

Функции – использование встроенных и создание собственных

Олег Булыгин

Нетология



Булыгин Олег

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



o.bulygin@netology.ru



<https://www.facebook.com/obulygin91>



О чем мы поговорим сегодня

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



Что такое функция?



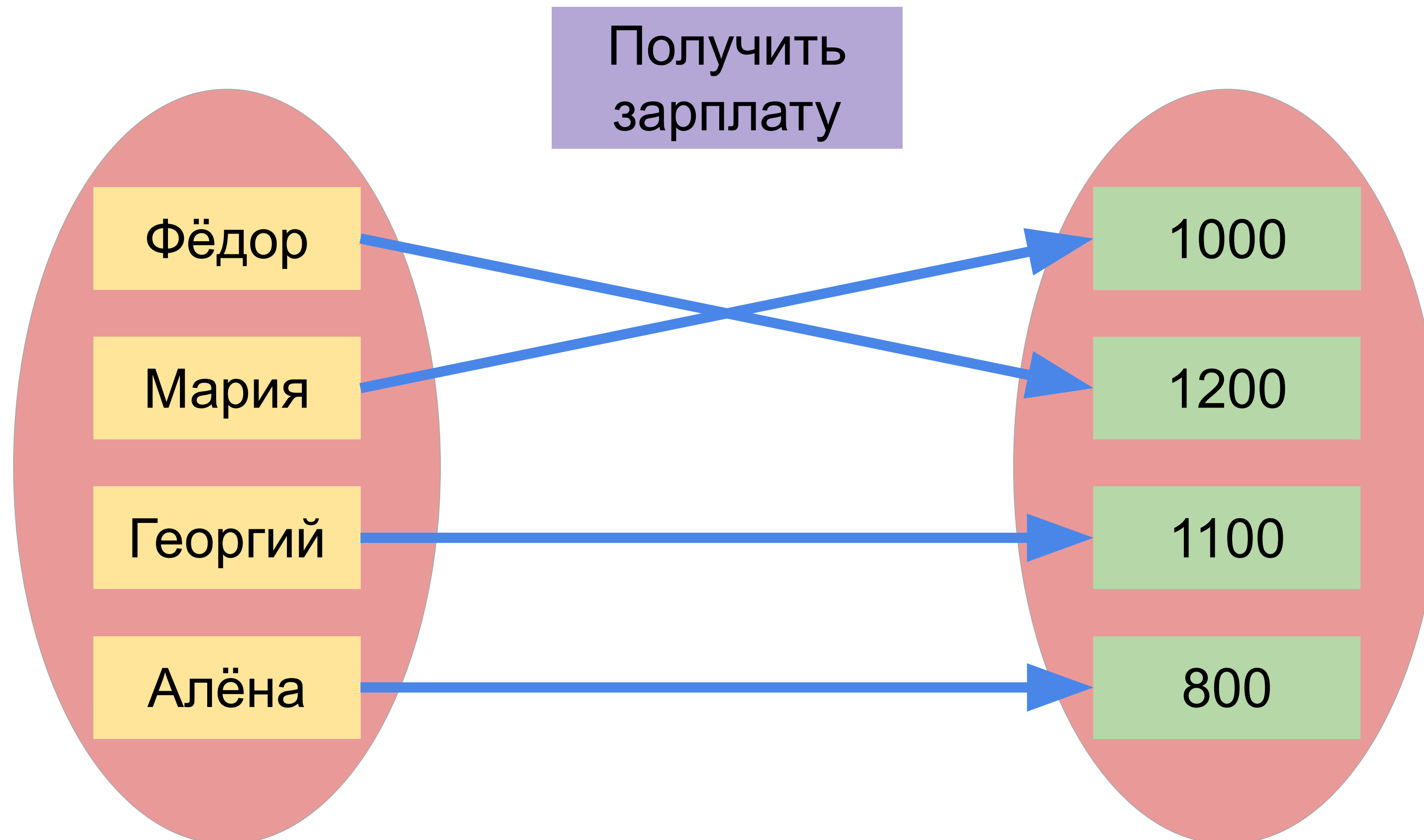
Что такое функция?

- В математике – соответствие между элементами; то как значение одной величины определяет значение другой;
- В программировании – 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?





Что такое функция?

