

Информатика, 7 химический класс, 2017 год. Python - 3

## 5. Функции

Почти всегда код, который вы написали, удобно оборачивать в отдельные блоки, которые потом вызываются в программе. Такие блоки называются *функциями*, а разделение кода на функции -- *функциональная декомпозиция*.

Функции *принимают параметры и возвращают значения*. В примере ниже параметры функции *add* -- числа *a* и *b*, а функция *мяукнуть* вызывается без параметров. Возвращаемое значение функции *add* -- значение переменной *c*, у функции *мяукнуть* возвращаемого значения нет.

```
In [1]: def мяукнуть():  
        print("мяу!")  
  
        def add(a, b):  
            c = a + b  
            return c
```

```
In [2]: мяукнуть()  
  
мяу!
```

```
In [3]: add(3, 7)  
  
Out[3]: 10
```

Разумеется, значения функций можно передавать в другие функции, а функции можно вызывать из других функций. Например, давайте вычислим сумму квадратов двух чисел, используя только функцию сложения и возведения в квадрат.

```
In [4]: def square(a):  
        return a * a  
  
        def square_sum(a, b):  
            return add(square(a), square(b))  
  
        print(square_sum(3, 4))  
  
25
```

Комментарий к задачам: обратите внимание, что все функции должны *возвращать* ответ, а не печатать его, если явно не указано иное (как в задаче 2).

## Задание 1

Реализуйте функцию *power(x, n)* для возведения числа *x* в степень *n*. Оператором "две звёздочки" пользоваться запрещено.

```
In [ ]: # ваш код
```

## Задание 2

Реализуйте функцию `print_triange(n)`. Функция печатает прямоугольный треугольник из звёздочек, лежащий на боку. Например, если  $n=3$ , то функция должна вывести такой треугольник:

```
In [ ]: *
      **
      ***
```

```
In [ ]: # ваш код
```

## Задание 3

Реализуйте функцию `mode(ar)`, которая находит самое часто встречающееся число в массиве. Все числа в данном массиве больше 1 и меньше 100.

```
In [ ]: # ваш код
```

## Задание 4

Напишите тесты к первым трём задачам (и к пятой, если справились к ней). Тесты представляют из себя несколько наборов входных данных, и несколько условных операторов (if), нужных, чтобы проверить ответ. Ниже написан пример трёх тестов для первой задачи. Напишите не менее трёх тестов к каждой задаче. Кратко поясните словами, чем они отличаются друг от друга.

```
In [ ]: # first task test
ans1 = power(7, 2)
ans2 = power(10, 5)
ans3 = power(1, 10)
if(ans1 == 49 and ans2 == 100000 and ans3 == 1):
    print("All tests passed")
else:
    print("Some of the tests doesn't work")
```

```
In [ ]: # ваш код
```

## Задание 5(\*)

*Ханойские башни*. Прочитайте статью [https://ru.wikipedia.org/wiki/Ханойская\\_башня](https://ru.wikipedia.org/wiki/Ханойская_башня) ([https://ru.wikipedia.org/wiki/Ханойская\\_башня](https://ru.wikipedia.org/wiki/Ханойская_башня)). Реализуйте функцию, которая принимает как параметр количество дисков на первом штыре и выводит последовательность операций, нужных, чтобы перенести все диски на третий штырь.

```
In [ ]: # ваш код
```