**Функции**

Функции

Функции в Python определяются с использованием ключевого слова def, за которым следует имя функции и параметры в круглых скобках.

Вызов функции выполняется путем указания имени функции, за которым следуют

аргументы в круглых скобках.

def greet (name) :

print( "Привет, " + name + "!")

greet("Вася")

Возврат значений

Функции могут возвращать значения с помощью ключевого слова return.

def add(a, b):

result = a + b

return result

x = add(2,5)

print( "Результат: ", x)

Возврат нескольких значений

Python также позволяет возвращать несколько значений, которые будут упакованы в кортеж.

def calculate(a, b):

sum result = a + b

diff\_result = a - b

return sum\_result, diff\_result

sum\_value, diff\_value = calculate(10, 5)

print("Сумма:", sum\_value)

print( "Разность: ", diff\_value)

Возврат None

Если в функции нет оператора return или он указан без значения, функция вернет None.

def say\_hello(name) :

print("Привет, " + name + "!")

result = say\_hello ("Анна")

print(result) # Вывод: None

Аргументы функций

Функции могут принимать аргументы. В Python есть два типа аргументов: позиционные и именованные (ключевые). Позиционные аргументы передаются в том порядке, в котором объявлены параметры функции. Именованные аргументы передаются с указанием имени параметра.

def show\_info( name, age):

print( "Имя:", name )

print("Возраст:", age)

show\_info(age=25, name= "Андрей")

Аргументы по умолчанию

Функции могут иметь аргументы по умолчанию. Если аргумент не передан при вызове функции, будет использовано значение по умолчанию.

def greet (name="Гость"):

print( "Привет, " + name + "!")

greet( ) # Привет, Гость!

greet("Анна") # Привет, Анна!

Вложенные функции

В Python можно определять вложенные функции, которые находятся внутри других функций.

def outer\_function( ):

print( "Внешняя функция" )

def inner\_function( ):

print( "Внутренняя функция")

inner function( )

outer\_function( )

Локальные и глобальные переменные

Переменные, определенные внутри функции, называются локальными и видны только внутри этой функции. Переменные, определенные за пределами функции, называются глобальными и видны во всей программе.

# Глобальная переменная

global\_var = 10

def modify\_global( ):

global global\_var

global\_var += 1

def use\_ local( ):

local\_var = 5

print("Значение локальной переменной: " local var)

modify\_ global( )

print("Значение глобальной переменной после изменения:",global\_var)

use\_local()

# Попытка доступа к локальной переменной вне функции вызовет ошибку

# print("Значение локальной переменной вне функции: , "local\_var)

Рекурсия

Функция может вызывать саму себя. Это называется рекурсией.

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

print(factorial(4)) # Результат 24