

Многие сайты по старым компьютерам были выполнены в соответствующем стиле – в стиле той эпохи, когда такая вещь, как Интернет, не казалась вполне обычным явлением. Тогда даже CSS был диковинкой, а JavaScript использовался довольно редко. Интерфейс был простым, с ограниченной цветовой гаммой. Однако как бы выглядел сайт такой тематики в более современном стиле, с использованием новых и актуальных технологий?

Сайт разбит на несколько категорий, по различным компонентам, используемых в компьютерах. На каждой странице сверху расположен список с миниатюрами экспонатов, которые можно выбрать и просмотреть. Снизу дается описание каждого экспоната. Для удобства мешающие элементы страницы можно скрыть. Данный сайт оптимизирован для работы с устройствами с сенсорными экранами, в частности с моноблоками и планшетами.

Отдельная категория посвящена мобильным телефонам и смартфонам. В некотором смысле телефоны можно ограничено считать узкоспециализированными компьютерами. Со временем они тоже развивались и развиваются сейчас, сегодня даже догоняя по возможностям современные ноутбуки. Поэтому они тоже попадают под рассмотрение с точки зрения исторического прогресса.

Сайт был разработан с использованием технологий HTML, CSS, JavaScript и JQuery.

Если подводить итоги, то можно сказать, что использование старой компьютерной техники набирает популярность среди любителей старины, ностальгирующих людей, и просто теми людьми, кто интересуется развитием компьютеров. Для того, чтобы двигаться дальше, необходимо не допускать ошибок, совершенных раньше, и без истории дальнейший прогресс невозможен.

Р.Р. Апасов (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.С. Давыдов**, канд. техн. наук, доцент

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ «CHART.JS» ДЛЯ СОЗДАНИЯ ДИАГРАММ В ПРИЛОЖЕНИИ ПО КОНТРОЛЮ АВИАПЕРЕВОЗОК

В веб-приложениях, содержащих множество таблиц с числовыми данными, большое значение имеет отображение данных. В приложении по контролю авиаперевозок для наглядного отображения данных используются диаграммы.

Один из способов создания и отображения диаграмм – построение диаграммы на стороне сервера. Серверный скрипт на основе данных генерирует статическое изображение диаграммы и передаёт его клиенту. Недостатками данного способа являются большие ресурсные затраты на передачу изображения, а также отсутствие динамики на диаграммах.

Более актуальным вариантом является использование готовых библиотек для построения диаграмм, конкретно библиотеки `chart.js`. Данная Javascript-библиотека является удобным инструментом, она не требует наличия каких-либо других зависимостей. `Chart.js` содержит много готовых типов диаграмм: линейный график (Line Chart), гистограмма (Bar Chart), лепестковая диаграмма (Radar Chart), кольцевая диаграмма (Doughnut Chart) и прочие известные решения.

Для отображения диаграммы `chart.js` использует элемент `html5 Canvas`. Идея элемента `Canvas` впервые выдвинута и реализована компанией Apple. До появления инструмента `Canvas` возможность рисования в браузерах была осуществима только при условии привлечения дополнительных подключаемых модулей, таких как Flash, плагины Adobe, языка векторной разметки VML либо других программных решений на основе JavaScript.

В HTML5 элемент `Canvas` определён как растровый холст, который может быть использован для отображения диаграмм или игровой графики при помощи скриптов, как правило, на языке Javascript.

Растровый холст представляет собой двумерную сетку. Координата 0,0 находится в левом верхнем углу холста. Вдоль оси X значения увеличиваются к правому краю холста. По оси Y значения увеличиваются к нижнему краю холста. Численные значения координат выражаются в пикселях.

Элемент `Canvas` фактически не помещается на страницу, вместо него формируется рисунок в двумерном контексте. Чтобы воспользоваться элементом `canvas`, необходимо создать соответствующий HTML тег, а затем задать ширину и высоту. Для получения доступа к двумерному контексту следует применить Javascript-функцию «`getContext`», после чего можно приступить к рисованию. Для изображения прямоугольников используются функции «`fillRect`» и «`strokeRect`». В качестве параметров в функцию передаются координаты верхнего левого угла прямоугольника, ширина и высота. Функция «`fillRect`» рисует закрашенный прямоугольник, функция «`strokeRect`» рисует прямоугольник без заливки. Стили заливки и контура устанавливаются при помощи методов «`fillStyle`» и «`strokeStyle`», поддерживается несколько способов задания цвета: шестнадцатеричные коды, `rgb()`, `rgba()`.

Кроме прямоугольников существует возможность рисовать фигуры по собственной схеме. Начинается рисование с вызова метода

«beginPath», который определяет начало схемы. Метод «lineTo» рисует линию от текущей позиции в указанную точку, но не выводит на холст. Для отображения линии используется метод «stroke». Для перемещения курсора в указанную точку используется метод «moveTo». Метод «closePath» замыкает текущий путь. Метод «fill» закрашивает и отображает фигуру, нарисованную при помощи линий.

Также в Canvas присутствует возможность рисовать дуги и кривые, что в совокупности позволяет создавать сложные изображения. Для отображения дуги используется функция «arc». В качестве параметров передаются координаты центра окружности, на которой будет лежать дуга, радиус окружности, начальный и конечный углы, заданные в радианах. Для отображения квадратичной кривой используется метод «quadraticCurveTo». Кривая строится по трём точкам, в качестве параметров функция принимает координаты положения курсора и координаты двух других точек.

Canvas предоставляет возможность создания кривых Безье. Для этого существует два метода: «quadraticCurveTo» и «bezierCurveTo». Первый метод используется для построения квадратичных кривых Безье. В качестве аргументов передаются координаты контрольной точки и конечной точки. Вторым методом используется для построения кубических кривых Безье. В качестве аргументов передаются координаты двух контрольных точек и координаты начальной точки.

Для очистки отдельной области холста используется метод «clearRect», который принимает в качестве параметров координаты верхней левой точки очищаемой области, ширину и высоту.

Таким образом, элемент Canvas обладает всеми необходимыми средствами для создания изображений на html-страницах без использования сторонних модулей.

Для создания диаграммы необходимо получить доступ к двумерному контексту элемента Canvas, после чего необходимо выбрать тип диаграммы и задать два параметра – входные данные и конфигурацию. Конфигурационный файл содержит большое количество опций, позволяющих изменять размер и отображение элементов диаграммы. Существуют глобальные опции и опции для отдельных типов диаграмм.

Библиотека для построения диаграмм chart.js предоставляет возможность создания различных комбинаций диаграмм, а также настройку их отображения. Благодаря тому, что диаграммы создаются на стороне клиента, они также способны легко изменяться, то есть диаграммы, построенные при помощи chart.js, являются динамическими.