

## Предоставление атрибутов только для записи

Вы также можете создать атрибуты только для записи, изменив способ реализации метода получения ваших свойств. Например, вы можете заставить свой метод получения вызывать исключение каждый раз, когда пользователь получает доступ к базовому значению атрибута.

Вот пример обработки паролей со свойством только для записи:

```
class User:
    def __init__(self, name, password):
        self.name = name
        self.password = password

    @property
    def password(self):
        raise AttributeError("Пароль можно только менять, нельзя смотреть")

    @password.setter
    def password(self, plaintext):
        salt = os.urandom(32)
        self._hashed_password = hashlib.pbkdf2_hmac(
            "sha256", plaintext.encode("utf-8"), salt, 100_000
        )
```

Инициализатор класса `User` принимает в качестве аргументов имя пользователя и пароль и сохраняет их в атрибут `.name` и в свойство `.password` соответственно. Мы используем свойство для управления тем, как ваш класс обрабатывает входной пароль. Метод получения вызывает `AttributeError` всякий раз, когда пользователь пытается получить текущий пароль. Это превращает `.password` в атрибут-свойство только для записи:

```
>>> jack = User("Jack", "secret_key")

>>> jack._hashed_password
b'7\x1f+\x02\xc4q\x93\xb6\x98\xb3\r\x9f\x9e\xa4v\nI\xde\x10\x11\x98\xb7\xcf\xff\x9c\x83f\xe4\x07\x8c\xce\xcg'

>>> jack.password
Traceback (most recent call last):
...
  raise AttributeError("Пароль можно только менять, нельзя смотреть")

>>> jack.password = "new_secret"

>>> jack._hashed_password
b'N\xd3f\xe5\x92,\xc4\xe6\xf29g\xe0\x96I\xd1\xf3^\xd6D\xb4\xbd\x89\xc8\x85s\x13\xa6YA\x08\x89\x89'
```

В этом примере мы создаем экземпляр пользователя `john` с начальным паролем через свойство `password`. Метод установки хэширует пароль и сохраняет его в защищенном атрибуте `._hashed_password`. Обратите внимание, что когда вы пытаетесь получить доступ к `.password` напрямую, вы получаете `AttributeError`. Наконец, присвоение нового значения `.password` запускает метод установки и создает новый хешированный пароль.

В методе установки `.password` мы используем `os.urandom()` для генерации 32-байтовой случайной строки в качестве соли вашей хеш-функции. Чтобы сгенерировать хешированный пароль используем `hashlib.pbkdf2_hmac()`. Затем вы сохраняете полученный хешированный пароль в закрытом атрибуте `._hashed_password`. Это гарантирует, что вы никогда не сохраните открытый текстовый пароль в каком-либо извлекаемом атрибуте.