

# Функции высшего порядка

Семинар 7

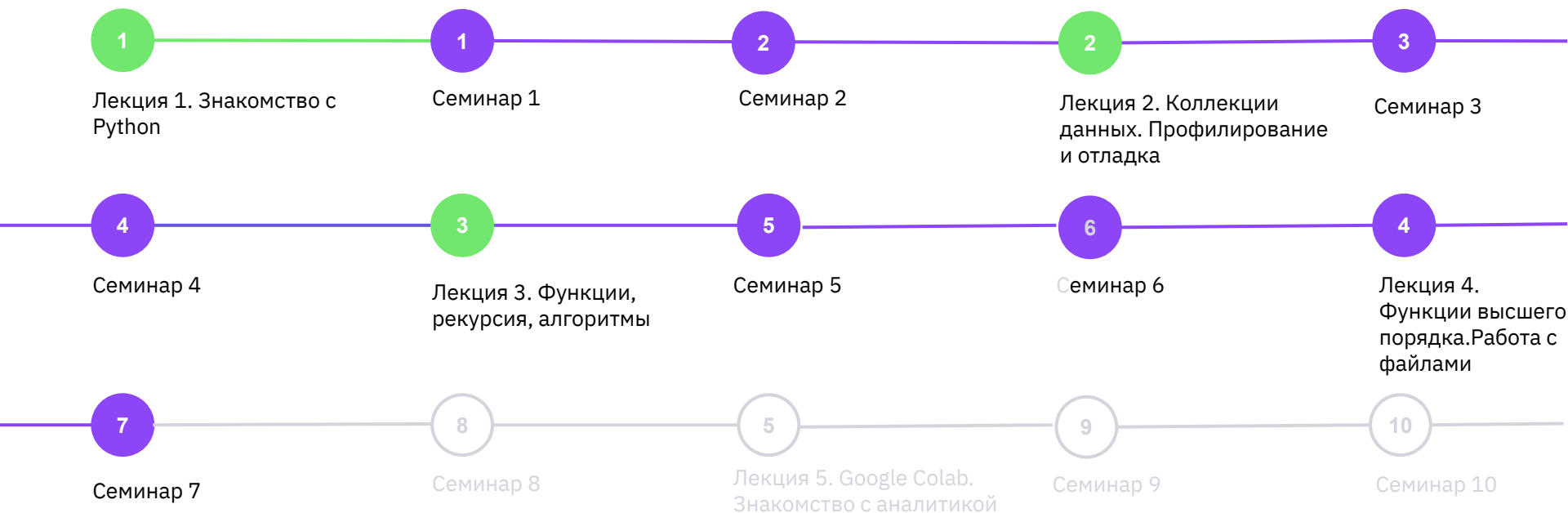




# Содержание урока



## План курса





## Что будет на уроке сегодня



Закрепить полученные знания с лекции по теме функции высшего порядка



Обсуждение домашнего задания



Решение в группе задач



Домашнее задание



Подведение итогов

## Домашнее задание

Задание	Пример
<p><b>Задача 1:</b> Заполните массив элементами арифметической прогрессии. Её первый элемент, разность и количество элементов нужно ввести с клавиатуры. Формула для получения n-го члена прогрессии: <math>a_n = a_1 + (n-1) * d</math>. Каждое число вводится с новой строки.</p>	<p><b>Ввод:</b> 7 2 5</p> <p><b>Вывод:</b> 7 9 11 13 15</p>
<p><b>Задача 2:</b> Определить индексы элементов массива (списка), значения которых принадлежат заданному диапазону (т.е. не меньше заданного минимума и не больше заданного максимума)</p>	<p><b>Ввод:</b> [-5, 9, 0, 3, -1, -2, 1, 4, -2, 10, 2, 0, -9, 8, 10, -9, 0, -5, -5, 7]</p> <p><b>Вывод:</b> [1, 9, 13, 14, 19]</p>





# Практика



## Задача №1. Решение в группах

У вас есть код, который вы не можете менять (так часто бывает, когда код в глубине программы используется множество раз и вы не хотите ничего сломать):

```
transformation = <???
```

```
values = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29] # или любой другой список
```

```
transormed_values = list(map(transformation, values))
```

Единственный способ вашего взаимодействия с этим кодом - посредством задания функции transformation.

Однако вы поняли, что для вашей текущей задачи вам не нужно никак преобразовывать список значений, а нужно получить его как есть.

Напишите такое лямбда-выражение transformation, чтобы transformed\_values получился копией values.

**Пример ввода и вывода данных представлены на следующем слайде**



20 минут



## Задача №1. Решение в группах

### Ввод:

```
values = [1, 23, 42, 'asdfg']  
transformed_values = list(map(trasformation, values))  
if values == transformed_values:  
    print('ok')  
else:  
    print('fail')
```

### Вывод:

ok





## Задача №1. Общее обсуждение

У вас есть код, который вы не можете менять (так часто бывает, когда код в глубине программы используется множество раз и вы не хотите ничего сломать):

```
transformation = <???
```

```
values = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29] # или любой другой список
```

```
transformed_values = list(map(transformation, values))
```

Единственный способ вашего взаимодействия с этим кодом - посредством задания функции transformation.

Однако вы поняли, что для вашей текущей задачи вам не нужно никак преобразовывать список значений, а нужно получить его как есть.

Напишите такое лямбда-выражение transformation, чтобы transformed\_values получился копией values.