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])`

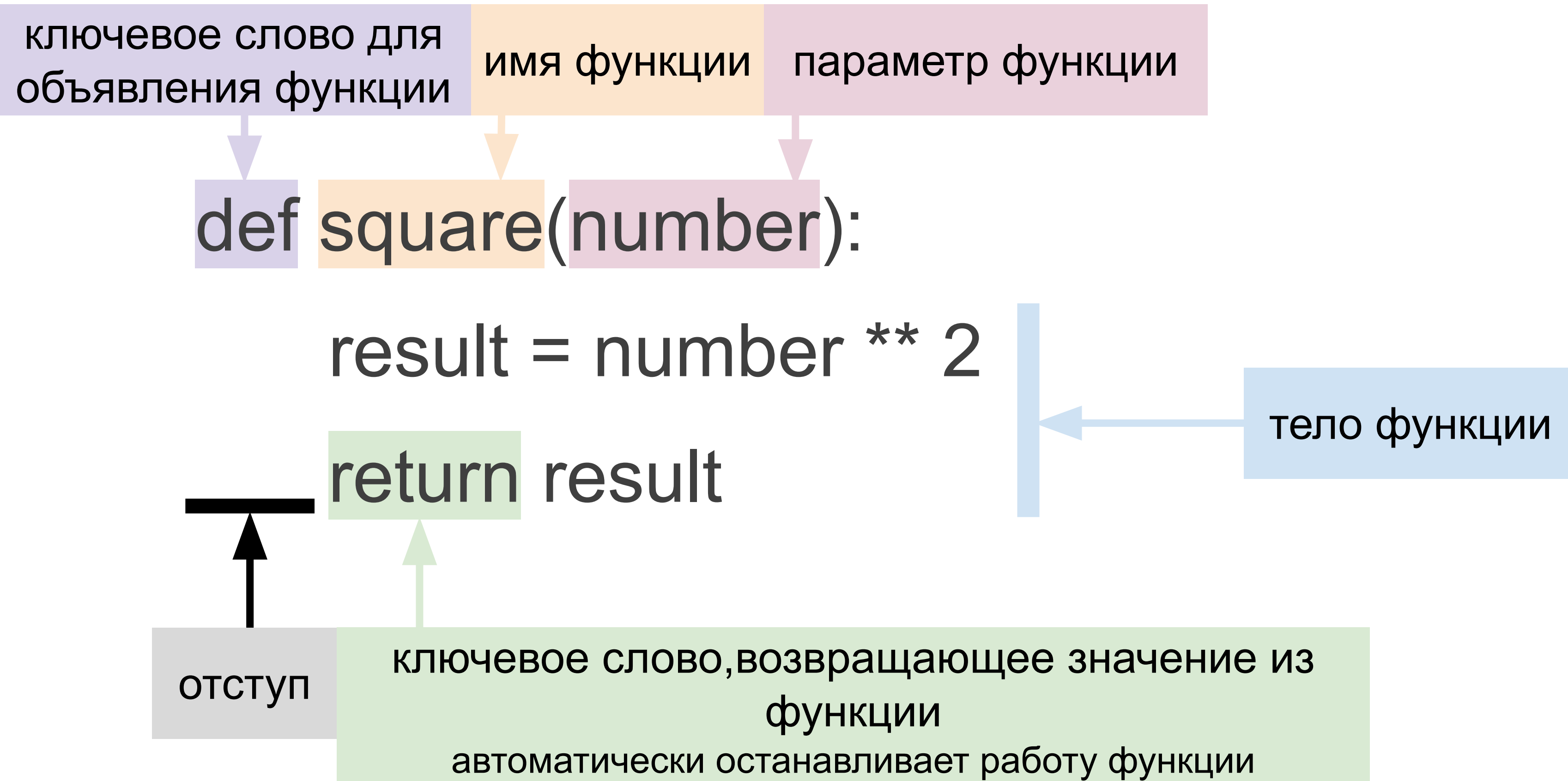
100 рублей

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

2000 рублей



Объявление функций в Python





Функция `help()`

Функция `help()` вызовет справку по нужной функции.



Docstring

Docstring – (сокращение 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

- 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

- ***None*** – специальный тип данных, который означает отсутствие значения.
- Если в функции нет ***return***, либо он пустой, то она возвращает ***None***.



Области видимости

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

Рассмотрим 3 типа области видимости:

- глобальная область видимости (global scope);
- локальная область видимости (local scope);
- нелокальная область видимости (nonlocal scope).



Области видимости

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

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`

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

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

Методы в Python – функции, который “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()*



Пакеты (библиотеки) и модули

Модуль – любой файл с расширением `.py` с заранее написанным кодом. Каждая программа может импортировать модуль и использовать определенные в нем функции, методы, новые типы данных и т.д.

Пакет (библиотека) – это набор (каталог) модулей Python.

Распространенные модули для анализа данных:

- Pandas;
- NumPy;
- Matplotlib;
- SciPy.



Импортирование пакетов

Anaconda включает в себя все необходимые нам пакеты, поэтому их отдельная установка не требуется.

Перед использованием пакета/модуля его необходимо импортировать при помощи ключевого слово ***import***, например:

```
import numpy
```

после чего его имя становится переменной и мы можем получить доступ к его атрибутам, например:

```
numpy.zeros(10)
```



Импортирование пакетов

Чтобы постоянно не писать название модуля/пакета при вызове его атрибутов, его можно импортировать под псевдонимом при помощи конструкции ***import ... as ...***, например:

import numpy as np

теперь функцию ***zeros()*** можно вызвать так: ***np.zeros(10)***

С помощью конструкции ***from ... import ...*** можно импортировать только конкретные нужные атрибуты пакета, например:

from numpy import zeros

Теперь функцию ***zeros*** можно вызвать без указания на название пакета: ***zeros(10)***



НЕТОЛОГИЯ
групп

Спасибо за внимание!

ОЛЕГ БУЛЫГИН



<https://www.facebook.com/obulygin91>



<https://vk.com/obulygin91>



<https://www.linkedin.com/in/obulygin>