

Цифриум

Технологическая карта

Умный код на Python. Базовый уровень

Модуль 3

Тема 4

Урок 5

**Интеграция API GPT в простую web-систему
(часть 1)**



Уважаемый коллега, обратите внимание!

Учёт присутствия ребёнка на уроке ведётся только по цифровому следу. Каждый ученик из группы на уроке должен передать цифровой след на нашей платформе.

Вы отвечаете за передачу следа учеником на нашей платформе во время урока.

Передавать цифровой след нужно в личном кабинете ученика.

Цифровой след засчитывается по итогу выполнения учеником трёх задач на платформе в ходе урока.

Обращаем внимание – в задачах с отправкой ответа в виде файла наша платформа поддерживает следующие форматы: .pdf, .doc, .png, .jpeg, .xlsx, .mp3, .py, .ipynb, .txt, .csv, .json, .xml, .sb3, .ino, .hex.

Контролируйте время урока, чтобы не пропустить момент, когда ученики должны залогиниться в личном кабинете и перейти к решению задач на платформе.

Проследите, чтобы каждый ученик залогинился в личном кабинете и выполнил 3 задачи на платформе. В противном случае, Вам не будет засчитано, что вы провели урок, занятие не будет оплачено, а ученику не будет выставлена отметка о присутствии на уроке.

Просим вас в ходе 45 минут урока проконтролировать, чтобы:

1. Каждый из учеников залогинился в личном кабинете.
2. Каждый ученик в разделе **«Решить задания»** отправил решения по 3 задачам – «Практическое задание 1», «Практическое задание 2» и «Практическое задание 3».

О занятии

Краткое содержание занятия:

1. Познакомитесь с основами HTML и JavaScript.
2. Узнаете, что такое DOM и как с ним работать.
3. Создадите веб-страницу с формой «Задай вопрос».
4. Напишете JavaScript-функцию для обработки событий.
5. Научитесь получать значения из полей ввода и выводить текст на страницу.
6. Узнаете, как отлаживать код с помощью консоли разработчика.

Цель:

Научиться создавать базовый web-интерфейс для взаимодействия с GPT через API.

Задачи:

1. Познакомиться с основами HTML и JavaScript: теги `<input>`, `<button>`.
2. Создать веб-страницу с формой «Задай вопрос» и подключённым `script.js`
3. Написать обработчик событий: чтение значения из поля и вывод ответа.
4. Научиться работать с DOM и отлаживать код.

Термины

HTML (HyperText Markup Language)

Язык разметки для создания структуры веб-страниц. Определяет, что будет на странице: заголовки, абзацы, кнопки, формы и т.д.

JavaScript (JS)

Язык программирования, который оживляет веб-страницы, делая их интерактивными. Позволяет реагировать на действия пользователя.

О ЗАНЯТИИ

DOM (Document Object Model)

Структура HTML-элементов как объекта, с которым можно работать через JavaScript. Представляет страницу как дерево элементов.

Тег

Элемент разметки HTML, заключённый в угловые скобки. Например: `<h1>`, `<p>`, `<button>`.

Форма

Элемент веб-страницы для ввода данных пользователем, обычно состоит из полей ввода (`input`) и кнопок (`button`).

Обработчик события

Функция JavaScript, которая запускается при определённом действии пользователя (клик, ввод текста и т.д.).

getElementById

Метод JavaScript для поиска HTML-элемента по его уникальному идентификатору (ID).

innerText

Свойство JavaScript для чтения или изменения текстового содержимого HTML-элемента.

value

Свойство JavaScript для получения значения из поля ввода (`input`).

Консоль разработчика

Инструмент в браузере для отладки кода, просмотра ошибок и вывода информации через `console.log()`.

Переменные окружения

Специальные переменные операционной системы для хранения конфиденциальной информации (например, API-ключей).

Необходимые материалы:

1. Браузер (Chrome, Firefox или другой).
2. Редактор кода (например, VS Code, Notepad++ или онлайн-редактор).
3. Доступ к интернету.

Темы и время

ЭТАП	ВРЕМЯ
Приветствие	5 мин.
Теория	25 мин.
Итоги занятия	5 мин.
Получение фидбека. Рефлексия. Проверка знаний. Домашнее задание	5 мин.
Вопрос-ответ	5 мин.
Итого	45 мин.

