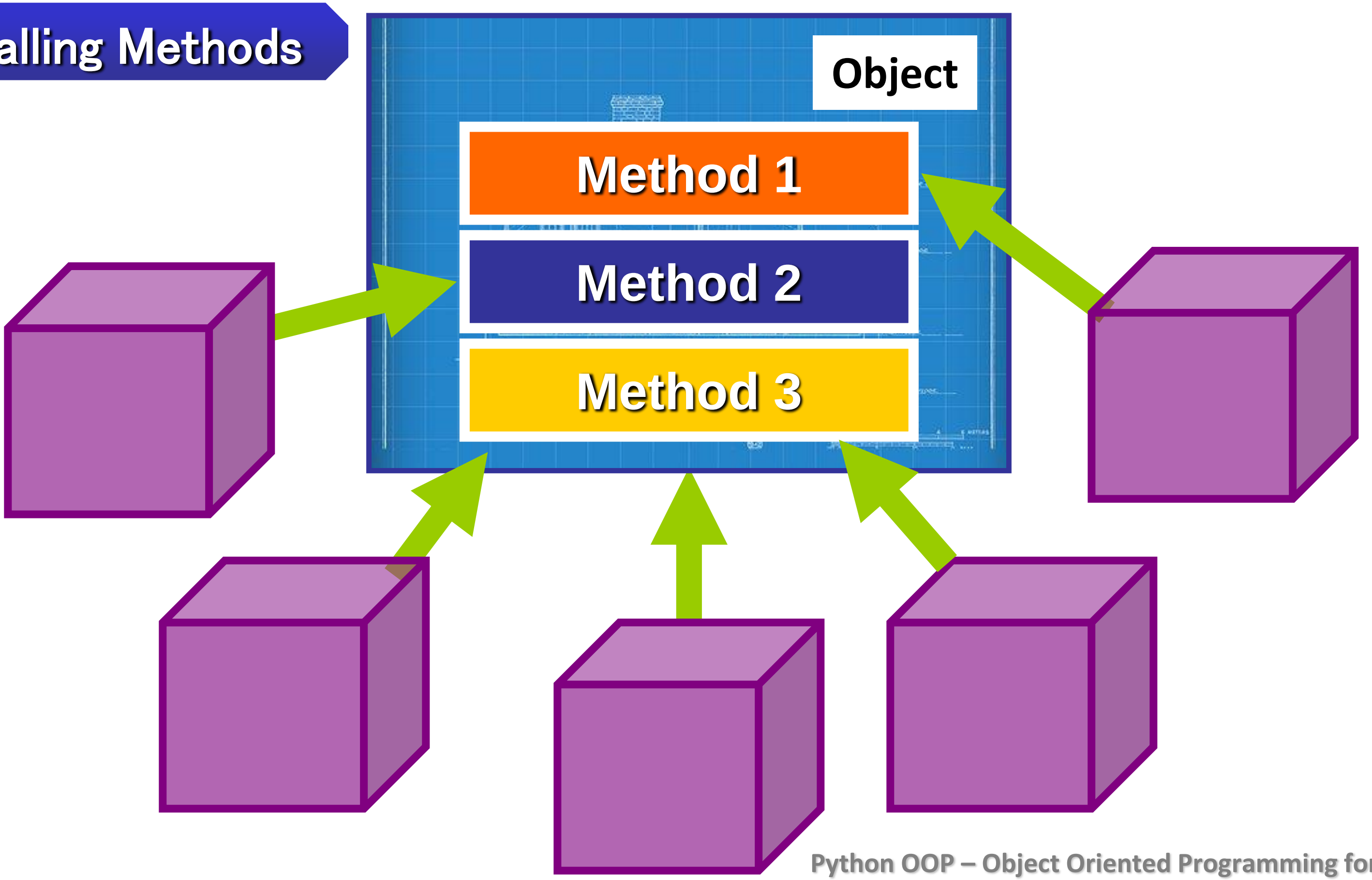
