

Skillbox

Декораторы с аргументами

Роман Булгаков

Спикер

Декораторы с аргументами

```
def timer_with_precision(precision: int):
    def timer_decorator(func: Callable) -> Callable:
        @functools.wraps(func)
        def wrapped(*args, **kwargs):
            started_at = time.time()
            result = func(*args, **kwargs)
            ended_at = time.time()
            run_time = round(ended_at - started_at, precision)
            print('Функция работала {} секунд(ы)'.format(run_time))
            return result
        return wrapped
    return timer_decorator
```

«Классическая» версия

```
def timer_with_precision(_func=None, *, precision: int = 10):
    def timer_decorator(func: Callable) -> Callable:
        @functools.wraps(func)
        def wrapped(*args, **kwargs):
            started_at = time.time()
            result = func(*args, **kwargs)
            ended_at = time.time()
            run_time = round(ended_at - started_at, precision)
            print('Функция работала {} секунд(ы)'.format(run_time))
            return result
        return wrapped

    if _func is None:
        return timer_decorator
    else:
        return timer_decorator(_func)
```

Модернизированная версия

Skillbox

**Спасибо
за внимание!**