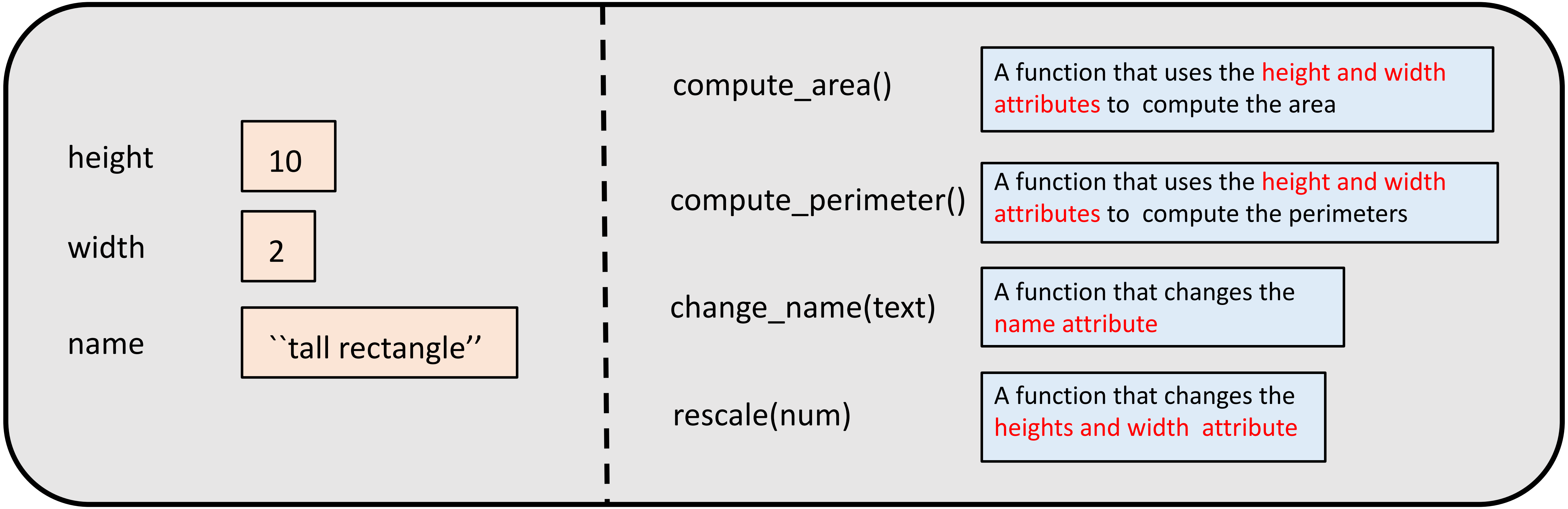
Objects and classes

A Rectangle object

3 ATTRIBUTES

4 METHODS



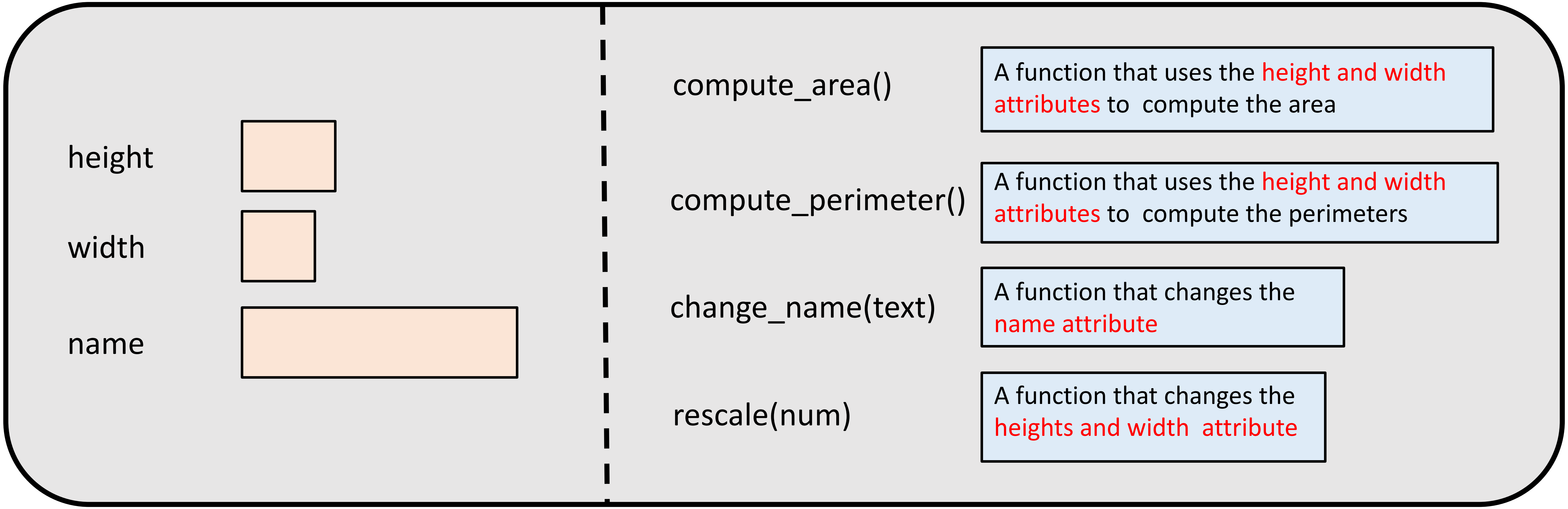
height	10
width	2
name	``tall rectangle''

compute_area()	A function that uses the height and width attributes to compute the area
compute_perimeter()	A function that uses the height and width attributes to compute the perimeters
change_name(text)	A function that changes the name attribute
