

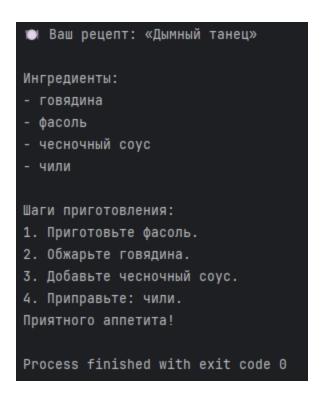
Практическая работа №1

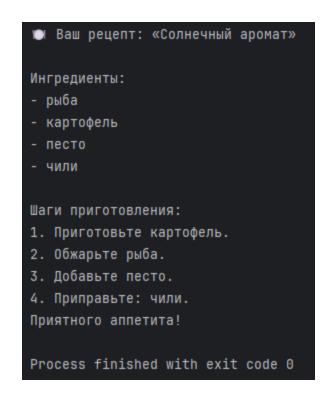
Время выполнения	@September 1, 2025 → September 12, 2025
Статус	В прогрессе

Задание: «Генератор случайного рецепта»

Разработайте программу, которая печатает случайно сгенерированный рецепт блюда: название (прилагательное + существительное), список ингредиентов (по одному из нескольких категорий) и пошаговые инструкции.

Пример выполнения программы:





Требования к функционалу

1. Структура программы

- Файл: recipe_generator_<Ваше_ФИО>_lab_01.py.
- Обязательные функции:
 - generate_dish_name() → str возвращает название блюда (например, "Огненный шторм").
 - generate_recipe() → str возвращает всю текстовую рецептуру: название, ингредиенты, шаги и заключение.
- В блоке запуска (if **name** == "**main**") печать результата: print(generate_recipe()).

2. Исходные наборы данных

- В коде должны быть списки (list[str]) с вариантами:
 - MAIN_INGREDIENTS основные ингредиенты (минимум 6 элементов).
 - SIDES гарниры (минимум 5 элементов).
 - SAUCES соусы (минимум 5 элементов).
 - SPICES специи (минимум 5 элементов).
 - ADJECTIVES прилагательные (минимум 6 элементов).
 - NOUNS существительные (минимум 6 элементов).

3. Поведение функций

Практическая работа №1

- generate_dish_name:
 - Выбирает случайное прилагательное и существительное.
 - Делает первую букву прилагательного заглавной (capitalize).
 - Возвращает строку формата "{Прилагательное} {существительное}".
- generate_recipe:
 - Выбирает по одному элементу из MAIN_INGREDIENTS, SIDES, SAUCES и SPICES с помощью random.choice.
 - Формирует название через generate_dish_name.
 - Собирает список шагов (минимум 4 шага), каждый шаг строка, содержащая выбранные элементы (например, "1. Приготовьте рис.", "2. Обжарьте курицу.").
 - Формирует итоговую многострочную строку, с разделами "Ингредиенты:" и "Шаги приготовления:", и возвращает её.

4. Стиль реализации и требования к коду

- Используйте аннотации типов, например list[str], def generate_recipe() → str.
- Используйте f-строки для форматирования.

Советы

- Ознакомьтесь с методическими указаниями, там расписана вся нужна информация для выполнения задания.
- f-строки позволяют собирать многострочные блоки, используйте склеивание строк или тройные кавычки по вкусу.
- Помните, что Python чувствителен к отступам.
- Для списка шагов удобно формировать list[str] и затем "\n".join(steps).

Ресурсы

https://metanit.com/python/tutorial/1.2.php (RU)

- Глава 1. Введение в Python:
 - Язык программирования Python
 - Установка Python и первая программа на Windows
 - Управление версиями Python на Windows, MacOS и Linux
 - Первая программа в PyCharm

• Глава 2. Основы Python:

- Введение в написание программ
- Переменные и типы данных
- <u>Функции</u>
- Параметры функции
- Оператор return и возвращение результата из функции
- Функция как тип, параметр и результат другой функции
- Списки
- Глава 7. Строки
 - Работа со строками
 - Основные методы строк
 - Форматирование
- https://fastapi.tiangolo.com/ru/python-types/

Практическая работа №1

https://www.geeksforgeeks.org/python/python-programming-language-tutorial/

1. Python Basics

- Introduction
- Input and Output
- <u>Variables</u>
- Operators
- Quiz: Basics, I/O
- <u>Keywords</u>
- Data Types
- Quiz: <u>Data Types</u>, <u>Numbers</u>, <u>Boolean</u>

2. Python Functions

- Functions
- Pass Statement in Function
- Global and Local Variables

3. Python Data Structures

- Strings
- List
- Quiz: <u>List</u>, <u>String</u>

https://fastapi.tiangolo.com/python-types/

Практическая работа №1