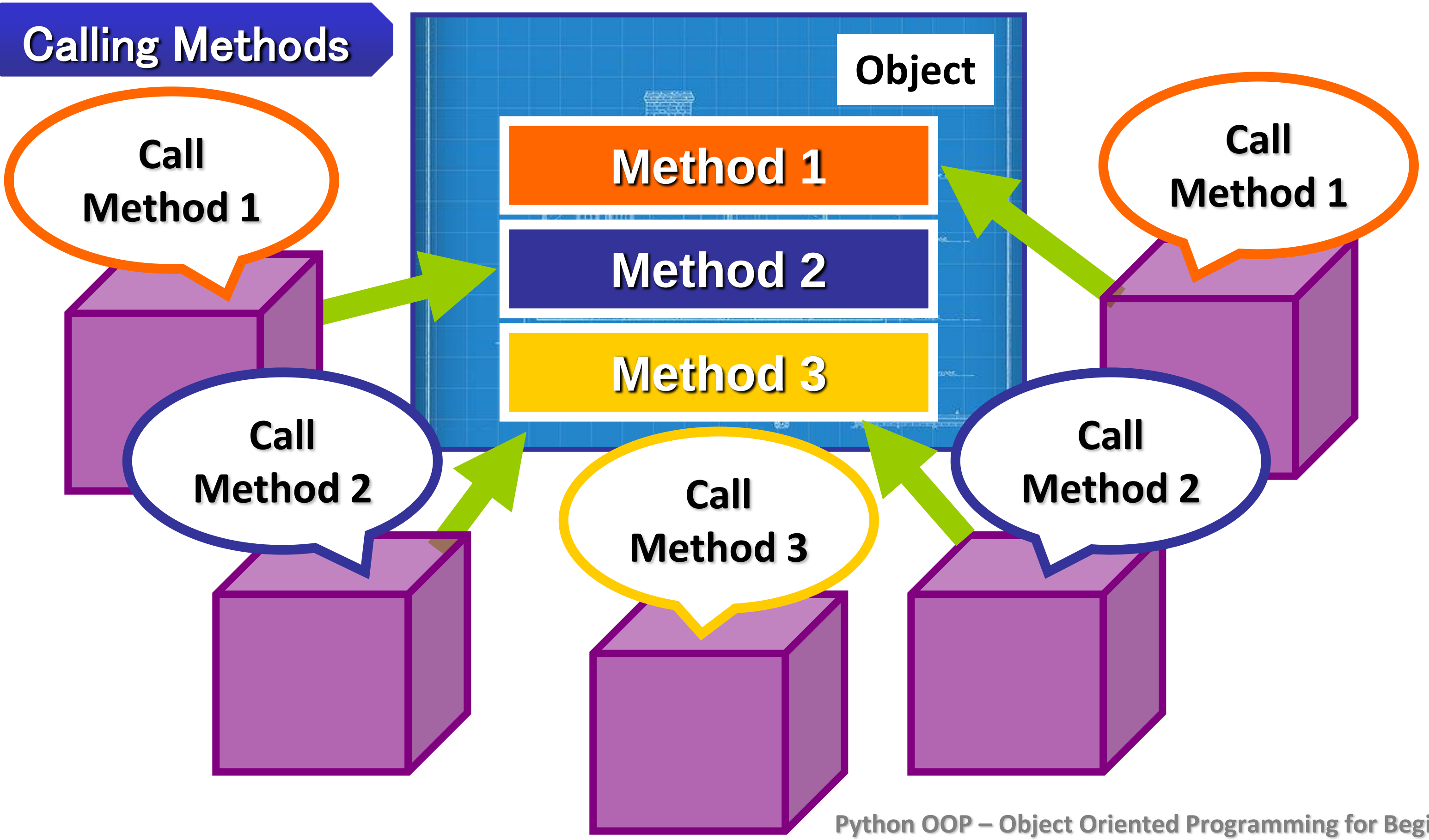
Lecture