rescale(num)	A function that changes the heights and width attribute

To create an object you first make a TEMPLATE which describe what are the attribute and methods

3 ATTRIBUTES

4 METHODS



height

width

name

compute_area()

A function that uses the height and width attributes to compute the area

compute_perimeter()

A function that uses the height and width attributes to compute the perimeters

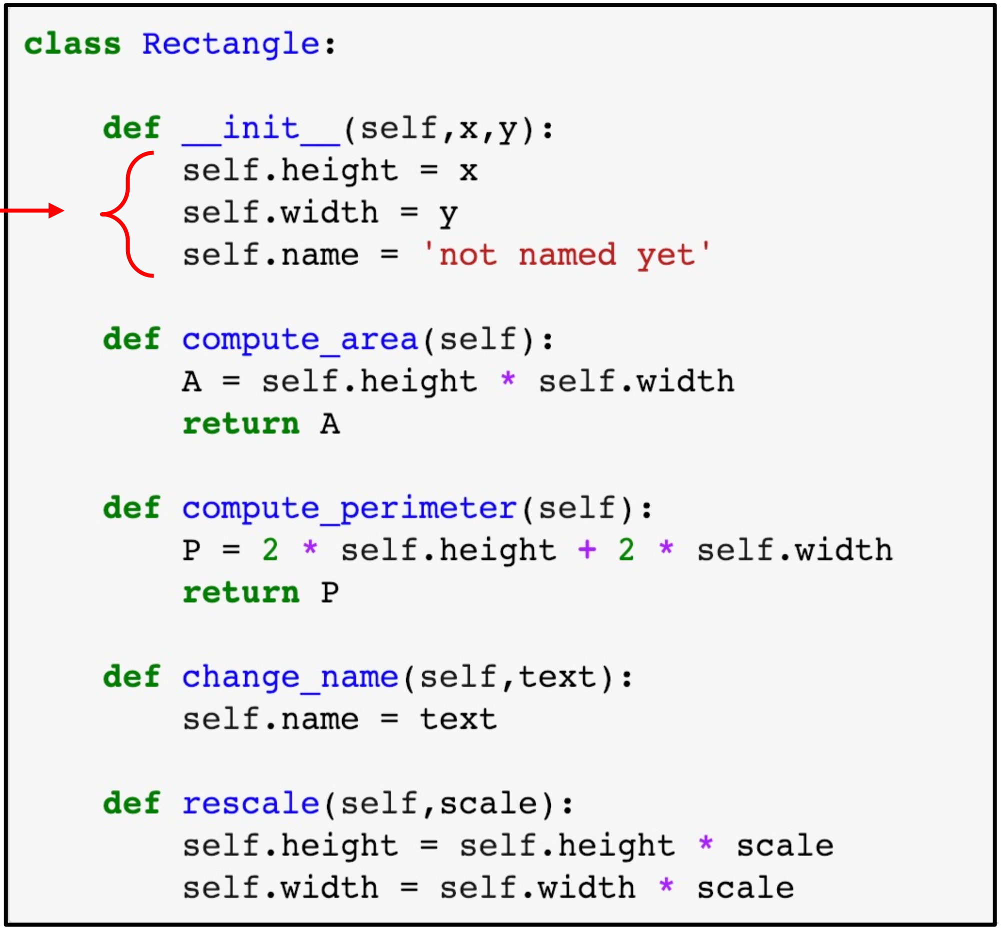
change_name(text)

