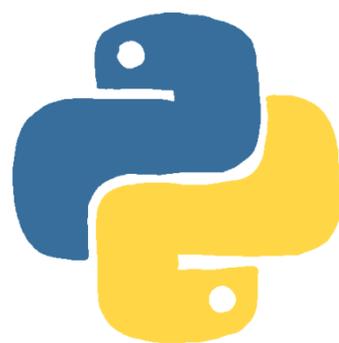




КАМПУС

ДЕТСТВА

КУРС PYTHON НАЧАЛО



ПЛАН ЗАНЯТИЙ ПО
ПРОГРАММИРОВАНИЮ НА
Python.

ОСНОВЫ, ЧАТ БОТ, ИГРА
НА PYGAME



Введение в Python.

- Установка Python и конструкции
- Условные операторы (if, elif, else)
- Циклы (for, while) и работа с диапазонами.
 - Функции и модули.
- Определение и вызов функций, аргументы и возвращаемые значения.
 - Модули и библиотеки.
(импортирование модулей, стандартная библиотека)
 - Структуры данных.
- Списки и кортежи: создание, методы и операции.
- Словари и множества: создание, методы и операции.

- Работа с файлами и исключениями.
 - Чтение и запись файлов.
- Обработка исключений (try, except).
 - ООП в Python.
- Основы объектно-ориентированного программирования: классы и объекты.
 - Наследование и полиморфизм.
 - Библиотеки и фреймворки.
- Введение в популярные библиотеки (NumPy, pandas, Matplotlib)
- Основы веб-разработки с Flask или Django.
 - Проектная работа
- Выбор темы для проекта, планирование и обсуждение.
- Презентация проектов и обратная связь.

Создание чат-бота на Python.

- Установка Python и необходимых инструментов (например, IDE, Jupyter Notebook).
- Основы синтаксиса Python: переменные, типы данных, операторы.
- Основные конструкции: условия, циклы, функции.
 - Работа с библиотеками.
 - Установка и использование pip.
- Обзор популярных библиотек для создания чат-ботов (например, requests, Flask, python-telegram-bot).

Основы API.

- Что такое API и как они работают.
- Примеры работы с REST API с помощью библиотеки requests.
- Создание простого приложения, которое делает запрос к API.

Создание простого чат-бота

- Выбор платформы для чат-бота (Telegram, Discord, Slack и т.д.).
- Настройка бота на выбранной платформе.
 - Написание простого бота, который отвечает на сообщения (например, "Привет" → "Здравствуйте!").
- Использование регулярных выражений для обработки текста.
 - Создание команд для бота (например, /start, /help).
- Реализация логики обработки сообщений.
 - Введение в базы данных (SQLite или другие).
 - Создание базы данных для хранения информации о пользователях.
- Чтение и запись данных из базы данных.
 - Добавление новых команд и функций.
- Реализация возможностей для ведения диалога (например, использование состояния).

Тестирование и отладка.

- Применение сторонних API (например, погода, новости).
 - Основы тестирования кода.
- Инструменты для отладки (например, логирование).
- Как исправлять ошибки и улучшать код.
 - Обзор платформ для развертывания (Heroku, AWS, DigitalOcean).
- Подготовка к развертыванию (настройка окружения).
 - Развертывание бота на выбранной платформе.
 - Создание собственного чат-бота с уникальной функциональностью.
- Презентация проекта и обмен опытом с другими участниками.

Введение в Python и Pygame.

- Установка Python и Pygame.
- Основы синтаксиса Python (переменные, типы данных, функции).
- Знакомство с Pygame: что это такое и для чего используется.

Создание окна игры

- Инициализация Pygame и создание основного окна.
 - Основные функции:
`pygame.display.set_mode()`,
`pygame.display.set_caption()`.
- Основной игровой цикл: обработка событий, обновление экрана, задержка.

Рисование на экране

- Использование функций для рисования:
`pygame.draw.rect()`, `pygame.draw.circle()`,
`pygame.draw.line()`.
- Задание цветов и фонового изображения.
- Обработка событий клавиатуры и мыши.

Работа с изображениями

- Загрузка изображений с помощью `pygame.image.load()`.
- Отображение изображений на экране.
 - Основы анимации: перемещение изображений.

Звук и музыка

- Загрузка и воспроизведение звуковых эффектов.
 - Добавление фоновой музыки.
 - Управление звуком (включение/выключение, изменение громкости).

Обработка столкновений

- Основы коллизий: прямоугольные области, круги.
 - Реализация столкновений между объектами.
- Примеры игр с использованием механики столкновений..

Создание простейшей игры:

- Проектирование простой игры (например, "Убегай от врага").
- Реализация игрового процесса, логики и правил.
- Тестирование и отладка игры.

Уровни и улучшения:

- Создание нескольких уровней в игре.
- Добавление новых объектов и врагов.
- Реализация системы очков и прогресса.

Интерфейс пользователя:

- Создание меню (начальное, пауза).
- Отображение информации о здоровье, очках и времени.
- Обработка событий в интерфейсе.

Итоговый проект:

- Разработка собственной игры с уникальной концепцией.
- Презентация проекта и обсуждение с участниками.
- Советы по улучшению и публикации игр.