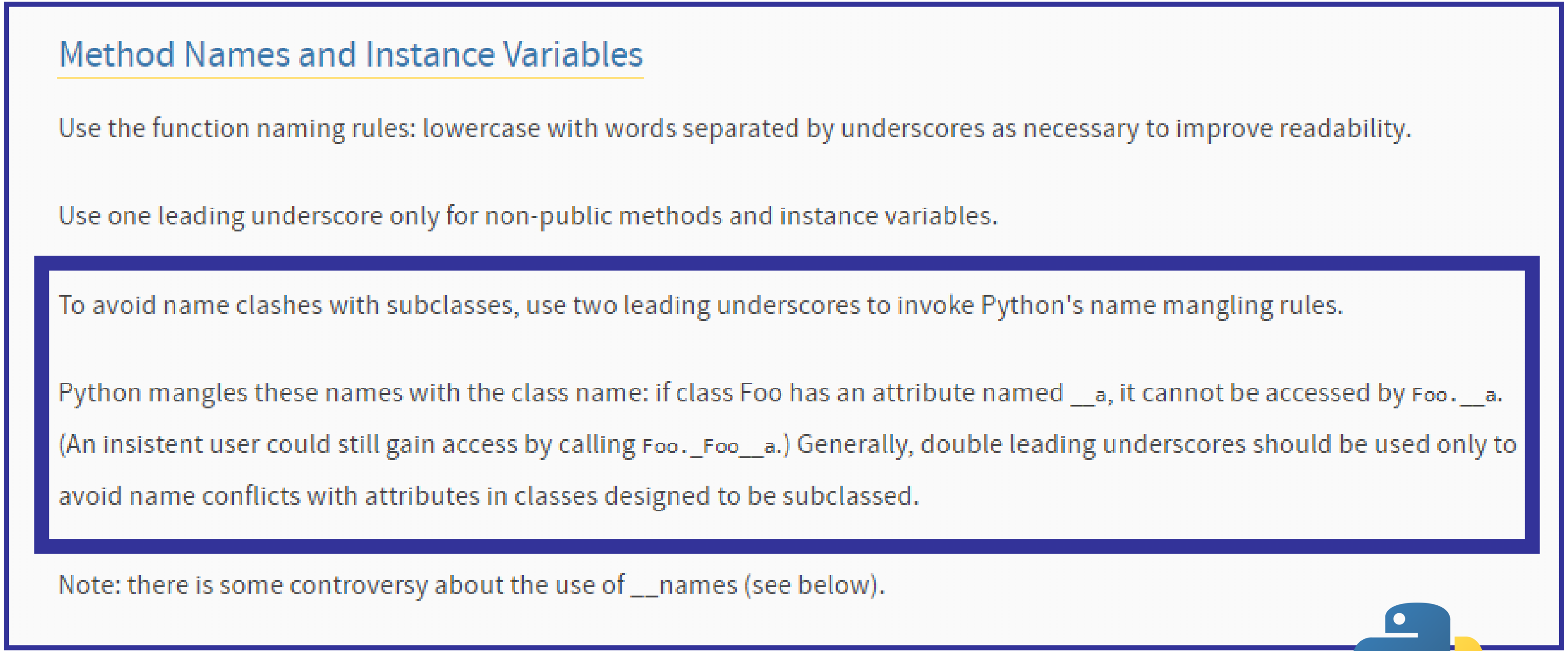
Name Mangling



```python
class Car:

    def __init__(self, engine_serial_num):
        self.__engine_serial_num = engine_serial_num
```

Name Mangling



Method Names and Instance Variables

Use the function naming rules: lowercase with words separated by underscores as necessary to improve readability.

Use one leading underscore only for non-public methods and instance variables.

To avoid name clashes with subclasses, use two leading underscores to invoke Python's name mangling rules.

Python mangles these names with the class name: if class Foo has an attribute named `__a`, it cannot be accessed by `Foo.__a`. (An insistent user could still gain access by calling `Foo._Foo__a`.) Generally, double leading underscores should be used only to avoid name conflicts with attributes in classes designed to be subclassed.

