

# Декораторы функций

**Декоратор** представляет собой некоторую функцию, аргументом которой является другая функция. Декоратор предназначен для добавления дополнительного функционала к данной функции без изменения содержимого последней.

## Создать декоратор

```
import time

def testTime(fn):
    def wrapper(*args, *kwargs):
        st = time.time()
        fn(*args, *kwargs)
        dt = time.time() - st
        print(f'Время работы: {dt} сек')
    return wrapper
```

```
@testTime
def getNOD(a, b):
    while a != b:
        if a > b: a -= b
        else: b -= a
    return a

@testTime
def getFastNOD(a, b):
    if a < b: a, b = b, a
    while b: a, b = b, a % b
    return a
```

**Замыкание** - когда вложенная функция ссылается на контекст внешней функции и потому имеет возможность обращаться к локальным переменным этого внешнего контекста