



Декоратор с параметрами: что ты такое?

Здесь мы “сами” создали декоратор.

```
def usage_alert(func):  
    def inner_func(*args, **kwargs):  
        print('Alert!', file=file)  
        return func(*args, **kwargs)  
    return inner_func
```

А здесь мы написали функцию, которая нам делает декораторы.

```
def usage_alert(file=sys.stdout):  
    def decorator(func):  
        def inner_func(*args, **kwargs):  
            print('Alert!', file=file)  
            return func(*args, **kwargs)  
        return inner_func  
    return decorator
```

синтаксический сахар 

декорируемая функция 

декоратор 

функция которая нам делает декоратор 

Подставляем декоратор на место после @

```
@usage_alert  
def make_greeting(name):  
    return f'Hello {name}!'
```

Вызвали нашу функцию, чтобы она нам
подставила декоратор на место после @

*

вызов функции

```
@usage_alert(sys.stderr)  
def make_greeting(name):  
    return f'Hello {name}!'
```

*

`sum([1, 2])` ⇒ 3

`usage_alert(sys.stderr)` ⇒ декоратор