

Функции — использование встроенных и создание собственных



Гежин Олег



Олег Гежин

Python - разработчик, специалист SQL, фрилансер.

План занятия

1. [Что такое функция?](#)
2. [Объявление функций в Python](#)
3. [Параметры функции](#)
4. [Области видимости](#)



Что такое функция?



Что такое функция?

В математике

это соответствие между элементами;

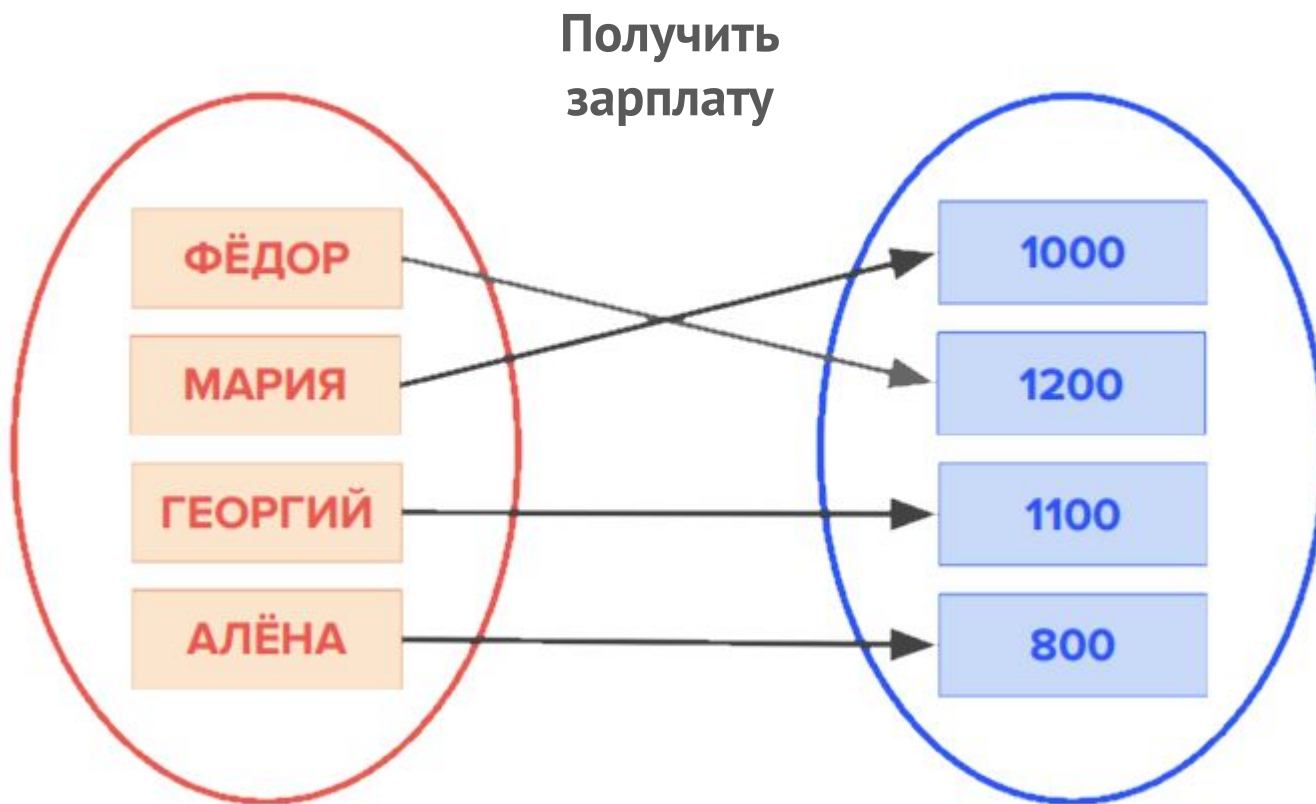
то как значение одной величины определяет значение другой;

В программировании

- a. это обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был назван (подпрограмма);
- b. 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 затрат

