

Нельзя сказать, чтобы доступная на сегодня веб-дизайнерам технология текстовой разметки — HTML с небольшой (из-за проблем совместимости) примесью CSS — была начисто лишена способности к разделению аспектов содержания и представления (стр. 21). Опыт, врожденная аккуратность и ответственное отношение к материалу, с которым приходится работать, позволяет отдельным дизайнерам практиковать в HTML стиль, вполне отвечающий требованиям идеологии SGML (или, что сейчас более актуально, XML).

Модульный HTML

Конечно, многим дизайнерам с преимущественно визуальным мышлением совсем не просто перестроиться на «ортогональный стиль» разметки. Так же как нельзя увидеть бестелесную душу, вам, возможно, трудно вообразить себе, как будет выглядеть документ, размеченный *только* логически, равно как и представить себе идеальную ортогональность — независимость такого «дистиллированного» содержимого от хранящегося отдельно оформления. Если даже примитивные «именованные стили» в текстовых процессорах считаются прерогативой «профессиональных пользователей», что уж говорить о более последовательных системах ортогональной разметки. Я думаю, что если бы умение воспринимать и создавать аспекты информации по отдельности было врожденным и не требовало обучения, язык SGML уже давно стал бы основным средством хранения и распространения текстов.

Режем по живому. Даже если не учитывать несовершенство HTML, в котором логический и визуальный аспекты оказались смешанными по причинам скорее историческим, соблюдение ортогональности — как и любая реализация некоей абстрактной идеи на практике — сталкивается и с вполне объективными трудностями. Бывают случаи, в которых раздельная линия между содержанием и оформлением может быть проведена по-разному; более того, иногда неудачное расщепление на аспекты документа, изначально (в сознании его автора) целостного, приводит к частичной потере информации и к невозможности в дальнейшем удовлетворительно состыковать получившиеся половинки.

Приведу пару примеров. В двумерных композициях с текстом и изображениями часть информации о связях между элементами может передаваться не последовательностью их расположения или какими-нибудь видимыми стрелками или рамками, а менее очевидными визуальными средствами — выравниванием, цветовыми переключками, контрастом. Если композиция эта создавалась изначально в графической среде, ее автор, возможно, просто не осознает некоторые из этих связей и, соответственно, не сможет «вербализовать» их при выделении структурной основы композиции.

С другой стороны, некоторые фрагменты текста относятся не к содержательной основе, а к оформительской надстройке документа: например, номер главы и само слово «Глава» в заголовке, постоянная часть перекрестных ссылок (т. е. сокращения типа «стр.» или «гл.»), любые повторяющиеся элементы, такие как колонтитулы на странице книги или панель навигации на веб-странице. Вынеся все это из текстовой основы документа в стилевые спецификации, вы не только упростите процедуру глобального изменения этих элементов во всем документе, но и приблизитесь к искомому идеалу ортогональности: ведь все, что при внимательном рассмотрении не принадлежит к уникальной информации документа, а лишь помогает воспринимать ее, правильнее отнести к аспекту представления, а не содержания.

Сборно-панельный сайт. Однако вернемся к HTML. Поскольку в случае этого языка одна и та же технология ответственна за оба аспекта разметки, необходимо придерживаться определенных правил, которые позволяют

если не разделить содержание и оформление, то по крайней мере сделать их хоть сколько-нибудь независимыми друг от друга.

На любом сайте, превышающем по размеру страницу и содержащем хотя бы одну серию повторяющихся или однотипных элементов, форматирующие коды HTML удобно собирать в унифицированные модули, или *блоки*, играющие роль своеобразных «тегов логической разметки», параметры оформления которых хранятся в них же самих. Внутреннее устройство таких блоков может быть в принципе любым — в частности, в них можно как угодно смешивать логические и визуальные теги HTML. Однако, чтобы построенный таким образом логический план разметки действительно облегчал создание и поддержку сайта, нужно придерживаться нескольких несложных правил:

- Экземпляры одного блока должны быть абсолютно идентичны, за исключением вставок изменяемого содержимого (например, текста заголовка в блоке заголовка).
- Общее количество разновидностей блоков должно быть минимальным, и после того как дизайн сайта устоялся, новые блоки могут вводиться в виде исключения — только когда на сайте появляется принципиально новое содержимое, не лезущее в старые «болванки».
- За пределами блоков не должно оставаться никаких «висячих» тегов, за исключением самых необходимых средств оформления текста (тег P и логические теги типа EM и STRONG).
- Каждый блок должен быть помечен в HTML-коде стандартным комментарием, который позволит легко опознать этот блок как при ручном редактировании, так и при автоматическом поиске.