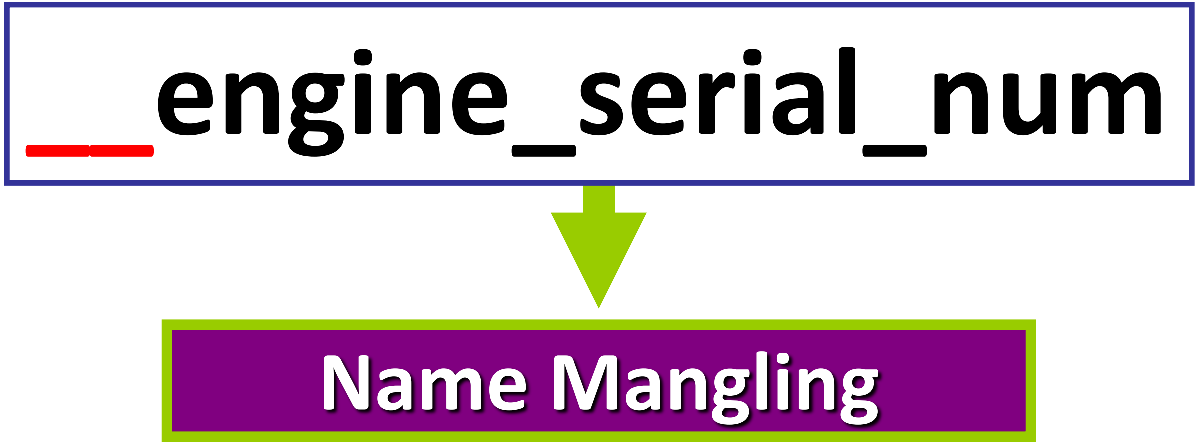
Note: there is some controversy about the use of __names (see below).

