МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ

КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

Таймер для шахмат  
  
  
По дисциплине «Проектный практикум»

Выполнил: студент гр. БВТ2201

Крюков Д.А

Проверил:

Москва, 2023 г.

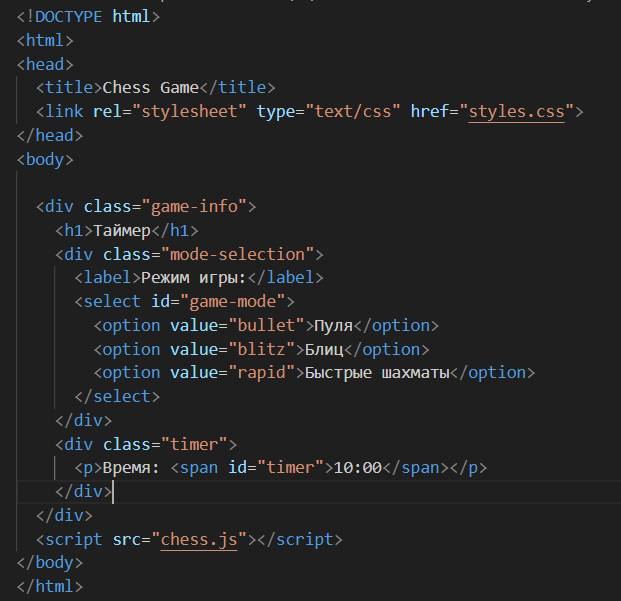
**Введение:**

Для реализации проекта нам понадобится HTML файл, в котором будет вся разметка страницы, выпадающие меню и прочие элементы на которые будет ссылаться наш JavaScript код.  
Также нам понадобится CSS файл(он необязателен,но глазу приятнее), который неободим для стилизации оформления элементов на нашей веб-странице.

Наконец нам понадобиться наш файл JavaScript в котором будет прописана вся логика и интерактивность.

**Основная информация**

1. Начну с HTML документа

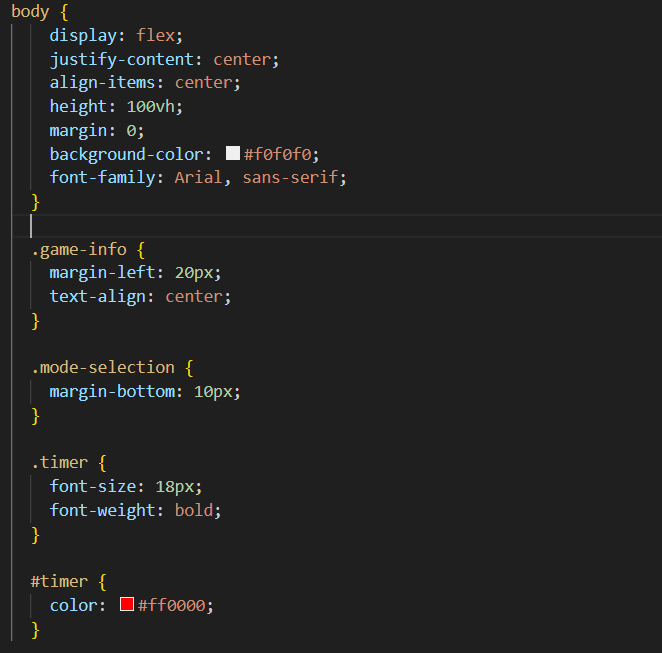


В начале мы объявляем что это именно HTML5 – документ, используя !DOCTYPE html, весь наш контент будет находиться между корневым элементом веб-страницы – <html>. Элемент <head> содержит информацию о документе, такую как: заголовок, страницы, стили и скрипты. Title устанавливает заголовок страницы, а <link> подключает внешний файл стилей

<styles.css> при помощи атрибута href(гипертекстовая ссылка, которая задает адрес на который надо перейти).<body> содержит фактическое содержание страницы, которое будет видно для пользователя. В <body> мы имеем блок с классом game info, внутри этого контейнера есть заголовок 1-го уровня h1,

вложенный блок с классом mode-selection, который представляет собой секцию для выбора режима, текстовую метку <label>, которая сопровождает выпадающее меню, сам элемент выпадающего меню <select> с идентификатором <game mode>, вложенный блок с классом timer с элементом абзаца <p> и тег скрипта <script src="chess.js"></script>, который подключает внешний файл chess.js для добавления логики и интерактивности к странице.