Ход занятия

Номер слайда	Пояснение к слайду
1	Приветствие учеников. Доброе утро, ребята! Сегодня мы начинаем очень интересное занятие, на котором создадим базовый web-интерфейс для взаимодействия с искусственным интеллектом GPT. Помните нашего Telegram-бота? Сегодня мы сделаем похожую вещь, но прямо в браузере. Представьте, что вы сможете общаться с нейросетью на своем собственном сайте. Это первый шаг к созданию настоящих веб-приложений. Не волнуйтесь, если какие-то термины покажутся новыми, мы разберем всё по шагам.
2	Титульный слайд. Тема урока: Интеграция API GPT в простую web-систему (часть 1). Это первая часть урока, где мы познакомимся с основами создания веб-интерфейса. Звучит серьезно, но на самом деле мы будем строить фундамент. Как дом нельзя построить без стен, так и сайт не работает без правильной структуры. Сегодня мы зложим этот фундамент, чтобы на следующем уроке подключить настоящую магию искусственного интеллекта.
3	Повторение прошлого занятия. На прошлом уроке вы научились интегрировать GPT с внешними сервисами через REST API и WebSocket, поняли различия между этими протоколами, настроили Telegram-бота и реализовали взаимодействие бота с GPT. Давайте вспомним: кто может сказать, чем отличался REST от WebSocket? А кто помнит, зачем нам нужен был токен бота? Обсудите с учениками, что им запомнилось больше всего. Это поможет нам связать прошлый опыт с сегодняшней задачей.
4	Анонс текущего занятия. Сегодня вы познакомитесь с основами HTML и JavaScript, узнаете, что такое DOM, создадите веб-страницу с формой для вопросов, напишете JavaScript-функцию для обработки событий и научитесь отлаживать код. Звучит много, но на самом деле это три

Ход занятия

	<p>кита веб-разработки. HTML создаст форму, JavaScript заставит её работать, а DOM поможет им понять друг друга. В конце урока у вас будет работающая страничка, которая реагирует на ваши действия.</p>
5	<p>Цели и задачи урока. Цель: научиться создавать базовый web-интерфейс для взаимодействия с GPT через API. Задачи: 1) познакомиться с основами HTML и JavaScript, 2) создать веб-страницу с формой, 3) написать обработчик событий. Объясните ученикам, что каждая задача важна для достижения цели. Без HTML не будет кнопок, без JavaScript не будет реакции, а без понимания задач сложно собрать всё вместе. Мы будем двигаться последовательно, чтобы никто не потерялся.</p>
6	<p>Что такое HTML. HTML – это каркас сайта, язык разметки. Представьте, что вы строите дом. HTML – это кирпичи и балки. С помощью HTML-тегов мы размещаем на странице заголовки, поля ввода, кнопки и контейнеры для вывода информации. Это основа любой веб-страницы. Спросите учеников: видели ли они раньше HTML-код? Возможно, кто-то уже пробовал менять текст на сайтах. Сегодня мы научимся создавать его с нуля.</p>
7	<p>Что такое JavaScript. JavaScript – это язык программирования, который оживляет страницу. Если HTML – это скелет, то JavaScript – это мышцы и нервы. С его помощью можно реагировать на действия пользователя, например, на клики или ввод текста. Он может получать и выводить текст, создавать интерактивные функции, изменять содержимое страницы динамически. JavaScript делает сайт интерактивным, превращая статичную картинку в работающую программу.</p>
8	<p>Что такое DOM. DOM (Document Object Model) – это способ видеть HTML-страницу как структуру объектов. Представьте, что каждый элемент на странице имеет свой адрес. С помощью JavaScript мы можем найти элемент по</p>

Ход занятия

	<p>этому адресу, прочитать и изменить его содержимое. Пример: мы можем сказать программе найди поле с именем userInput и возьми оттуда текст. Это как адресная книга для элементов вашего сайта, чтобы код знал, куда отправлять данные.</p>
9	<p>Пишем HTML-страницу (index.html). Мы будем писать код HTML-страницы с полем ввода, кнопкой и параграфом для вывода ответа. Объясните структуру: у элементов есть ID, это их уникальное имя, чтобы JavaScript мог их найти. А атрибут onclick означает, что при клике на кнопку запустится определенная функция. У элемента input имеется атрибут placeholder, который используется для вывода подсказки для пользователя внутри элемента. Внутри тега button указывается текст, который будет отображаться на кнопке. Мы создадим форму, куда пользователь сможет вписать вопрос, и место, где появится ответ. Это интерфейс нашего будущего общения с ИИ. Продемонстрируйте код html</p>
10	<p>JavaScript обработчик событий (script.js). Здесь мы рассмотрим функцию askGPT, которая будет управлять процессом. Она делает две важные вещи: сначала получает текст из поля ввода, чтобы понять, что спросил пользователь с помощью метода getElementById, а потом выводит этот текст на страницу с помощью метода innerText, чтобы мы видели результат. Это базовый пример обработки события клика. Пока мы просто передаем текст внутри страницы, но скоро научимся отправлять его настоящему искусственному интеллекту. Продемонстрируйте код JS Скажите, что применяется условие для проверки на пустой ввод данных</p>
11	<p>Самостоятельная работа. Задание для учеников: 1) измените надпись на кнопке, 2) установите новый placeholder для поля ввода, 3) добавьте проверку на пустой ввод, 4) проверьте, как работает JavaScript-ответ. Цель:</p>

