

Getters e Setters

domingo, 9 de outubro de 2022 14:26

Os getters e setters são usados quando temos algum dado que foi alterado na base de dados e está com a tipagem diferente, logo para não precisar fazer um typecast usamos esses 2 métodos.

o getter obtém o valor e o setter ele configura o valor, o getter pega o valor antes que o valor vá até o construtor `__init__`

#getters é representado por `@property` que é um decorador também.

criamos um decorador com o property para pegar o valor que você deseja que seja alterado.

```
class NovaClasse:

    def __init__(self, numero, dobra):
        self.numero = numero
        self.dobra = dobra

    def dobra_valor(self):
        return f"O valor de {self.numero} em dobro é {self.numero*self.dobra}"

    @property
    def numero(self):
        return self._numero

    @numero.setter
    def numero(self, valor):
        if isinstance(valor, str):
            valor = int((valor.replace("R$", "")))
        self._numero = valor
```

criamos uma função com o nome do objeto que queremos pegar e alterar com o decorador property e damos um return com um novo nome para o objeto.

Original

```
def __init__(self, numero, dobra):
    self.numero = numero
    self.dobra = dobra
```

Dentro do decorador getter - @property

```
@property
def numero(self):
    return self._numero
```

e para usar o setter usamos o decorador com o nome da variável e o setter após, criamos uma função igual a do getter com um novo parâmetro, e dentro do setter transformamos o novo getter igual o parâmetro adicionado, além de para modificar usamos o isinstance(novo parâmetro do setter, tipo de variável que queremos modificar (str)).

Na prática:

```
@property
def numero(self):
    return self._numero

@numero.setter
def numero(self, valor):
    if isinstance(valor, str):
        valor = int((valor.replace("R$", "")))
    self._numero = valor
```

Quando usamos o @property estamos fazendo com que um método se torne um atributo ou seja quando for chamar devemos chamar como um atributo.