

## Предоставление атрибутов только для чтения

Пожалуй самый элементарный вариант использования `property` — предоставить атрибуты только для чтения в ваших классах. Допустим, вам нужен неизменяемый класс `Point`, который не позволяет пользователю изменять исходное значение его координат `x` и `y`. Для достижения этой цели вы можете создать `Person`, как в следующем примере:

```
class Person:
    def __init__(self, name, age):
        self._name = name
        self._age = age

    @property
    def name(self):
        return self._name

    @property
    def age(self):
        return self._age
```

Здесь вы сохраняете входные аргументы в атрибутах `_name` и `_age`. Используем подчеркивания (`_`) в именах для того, чтобы сообщить другим разработчикам, что они являются закрытыми атрибутами и к ним нельзя обращаться с помощью записи через точку, например, в `person.age`. Далее, определяем два метода получения через декоратор `@property`. Теперь у вас есть два свойства только для чтения

```
>>> person = Person('Jack', 33)

>>> # Считываем значения
>>> person.name
Jack
>>> person.age
33

>>> # Пытаемся записать новое значение
>>> person.age = 42
Traceback (most recent call last):
...
AttributeError: can't set attribute
```