

Day 08 — Python Bootcamp

Продвинутые функции

Аргументы по умолчанию, *args, **kwargs, lambda

Contents

1. [Chapter I](#)
 - 1.1. [General rules](#)
 2. [Chapter II](#)
 - 2.1. [Rules of the day](#)
 3. [Chapter III](#)
 - 3.1. [Intro](#)
 4. [Chapter IV](#)
 - 4.1. [Exercise 00: Default Argument](#)
 - 4.2. [Exercise 01: Multiple Arguments \(*args\)](#)
 - 4.3. [Exercise 02: Keyword Arguments \(**kwargs\)](#)
 - 4.4. [Exercise 03: Combine args and kwargs](#)
 - 4.5. [Exercise 04: Simple Lambda](#)
 - 4.6. [Exercise 05: Lambda with Sorting](#)
 5. [Chapter V](#)
 - 5.1. [Reading and tips](#)
-

Chapter I

General rules

- Используйте Python версии **3.8+**.
 - Каждое задание — отдельный файл *.py.
 - Все функции должны быть объявлены с помощью def или lambda (где указано).
 - Код должен быть читаемым и аккуратным.
 - Запрещено использовать темы, которые ещё не изучались (классы, декораторы и т.д.).
-

Chapter II

Rules of the day

Сегодня вы изучаете:

- аргументы по умолчанию
- *args
- **kwargs
- анонимные функции lambda

Фокус дня: - гибкость функций - понимание передачи аргументов

Chapter III

Intro

Вчера ты научился создавать функции.

Но Python улыбнулся и сказал:

- А что, если аргументов будет **неизвестное количество?**
- Или если я хочу написать функцию **в одну строку?**

Python ответил:

- Тогда ты готов к продвинутым функциям.

Сегодня ты сделаешь свои функции **гибкими и мощными**.

Chapter IV

Exercise 00: Default Argument

Создайте функцию `power(base, exponent=2)`, которая: - принимает число `base` - возводит его в степень `exponent` - если степень не передана — используется 2

Выведите результат вызова функции с одним и двумя аргументами.

Имя файла: `ex00.py`

Exercise 01: Multiple Arguments (*args)

Создайте функцию `sum_all(*args)`, которая: - принимает любое количество чисел - возвращает их сумму

Выведите результат работы функции.

Имя файла: `ex01.py`

Exercise 02: Keyword Arguments (**kwargs)

Создайте функцию `print_user(**kwargs)`, которая: - принимает любые именованные аргументы - выводит их в формате:

`key: value`

Имя файла: `ex02.py`

Exercise 03: Combine args and kwargs

Создайте функцию `describe_person(*args, **kwargs)`, которая: - выводит все позиционные аргументы - затем выводит все именованные аргументы

Имя файла: `ex03.py`

Exercise 04: Simple Lambda

Создайте `lambda`-функцию, которая: - принимает число - возвращает его квадрат

Выведите результат вызова `lambda`.

Имя файла: `ex04.py`

Exercise 05: Lambda with Sorting

Создайте список кортежей:

`[("Alice", 25), ("Bob", 20), ("Charlie", 30)]`

Отсортируйте список по возрасту, используя `lambda`.

Выведите отсортированный список.

Имя файла: `ex05.py`

Chapter V

Reading and tips

- Аргументы по умолчанию упрощают вызов функций
- `*args` — кортеж позиционных аргументов

- `**kwargs` — словарь именованных аргументов
- `lambda` — короткая функция без имени

Полезно почитать: - <https://docs.python.org/3/tutorial/controlflow.html#more-on-defining-functions> - <https://realpython.com/python-lambda/>

 Совет: если функция становится сложной — лучше использовать `def` вместо `lambda`.

Удачи! 