

Разбор домашнего задания

Роман Булгаков

Спикер курса

Skillbox

Задача «Свой zip»

Входные данные:

- строка
- кортеж чисел

Выходные данные:

- генератор из пар «символ — число»

Строка: abc

Кортеж: (10, 20, 30, 40)

Результат:

Генератор (a, 10) (b, 20) (c, 30)

Рекурсия

Роман Булгаков

Спикер курса

Skillbox

Задача «Факториал числа»

Условие задачи: найти факториал числа без использования циклов.

$$5! = 1 * 2 * 3 * 4 * 5$$

$$5! = 5 * \underbrace{(4 * 3 * 2 * 1)}_{4!} = 5 * \underline{4!}$$

$$\underline{4!} = 4 * 3 * 2 * 1 = 4 * \underline{3!}$$

$$\underline{3!} = 3 * 2 * 1 = 3 * \underline{2!}$$

$$\underline{2!} = 2 * \underline{1!}$$

$$\underline{1!} = 1$$

Решение



Общая формула:

$$n! = n * (n - 1)!$$

$$\underline{1!} = 1$$

$$\underline{2!} = 2 * \underline{1!} = 2 * 1 = 2$$

$$\underline{3!} = 3 * \underline{2!} = 3 * 2 = 6$$

$$\underline{4!} = 4 * \underline{3!} = 4 * 6 = 24$$

$$\underline{5!} = 5 * \underline{4!} = 5 * 24 = 120$$

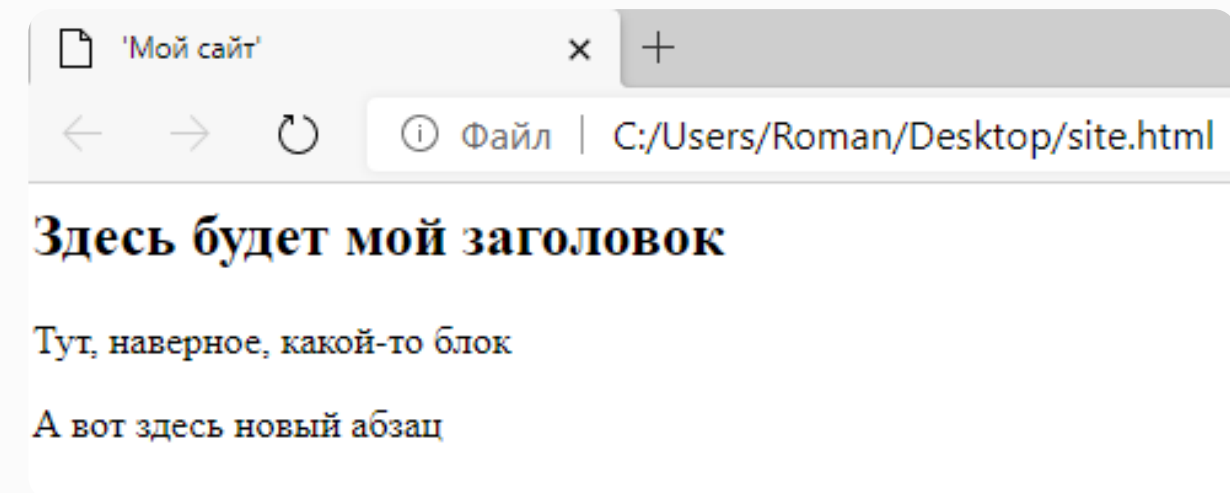
Задача «Код сайта»

Входные данные:

- словарь — структура сайта
- искомый ключ

Выходные данные:

- значение искомого ключа
(None, если такого ключа нет)



```
site = {  
    'html': {  
        'head': {  
            'title': 'Мой сайт'  
        },  
        'body': {  
            'h2': 'Здесь будет мой заголовок',  
            'div': 'Тут, наверное, какой-то блок',  
            'p': 'А вот здесь новый абзац'  
        }  
    }  
}
```

Ошибки рекурсии

```
def factorial(num):  
    fact_n_minus_1 = factorial(num - 1)  
    return num * fact_n_minus_1
```

Неправильное оформление выхода

```
def factorial(num):  
    if num == 1:  
        return 1  
    fact_n_minus_1 = factorial(num)  
    return num * fact_n_minus_1
```