Работа с таким модульным сайтом происходит в одном из двух режимов, соответствующих двум ортогональным аспектам его разметки. В «режиме содержания» можно как угодно редактировать существующий текст или добавлять новый, копируя при необходимости нужные блоки, но ни в коем случае не залезая *внутри* этих блоков. Эта повседневная работа по обновлению и расширению сайта не требует никакой дизайнерской квалификации, и создатель сайта вполне может перепоручить ее обслуживающему персоналу сайта.

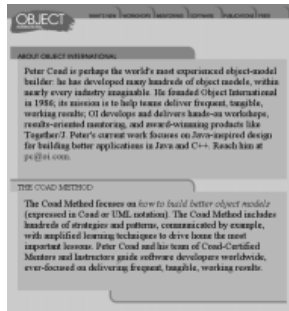


Рис. 1 «Модульная» страница на сайте www.oi.com

Наоборот, редактирование «плана представления» после того, как сайт создан и запущен, в идеале должно быть событием исключительным, осуществляющимся только под контролем дизайнера. (Например, если вдруг выяснилось, что какой-то заголовок ведет себя неправильно, когда его текст превышает по длине некую заранее планировавшуюся величину, может понадобиться изменить устройство заголовочного блока.) Это можно делать *только* глобальным поиском и заменой во всех файлах сайта — ведь если вы поправите вручную одну из копий блока, ее уже не найдет следующий автоматический поиск, и рассинхронизация поползет по сайту, как раковая опухоль. Программа, которой вы пользуетесь для редактирования HTML-кода, должна уметь искать и заменять *многострочные* блоки текста и пользоваться *регулярными выражениями* (*regular expressions*) в тех случаях, когда блок содержит вставки, изменяющиеся от одной копии блока к другой. Обе эти возможности поддерживает, например, редактор HomeSite (www.allaire.com).

Например. Описанные выше принципы были взяты за основу в дизайне сайта www.oi.com (рис. 1). Этот корпоративный сайт по объему и частоте обновления своего материала близок к контент-сайтам (стр. 182), и возможность свободно редактировать содержимое, оставляя нетронутым дизайн, для него особенно важна. Вот, к примеру, как выглядит блок, создающий стандартный внутритекстовый заголовок:

```
<!-- framed heading -->
<table border=0 cellpadding=0 cellspacing=0><tr>
<td bgcolor=ffaf60></td>
<td bgcolor=ffaf60></td>
<td bgcolor=d8d8d8 align=right valign=top rowspan=2>
</td>
</tr><tr>
<td bgcolor=d8d8d8></td>
<td bgcolor=d8d8d8 valign=bottom><small>THE COAD METHOD</small></td>
</tr></table>
```

В начале блока ставится комментарий-идентификатор, а в предпоследней его строке мы видим единственный фрагмент, изменяющийся от одного заголовка к другому, — его текст (в данном случае «THE COAD METHOD»). Между собой блоки удобно разделять пустыми строками. Вся страница, показанная на рис. 1, состоит из следующих блоков (приведены только строки с комментариями):

```
<!-- top navigation -->
<!-- solid heading -->
<!-- open text block -->
Peter Coad is perhaps ... Reach him at pc@oi.com.
<!-- close text block -->
<!-- framed heading -->
<!-- open text block -->
```

The Coad Method focuses on ... frequent, tangible, working results.

```
<!-- close text block -->
```

```
<!-- decorated close -->
```

Модульный HTML — не только имитация имеющегося в других языках программирования структурного подхода и не только единственная реальная возможность приспособить этот язык к созданию объемных и часто обновляемых сайтов. Это еще и необходимый промежуточный этап будущей миграции к языку XML (о котором мы будем говорить чуть ниже): тем же самым глобальным поиском вы в любой момент можете заменить «псевдотеги» структурных блоков HTML на настоящие структурные теги XML, разработав для них соответствующие стилевые спецификации. Такая конверсия гораздо полнее отвечает целям и духу XML, чем приходящий в голову первым буквальный, «тег в тег» перевод HTML в формально корректный, но совершенно бессмысленный XML (стр. 51), — ведь большинству визуально-ориентированных тегов HTML в структурном языке XML нет и не может быть никаких соответствий.