Написание корректного кода

Валидация документов предназначена не только для того чтобы удостовериться, что код соответствует спецификации HTML, но и с целью устранения имеющихся ошибок и замечаний в документе. Между тем, формирование определенной культуры написания кода позволяет существенно снизить или даже вообще избавиться от возможных ошибок. Такая культура складывается из знания спецификаций и типовых ляпов разработчиков, которых надо избегать.

По адресу <http://www.w3.org/TR/html401/> ознакомиться с правилами HTML версии 4.01 может каждый, здесь же мы рассмотрим рядовые ошибки и научимся, как же их обходить.

Ошибки в коде обычно возникают по следующим причинам:

* на странице не задан **<!DOCTYPE>**;
* опечатка (неверно написан тег или его атрибут);
* не указан обязательный атрибут тега;
* используется тег или его атрибут, который не входит в спецификацию;
* неверное вложение тегов.

Далее разберем эти ошибки подробнее.

**Не указан <!DOCTYPE>**

Элемент **<!DOCTYPE>** располагается в первой строке кода документа и сообщает браузеру, как интерпретировать код и отображать данную веб-страницу. Разница между страницей с **<!DOCTYPE>** и без него может быть очень существенной, к тому же валидатор в первую очередь проверяет наличие этого элемента в коде.

**Опечатки**

Очевидно, что самая простая для исправления ошибка возникает из-за опечатки, когда допущено неверное написание требуемого тега. После валидации выдается тип ошибки и номер строки в коде, где она имеется, так что остается только поменять значение на корректное.

**Не указан обязательный атрибут тега**

У некоторых тегов имеются атрибуты, которые обязательно должны присутствовать. Например, нельзя просто указать тег <style>, необходимо писать <style type="text/css">.

**Атрибут или значение не входит в спецификацию**

В порыве завоевать рынок пользователей разработчики браузеров добавляли в них специальные теги, не входящие в спецификацию HTML, но расширяющие возможности веб-страниц. Со временем часть таких тегов была включена в спецификацию, но многие так и остались «за бортом». При этом поддержка браузером осталась, так что результат работы тега наблюдать можно, но валидацию документ не пройдет. Типичным примером подобного тега является **<marquee>** придуманный компанией Microsoft и понимаемый всеми современными браузерами. Вот только в спецификацию этот тег не включен.

**Неверное вложение тегов**

Ошибка с вложением одного контейнера внутрь другого может быть вызвана следующими причинами:

* блочный элемент располагается внутри строчного, когда должно быть наоборот — строчные элементы допустимо помещать внутрь блочных;
* пересечение тегов, например, как это показано в следующем примере: <strong><em>текст</strong></em>. Здесь закрывающий тег **</strong>** помещается в контейнер **<em>**, тогда как он должен следовать только после тега **</em>**;
* не соблюдается порядок вложения тегов. В определенных элементах вроде списка и таблицы принципиальное значение имеет порядок следования тегов. Перестановка тегов местами может привести к неверному отображению объекта и появлению ошибок при валидации документа.

Напоследок отметим еще раз простые правила написания кода, соблюдение которых поможет существенно сократить количество ошибок или обойтись без них.

**Закрывайте все теги**

Хотя HTML и не требует присутствия некоторых закрывающих тегов, их наличие поможет сохранить строгость кода и четко определить порядок следования тегов.

**Указывайте значения атрибутов тегов в кавычках**

Валидатор во многих случаях пропустит значения атрибутов указанных без всяких кавычек, тем не менее, кавычки лучше писать всегда. Во-первых, подобный навык поможет для устранения возможных ошибок связанных с атрибутами тегов. А во-вторых, поможет легче перейти на XHTML (Extensible Hypertext Markup Language, расширяемый язык разметки гипертекста), синтаксически более строгую версию HTML. В XHTML кавычки выступают обязательным элементом синтаксиса.

**Коллекционируйте заготовки**

Большинство элементов веб-страницы достаточно шаблонно, поэтому имея в своем запасе набор проверенных заготовок на разные случаи, можно сократить затраты времени и быть уверенным, что код корректный.

**Используйте блочные элементы**

Нельзя так просто вставить текст в код документа, он должен располагаться внутри абзаца (тег **<p>**) или другого блочного элемента. В тех случаях, когда вы не знаете, какой блочный тег использовать, добавляйте универсальный элемент **<div>**.

**Переключайте <!DOCTYPE>**

В HTML-коде обычно применяется строгий **<!DOCTYPE>**, который наиболее полно соответствует спецификации. Однако он же и требует соблюдения всех, самых жестких правил написания кода. В тех случаях, когда это сложно или затратно по времени, переключайтесь на переходный **<!DOCTYPE>**.