Вызов с неправильными параметрами

Передача изменяемых и неизменяемых данных в функцию

Роман Булгаков

Спикер курса

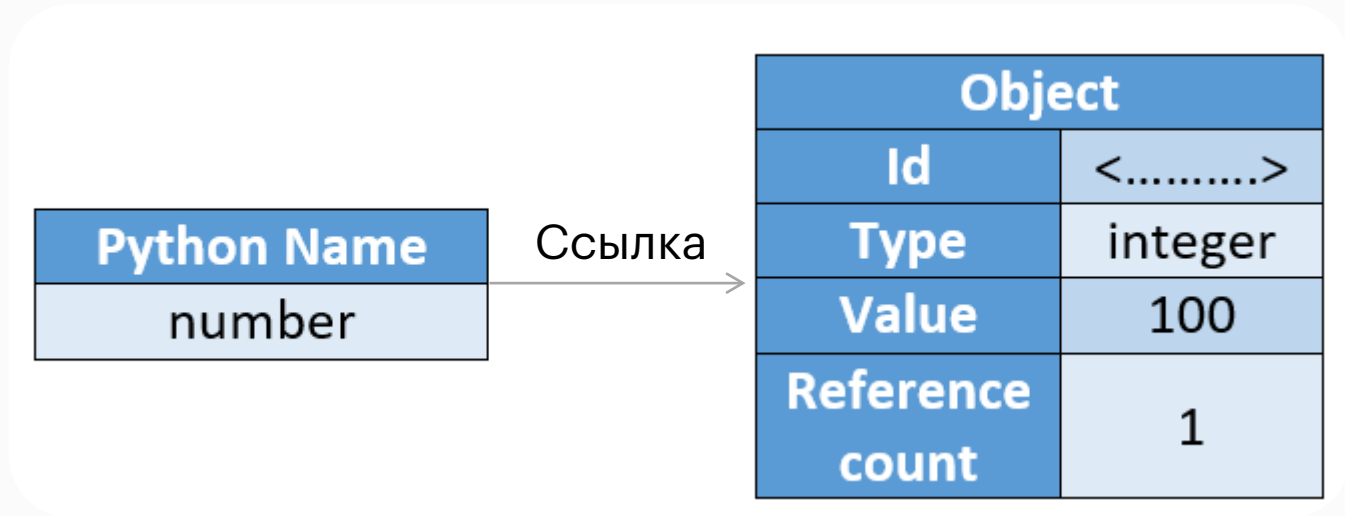
Skillbox

Объекты в Python

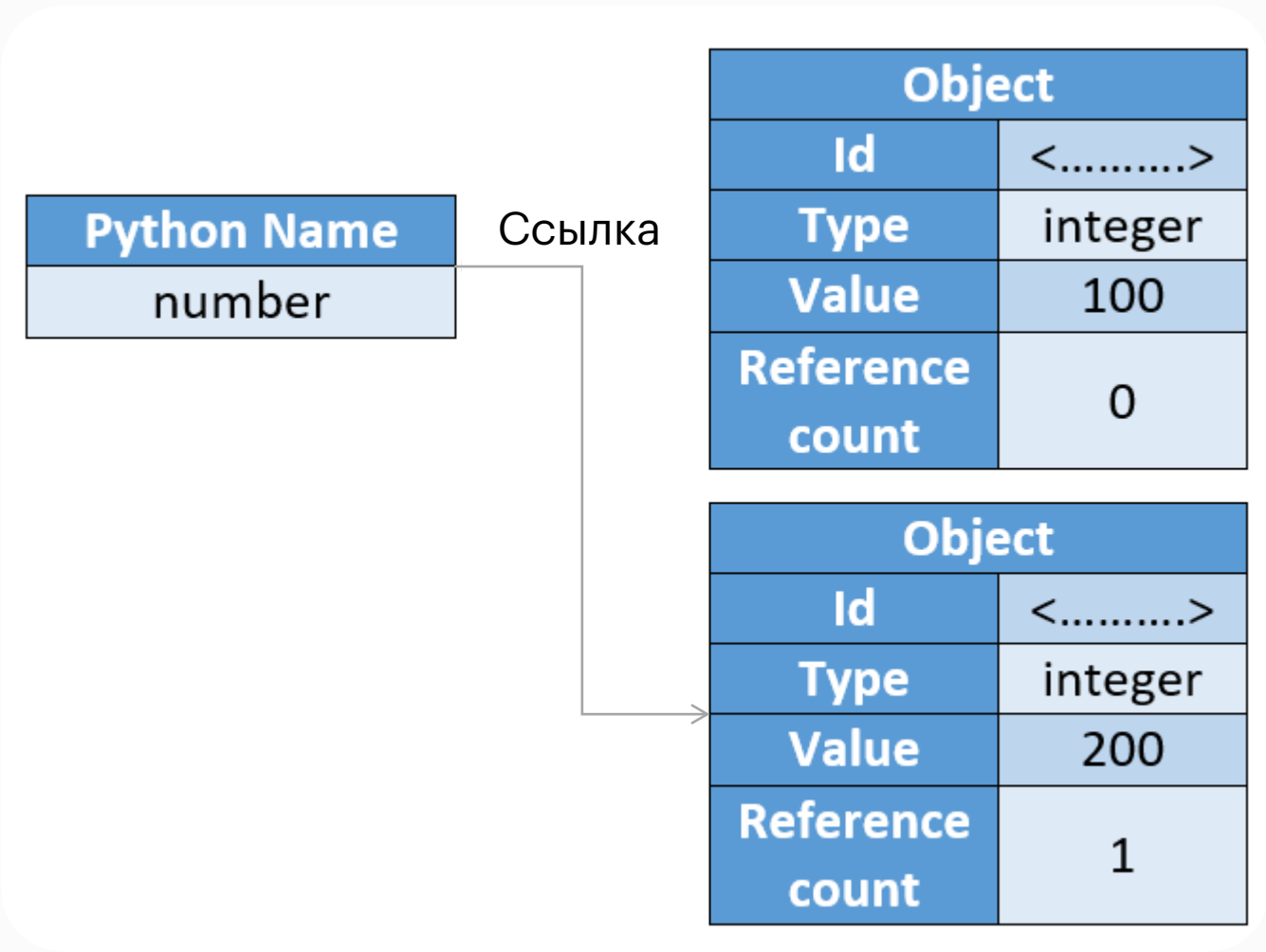
number = 100



Имя Объект



number = 200



Типы данных в Python

Неизменяемые (immutable) типы данных:

int, float, str, tuple, bool

try_to_change_values(my_list, number)

Значение

```
def try_to_change_values(some_list, num):  
    for i_index, i_val in enumerate(some_list):  
        some_list[i_index] += 10  
    num += 10
```

Новый объект с именем num

Изменяемые (mutable) типы данных:

list, dict, set

try_to_change_values(my_list, number)

Ссылка

```
def try_to_change_values(some_list, num):  
    for i_index, i_val in enumerate(some_list):  
        some_list[i_index] += 10  
    num += 10
```

Тот же объект с именем some_list

Именованные аргументы и значения по умолчанию

Роман Булгаков

Спикер курса

Skillbox

Задача «Работа с файлом»

