

Анализ реализации новых WEB-стандартов в массовом программном обеспечении

Рассмотрены вопросы поддержки новых возможностей по выполнению верстки WEB-страниц в наиболее распространенных браузерах. Приведены примеры текстов страниц для проверки и результаты проверки.

Ключевые слова: *верстка, WEB-стандарты, таблицы стилей.*

В ответ на расширение номенклатуры и возможностей устройств, с которых пользователи выходят в Интернет, расширяются и возможности программных средств отображения данных. При этом разработчики программного обеспечения, с одной стороны, предлагают новые решения, а с другой, вынуждены следовать стандартам для обеспечения совместимости на разных платформах и системах. Со своей стороны, стандарты WEB тоже постоянно развиваются, впитывая и обобщая опыт разработчиков, поддерживая или отвергая различные решения. Эти процессы идут постоянно и зачастую несогласованно. В результате разработчики WEB-ресурсов вынуждены не только тестировать создаваемый продукт для большого числа различных вариантов, но и принимать специальные меры для обеспечения отображения контента на всей номенклатуре устройств и программного обеспечения. Несмотря на то, что ситуация в этой сфере меняется достаточно быстро, соотношение между принятыми стандартами и их реализацией в массовом ПО представляет интерес.

Ведя речь о направлениях развития средств создания WEB-ресурсов и стандартов, обычно, выделяют базовые компоненты HTML, CSS, API, JavaScript. Таких направлений и новшеств в этих компонентах достаточно много, например, средства семантической разметки, средства мультимедиа, графики, новые модули CSS, геолокация, локальные хранилища данных, средства асинхронного взаимодействия с сервером и многое другое. Часть из этих новых направлений пока не слишком востребована, часть поддерживается лишь частично, а часть вообще пока остаются пожеланиями. Более того, многие возможности дополняют или даже дублируют друг друга. Похожие тенденции можно наблюдать практически со всеми средствами.

Необходимо отметить еще один момент, а именно широкое распространение фреймворков, как

подхода к повторному использованию кода и решению проблем с неодинаковой поддержкой базовых компонент в различных средах. Если проанализировать современные подходы к разработке, обнаружим, что практически во всех использованы ранее сделанные наработки (шаблоны, фрейворки и т.п.). Трудно обнаружить, например, страницы, сверстаные только с использованием базовых средств. Однако эти решения позволяют преодолеть проблемы лишь частично, порождая новые, но уже на уровне архитектуры [1]. Потому остается очень важной поддержка именно базовых средств. Чем более развита база, тем проще надстройка.

Проанализируем степень поддержки процесса верстки страниц в современных браузерах, как самом массовом программном обеспечении для работы с WEB.

Что влияет на этот процесс? Во-первых, подготовка и практика работы специалистов. Многие используют наработанные способы. Например, несмотря на общую тенденцию к семантической верстке, можно встретить как высказывания о допустимости, так и использование для позиционирования элементов тега **table**. Во-вторых, современные требования, самым существенным из которых является адаптация контента к условиям просмотра. И, в-третьих, уровень поддержки в массовом программном обеспечении. Современные стандарты предлагают множество новых средств, которые можно использовать при верстке. Для анализа остановимся на модулях Exclusions and Shapes, Grid Layout, Flexible Box, как таких, что имеют непосредственное отношение к верстке, конкурируют с современным подходом на основе позиционируемых и обтекаемых разделов или продолжают его.

Модуль CSS Multi-Columns предназначен для размещения контента в несколько колонок. Основное свойство `column-count` задает число колонок, на которое разбивается контент. Рисунки, ширина которых превышает ширину колонки обрезаются. Свойство `column-width` задает ширину колонки, может использоваться самостоятельно и вместе с `column-count`, значение которого в этом случае интерпретируется как максимальное число колонок. Кроме того, можно изменять ширину пустого пространства между колонками (`column-gap`), устанавливать разделительную линию (`column-rule`), растягивать элемент на несколько колонок (`column-span`), управлять высотой колонки (`height` или `max-height`).

Код страницы для проверки
`<div class="sec">`
размещаемый текст, включая рисунки

...
`</div>`

Таблица стилей:
`.sec {`
`column-count: 3;`
`-webkit-column-count: 3;`
`}`

Браузеры Internet Explorer 11 и Edge поддерживают эти свойства непосредственно, Google Chrome – с вендорным префиксом `-webkit`.

CSS Flexible Box Layout [2] - это модуль, содержащий свойства, управляющие заполнением блоками пространства родительского элемента. Заполнение выполняется равномерно или в некоторой пропорции. При этом размеры элементов внутри родительского будут определяться автоматически. Таким образом, решаются проблемы учета размера, порядка и выравнивания элементов по горизонтали и вертикали, распределения свободного места между элементами. Это позволяет адаптировать отображение, например, при изменении ориентации, растягивании или сжатии экрана на мобильных устройствах.

Дочерние элементы могут быть объединены в строки или распределены по колонкам, можно управлять выравниванием, отступами и размерами этих элементов.

Эти свойства позволяют автоматически заполнять выделенное место и предотвратить выход за границы родительского контейнера, что существенно упрощает процесс верстки.

Код для проверки поддержки технологии браузером приведен ниже.