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

100 рублей

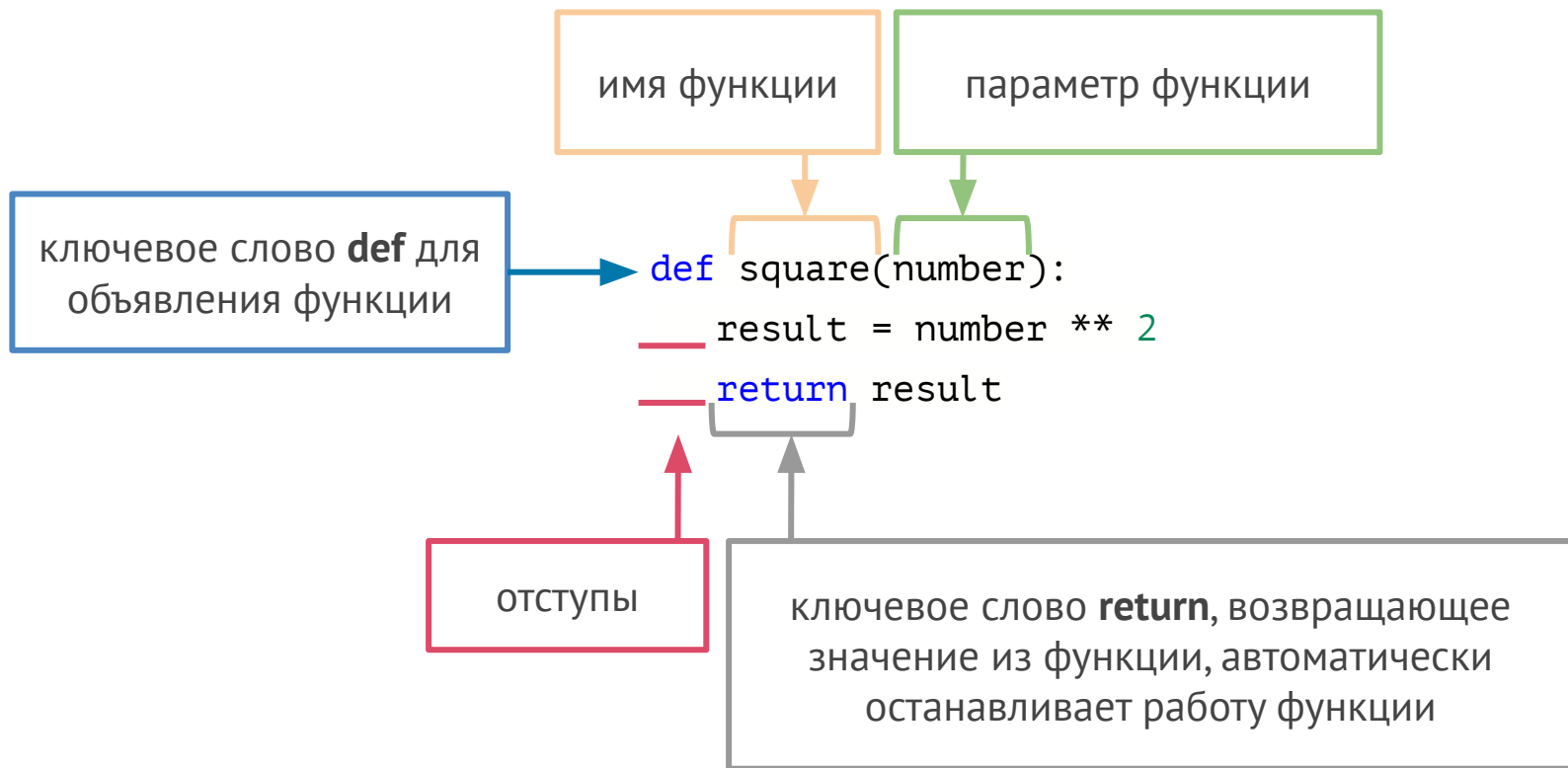
`сходить_в_магазин('DNS', [мышь, клавиатура])`

2000 рублей



Объявление функций в Python

Объявление функций в Python



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.

```
def function(a, b):  
    """function(a, b) -> list"""  
    return [a, b]
```



