## Предоставление атрибутов для чтения и записи

Вы также можете использовать `property` , чтобы предоставить управляемым атрибутам возможность и для чтения и для записи. На практике вам просто нужно предоставить соответствующий метод получения («чтение» он же «getter») и метод установки («запись» он же «setter») для ваших свойств, чтобы создать управляемые атрибуты для чтения и записи.

Допустим, вы хотите, чтобы ваш класс `Square` имел атрибут `.area` . Однако получение стороны и площади в инициализаторе класса квадрата кажется ненужным, потому что вы можете вычислить одно, используя другое. Вот `Square` , который управляет `.side` и `.area` как атрибутами для чтения и записи:

```
class Square:
    def __init__(self, side):
        self.side = side

    @property
    def side(self):
        return self._side

    @side.setter
    def side(self, value):
        self._side = float(value)

    @property
    def area(self):
        return self.side ** 2

    @area.setter
    def area(self, value):
        self.side = value ** 0.5
```

Здесь мы создаем класс `Square` со свойством-атрибутом `.side` для чтения и записи. В этом случае метод получения(getter) просто возвращает значение стороны квадрата. Метод setter преобразует входное значение стороны и присваивает его закрытой переменной `._side` , которую вы используете для хранения окончательных данных.

В этой новой реализации `Square` и его свойства `.side` следует отметить одну тонкую деталь. В этом случае инициализатор класса присваивает входное значение свойству `.side` напрямую, а не сохраняет его в выделенном непубличном атрибуте, таком как `._side` .

Почему? Потому что вам нужно убедиться, что каждое значение, предоставленное как сторона квадрата, включая значение инициализации, проходит через метод установки и преобразуется в число с плавающей запятой.

`Square` также реализует атрибут `.area` как свойство. Метод getter вычисляет площадь, используя сторону квадрата. Метод setter делает нечто любопытное. Вместо сохранения входного значения площади в специальном атрибуте он вычисляет сторону квадрата и записывает результат вновь в свойство `.side` .

```
>>> sq = Square(42)

>>> # Считываем значения
>>> sq.side
42.0

>>> sq.area
1764.0

>>> # записаем новое значение
>>> sq.area = 100

>>> sq.side
10.0
```