2. CSS файл.



body: Этот селектор применяется ко всем элементам <body> на веб-странице.

display: flex;: Это свойство задает значение "flex" для отображения элементов <body>. Это используется для создания гибкой структуры страницы с целью выравнивания элементов по горизонтали и вертикали.

justify-content: center;: Это свойство задает значение "center" для горизонтального выравнивания элементов внутри <body>, что помещает их в центре горизонтальной оси.

align-items: center;: Это свойство задает значение "center" для вертикального выравнивания элементов внутри <body>, что помещает их в центре вертикальной оси.

height: 100vh;: Это задает высоту элемента <body> равной 100 вьюпортам (высоте экрана браузера), что обеспечивает вертикальное выравнивание на всю высоту экрана.

margin: 0;: Это устанавливает нулевые отступы по умолчанию для элемента <body>, чтобы убрать внешние отступы.

background-color: #f0f0f0;: Это задает фоновый цвет для элемента <body>, который становится светло-серым (#f0f0f0).

font-family: Arial, sans-serif;: Это определяет шрифт для текста на странице. Если шрифт Arial недоступен, браузер будет использовать стандартный шрифт без засечек (sans-serif).

.game-info: Этот селектор применяется ко всем элементам с классом "game-info".

margin-left: 20px;: Это задает левый отступ элементу "game-info" равный 20 пикселям, чтобы создать отступ от левого края.

.mode-selection: Этот селектор применяется ко всем элементам с классом "mode-selection".

margin-bottom: 10px;: Это задает нижний отступ элементу "mode-selection" равный 10 пикселям, что создает вертикальный отступ под этим элементом.

.timer: Этот селектор применяется ко всем элементам с классом "timer".

font-size: 18px;: Это задает размер шрифта для элементов "timer" равным 18 пикселям.

font-weight: bold;: Это делает текст элементов "timer" жирным (полужирным).

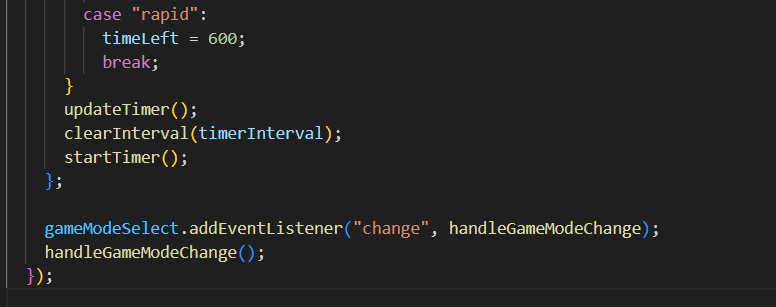
#timer: Этот селектор применяется к элементу с идентификатором "timer".

color: #ff0000;: Это задает красный цвет текста внутри элемента "timer" (#ff0000).

В итоге, этот CSS код задает стили и оформление для различных элементов на странице, включая фон, выравнивание, размеры, отступы и шрифты.

3. JavaScript файл





document.addEventListener("DOMContentLoaded", function () { ... });: Этот фрагмент кода устанавливает обработчик события "DOMContentLoaded". Это событие срабатывает, когда весь HTML-документ полностью загружен и готов к манипуляциям JavaScript. Весь код находится внутри этого обработчика, чтобы убедиться, что он начнет выполняться только после загрузки страницы.

const gameModeSelect = document.getElementById("game-mode");: Эта строка находит HTML элемент с идентификатором "game-mode" и сохраняет его в переменной gameModeSelect. Этот элемент представляет собой выпадающее меню, в котором пользователь может выбрать режим игры (bullet, blitz или rapid).

const timerElement = document.getElementById("timer");: Эта строка находит HTML элемент с идентификатором "timer" и сохраняет его в переменной timerElement. Этот элемент будет использоваться для отображения оставшегося времени игры.

let gameMode = "bullet";: Эта строка инициализирует переменную gameMode значением "bullet". По умолчанию выбран режим игры "bullet".

let timeLeft = 600;: Эта строка инициализирует переменную timeLeft значением 600, что представляет собой количество секунд (10 минут) по умолчанию для игры.

let timerInterval;: Эта строка объявляет переменную timerInterval, которая будет использоваться для хранения идентификатора интервала таймера.

const updateTimer = () => { ... };: Это объявление функции updateTimer. Эта функция обновляет текстовое содержимое элемента timerElement для отображения текущего оставшегося времени в минутах и секундах. Функция вычисляет минуты и секунды на основе значения timeLeft.

const startTimer = () => { ... };: Это объявление функции startTimer. Эта функция запускает интервал, который каждую секунду уменьшает значение timeLeft и вызывает функцию updateTimer для обновления отображаемого времени. Если время истекло, интервал очищается.

const handleGameModeChange = () => { ... };: Это объявление функции handleGameModeChange. Эта функция вызывается при изменении выбранного режима игры в выпадающем меню. В зависимости от выбранного режима, она обновляет timeLeft, останавливает текущий таймер (если он запущен) и запускает новый таймер.

gameModeSelect.addEventListener("change", handleGameModeChange);: Это добавляет прослушиватель события "change" к элементу gameModeSelect, который будет вызывать функцию handleGameModeChange, когда выбранный режим игры изменяется.

handleGameModeChange();: Это вызывает функцию handleGameModeChange при начальной загрузке страницы для установки начальных значений времени и запуска таймера.