D3.js

**D3** позволяет привязать произвольные данные к модели DOM, а затем применить к документу управляемые данными преобразования. Например, D3 можно использовать для создания HTML-таблицы из массива чисел. Или используйте те же данные для создания интерактивной линейчатой диаграммы SVG с плавными переходами и взаимодействием.

D3 не является монолитной структурой, которая стремится обеспечить все мыслимые функции. Вместо этого D3 решает суть проблемы: эффективное манипулирование документами на основе данных. Это позволяет избежать проприетарного представления и обеспечивает исключительную гибкость, предоставляя все возможности веб-стандартов, таких как HTML, SVG и CSS. С минимальными накладными расходами D3 чрезвычайно быстр, поддерживая большие наборы данных и динамическое поведение для взаимодействия и анимации. Функциональный стиль D3 позволяет повторно использовать код через разнообразную коллекцию [официальных](https://github.com/d3/d3/blob/main/API.md) и [разработанных сообществом](https://www.npmjs.com/browse/keyword/d3-module) модулей.

D3.js представляет библиотеку на языке JavaScript для обработки и визуализации данных. Само название D3 расшифровывается как Data-Driven Documents и как бы делает упор на управление данными, хотя ключевой функциональностью библиотеки являются мощные возможности для их визуализации.

Библиотека D3.js основана прежде всего на использовании JavaScript, SVG и CSS в противовес другим подобным библиотекам, которые вместо SVG используют элемент canvas и его возможности. Если стандартные механизмы рисования, например, элемент canvas, полагаются на пиксели, то svg использует векторы. Применение SVG позволяет создавать структуры с насыщенной графикой, обладающие анимацией и возможностями взаимодействия.

По сравнению с пиксельной графикой SVG обладает рядом преимуществ. В частности, SVG основан на xml, что делает его более читабельным. Кроме того, код SVG более легковесный по сравнению файлами изображений.

Во многом благодаря этому D3 в настоящее время является одним из наиболее популярных фреймворков, используемых для графической обработки данных и создания всякого рода диаграмм и графиков.

В отличие от других подобных JavaScript-библиотек D3 не использует jQuery для работы со структурой DOM, хотя в то же время реализует похожие концепции для работы с элементами.

Что потребуется для разработки с D3.js? Во-первых, нам надо загрузить саму библиотеку с официального сайта http://d3js.org/.

Также можно ссылаться на библиотеку напрямую из CDN с официального сайта:

<script src="http://d3js.org/d3.v3.js"></script>

SVG и D3.js

Как уже говорилось, D3.js использует SVG, предполагающий применение векторной графики для визуализации данных. Сделаем небольшой пример с SVG и D3.js. Создадим небольшую html-страницу:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<script src="http://d3js.org/d3.v3.min.js"> </script>

</head>

<body>

<h3>SVG</h3>

<svg>

<rect width="50" height="200" style="fill:red;" />

</svg>

<h3>D3.js</h3>

<script>

d3.select("body")

.append("svg")

.append("rect")

.attr("width", 50)

.attr("height", 200)

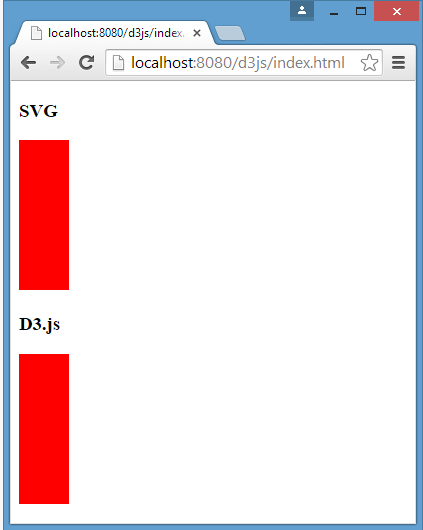
.style("fill", "red");

</script>

</body>

</html>

Запустим эту страницу в браузере. Мы получим два идентичных красных столбика:



Вначале подключаем из официального CDN саму библиотеку:

<script src="http://d3js.org/d3.v3.min.js"> </script>

Затем создается простейшая графика на svg. Если вы не знакомы с SVG, работать с данной технологией будет относительно не сложно. SVG позволяет использовать готовые блоки (линии, круги, прямоугольники и т.д.) для создания графики. В данном случае с помощью элемента rect выводится прямоугольник.

Далее идет собственно работа D3.js:

d3.select("body") // получаем элемент body

.append("svg") // добавляем элемент svg

.append("rect") // добавляем в svg элемент rect

.attr("width", 50) // установка у rect ширины

.attr("height", 200) // установка у rect высоты

.style("fill", "red"); // установка у rect стиля

Фактически здесь будет формироваться с помощью цепочки вызовов тот же элемент svg с идентичным html-кодом, у которого устанавливаются аналогичные атрибуты и стиль. Только в данном случае мы уходим от прямого использования технологии SVG в сторону javascript.

Каковы возможности D3.js? Чтобы понять, что можно делать с помощью библиотеки, достаточно посмотреть ряд примеров, которые приводятся разработчиками на странице https://github.com/mbostock/d3/wiki/Gallery. Это прежде всего, но не только, графики, диаграммы, карты с богатой функциональностью и многое другое.