

День 1: Настройка среды

- Теория: [Понимание потока GitHub](#)
- Задание: Установите Git, создайте учетную запись GitHub и настройте VS Code.
- Выполнить: Сделайте снимок своей установки и загрузите его в README.md нового репозитория.

День 2: Введение в Git

- Теория: [Справочник по Git](#)
- Задание: Изучите учебник по Git.
- Выполнить фиксацию: Инициализируйте локальный репозиторий, добавьте файл .gitignore и сделайте свою первую фиксацию.

День 3: Ваш первый репозиторий

- Теория: [GitHub Hello World](#)
- Задание: Создайте новый репозиторий GitHub.
- Принять решение: Добавьте файл README.md с описанием ваших планов на 30 дней.

День 4: "Здравствуй, мир!" на языке Python

- Теория: [Python для начинающих](#)
- Задание: Написать скрипт "Hello, World!" на языке Python.
- Выполнить: Поместите файл hello_world.py в свой репозиторий.

День 5: Переменные и типы данных в Python

- Теория: [Переменные и типы данных в Python](#)
- Задание: Напишите сценарий на языке Python, в котором используются различные типы данных и переменные.
- Выполнить: Добавить файл variables.py в свой репозиторий.

День 6: Работа со строками в Python

- Теория: [Манипулирование строками в Python](#)
- Задание: Создать скрипт Python, который манипулирует строками (конкатенация, нарезка и т.д.).
- Выполнить: Поместите файл string_operations.py в свой репозиторий.

День 7: Основы математики в Python

- Теория: [Арифметические операции в Python](#)
- Задание: Написать скрипт на языке Python, выполняющий различные арифметические операции.
- Примите решение: Добавить arithmetic.py в свой репозиторий.

День 8: Списки в Python

- Теория: [Списки Python](#)
- Задание: Поиграйте со списками Python: сортировка, добавление и индексация.
- Выполнить: Записать файл lists.py.

День 9: Словари Python

- Теория: [Словари Python](#)
- Задание: Создать скрипт, использующий словари для хранения и получения данных.
- Выполнить: Добавить dictionaries.py в свой репозиторий.

День 10: Условные операторы в Python

- Теория: [Поток управления в Python](#)
- Задание: Напишите сценарий, использующий операторы if-else.
- Выполнить: Поместите файл conditionals.py в свой репозиторий.

День 11: Циклы Python

- Теория: [Циклы Python](#)
- Задание: Использовать циклы for и while в сценарии Python.
- Выполнить: Добавить файл loops.py в свой репозиторий.

День 12: Функции в Python

- Теория: [Функции Python](#)
- Задание: Создать скрипт на языке Python с несколькими функциями, выполняющими задачи.
- Примите решение: Поместите functions.py в свой репозиторий.

День 13: Лямбды Python

- Теория: [Лямбда-функции Python](#)
- Задание: Поэкспериментировать с лямбда-функциями.
- Выполнить: Добавить файл lambda.py с примерами лямбда-функций.

День 14: Импортирование модулей в Python

- Теория: [Модули Python](#)
- Задание: Напишите скрипт, который импортирует и использует хотя бы один модуль.
- Выполнить: Поместите файл modules.py в свой репозиторий.

День 15: Работа с файлами в Python

- Теория: [Файловый ввод/вывод в Python](#)
- Задание: Написать скрипт на Python, который читает из файлов и записывает в них.
- Выполнить: Добавьте файл file_handling.py и текстовый файл, с которым вы работали.

День 16: Введение в обработку ошибок

- Теория: [Обработка исключений в Python](#)
- Задание: Использовать блоки try-except в сценарии.
- Выполнить: Выложите файл error_handling.py со своими примерами.

День 17: Понимание списков в Python

- Теория: [Понимание списков в Python](#)
- Задание: Записать в скрипте списковые вычисления.
- Выполнить: Добавить list_comprehensions.py в свой репозиторий.

День 18: Создание базового класса в Python

- Теория: [Классы и объекты Python](#)
- Задание: Определить класс с базовой функциональностью.
- Выполнить: Выложите файл classes.py с определением класса и примерами его использования.

День 19: Наследование в Python

- Теория: [Наследование в Python](#)
- Задание: Написать скрипт на языке Python, демонстрирующий наследование.
- Выполнить: Добавьте файл inheritance.py в свой репозиторий.

День 20: Введение в веб-скраппинг с помощью Python

- Теория: [Beautiful Soup: Веб-скраппинг](#)
- Задача: Соскрести данные с веб-страницы.
- Выполнить: Поместите web_scraping.py и все созданные вами файлы данных.

День 21: Python и API

- Теория: [Работа с API в Python](#)
- Задание: Использовать публичный API для получения данных и их обработки.
- Выполнить: Добавьте api_interaction.py в свой скрипт.

День 22: Настройка простого веб-сервера с помощью Flask

- Теория: [Быстрый старт Flask](#)
- Задание: Создать базовый веб-сервер с помощью Flask.
- Выполнить: Поместите папку, содержащую ваше приложение Flask.

День 23: Основы HTML/CSS

- Теория: [HTML и CSS](#)
- Задание: Создайте простую HTML-страницу со стилями CSS.
- Выполнить: Добавьте index.html и styles.css в свой репозиторий.

День 24: Основы JavaScript

- Теория: [Основы JavaScript](#)
- Задание: Напишите базовый JavaScript для взаимодействия с HTML-страницей.
- Выполнить: Разместите script.js и обновите свой index.html, чтобы включить его.

День 25: Интеграция JavaScript с HTML/CSS

- Теория: [Интерактивные веб-сайты](#)
- Задание: Сделайте свою веб-страницу интерактивной с помощью JavaScript.
- Выполнить: Обновите script.js с помощью интерактивных функций и зафиксируйте изменения.

День 26: Введение в базы данных

- Теория: [Самоучитель SQL](#)
- Задание: Написать базовые SQL-запросы к базе данных примера.
- Выполнить: Добавьте файл с вашими SQL-запросами.

День 27: Использование Git и GitHub для контроля версий

- Теория: [Продвинутый Git](#)
- Задачи: Ветвление, слияние изменений и использование запросов на притяжение.
- Задание: Слияние ветки фич с основной веткой.

День 28: Обзор кода и совместная работа с GitHub

- Теория: [Совместная работа на GitHub](#)
- Задание: Рецензирование чужого кода или совместная работа над небольшой функцией.
- Принять решение: Открыть запрос на исправление (pull request) со своим вкладом в проект.

День 29: Развертывание простого приложения

- Теория: [Введение в Heroku](#)
- Задание: Разверните один из своих проектов (например, приложение на Flask) на Heroku или аналогичной платформе.
- Выполнить: Добавьте файлы развертывания и секцию 'Deployed' в README с прямой ссылкой.

День 30: Рефлексия и документирование своего учебного пути

- Теория: [Написание рефлексивных эссе](#)
- Задание: Напишите размышление о своем учебном пути и будущих целях.
- Выполнить: Обновите README.md, добавив в него свои размышления, и внесите в код все последние улучшения.