## \*\*📌 План обучения Python (от нуля до продвинутого уровня)\*\*

### \*\*🔹 Этап 1: Основы Python (2-4 недели)\*\*

\*\*Теория:\*\*

1. Установка Python и настройка среды (IDLE, PyCharm, VS Code).

2. Синтаксис: переменные, типы данных (`int`, `float`, `str`, `bool`), операторы.

3. Базовые структуры данных: списки (`list`), кортежи (`tuple`), множества (`set`), словари (`dict`).

4. Условные операторы: `if-elif-else`.

5. Циклы: `for`, `while`, `break`, `continue`.

6. Функции: создание, аргументы, возврат значений.

7. Работа с файлами: чтение/запись (`open()`, `read()`, `write()`).

\*\*Практика:\*\*

- Задачи на условные операторы (калькулятор, проверка числа на четность).

- Работа со списками (поиск max/min, сортировка).

- Чтение файла и подсчет слов/символов.

\*\*Мини-проект:\*\* Консольный "Калькулятор" или "Викторина с вопросами".

---

### \*\*🔹 Этап 2: Углубленное изучение (3-5 недель)\*\*

\*\*Теория:\*\*

1. Исключения: `try-except-finally`.

2. Модули и пакеты (`import`, создание своих модулей).

3. Работа с датами и временем (`datetime`).

4. Регулярные выражения (`re`).

5. Основы ООП: классы, объекты, методы, наследование.

6. Декораторы и генераторы.

\*\*Практика:\*\*

- Обработка ошибок ввода пользователя.

- Парсинг текста (поиск email/телефонов в строке).

- Создание класса "Банковский счет" с методами.

\*\*Мини-проект:\*\* "Планировщик задач" (сохранение дел в файл).

---

### \*\*🔹 Этап 3: Продвинутые темы (4-6 недель)\*\*

\*\*Теория:\*\*

1. Работа с API (запросы `requests`, JSON).

2. Многопоточность (`threading`) и асинхронность (`asyncio`).

3. Базы данных: SQLite, базовый SQL.

4. Тестирование кода (`unittest`, `pytest`).

5. Виртуальные окружения (`venv`).

\*\*Практика:\*\*

- Получение данных с публичного API (например, погода).

- Многопоточный парсинг сайта (с сохранением в CSV).

- Написание тестов для функций.

\*\*Мини-проект:\*\* "Бот для погоды" (через API OpenWeatherMap).

---

### \*\*🔹 Этап 4: Специализация (выберите направление)\*\*

#### \*\*🎯 Варианты:\*\*

1. \*\*Веб-разработка\*\* (Django/Flask):

- Создание блога с CRUD.

- Деploy на Heroku.

2. \*\*Анализ данных\*\* (Pandas, Matplotlib):

- Анализ датасета (например, Titanic).

- Визуализация статистики.

3. \*\*Автоматизация\*\* (Selenium, BeautifulSoup):

- Парсинг сайта.

- Бот для Telegram.

4. \*\*Машинное обучение\*\* (Scikit-learn):

- Предсказание цен на жилье.

\*\*Финальный проект:\*\* Выбор зависит от специализации (например, "Веб-приложение для учета финансов").

---

### \*\*🔹 Этап 5: Развитие навыков\*\*

1. Изучение алгоритмов (сортировки, поиска).

2. Участие в хакатонах или соревнованиях (LeetCode, CodeWars).

3. Чтение чужого кода (GitHub).

4. Оптимизация производительности.

---

### \*\*📌 Советы:\*\*

- \*\*Регулярность\*\*: 3-5 часов в неделю.

- \*\*Практика > теория\*\*: 70% времени на код.

- \*\*Git\*\*: с первых проектов создавайте репозитории.

- \*\*Сообщество\*\*: задавайте вопросы (Stack Overflow, форумы).

Пример графика на неделю:

- Понедельник: Теория (ООП).

- Среда: Практика (классы).

- Пятница: Мини-проект (игра "Угадай число").

Через 3-6 месяцев сможете уверенно писать на Python! 🚀