Текст страницы:
`<div class="box_">`
`<header class="header_">`
`<h1>Здесь`
должен быть заголовок`</h1>`

`</header>`
`<div class="body_">`
`<div class="content_">`
`<h2>Здесь основное содержание</h2>`
`<p>Здесь произвольный текст</p>`
`</div>`
`<div class="sidebar_ sidebar_first">`
`<h4>Первая боковая панель</h4>`
`</div>`
`<div class="sidebar_ sidebar_second">`
`<h4>Вторая боковая панель</h4>`
`</div>`
`</div>`
`<footer class="footer_">`
`<small>А это подвал</small>`
`</footer>`
`</div>`

Таблица стилей:

```
.box_ {display: flex; flex-direction: column;}
.footer_, .header_ {background-color: #faf8f0; flex: 0 0 100%; padding: 12px; text-align: center;}
.footer_, .header_ h1 { color: #0f0f0f;}
.body_ {display: flex; flex-direction: column;}
.sidebar_ {background-color: #e7e7e7; padding: 12px;}
.sidebar_first { order: 1;}
.sidebar_second { order: 3;}
.content_ { order: 2; padding: 12px;}
@media all and (min-width: 720px) {
  body_ {flex-direction: row;}
  .sidebar_first { flex: 0 0 150px;}
  .sidebar_second { flex: 0 0 210px;}
  .content_ {flex: 1 1 0;}
}
@media all and (min-width: 960px) {
  .sidebar_first { flex: 0 0 180px;}
  .sidebar_second { flex: 0 0 250px;}
}
```

Пример демонстрирует использование Flexible Box для верстки с адаптацией контента к размерам экрана. Последние версии массовых браузеров достаточно полно поддерживают эти возможности.

Может показаться, что Flexible Box и Multi-Columns решают одну и ту же задачу, но это не так. В первом случае идет управление блоковыми элементами, их положением, выравниванием, размерами без изменения внутренней структуры блоков. Во втором случае изменяется контент, его структура (распределение его по колонкам).

Окончательный вывод об использовании этих возможностей можно будет сделать после накопления опыта [1].

Достаточно часто верстка выполняется на основе сетки, некоторой регулярной структуры, состоящей из строк и столбцов. Простейшим и дружно осуждаемым способом ее создания является `table`. Сетка используется во многих

фреймворках. Поэтому включение модуля Grid Layout [3] в спецификацию CSS3 вызывает закономерный интерес. Grid Layout это реализация подхода к табличной верстке. Контент на странице распределяется по строкам и столбцам таблицы с помощью, соответствующих свойств и может быть связан с семантическими элементами (header, footer, и т.п.). По сути, это подход к выстраиванию элементов по некоторой виртуальной сетке.

Сама сетка определяется в свойствах контейнера. Дочерние элементы привязываются к сетке по номеру столбца и строки, можно занимать несколько ячеек. Можно использовать разные сетки для разных устройств.

Используя пример легко убедиться, что даже последние версии браузеров эти возможности пока не реализуют.

Код разметки:

```
<div class="grid">
  <div class="grid-item grid-item1">item 1</div>
  <div class="grid-item grid-item2">item 2</div>
  <div class="grid-item grid-item3">item 3</div>
</div>
```

Таблица стилей:

```
.grid {
  display: grid;
  grid-template-columns: 100px 1fr 100px;
}
```

Существуют еще несколько модулей, с которыми связаны достаточно многообещающие возможности. Это Template Layout, CSS Exclusions и CSS Regions.

Список литературы

1. Николас Бевакка Хватит разрушать веб [Электронный ресурс]. - Электрон. дан. - Режим доступа: <http://web-standards.ru/articles/stop-breaking-the-web/>
2. Полное руководство по Flexbox [Электронный ресурс]. - Электрон. дан. - Режим доступа: <http://frontender.info/a-guide-to-flexbox/#polnoerukovodstvopoflexbox>
3. Введение в CSS3 Grid Layout. Работаем с сетками [Электронный ресурс]. - Электрон. дан. - Режим доступа: <https://habrahabr.ru/company/microsoft/blog/140715/>

Рецензент: д-р эконом. наук, проф. А.И. Пушкар, Харьковский национальный экономический университет им. Семе́на Кузнеца, Харьков.

Автор: МОЛЧАНОВ Виктор Петрович

Харьковский национальный экономический университет им. Семе́на Кузнеца, Харьков, кандидат технических наук, доцент, доцент кафедры КСuT.

Раб. тел. – 702-06-74 (4-01), дом. тел. 336-06-98, E-mail – Viktor.Molchanov@hneu.net

Аналіз реалізації WEB-стандартів в масовому програмному забезпеченні

В.П. Молчанов

Розглянуто питання підтримки нових можливостей щодо виконання верстки WEB-сторінок в найбільш поширених браузерах. Наведено приклади текстів сторінок для перевірки та результати перевірки.

Ключові слова: верстка, WEB-стандарти, таблиці стилів.

Analysis of the implementation of new standards in the WEB-mass software

V. Molchanov

The issues to support new opportunities for the implementation of the layout of WEB-pages in the most common browsers. Examples of text pages to test and test results.

Keywords: make up in pages, WEB-standards, the stylesheet