PEP 8

Name Mangling

__engine_serial_num

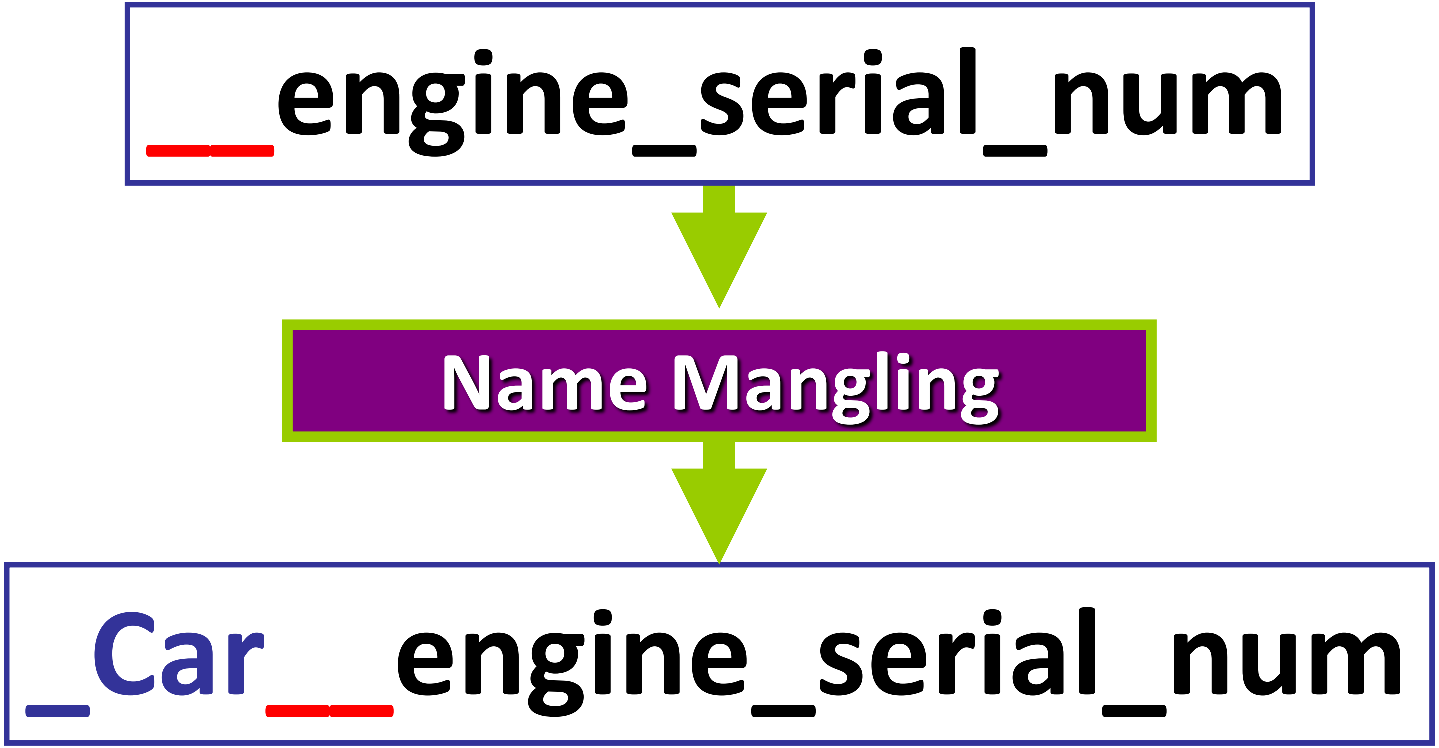
Name Mangling



__engine_serial_num

Name Mangling

Name Mangling



__engine_serial_num

Name Mangling

_Car__engine_serial_num

Name Mangling

```
>>> class Car:

        def __init__(self, engine_serial_num):
            self.__engine_serial_num = engine_serial_num


>>> my_car = Car("53094982")
>>> my_car.__engine_serial_num
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    my_car.__engine_serial_num
AttributeError: 'Car' object has no attribute '__engine_serial_num'
```

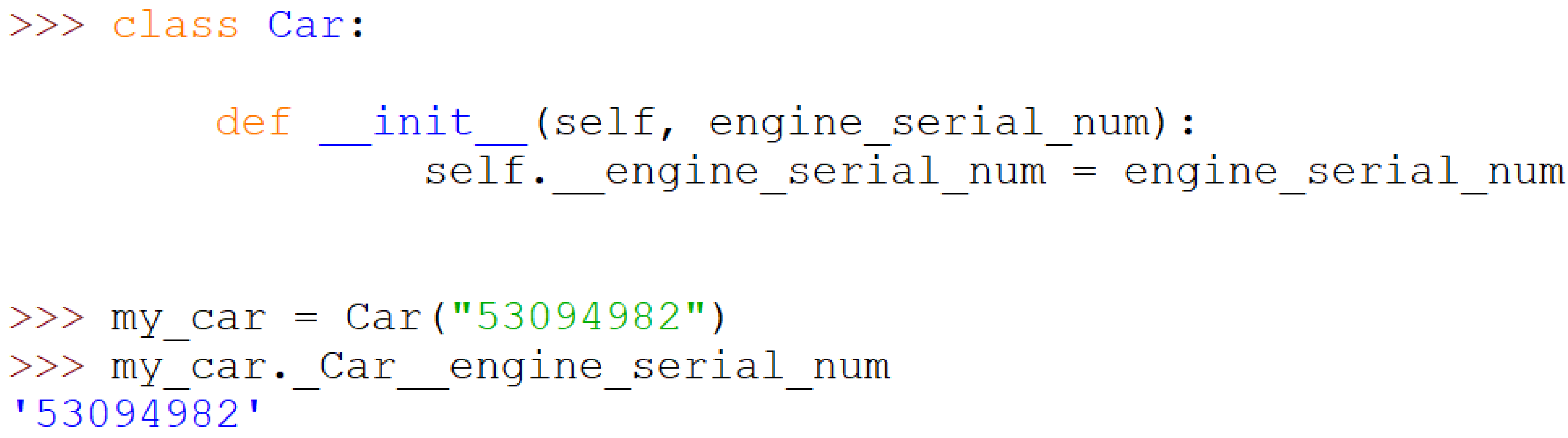
Name Mangling

```
>>> class Car:

        def __init__(self, engine_serial_num):
            self.__engine_serial_num = engine_serial_num


>>> my_car = Car("53094982")
>>> my_car.__engine_serial_num
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    my_car.__engine_serial_num
AttributeError: 'Car' object has no attribute '__engine_serial_num'
```