A function that changes the name attribute

rescale(num)

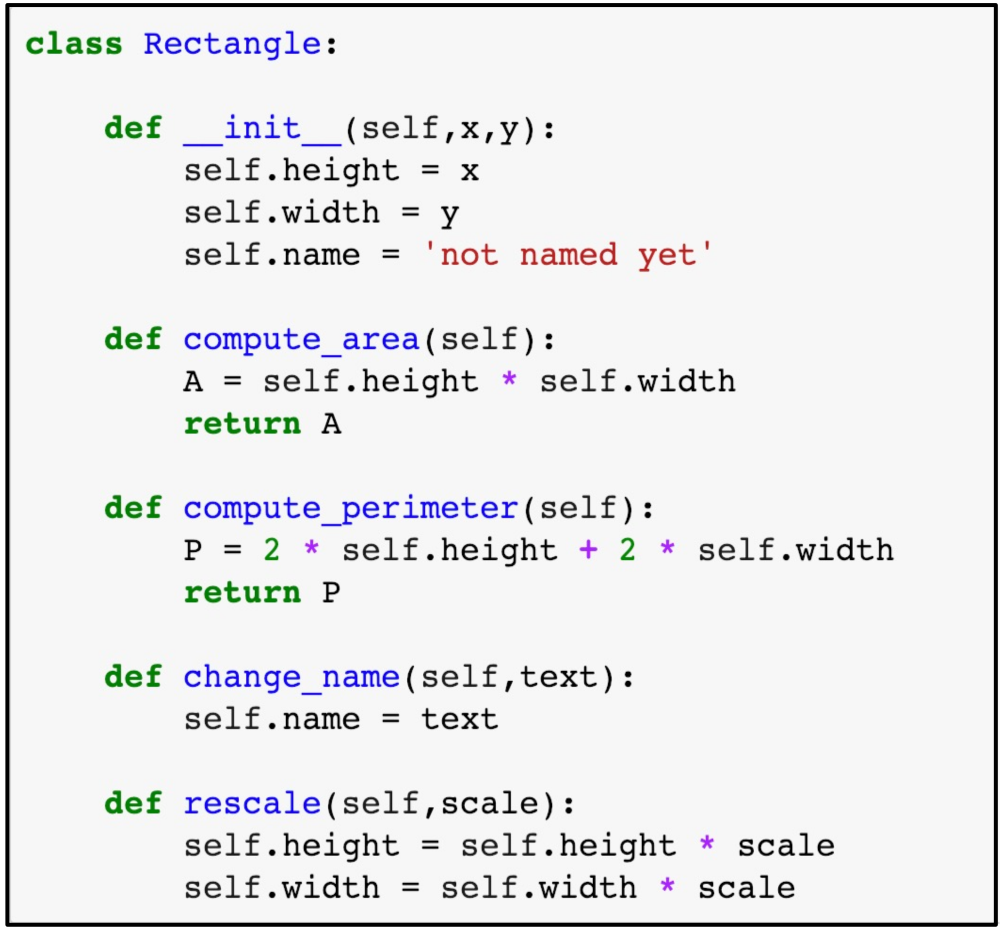
A function that changes the heights and width attribute

