**Лекция. Функции в Python**

Функции являются важным аспектом программирования, который позволяет нам организовывать и переиспользовать код более эффективно. Они дают возможность разбить программу на более мелкие и управляемые части. Позвольте поделиться с вами несколькими ключевыми принципами использования функций в Python.

1️⃣

Определение функции: Чтобы создать функцию, мы используем ключевое слово "def" и указываем имя функции вместе с круглыми скобками. В скобках мы можем указать аргументы функции. Затем следует двоеточие и блок кода функции с отступом.

def my\_function():

# Здесь находится код функции

2️⃣

Вызов функции: Для вызова функции мы просто указываем ее имя с круглыми скобками.

my\_function()

3️⃣

Аргументы функции: Функции могут принимать аргументы - значения, которые передаются им при вызове. Аргументы позволяют функциям работать с различными данными. Аргументы могут быть обязательными или необязательными.

def greet(name):

print("Привет, " + name + "!")

4️⃣

Возвращаемое значение: Функции могут возвращать значение с помощью ключевого слова "return". Возвращенное значение может быть использовано в других частях программы.

def multiply(x, y):

return x \* y

5️⃣

Модули и импорт: Функции часто организовываются в модули, которые представляют собой файлы с Python-кодом. Чтобы использовать функции из модуля, необходимо импортировать модуль с помощью ключевого слова "import".

import mymodule

mymodule.my\_function()

Это основные принципы работы с функциями в Python. Использование функций значительно упрощает организацию кода и повторное использование.

Давайте рассмотрим ещё несколько важных аспектов работы с функциями в Python.

6️⃣

Аргументы по умолчанию: В Python мы можем задать значения по умолчанию для аргументов функции. Это позволяет вызывать функцию без указания всех аргументов.

def greet(name, greeting="Привет"):

print(greeting + ", " + name + "!")

7️⃣

Именованные аргументы: При вызове функции мы можем использовать имена аргументов и их значения, что улучшает читаемость кода и позволяет указывать аргументы в любом порядке.

greet(name="Алиса", greeting="Здравствуй")

greet(greeting="Привет", name="Боб")

8️⃣

Рекурсия: В Python функции могут вызывать сами себя. Это называется рекурсией. Рекурсивные функции полезны для задач, которые могут быть разбиты на подзадачи.

|  |
| --- |
| def countdown(n):  if n <= 0:  print("Готово!")  else:  print(n)  countdown(n-1) |

9️⃣

Область видимости: Переменные, определенные внутри функции, имеют локальную область видимости и доступны только внутри функции. Глобальные переменные, объявленные вне функций, могут быть доступны из любого места программы.

|  |
| --- |
| x = 10  def my\_function():  y = x + 5  print(y) |

Это лишь некоторые из аспектов, связанных с функциями в Python. Они делают код более организованным, повторно используемым и модульным.

Если у вас есть вопросы, не стесняйтесь задавать - я всегда готов помочь!

**Практические задания**

Задача 1: Напишите функцию **calculate\_average**, которая принимает список чисел в качестве аргумента и возвращает их среднее значение.

Задача 2: Реализуйте функцию **is\_prime**, которая принимает целое число в качестве аргумента и проверяет, является ли оно простым числом (имеет только два делителя: 1 и само число). Функция должна вернуть булево значение (True или False).

Задача 3: Напишите функцию **reverse\_string**, которая принимает строку в качестве аргумента и возвращает её в обратном порядке.

Задача 4: Создайте функцию **count\_vowels**, которая принимает строку в качестве аргумента и возвращает количество гласных букв в этой строке. Гласными считаются буквы 'a', 'у', 'э', 'o' я 'и’ (в нижнем регистре).

Задача 5: Напишите функцию **is\_palindrome**, которая принимает строку в качестве аргумента и проверяет, является ли она палиндромом. Палиндромом считается строка, которая читается одинаково в обоих направлениях. Функция должна возвращать булево значение (True или False).

Задача 6: Создайте функцию **find\_max**, которая принимает список чисел в качестве аргумента и возвращает наибольшее число из списка.