Name Mangling

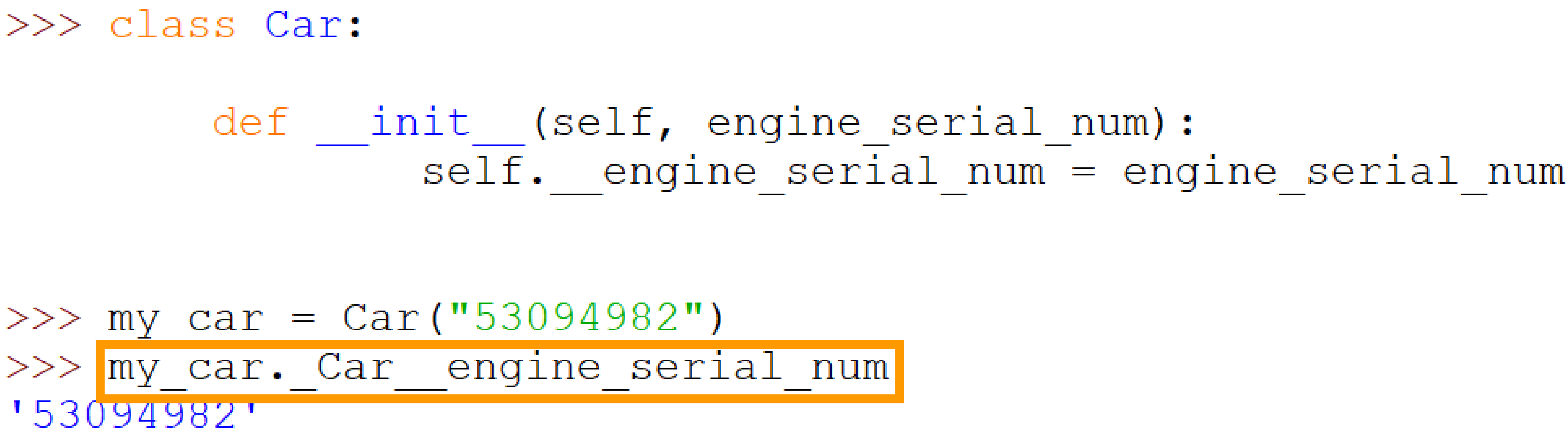


```
>>> class Car:

        def __init__(self, engine_serial_num):
                self.__engine_serial_num = engine_serial_num


>>> my_car = Car("53094982")
>>> my_car._Car__engine_serial_num
'53094982'
```

Name Mangling

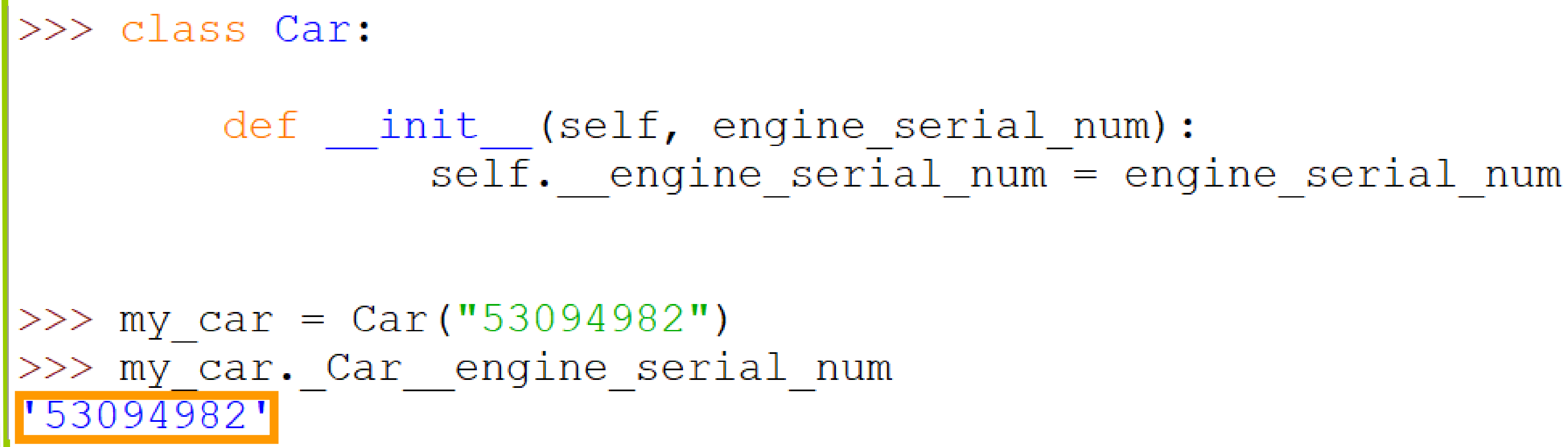


```
>>> class Car:

        def __init__(self, engine_serial_num):
                self.__engine_serial_num = engine_serial_num


>>> my_car = Car("53094982")
>>> my_car._Car__engine_serial_num
'53094982'
```

