

Правила оформления HTML-кода

1. Синтаксис

- 1.1. Два пробела в качестве отступа
- 1.2. Теги и атрибуты записываются в нижнем регистре
- 1.3. Отдельные логические блоки отбиваются пустой строкой
- 1.4. Используйте только двойные кавычки
- 1.5. Не ставим пробелов перед и после символа =
- 1.6. Между атрибутами один пробел
- 1.7. Одиночные теги без закрывающего слэша

2. Валидность

3. !DOCTYPE

4. Кодировка

5. Подключение стилей

6. Подключение скриптов

7. Атрибуты и их порядок

8. Логические атрибуты

9. Подписи input

10. Атрибут языка

1. Синтаксис

1.1 Два пробела в качестве отступа

Отступы позволяют визуально оценить структуру документа и быстро переключаться между его фрагментами. Размер отступа настраивается в редакторе. Также во многих редакторах можно включить отображение пробельных символов и преобразовать отступы.

Хорошо

Markup

```
1  <!DOCTYPE html>
2  <html lang="ru">
3    <head>
4      <meta charset="UTF-8">
5      <title>Ваша страница</title>
6    </head>
7    <body>
8      <h1>Страница</h1>
9      <p>Текст</p>
10   </body>
11 </html>
```

Плохо

Markup

```
1  <!DOCTYPE html>
2  <html lang='ru'>
3    <head>
4      <meta charset='UTF-8'>
5      <title>Ваша страница</title>
6    </head>
7    <body>
8    <h1>Страница</h1>
9    <p>Текст</p>
10   </body>
11 </html>
```

1.2 Теги и атрибуты записываются в нижнем регистре

Символы нижнего регистра не привлекают к себе большого внимания, и вам легче будет найти нужный.

Хорошо

```
1 <div class="user">
2   
3   <a href="/profile/">Перейти в профиль</a>
4 </div>
```

Плохо

```
1 <DIV Class="user">
2   <Img Src="./i/userpic.png" titlE="Иванов Иван">
3   <A href="/profile/">Перейти в профиль</a>
4 </DIV>
```

1.3 Отдельные логические блоки отбиваются пустой строкой

Это облегчает работу с кодом и визуально создает структуру документа.

Хорошо

```
1  <h1>Заголовок первого уровня</h1>
2  <p>Первый параграф</p>
3
4  <h2>Заголовок второго уровня</h2>
5  <ul>
6    <li>Первый пункт</li>
7    <li>Второй пункт</li>
8    <li>Третий пункт</li>
9  </ul>
10
11 <h3>Заголовок третьего уровня</h3>
12 <figure>
13   <figcaption>Блок с картинкой</figcaption>
14   
15 </figure>
```

Плохо

```
1  <h1>Заголовок первого уровня</h1>
2  <p>Первый параграф</p>
3  <h2>Заголовок второго уровня</h2>
4  <ul>
5    <li>Первый пункт</li>
6    <li>Второй пункт</li>
7    <li>Третий пункт</li>
8  </ul>
9  <h3>Заголовок третьего уровня</h3>
10 <figure>
11   <figcaption>Блок с картинкой</figcaption>
12   
13 </figure>
```

1.4 Используйте только двойные кавычки

При написании значений атрибутов используйте только двойные кавычки.

Хорошо

```
1 | <div class="first">
2 |   
3 | </div>
```

Плохо

```
1 | <div class='first'>
2 |   <img src='first-img.jpg' alt='Первая картинка'>
3 | </div>
```

1.5 Не ставим пробелов перед и после символа =

Поскольку правая часть непосредственно относится к левой, то атрибут и его значение должны быть написаны без пробелов.

Хорошо

```
1 | <p id="first" class="post">Это хороший пример</p>
```

Markup

Плохо

```
1 | <p id ="first" class= "post">Это плохой пример</p>
```

Markup

1.6 Между атрибутами один пробел

Не используем переносов строк между атрибутами, только один пробел. Перенос строк принят в css-документах, но не в html-разметке. Пишите все атрибуты элемента в строку.

