



Функции

использование встроенных и создание собственных



Олег Булыгин



Олег Булыгин

- Преподаватель на курсах “Основы языка программирования Python”, “Продвинутый Python”, “Python для анализа данных” в Нетологии
- Начальник бюро планирования и управления в АО “НПО автоматики”

О чём мы поговорим сегодня

3

1. Что такое функция?
2. Объявление функций в Python
3. Параметры функции
4. Области видимости



Что такое
функция?

Что такое **функция**?

5

01 В математике

это соответствие между элементами; то как значение одной величины определяет значение другой;

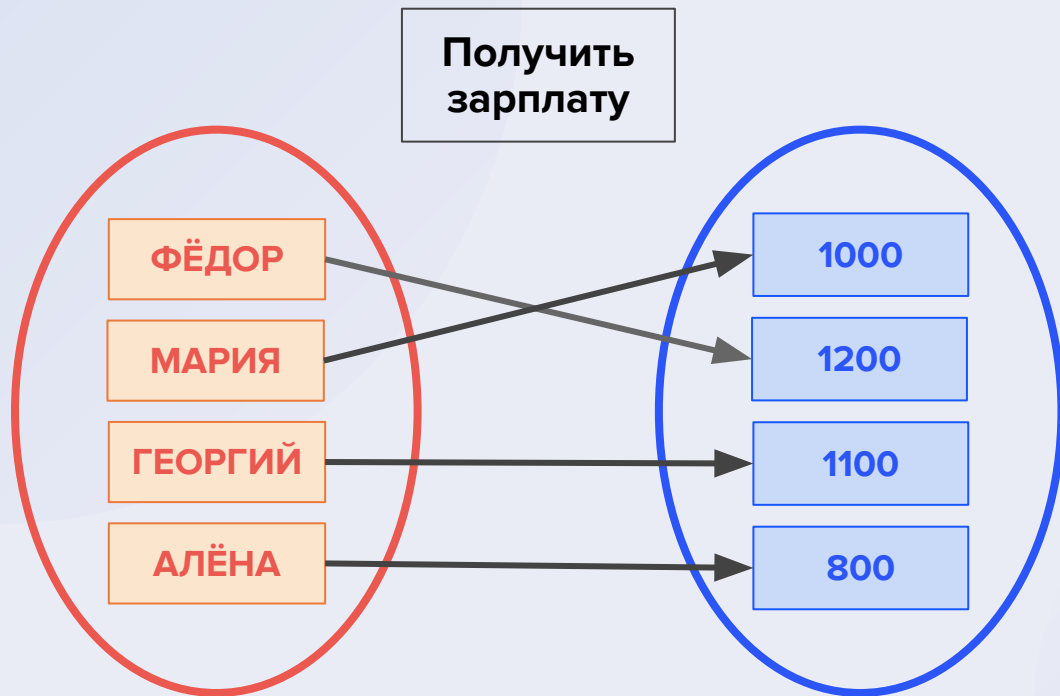
02 В программировании

это обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был назван (подпрограмма); объект, принимающий аргументы и возвращающий значение.

Функции помогают избежать дублирования кода, улучшить его структурированность и читаемость.

Что такое функция?

6



Что такое **функция**?

7

Функция `сходить_в_магазин('магазин', список покупок)`

- 1 Встать с дивана
- 2 Найти магазин на карте
- 3 Доехать до магазина
- 4 Купить товары по списку