Параметры функции



Области видимости

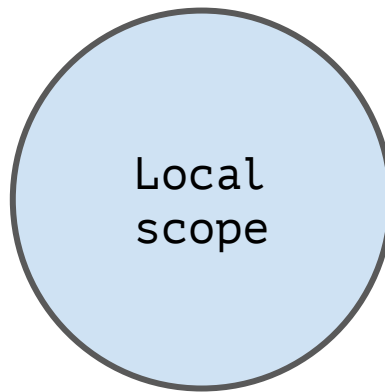
Область видимости

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

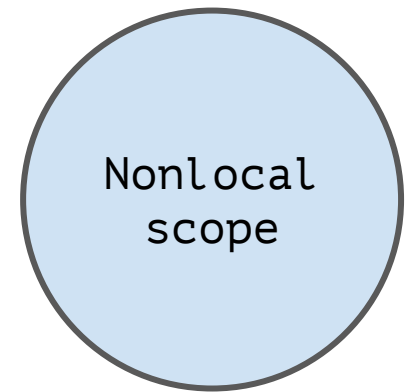
Рассмотрим 3 типа области видимости:



**Глобальная
область видимости**



**Локальная
область видимости**



**Нелокальная
область видимости**




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 её 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 – это 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()`

Домашнее задание

Я работаю секретарем и мне постоянно приходят различные документы. Я должен быть очень внимателен чтобы не потерять ни один документ. Каталог документов хранится в следующем виде:

...

```
documents = [  
{"type": "passport", "number": "2207 876234", "name": "Василий Гупкин"},  
{"type": "invoice", "number": "11-2", "name": "Геннадий Покемонов"},  
{"type": "insurance", "number": "10006", "name": "Аристарх Павлов"}  
]  
...
```

Перечень полок, на которых находятся документы хранится в следующем виде:

...

```
directories = {  
    '1': ['2207 876234', '11-2', '5455 028765'],  
    '2': ['10006'],  
    '3': []  
}  
...
```

Задача №1

Необходимо реализовать пользовательские команды, которые будут выполнять следующие функции:

* `р` – people – команда, которая спросит номер документа и выведет имя человека, которому он принадлежит;

* `s` – shelf – команда, которая спросит номер документа и выведет номер полки, на которой он находится;

Правильно обработайте ситуации, когда пользователь будет вводить несуществующий документ.

* `l` – list – команда, которая выведет список всех документов в формате passport "2207 876234" "Василий Гупкин";

* `a` – add – команда, которая добавит новый документ в каталог и в перечень полок, спросив его номер, тип, имя владельца и номер полки, на котором он будет храниться. **Корректно обработайте ситуацию, когда пользователь будет пытаться добавить документ на несуществующую полку.**

Внимание: p, s, l, a - это пользовательские команды, а не названия функций. Функции должны иметь выразительное название, передающие её действие.

Задача №2. Дополнительная (не обязательная)

* ``d`` – delete – команда, которая спросит номер документа и удалит его из каталога и из перечня полок. Предусмотрите сценарий, когда пользователь вводит несуществующий документ;

* ``m`` – move – команда, которая спросит номер документа и целевую полку и переместит его с текущей полки на целевую. Корректно обработайте кейсы, когда пользователь пытается переместить несуществующий документ или переместить документ на несуществующую полку;

* ``as`` – add shelf – команда, которая спросит номер новой полки и добавит ее в перечень. Предусмотрите случай, когда пользователь добавляет полку, которая уже существует.;

Задача №3

Почитать про lambda-функции. И что такое ``*args`` и ``**kwargs``.

Задача №4

Для подготовки к следующей лекции прочитайте про [классы](#).

Инструкция по выполнению домашнего задания:

1. Зарегистрируйтесь на сайте [Repl.IT](#).
2. Перейдите в раздел **my repls**.
3. Нажмите кнопку **Start coding now!**, если приступаете впервые, или **New Repl**, если у вас уже есть работы.
4. В списке языков выберите Python.
5. Код пишите в левой части окна.
6. Посмотреть результат выполнения файла можно, нажав на кнопку **Run**. Результат появится в правой части окна.
7. После окончания работы нажмите кнопку **Share** и скопируйте ссылку из поля **Share link**.
8. В личном кабинете на сайте [netology.ru](#) в поле комментария к домашней работе вставьте скопированную ссылку и отправьте работу на проверку.

Помимо ссылки, пожалуйста, прикрепите к вашему решению любой файл (например, скриншот любой части вашего кода).

**Задавайте вопросы и
пишите отзыв о лекции!**

Олег Гежин