Хорошо

```
1 <figure class="postItem">
2   
3   <figcaption>Логотип компании</figcaption>
4 </figure>
```

Плохо

```
1 <figure class="postItem">
2   
5   <figcaption>Логотип компании</figcaption>
6 </figure>
```

1.7 Одиночные теги без закрывающего слэша

Не используйте закрывающий слэш у одиночных тегов (``, `
` и другие). Этот элемент является пережитком прошлого, когда HTML был более строгим, а браузеры не умели распознавать и исправлять ошибки.

Хорошо

```
1 
2 <br>
```

Плохо

```
1 
2 <br />
```

2. Валидность

Html-документ должен проходить проверку на валидность. Для проверки используйте [валидатор](#).

3. !DOCTYPE

Первой строчкой в HTML-документе должен идти актуальный `doctype`. Это необходимо чтобы браузер верно отображал страницу. Это обеспечит единообразное отображение во всех современных браузерах.

```
1 <!DOCTYPE html>
2 <html lang="ru">
3   <head>...</head>
4   <body>...</body>
5 </html>
```

4. Кодировка

Кодировка символов в html-документе всегда должна быть указана явно. Это обеспечит верное отображение текста. Кодировка utf-8 подходит всегда, за редким исключением.

```
1 <head>
2   <meta charset="utf-8">
3   <title>Заголовок</title>
4 </head>
```

5. Подключение стилей

Файлы со стилями подключаются внутри тега `<head>` при помощи `<link>`. Атрибут `type` для тега `<link>` указывать не нужно, так как значение `text/css` устанавливается по умолчанию.

Хорошо: стилевой файл подключён в секции head

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 |   <head>
4 |     <link rel="stylesheet" href="style.css">
5 |   </head>
6 |   <body>...</body>
7 | </html>
```

Плохо: стилевой файл подключён в секции body

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 |   <head>...</head>
4 |   <body>
5 |     <link rel="stylesheet" href="style.css">
6 |   </body>
7 | </html>
```

6. Подключение скриптов

Скрипты при загрузке блокируют отображение содержимого страницы. По этой причине следует подключать их в самом конце html-документа.

При подключении скриптов в теге `<script>` атрибут `type` указывать не нужно, так как значение `text/javascript` устанавливается по умолчанию.

Хорошо: скрипт подключается перед `</body>`

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 |   <head>...</head>
4 |   <body>
5 |     <!-- Содержимое страницы -->
6 |     <script src="app.js"></script>
7 |   </body>
8 | </html>
```

Плохо: скрипт подключается в секции `<head>`

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 |   <head>
4 |     <script src="app.js"></script>
5 |   </head>
6 |   <body>...</body>
7 | </html>
```


7. Атрибуты и их порядок

У HTML-элементов атрибут `class` пишется первым. Единообразное написание помогает легче считывать код и по классам быстрее разбираться в назначении блоков.

Остальные атрибуты могут быть расставлены в любом порядке, но тоже единообразно для схожих элементов.

```
<a class="element element-big" id="element" href="/" data-name="element">Ссылка</a>

<input class="form-control" type="text" name="test">


```

8. Логические атрибуты

Для логических атрибутов (например, `checked`, `disabled`, `required`) значение не указывается, а сами атрибуты указываются последними и в единообразной последовательности во всём документе.

```
1 <!-- checked="checked" необязательно -->
2 <input type="checkbox" required checked>
3
4 <input type="text" disabled>
5
6 <select>
7   <option value="1" selected>1</option>
8 </select>
```

9. Подписи `input`

Для улучшения опыта пользователя, при нажатии на подпись поля, само поле должно активироваться. Для этого поле формы связывается с описанием при помощи идентификатора и атрибута `for` тега `<label>`.

Хорошо: элемент формы `radio` связан с подписью через `id` и `for`

```
1 | <input type="radio" id="choose">
2 | <label for="choose">Радио кнопка</label>
```

Хорошо: элемент формы `radio` и подпись обернуты в `label`

```
1 | <label>
2 |   <input type="radio"> Радио кнопка
3 | </label>
```

Плохо: подпись не связана с элементом формы

```
1 | <input type="radio" id="choose"> Радио кнопка
```

10. Атрибут языка

Для элемента `<html>` в атрибуте `lang` должен указываться верный язык документа. Это помогает системам перевода определить, какие использовать языковые правила.

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 |   <head>...</head>
4 |   <body>...</body>
5 | </html>
```