Calling Methods

Calling Methods



Calling Methods



Calling Methods

```
<obj_var>.<method>(<params>)
```

Calling Methods

```
<obj_var>.<method>(<params>)
```

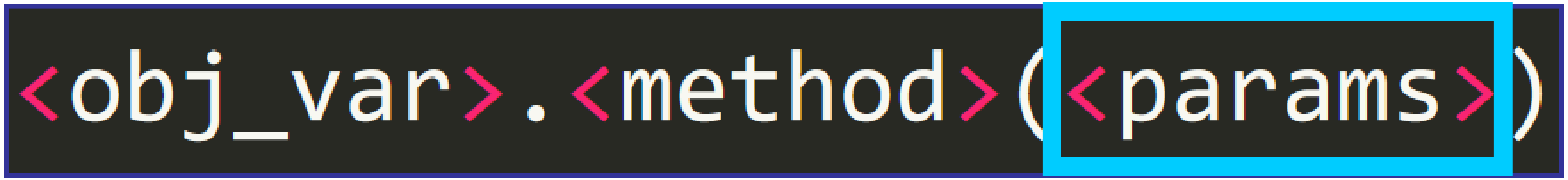
Calling Methods

```
<obj_var>.<method>(<params>)
```

Calling Methods

```
<obj_var>.<method>(<params>)
```

Calling Methods



```
<obj_var>.<method>(<params>)
```

Calling Methods

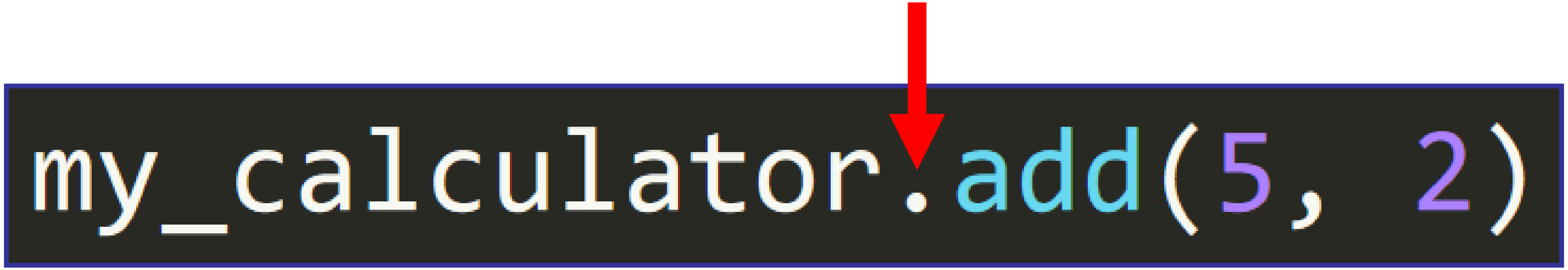
```
my_calculator.add(5, 2)
```

Calling Methods

Variable

```
my_calculator.add(5, 2)
```

Calling Methods



```python
my_calculator.add(5, 2)
```

Calling Methods

Name

```
my_calculator.add(5, 2)
```

Calling Methods

Arguments

```
my_calculator.add(5, 2)
```

Calling Methods

```
def add(self, a, b):
    return a + b
```

```
my_calculator.add(5, 2)
```

Calling Methods

```
def add(self, a, b):
    return a + b
```

```
my_calculator.add(5, 2)
```

Calling Methods

```
def add(self, a, b):
    return a + b
```

```
my_calculator.add(5, 2)
```

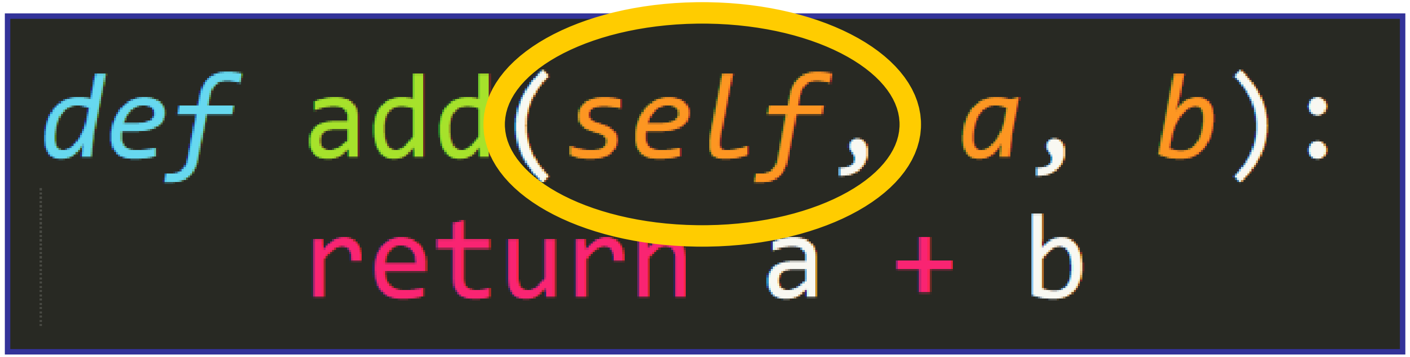
Calling Methods

“skipped”

