

Занятие №7

Задание №1 (простое)

1. На map, filter, reduce написать программу, которая проверяет, является ли строка палиндромом
2. Написать декоратор, который замеряет время работы функции и выводит его
3. Переписать декоратор так, чтобы его можно было включать и выключать
4. Написать три декоратора, каждый декоратор выводит свое имя, так, чтобы можно было написать их один над другим:

```
@first_decorator
@second_decorator
@third_decorator
def foo(*args, **kwargs):
    print(args, kwargs)
```

Задание №2 (посложнее)

Написать декоратор, который кеширует результат исполнения функции. При повторном вызове с теми же аргументами функция не вызывается, а результат выдается из кэша.

Например:

```
In [4]: @cached
...: def sum(x, y):
...:     result = x + y
...:     print('Function called, result ', result)
...:     return result
```

```
In [5]:
```

```
In [5]: sum(1, 2)
Function called, result  3
Out[5]: 3
```

```
In [6]: sum(1, 2)
Out[6]: 3
```

```
In [7]: sum(3, 5)
Function called, result  8
Out[7]: 8
```

```
In [8]: sum(3, 5)
Out[8]: 8
```

Задание №3 (посложное)

Написать декоратор check, который проверяет типы входных аргументов декорируемой функции и тип результата

Задание №4 (посложное)

Написать декоратор, который “пропускает” из функции только один типа исключения (RightError), все остальные превращая в исключения другого типа(WrongError)