Ход занятия

	<p>научиться связывать HTML и JavaScript. Дайте ученикам 5-7 минут на выполнение. Не бойтесь экспериментировать с текстом!</p>
12	<p>Проверка понимания. Задайте ученикам вопросы: Что произойдёт при пустом вводе? Как работает свойство <code>innerText</code>? Что показывает <code>console.log()</code>? Где найти ошибки в коде? Обсудите: почему важно проверять значение до отправки? Это закрепит понимание материала.</p>
13	<p>Итоги занятия. Подведите итоги: сегодня вы научились создавать базовый web-интерфейс, познакомились с HTML и JavaScript, создали веб-страницу с формой, написали обработчик событий, научились работать с DOM и отлаживать код. Спросите учеников, что было самым интересным. Вы большие молодцы, теперь у вас есть своя веб-страница. Это уже не просто код в тетради, это работающий интерфейс, который виден в браузере.</p>
14	<p>Анонс следующего занятия. На следующем уроке вы изучите отправку запросов к GPT API с помощью <code>fetch</code>, научитесь обрабатывать JSON-ответы, добавите индикатор загрузки, проведёте практику с API-ключом и создадите веб-страницу, которая по-настоящему общается с искусственным интеллектом. Это будет очень увлекательно! Представьте, что ваша страница сама станет умной. Обязательно выполните домашнее задание, чтобы быть готовыми к магии следующего урока. Всем спасибо, занятие окончено!</p>

Практика

Вопросы для самопроверки

- Для чего нужен язык HTML?
- Что делает JavaScript на веб-странице?
- Что такое DOM?
- Как найти HTML-элемент по его ID в JavaScript?
- Как получить значение из поля ввода `<input>`
- Как изменить текст внутри элемента `<p>`

Задание 1

Создайте HTML-страницу с формой для ввода вопроса и JavaScript-функцию, которая отображает введённый текст.

Примеры выполнения заданий

Проверь себя по следующим критериям:

- HTML – язык разметки для создания структуры веб-страницы (заголовки, абзацы, формы и т.д.).
- JavaScript делает страницу интерактивной: реагирует на действия пользователя, изменяет содержимое, отправляет запросы и т.п.
- DOM (Document Object Model) – программное представление HTML-документа в виде дерева объектов, с которым можно работать через JavaScript.
- `const el = document.getElementById('myId');`
- `const value = document.getElementById('inputId').value;`
- `document.getElementById('pld').textContent = 'Новый текст';`(или `.innerHTML`, если нужен HTML).

Задание 1

Открыть текстовый редактор (VS Code, Notepad++ или любой другой).

Создать файл `index.html`.

Написать HTML-код с формой (`input`, `button`, `p`).

Создать файл `script.js` в той же папке.

Написать функцию `askGPT()` для обработки ввода.

Открыть `index.html` в браузере (двойной клик по файлу).

Проверить работу: ввести текст и нажать кнопку.

При ошибках открыть консоль разработчика (F12 в браузере).

Полное решение:

Файл: `index.html`

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Примеры выполнения заданий

```
<title>Вопрос к GPT</title>

</head>

<body>

  <h1>Задай вопрос</h1>

  <input id="userInput" type="text" placeholder="Напишите вопрос здесь">

  <button onclick="askGPT()">Отправить</button>

  <p id="response"></p>

  <script src="script.js"></script>

</body>

</html>
```

Файл: script.js

```
function askGPT() {

  const question = document.getElementById("userInput").value;

  const responseField = document.getElementById("response");
```

Сегодня на занятии вы:

- Научились создавать базовый web-интерфейс для взаимодействия с GPT.
- Познакомились с основами HTML и JavaScript: теги `<input>`, `<button>`.
- Создали веб-страницу с формой «Задай вопрос» и подключённым `script.js`
- Написали обработчик событий: чтение значения из поля и вывод ответа.
- Научились работать с DOM: находить элементы и изменять их содержимое.
- Узнали, как отлаживать код с помощью консоли разработчика.

На следующем занятии вы:

- Повторите HTML-форму и JavaScript-функцию из этого занятия.
- Изучите отправку запросов к GPT API с помощью `fetch`.
- Научитесь обрабатывать JSON-ответы от сервера.
- Создадите веб-страницу, которая по-настоящему общается с искусственным интеллектом.

Получение фидбека. Рефлексия. Проверка знаний

Попросите ребят решить интерактивные задачи по теме урока.

Ответы к интерактивным задачам

Практическое задание 1

Как называется язык разметки для создания структуры веб-страниц?

HTML

Практическое задание 2

Что такое DOM в контексте веб-разработки?

- Язык программирования для создания игр.
- **Структура HTML-элементов как объекта, с которым можно работать через JavaScript.**
- База данных для хранения информации о пользователях.
- Графический редактор для веб-дизайна.

Практическое задание 3

Какие элементы необходимы для создания простой интерактивной веб-формы? (выберите все правильные варианты)

- **Поле ввода <input>**
- **Кнопка <button>**
- **JavaScript-функция для обработки события**
- База данных SQL

Ответы к интерактивным задачам

Вопрос-ответ

Если во время подготовки или проведения урока возникают вопросы, их необходимо адресовать в методический чат.