Зафиксировать сумму трат

`сходить_в_магазин('Десяточка', [молоко, хлеб])`

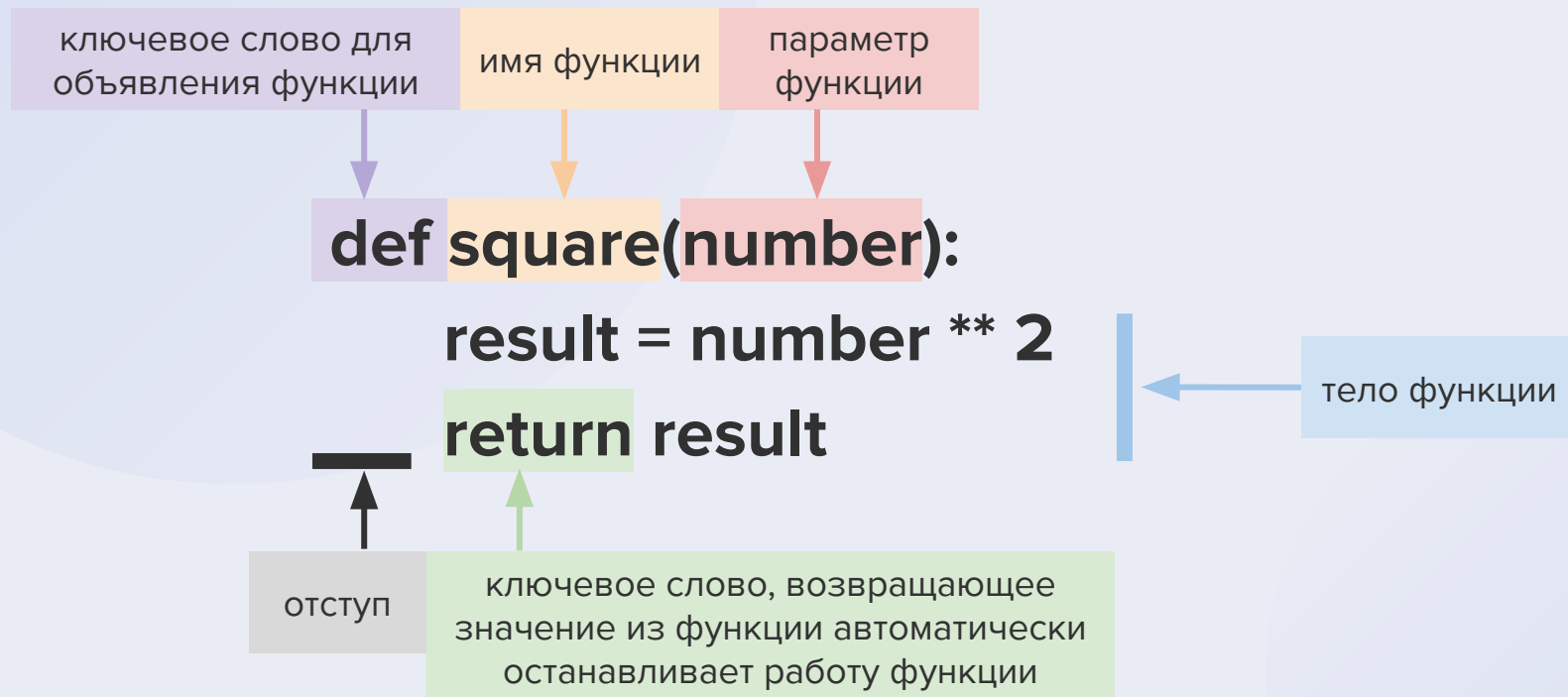
100 рублей

`сходить_в_магазин('DNS', [мышь, клавиатура])`

2000 рублей

Объявление функций в Python

8





Функция `help()`

вызывает справку по нужной функции

Docstring

(сокр. от *documentation string*, строка документации)
встроенное средство документирования модулей,
функций, классов и методов.

Сразу после определения указывается строковое
значение, которое и будет docstring'ом.

Параметры функции

11

- Функция может принимать более 1 параметра (а может не принимать параметры вообще).
- Для всех параметров функций можно указывать значения по-умолчанию, это дает возможность вызвать функцию с меньшим числом параметров.

Тип данных **None**

12

None — специальный тип данных, который означает отсутствие значения.

Если в функции нет **return**, либо он пустой, то она возвращает **None**.

Область видимости

13

Область видимости (scope) определяет контекст объекта, в рамках которого его можно использовать.

Рассмотрим 3 типа области видимости:

**Global
scope**

**Глобальная
область видимости**

**Local
scope**

**Локальная
область видимости**

**Nonlocal
scope**

**Нелокальная
область видимости**

Глобальная область видимости

Глобальный контекст

подразумевает, что переменная является глобальной, она определена вне любой из функций и доступна любой функции в программе.

Локальная область видимости

В отличие от глобальных переменных **локальная переменная** определяется внутри функции и доступна только из этой функции, то есть имеет локальную область видимости.

Если Python не может найти нужную переменную в локальной области видимости, то тогда (и только тогда) он будет искать ее в области видимости уровня выше.

Операторы **global** и **nonlocal**

15

Оператор **global**

позволяет создать глобальную переменную в локальном контексте.

Оператор **nonlocal**

позволяет изменить переменную в области видимости более высокого уровня (которая, в свою очередь, является локальной областью видимости для других переменных).

Анонимные функции

16

Анонимные функции создаются при помощи инструкции *lambda* и используются для более краткой записи функций с одним выражением. Выполняются быстрее обычных и не требуют инструкции *return*:

*lambda x, pow: x**pow*

Методы

в Python – функции, которые “принадлежат” к определенному объекту.

У каждого типа объектов есть свои методы.

Примеры методов
списков:

- `.index()`
- `.count()`
- `.append()`
- `.remove()`
- `.reverse()`

Примеры методов
строк:

- `.capitalize()`
- `.upper()`
- `.lower()`
- `.replace()`
- `.count()`

Примеры методов
словарей:

- `.keys()`
- `.values()`
- `.items()`



Функции

использование встроенных и создание собственных

Вопросы?

Олег Булыгин

Соцсеть
fb.com/obulygin91

Почта
obulygin91@ya.ru