Проверка валидности

Посмотрите на пример кода ниже и попытайтесь найти ошибки синтаксиса.

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="UTF-8">
    <title>Барбершоп «Бородинский»</title>
  </head>
  <body>
    <header class="main-header">
      <nav class="main-navigation">
        <ul class="site-navigation">
          <p><a href="info.html">Информация</a></p>
          <li><a>Новости</a></li>
        </ul>
      </nav>
    </header>
    <section class="features">
      <ul class="features-list">
        <li class="feature-item">
          <strong>Быстро</strong>
          <p>Мы делаем свою работу быстро!</p>
        </li>
      </div>
    </section>
  </body>
</html>
```

Код с ошибками в разметке

Правильный ответ

Если вы проделали это упражнение, то должны были найти две ошибки: неправильно закрытый тег и неправильно вложенный тег. Они подсвечены.

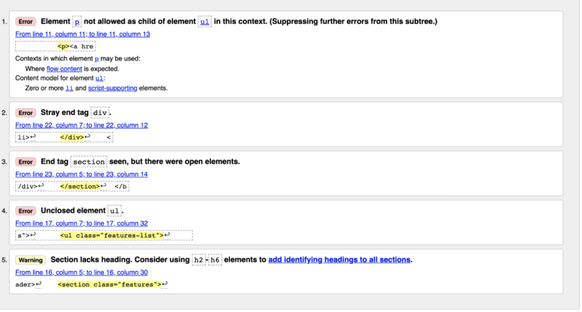
```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="UTF-8">
    <title>Барбершоп «Бородинский»</title>
  </head>
  <body>
    <header class="main-header">
      <nav class="main-navigation">
        <ul class="site-navigation">
          <p><a href="info.html">Информация</a></p>
          <li><a>Новости</a></li>
        </ul>
      </nav>
    </header>
    <section class="features">
      <ul class="features-list">
        <li class="feature-item">
          <strong>Быстро</strong>
          <p>Мы делаем свою работу быстро!</p>
        </li>
      </div>
    </section>
  </body>
</html>
```

Ошибки в разметке

И что, действительно верстальщики каждый раз проверяют свой код именно так, вручную? Конечно же, нет. Строгий синтаксис тем и хорош, что его достаточно легко проверять в автоматическом режиме. Поэтому есть специальный сервис для проверки, или валидации, вашего HTML-кода. Называется он — валидатор W3C.

Валидатор берёт ваш код и проверяет его на формальные, строгие правила: правильно ли закрыты теги, правильно ли вложены теги и так далее. Для этого валидатор использует всё те же два поля, с которыми мы работали: Categories и Content model — и универсальное правило про правильный порядок вложенности.

Валидатор проверяет всю разметку по спецификации HTML. Чтобы правильно валидировать документ, он использует `<DOCTYPE>`. Если вы написали `<DOCTYPE HTML>` — вы работаете по современной версии HTML 5, в которой в ссылки можно вкладывать заголовок, потому что у ссылки прозрачная модель контента. Но если у вас `<DOCTYPE>` старый, то в этом случае валидатор выдаст ошибку.



Пример работы валидатора

Почему валидатор нашёл сразу пять строгих ошибок? Зачастую одна ошибка, например, незакрытый тег, вызывает целый каскад последующих. Чтобы правильно исправлять ошибки при валидации, необходимо делать это сверху вниз. Как только вы замените `<p>` на `<li>` или правильно закроете `<div>`, часть последующих ошибок может исчезнуть.

Отличие ошибки и предупреждения

Ошибки помечаются как **error**, и их нужно обязательно исправлять. Это несоответствие формальному синтаксису языка.

Предупреждения помечаются как **warning**. И это те вещи, которые валидатор проверяет, смотря не на метаданные тега, а на описание, то есть, на его предназначение. Эти предупреждения менее строгие, исправлять их необязательно.

Например, у тега `<section>` есть рекомендации от разработчиков, о том, что неплохо в каждом теге `<section>` иметь заголовок. Если валидатор видит этот тег и не видит в нём заголовка, он может показать вам предупреждение. Необходимо ли это править? Не всегда. Error — обязательный к исправлению, warning — обязательный к размышлению.

Зачем вам, как профессионалам, нужно знать, как устроена спецификация? Дело в том, что иногда встречаются нестандартные вопросы, на которые вы не сможете ответить, используя справочники. Например, можно ли вложить внутрь `<footer>` тег `<section>`? Зная систему, по которой устроен стандарт, вы легко найдёте ответ.

Ознакомились со статьёй?

Сохранить прогресс

Работа со спецификацией
Часто используемые теги и типовые ошибки