Pystart.pl

Dekoratory

lekcja dwudziesta siódma

PyStart #27 Dekoratory

Do czego służą dekoratory?

- ➔ Modyfikują działanie funkcji:
  - przechwytują wynik
  - zwracają go w innej postaci
- ➔ Pozwalają na reużywalną zmianę działania funkcji.
- ➔ Do wywołania dekoratora służy symbol @

PyStart #27 Dekoratory

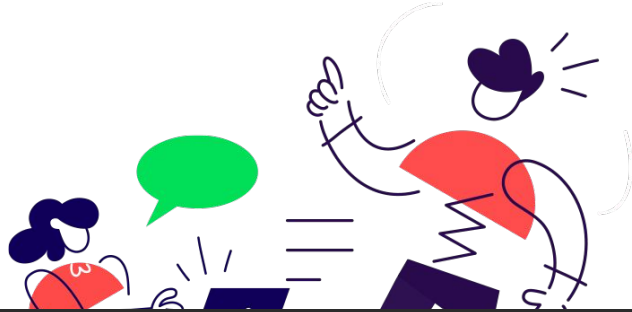
Do czego służą dekoratory?

```python
def to_string(function):
    def wrapper(*args, **kwargs):
        value = function(*args, **kwargs)
        if isinstance(value, list):
            return [str(element) for element in value]
        else:
            return str(value)

    return wrapper

```

PyStart #27 Dekoratory

Do czego służą dekoratory?

```
12  @to_string
13  def return_int():
14      return 1
15
16
17  @to_string
18  def return_list():
19      return [1, 2, 3]
```



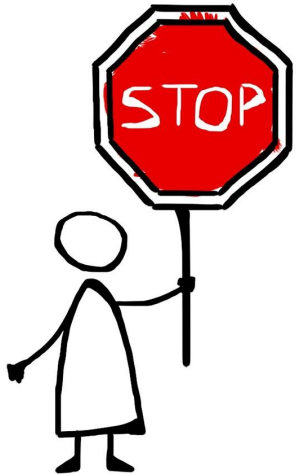
```
1 <class 'str'>
['1', '2', '3']
```

PyStart #27 Dekoratory

Zadania dla nabrania wprawy

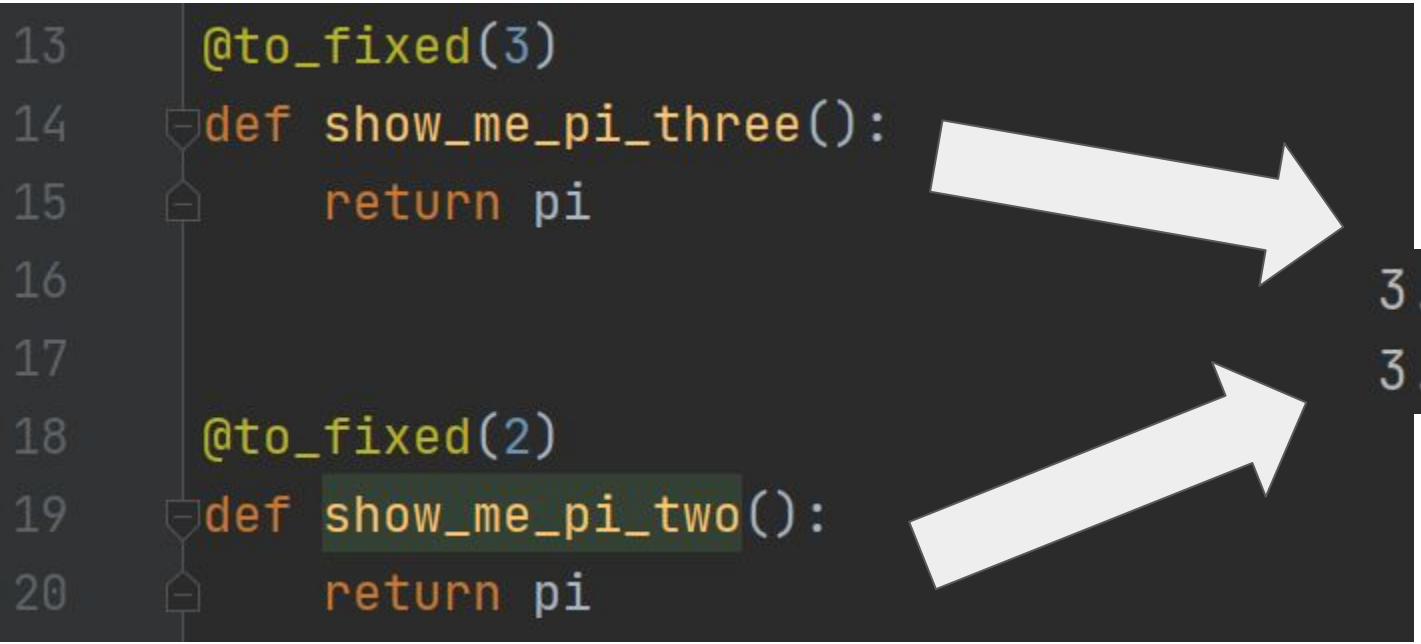
27.1

1. Przygotuj dekorator, który zaokrągli zwracaną z funkcji wartość do dwóch miejsc po przecinku.
2. Przygotuj dekorator, który zwróci informację ile sekund wykonywała się dana funkcja, z jakimi została uruchomiona argumentami. Przyda Ci się do tego funkcja time z biblioteki time.
   from time import time



PyStart #27 Dekoratory

Dekorator z argumentem



PyStart #27 Dekoratory

Dekorator z argumentem

```python
from math import pi


def to_fixed(number):
    def wrapper(func):
        def inner_wrapper(*args, **kwargs):
            return round(func(*args, **kwargs), number)
        return inner_wrapper

    return wrapper
```