

## Лабораторная работа № 7

Тема: Функции — использование встроенных и создание собственных.

Цель работы: научиться работать с функциями в Python.

Про основные встроенные функции можно почитать тут:

<https://pythonru.com/osnovy/vstroennye-funkcii-python>

Создание собственных функций в Python позволяет вам определить собственные блоки кода, которые могут быть многократно использованы в вашей программе.

```
1  # Определение функции:
2  # Для определения функции используется ключевое слово def,
3  # за которым следует имя функции и круглые скобки с параметрами.
4  def my_function(parameter1, parameter2):
5      # Тело функции
6      result = parameter1 + parameter2
7      return result
8
9  # Вызов функции:
10 # Вызывайте функцию по её имени, передавая необходимые аргументы.
11 result_value = my_function(10, 20)
12 print(result_value) # Вывод: 30
13
14 # Параметры функции:
15 # Функции могут принимать параметры.
16 # Параметры указываются в круглых скобках при определении функции.
17 def greet(name):
18     print(f"Привет, {name}!")
19
20 greet("Алиса") # Вывод: Привет, Алиса!

22 # Возвращаемое значение:
23 # Функции могут возвращать значение с использованием ключевого слова return.
24 # Если return отсутствует, функция вернет None.
25 def multiply(a, b):
26     result = a * b
27     return result
28
29 product = multiply(5, 3)
30 print(product) # Вывод: 15
31
32 # Значения по умолчанию для параметров:
33 # Можно задать значения по умолчанию для параметров функции,
34 # что делает их необязательными при вызове.
35 def power(base, exponent=2):
36     result = base ** exponent
37     return result
38
39 squared = power(5)
40 cubed = power(5, 3)
41
42 # Передача аргументов по ключевым словам:
43 # Вы можете передавать аргументы по ключевым словам,
44 # что делает вызов более явным и понятным.
45 result = power(base=2, exponent=3)
```

```

47 # Локальные и глобальные переменные:
48 # Переменные, определенные внутри функции, являются локальными для этой функц
49 # Глобальные переменные доступны из любой части программы.
50 global_variable = 10
51
52 def my_function():
53     local_variable = 5
54     print(global_variable) # Доступ к глобальной переменной
55     print(local_variable)
56
57 my_function()
58
59 # Лямбда-функции (анонимные функции):
60 # Лямбда-функции - это короткие функции, которые могут быть определены
61 # в одной строке.
62 square = lambda x: x**2
63 print(square(3)) # Вывод: 9
64

```

Практика.

Задание 1.

Простые числа. Напишите функцию, которая принимает число и проверяет, является ли оно простым. Затем напишите функцию, которая использует ранее объявленную функцию для определения простых чисел в заданном диапазоне.

Задание 2.

Генерация Пароля: Напишите функцию, которая генерирует случайный пароль заданной длины. Пароль должен включать в себя буквы (верхний и нижний регистр), цифры и специальные символы.

Задание 3.

Игра "Камень, Ножницы, Бумага": Создайте функцию, которая реализует игру "Камень, Ножницы, Бумага". Функция должна принимать выбор игрока и случайным образом выбирать выбор компьютера. Затем определите победителя в этом раунде.

Задание 4\* (необязательное).

Создайте функцию для игры "Виселица", где компьютер случайным образом выбирает слово, а игрок должен угадать его, вводя буквы. У игрока есть ограниченное количество попыток.

Задание 5\* (необязательное).

Секретарю постоянно приходят различные документы. Он 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'],
    '2': ['10006'],
    '3': []
}

```

Необходимо реализовать пользовательские команды, которые будут выполнять следующие функции:

**p** - reople – 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.

**d** – delete – команда, которая спросит номер документа и удалит полностью документ из каталога и его номер 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.

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.