3 attributes height, width and name

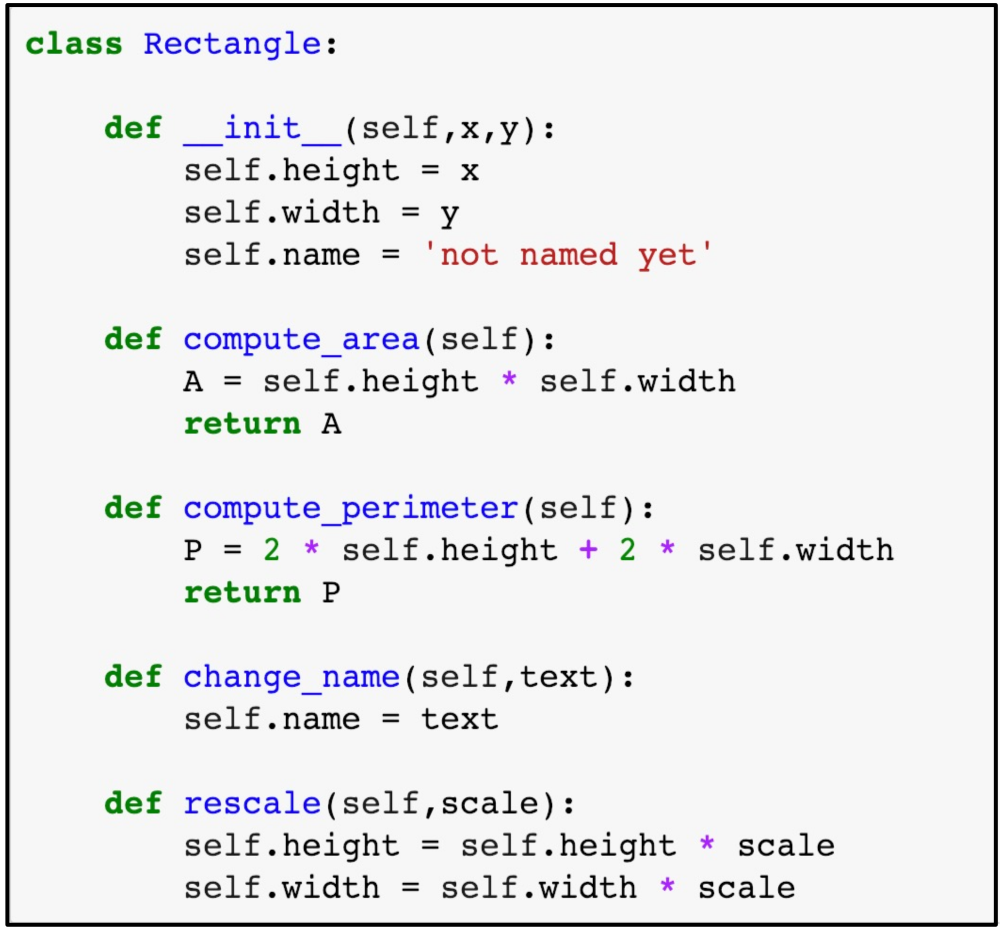


```
class Rectangle:

    def __init__(self,x,y):
        self.height = x
        self.width = y
        self.name = 'not named yet'

    def compute_area(self):
        A = self.height * self.width
        return A

    def compute_perimeter(self):
        P = 2 * self.height + 2 * self.width
        return P

    def change_name(self,text):
        self.name = text

    def rescale(self,scale):
        self.height = self.height * scale
        self.width = self.width * scale
```