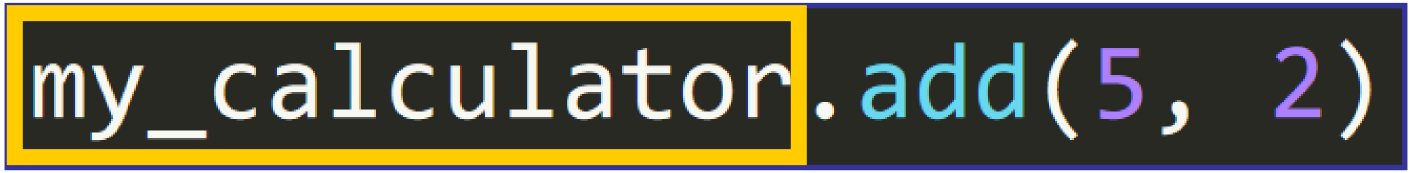
```python
def add(self, a, b):
    return a + b
```

```python
my_calculator.add(5, 2)
```

Calling Methods



```
def add(self, a, b):
    return a + b
```



```
my_calculator.add(5, 2)
```

Calling Methods

```
>>> class Calculator:

        def __init__(self, year, serial_num):
                self.year = year
                self._serial_num = serial_num

        def add(self, a, b):
                return a + b


>>> my_calculator = Calculator(2010, "4251315")
>>> print(my_calculator.add(5, 4))
9
```

Calling Methods

```
>>> class Calculator:

        def __init__(self, year, serial_num):
            self.year = year
            self._serial_num = serial_num

        def add(self, a, b):
            return a + b


>>> my_calculator = Calculator(2010, "4251315")
>>> print(my_calculator.add(5, 4))
9
```

Calling Methods

```
>>> class Calculator:

        def __init__(self, year, serial_num):
                self.year = year
                self._serial_num = serial_num

        def add(self, a, b):
                return a + b


>>> my_calculator = Calculator(2010, "4251315")
>>> print(my_calculator.add(5, 4))
9
```

Calling Methods

self

```
patient1.display_data()
```

```
class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")
```

Calling Methods

self

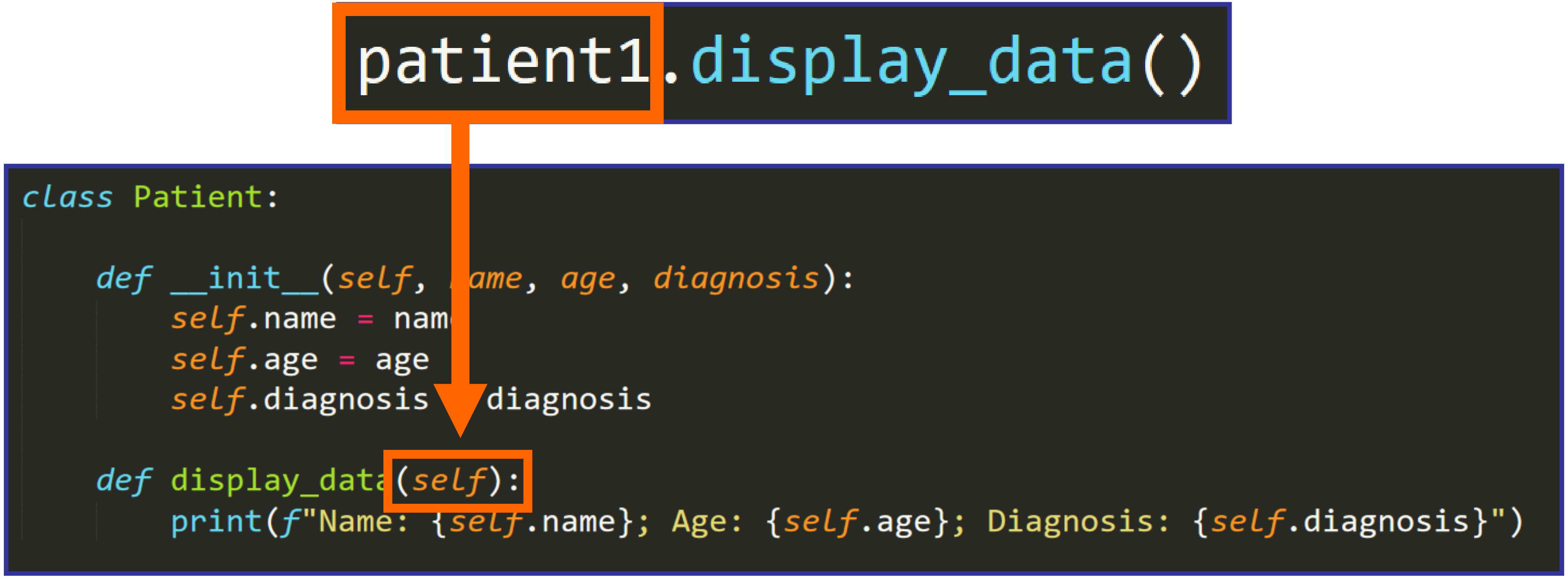
```
patient1.display_data()
```

```
class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")
```