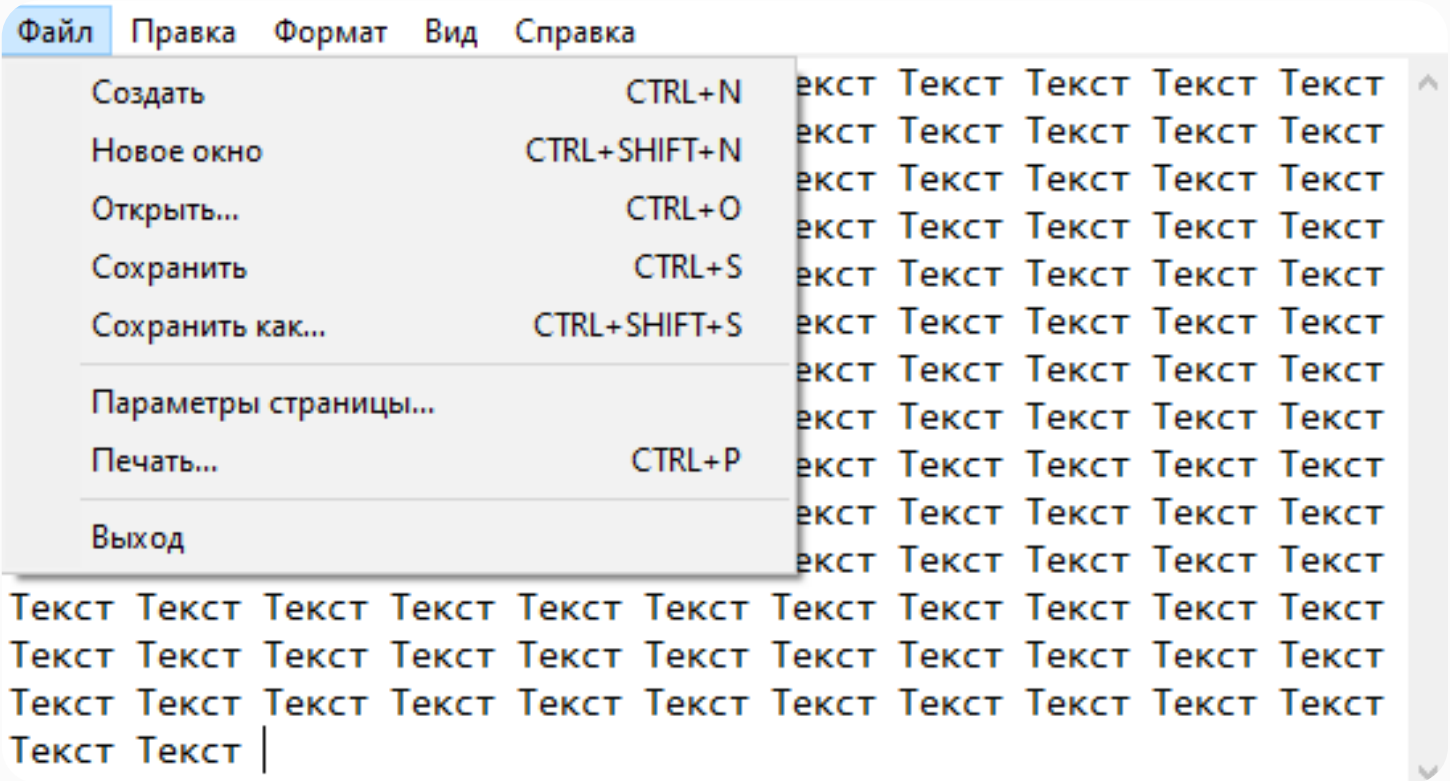
Входные данные:

функция, в которую передаётся:

- вопрос
- сообщение о неверном вводе
- количество попыток

Выходные данные:

- если «да» — вернуть 1; если «нет» — вернуть 0
- «Осталось попыток» и «Неверный ввод»



Итоги урока

- `def ask_user(
 question,
 complaint="Неверный ввод.",
 retries=4):`
- `ask_user('Записать файл?', retries=2)`
- *Ошибка 1:*
`ask_user(retries=2, 'Записать файл?')`
- *Ошибка 2:*
`ask_user('Записать файл',
 question='Удалить файл?')`

Аргументы `*args` и `**kwargs`

Роман Булгаков

Спикер курса

Skillbox

Задача «Финансовый документ»

Условие задачи:

- величина налога
- функция с параметрами:
 - налог
 - цены
 - доп. информация

Выходные данные:

- сумма цен с учётом налога и доп. информация



Использование *args и **kwargs

```
def print_storage(name, good_1=None, good_2=None, good_3=None):  
    print('Storage name:', name)  
    print(good_1, "is stored in section 1")  
    print(good_2, "is stored in section 2")  
    print(good_3, "is stored in section 3")
```

Заранее прописанные параметры

```
def print_storage(name, *args):  
    print('Storage name:', name)  
    for i_sec, i_good in enumerate(args):  
        print("{} is stored in section {}".format(  
            i_good, i_sec + 1  
        ))
```

Параметры передаются «кучей»

Итоги модуля

- `def factorial(num):`
`return num * factorial(num - 1)`
- `number = 100` — тип immutable

{
Имя

{
Объект

`my_lst = [1, 2, 3]` — тип mutable
- `def ask_user(`
 `question,`
 `complaint="Неверный ввод.",`
 `retries=4):`

 ...
 `ask_user('Записать файл?', retries=2)`