4 methods

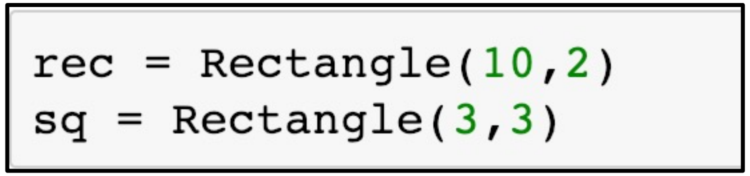


```python
class Rectangle:

    def __init__(self,x,y):
        self.height = x
        self.width = y
        self.name = 'not named yet'

    def compute_area(self):
        A = self.height * self.width
        return A

    def compute_perimeter(self):
        P = 2 * self.height + 2 * self.width
        return P

    def change_name(self,text):
        self.name = text

    def rescale(self,scale):
        self.height = self.height * scale
        self.width = self.width * scale
```



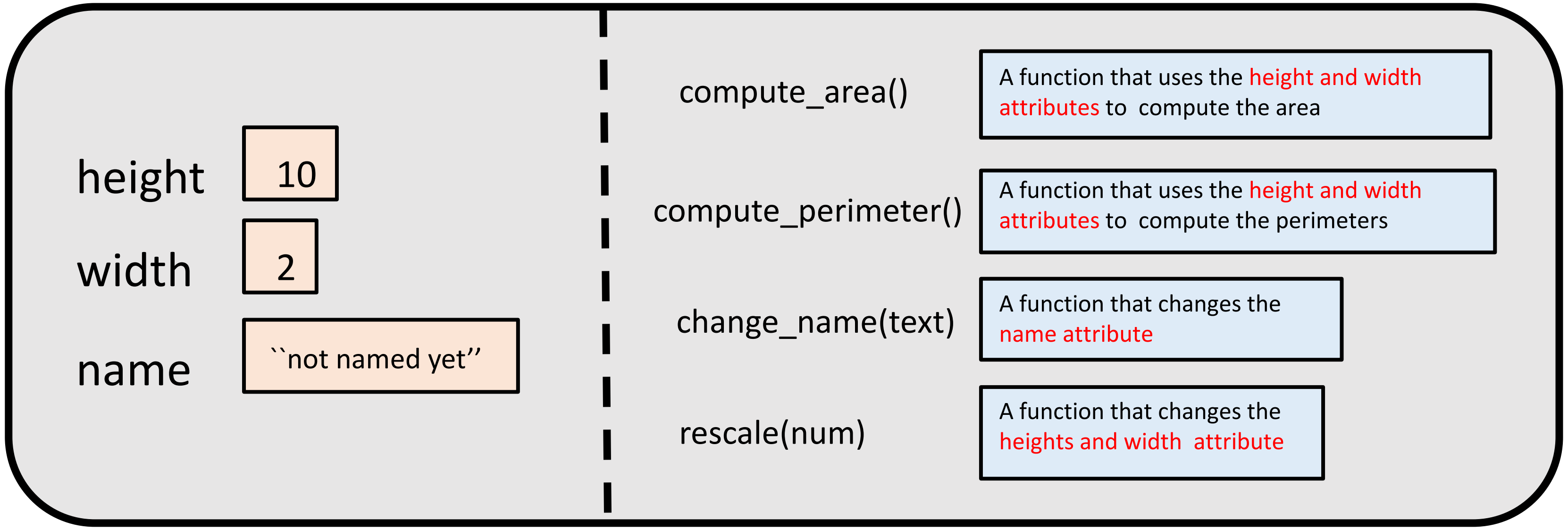
```python
class Rectangle:

    def __init__(self,x,y):
        self.height = x
        self.width = y
        self.name = 'not named yet'

    def compute_area(self):
        A = self.height * self.width
        return A

    def compute_perimeter(self):
        P = 2 * self.height + 2 * self.width
        return P

    def change_name(self,text):
        self.name = text

    def rescale(self,scale):
        self.height = self.height * scale
        self.width = self.width * scale
```