Calling Methods



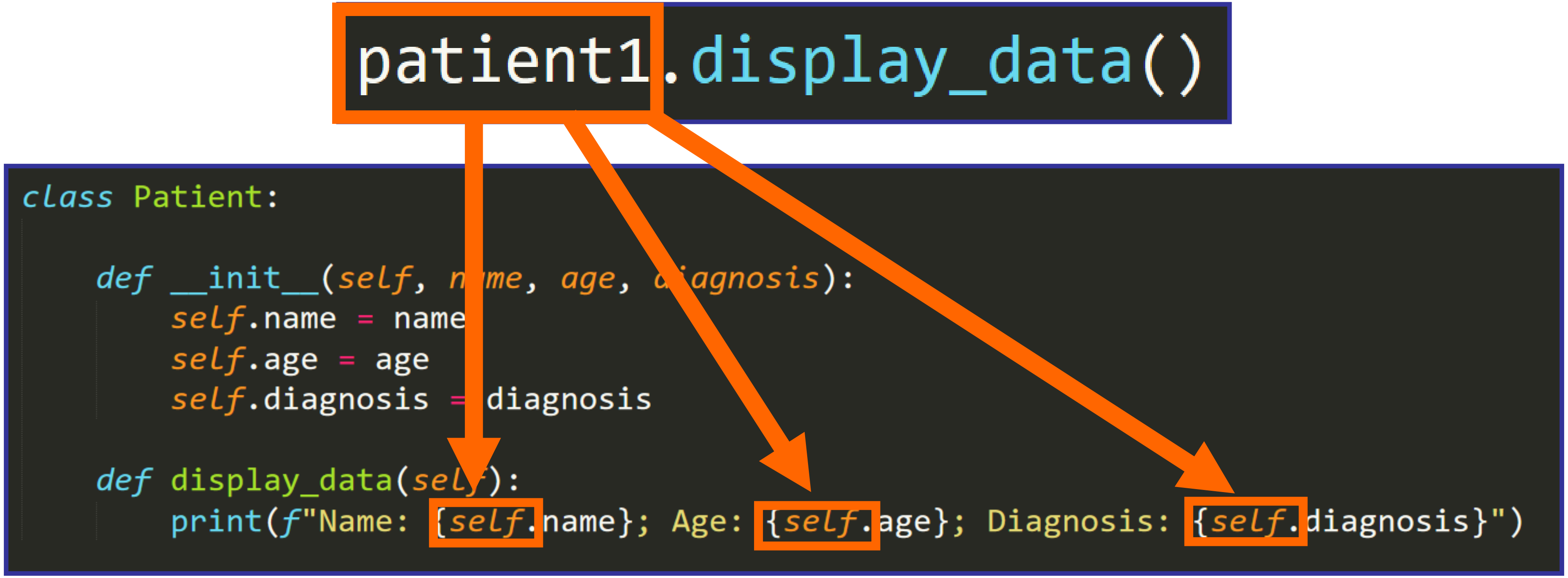
```
patient1.display_data()
```

```
class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")
```

Calling Methods



```
patient1.display_data()
```

```
class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")
```

Calling Methods

```
>>> class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")


>>> patient1 = Patient("Daniel", 56, "Femur Fracture")
>>> patient1.display_data()
Name: Daniel; Age: 56; Diagnosis: Femur Fracture
```

Calling Methods

```
>>> class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")


>>> patient1 = Patient("Daniel", 56, "Femur Fracture")
>>> patient1.display_data()
Name: Daniel; Age: 56; Diagnosis: Femur Fracture
```

Calling Methods

```
>>> class Patient:

    def __init__(self, name, age, diagnosis):
        self.name = name
        self.age = age
        self.diagnosis = diagnosis

    def display_data(self):
        print(f"Name: {self.name}; Age: {self.age}; Diagnosis: {self.diagnosis}")


>>> patient1 = Patient("Daniel", 56, "Femur Fracture")
>>> patient1.display_data()
Name: Daniel; Age: 56; Diagnosis: Femur Fracture
```

Calling Methods

```
<obj_var>.<method>(<params>)
```

Alternative Syntax