5 минут



## Задача №2. Решение в группах

Планеты вращаются вокруг звезд по эллиптическим орбитам. Назовем самой далекой планетой ту, орбита которой имеет самую большую площадь. Напишите функцию `find_farthest_orbit(list_of_orbits)`, которая среди списка орбит планет найдет ту, по которой вращается самая далекая планета. Круговые орбиты не учитывайте: вы знаете, что у вашей звезды таких планет нет, зато искусственные спутники были запущены на круговые орбиты. Результатом функции должен быть кортеж, содержащий длины полуосей эллипса орбиты самой далекой планеты. Каждая орбита представляет из себя кортеж из пары чисел - полуосей ее эллипса. Площадь эллипса вычисляется по формуле  $S = \pi \cdot a \cdot b$ , где  $a$  и  $b$  - длины полуосей эллипса. При решении задачи используйте списочные выражения. Подсказка: проще всего будет найти эллипс в два шага: сначала вычислить самую большую площадь эллипса, а затем найти и сам эллипс, имеющий такую площадь. Гарантируется, что самая далекая планета ровно одна

**Пример ввода и вывода данных представлены на следующем слайде**



20 минут



## Задача №2. Решение в группах

### Ввод:

```
orbits = [(1, 3), (2.5, 10), (7, 2), (6, 6), (4, 3)]  
print(*find_farthest_orbit(orbits))
```

### Вывод:

```
2.5 10
```



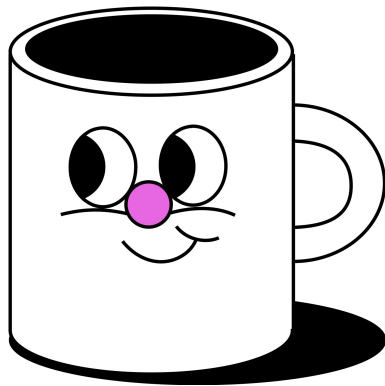
## Задача №2. Общее обсуждение

Планеты вращаются вокруг звезд по эллиптическим орбитам. Назовем самой далекой планетой ту, орбита которой имеет самую большую площадь. Напишите функцию `find_farthest_orbit(list_of_orbits)`, которая среди списка орбит планет найдет ту, по которой вращается самая далекая планета. Круговые орбиты не учитывайте: вы знаете, что у вашей звезды таких планет нет, зато искусственные спутники были запущены на круговые орбиты. Результатом функции должен быть кортеж, содержащий длины полуосей эллипса орбиты самой далекой планеты. Каждая орбита представляет из себя кортеж из пары чисел - полуосей ее эллипса. Площадь эллипса вычисляется по формуле  $S = \pi \cdot a \cdot b$ , где  $a$  и  $b$  - длины полуосей эллипса. При решении задачи используйте списочные выражения. Подсказка: проще всего будет найти эллипс в два шага: сначала вычислить самую большую площадь эллипса, а затем найти и сам эллипс, имеющий такую площадь. Гарантируется, что самая далекая планета ровно одна



5 минут

## Перерыв



<<5:00->>



## Задача №3. Решение в группах

Напишите функцию `same_by(characteristic, objects)`, которая проверяет, все ли объекты имеют одинаковое значение некоторой характеристики, и возвращают `True`, если это так. Если значение характеристики для разных объектов отличается - то `False`. Для пустого набора объектов, функция должна возвращать `True`. Аргумент `characteristic` - это функция, которая принимает объект и вычисляет его характеристику.

### Ввод:

```
values = [0, 2, 10, 6]
```

```
if same_by(lambda x: x % 2, values):
```

```
    print('same')
```

```
else:
```

```
    print('different')
```

### Вывод:

```
same
```



20 минут



## Задача №3. Общее обсуждение

Напишите функцию `same_by(characteristic, objects)`, которая проверяет, все ли объекты имеют одинаковое значение некоторой характеристики, и возвращают `True`, если это так. Если значение характеристики для разных объектов отличается - то `False`. Для пустого набора объектов, функция должна возвращать `True`. Аргумент `characteristic` - это функция, которая принимает объект и вычисляет его характеристику.

### Ввод:

```
values = [0, 2, 10, 6]
```

```
if same_by(lambda x: x % 2, values):
```

```
    print('same')
```

```
else:
```

```
    print('different')
```

### Вывод:

```
same
```



5 минут



Вопросы?

Вопросы?



Вопросы?







# Домашнее задание

### Задание

**Задача 1:** Винни-Пух попросил Вас посмотреть, есть ли в его стихах ритм. Поскольку разобраться в его кричалках не настолько просто, насколько легко он их придумывает, Вам стоит написать программу. Винни-Пух считает, что ритм есть, если число слогов (т.е. число гласных букв) в каждой фразе стихотворения одинаковое. Фраза может состоять из одного слова, если во фразе несколько слов, то они разделяются дефисами. Фразы отделяются друг от друга пробелами. Стихотворение Винни-Пух вбивает в программу с клавиатуры. В ответе напишите “Парам пам-пам”, если с ритмом все в порядке и “Пам парам”, если с ритмом все не в порядке

**Ввод:**

пара-ра-рам рам-пам-папам па-ра-па-дам

**Вывод:**

Парам пам-пам



### Задание

**Задача 2:** Напишите функцию `print_operation_table(operation, num_rows=6, num_columns=6)`, которая принимает в качестве аргумента функцию, вычисляющую элемент по номеру строки и столбца. Аргументы `num_rows` и `num_columns` указывают число строк и столбцов таблицы, которые должны быть распечатаны. Нумерация строк и столбцов идет с единицы (подумайте, почему не с нуля). Примечание: бинарной операцией называется любая операция, у которой ровно два аргумента, как, например, у операции умножения.

#### Ввод:

```
print_operation_table(lambda x, y: x * y)
```

#### Вывод:

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36





## Рефлексия



**Был урок полезен вам?**



**Узнали вы что-то новое?**



**Что было сложно?**



Спасибо за внимание!