Name Mangling

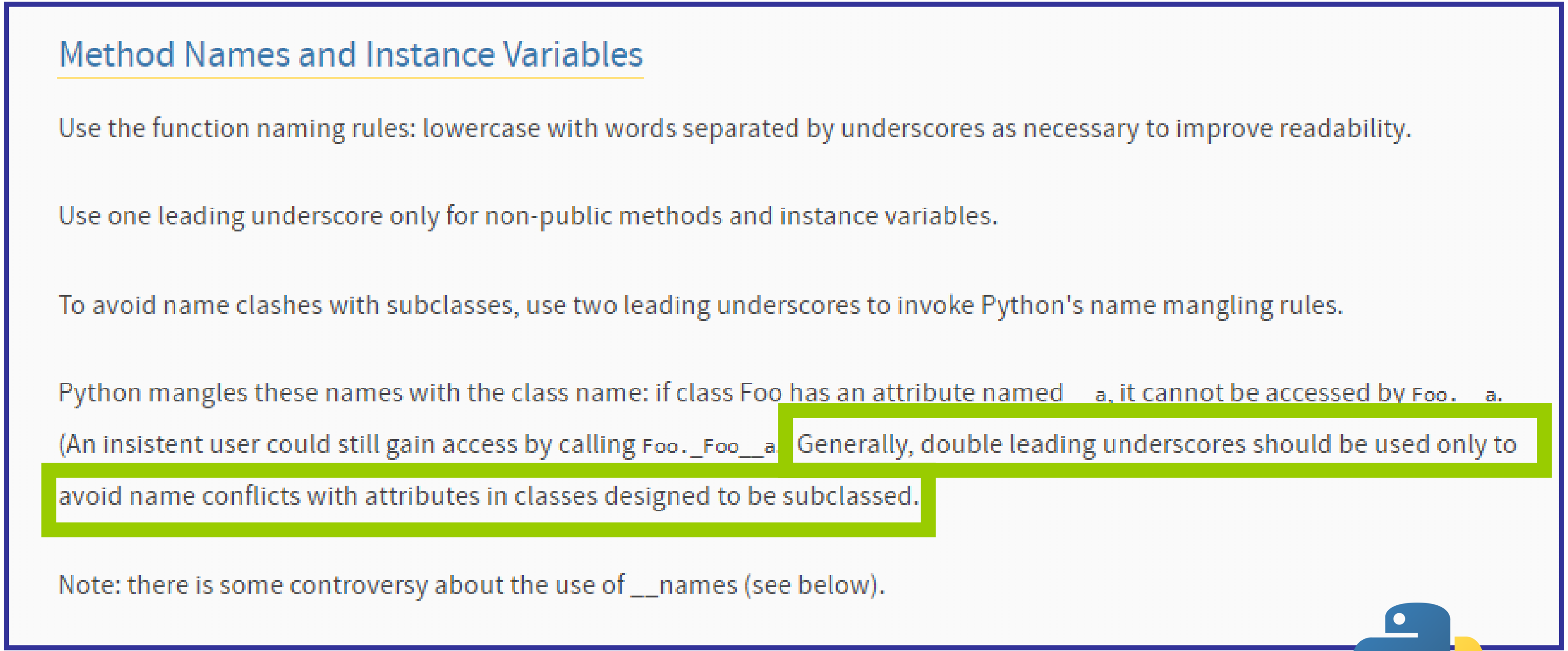


```
>>> class Car:

        def __init__(self, engine_serial_num):
                self.__engine_serial_num = engine_serial_num


>>> my_car = Car("53094982")
>>> my_car._Car__engine_serial_num
'53094982'
```

Name Mangling



Method Names and Instance Variables

Use the function naming rules: lowercase with words separated by underscores as necessary to improve readability.