Once the template is made,
You can instantiate multiple objects



```python
rec = Rectangle(10,2)
sq = Rectangle(3,3)
```

rec =



height 10

width 2

name ``not named yet’’

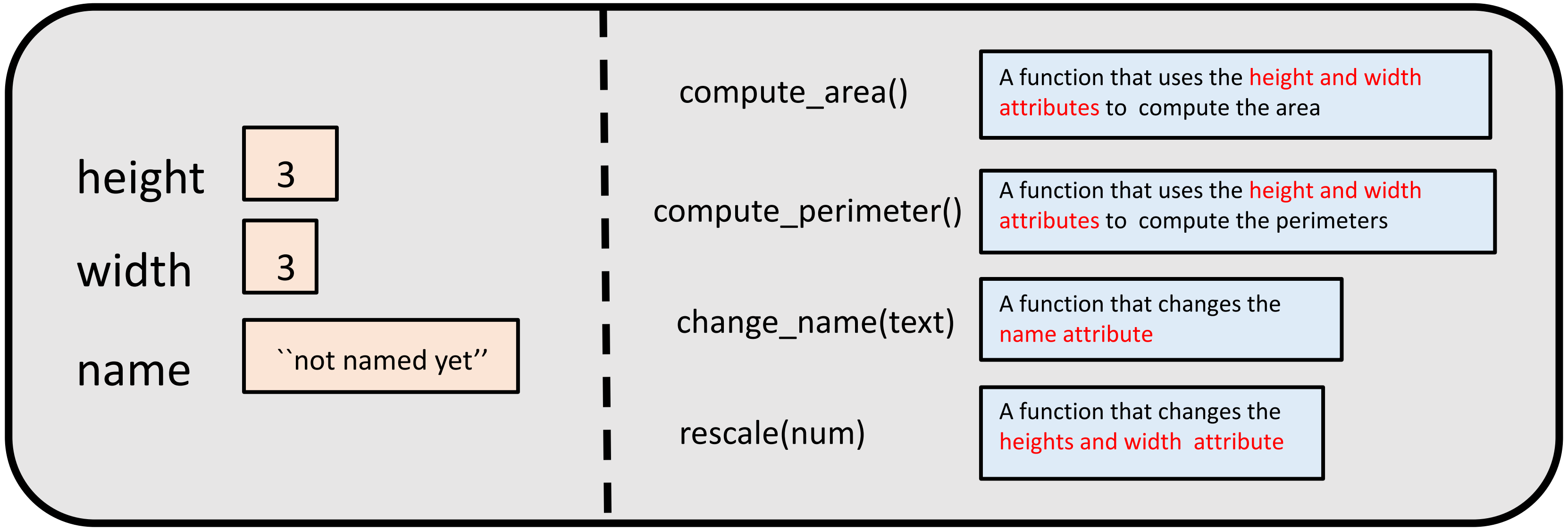
compute_area() A function that uses the height and width attributes to compute the area

compute_perimeter() A function that uses the height and width attributes to compute the perimeters

change_name(text) A function that changes the name attribute

rescale(num) A function that changes the heights and width attribute

sq =



height 3

width 3

name ``not named yet’’

compute_area() A function that uses the height and width attributes to compute the area

compute_perimeter() A function that uses the height and width attributes to compute the perimeters

change_name(text) A function that changes the name attribute

rescale(num) A function that changes the heights and width attribute