

Функции

Что такое функция?

- Функция - это блок кода, который:
 - выполняет определенную задачу,
 - может принимать входные данные (аргументы),
 - может возвращать результат (return),
 - может быть вызван многократно.
- ```
def greet(name, age): # name и age - параметры
 print(f"Привет, {name}!")
```

```
name = "Анна"
```

```
age = 18
```

```
greet(name, age) # name и age - аргументы
```



# Передача и прием аргументов

- `def greet(name, age=18):`    # 18 – Значение по умолчанию  
    `print(f"Привет, {name}! Тебе {age} лет.")`

```
greet("Анна") # Активируем значение по умолчанию
greet("Лена", 25) # Позиционные аргументы
greet(age=21, name="Соня") # Именованные аргументы
greet("Оля", age=28) # Смешанные аргументы
```

- В случае использования смешанных аргументов именованные аргументы должны идти после позиционных.

# Возврат значений

- ```
def add(x, y):  
    sum = x + y  
    return sum
```

```
print(add(4, 7))
```

- Детали:
 - может возвращать любое значение
 - может возвращать несколько значений (кортеж)
 - Без `return` возвращает `None`

Область видимости переменных

- `x = 10` `#` Глобальная

```
def func():  
    global x    # Подключаем глобальную  
    x = 20      # Изменяем глобальную  
    y = 30      # Создаем локальную
```

```
func()  
print(x)    # 20  
print(y)    # Будет ошибка
```

Анонимные функции (лямбда)

- Однострочная функция без имени
- Обычная функция:

```
def double(x):  
    return x * 2  
print(double(5))
```
- Та же функциональность, но в виде лямбда функции:

```
double = lambda x: x * 2  
print(double(5))
```


Рекурсия

- функция вызывает саму себя

```
def factorial(n):  
    if n == 1:                # УСЛОВИЕ ВЫХОДА  
        return 1  
    else:  
        return n * factorial(n-1)  
  
print(factorial(5))  # 120
```

***args и **kwargs**

- *args - собирает позиционные аргументы в кортеж
**kwargs - собирает именованные аргументы в словарь
- ```
def print_info(*args, **kwargs):
 print("Позиционные:", args) # (1, 2, 3)
 print("Именованные:", kwargs) # {'name': 'Оля', 'age': 8}
```

  

```
print_info(1, 2, 3, name="Оля", age=8)
```
- Пример:  

```
print("first", "second", "third", sep="_", end="\n")
```



# Docstrings

- строки документации, описывающие работу функции.

```
def add(a, b):
 """
 Возвращает сумму двух чисел.

 :param a: Первое число.
 :param b: Второе число.
 :return: Сумма a и b.
 """
 return a + b
```

```
help(add) # Выводит документацию
```

# Аннотация типа

- Аннотация - указание ожидаемых типов параметров и возвращаемого значения.
- Детали:
  - не влияют на выполнение кода
  - помогают IDE и инструментам проверки кода
  - улучшают читаемость

```
name: str - аннотация типа; -> None - аннотация возвращения
def greet(name: str, times: int = 1) -> None:
 for _ in range(times):
 print(f"Привет, {name}!")

greet("Мир", 3)
```