Use one leading underscore only for non-public methods and instance variables.

To avoid name clashes with subclasses, use two leading underscores to invoke Python's name mangling rules.

Python mangles these names with the class name: if class Foo has an attribute named a, it cannot be accessed by `Foo. a`. (An insistent user could still gain access by calling `Foo._Foo__a`. Generally, double leading underscores should be used only to avoid name conflicts with attributes in classes designed to be subclassed.

Note: there is some controversy about the use of __names (see below).

PEP 8

Name Mangling

✓ __<attribute>

Name Mangling

- ✓ __<attribute>
- ✓ __<attribute>_

Name Mangling

✓ __<attribute>

✓ __<attribute>_

Continue name with multiple words

Name Mangling

✓ __<attribute>

✓ __<attribute>_

✗ __<attribute>__

Name Mangling

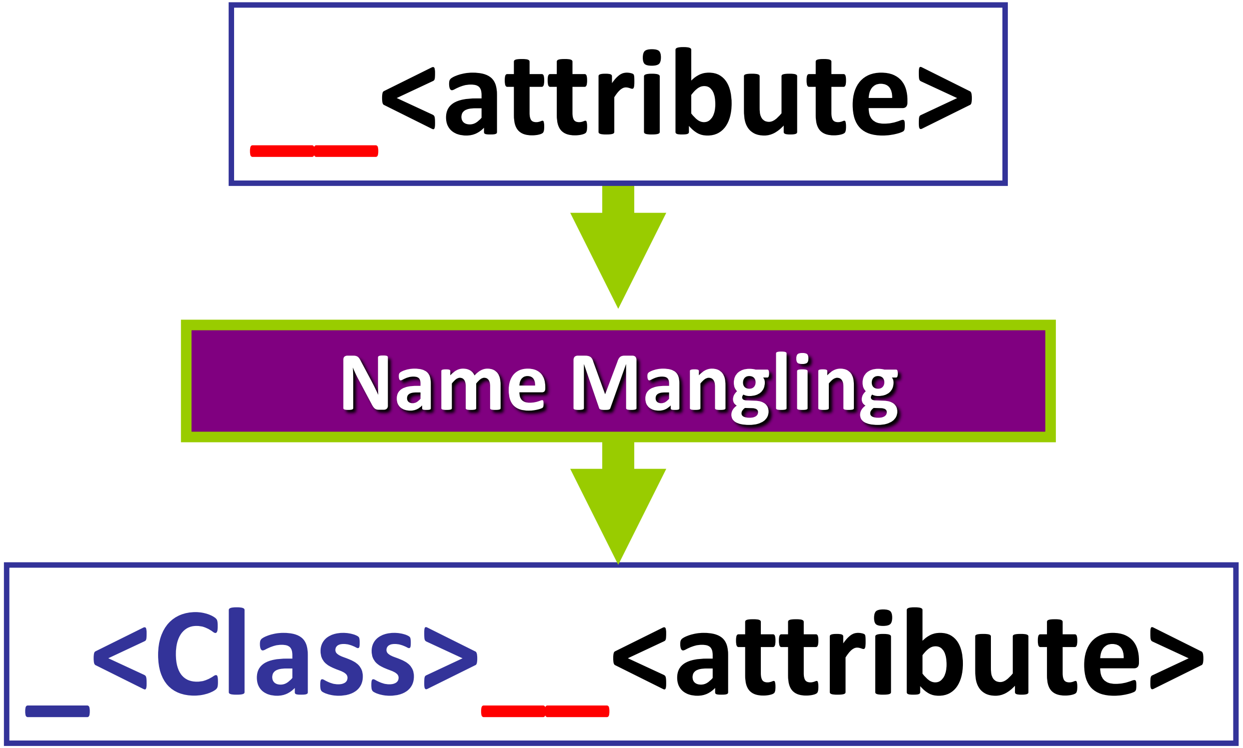
✓ __<attribute>

✓ __<attribute>_

✗ __<attribute>__

Magic
Method
s

Name Mangling



__<attribute>

Name Mangling

_<Class>__<attribute>

Time to Practice