

HTML таблицы+img

Если правильно представить таблицу HTML, интерпретировать её данные смогут даже люди, имеющие проблемы со зрением.

Содержание

- **Оформление таблиц**
- **Продвинутое техники редактирования таблиц**
- **Мультимедиа и встраивание - тег `img`**

Что такое таблица ?

Таблица - это структурированный набор данных, состоящий из строк и столбцов (**табличных данных**).

Таблицы позволяют быстро и легко посмотреть значения, показывающие некоторую взаимосвязь между различными типами данных, например -

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Public Swim 06:30 - 10:30	Public Swim 06:30 - 09:00	Public Swim 06:30 - 09:00	Public Swim 06:30 - 11:15	Public Swim 06:30 - 09:00	Lane Swim 08:00 - 09:00	Lane Swim 08:00 - 09:00
Aquacise 10:30 - 11:15	Aqua Jog 09:15 - 10:00	Education Swimming Lessons 09:00 - 12:00	Aquacise 11:15 - 12:00	Education Swimming Lessons 09:00 - 12:00	Oldham Active Kids Swimming Lessons 09:00 - 13:00	Public Swim 09:00 - 11:00
Lane Swim 11:30 - 13:00	Parent & Baby Class 09:30 - 10:15	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Parent and Baby 12:00 - 12:45	Aquacise 11:00 - 11:45
Education Swimming Lessons	Public Swim 10:00 - 11:45	Public Swim 13:00 - 14:00	Education Swimming Lessons	Oldham Active Kids Swimming	Public Swim 13:00 - 17:00	Public Swim 11:45 - 13:00

Оформление таблиц

Чтобы эффективно представлять таблицы в веб, необходимо придать им хорошую структуру в HTML и применить к ним таблицы стилей ([CSS](#)).

Когда не надо использовать таблицы HTML?

HTML-таблицы следует использовать для табличных данных — это то, для чего они предназначены.

Многие используют таблицы HTML для оформления веб-страниц, например, одна строка для заголовка, одна для содержимого, одна для сносок, и тому подобное.

Это происходило из-за плохой поддержки CSS в разных браузерах; в наше время такое встречается гораздо реже, но иногда всё же попадает.

Использование таблиц в целях оформления вместо [методов CSS](#) является плохой идеей.

Причины по которым следует избегать использование таблиц для оформления сайта

- **Таблицы, используемые для оформления, уменьшают доступность страниц для людей, имеющих проблемы со зрением:** [Скринридеры \(Screenreaders\)](#), используемые ими, интерпретируют HTML-теги и читают содержимое пользователю. Поскольку таблицы не являются средством для представления структуры таблицы, и разметка получается сложнее, чем при использовании методов CSS, скринридеры вводят пользователей в заблуждение.
- **Таблицы создают путаницу тегов:** Как уже упоминалось, оформление страниц с помощью таблиц даёт более сложную структуру разметки, чем специально предназначенные для этого методы. Соответственно, такой код труднее писать, поддерживать и отлаживать.
- **Таблицы не реагируют автоматически на тип устройства:** У надлежащих контейнеров (например, [<header>](#), [<section>](#), [<article>](#), или [<div>](#)) ширина по умолчанию равна 100% от их родительского элемента. У таблиц же размер по умолчанию подстраивается под их содержимое, так что чтобы они одинаково хорошо работали на разных типах устройств необходимо принимать дополнительные меры.

Пример таблицы

Содержимое любой таблицы заключается между двумя тегами :

[<table></table>](#).

Самым маленьким контейнером в таблице является ячейка, она создаётся элементом [<td>](#) ('td' - сокращение от 'table data'). Введите внутри тегов table следующее:

```
<td>Hi, I'm your first cell.</td>
```

Чтобы получить строку из четырёх ячеек, необходимо скопировать эти теги три раза. Обновите содержимое таблицы так, чтобы она выглядела следующим образом:

```
<td>Hi, I'm your first cell.</td>
```

```
<td>I'm your second cell.</td>
```

```
<td>I'm your third cell.</td>
```

```
<td>I'm your fourth cell.</td>
```

Ячейки не располагаются одна под другой, на самом деле они автоматически выравниваются по отношению к другим ячейкам той же строки.

Каждый элемент `<td>` создаёт отдельную ячейку, а все вместе они создают первую строку. Каждая добавленная ячейка удлиняет эту строку.

Чтобы эта строка перестала расти, а новые ячейки перешли на вторую

Добавление заголовков с помощью элементов <th>

Чтобы опознавать заголовки таблицы в качестве заголовков, визуально и семантически, можно использовать элемент `<th>` ('th' сокращение от 'table header').

Он работает в точности как `<td>`, за исключением того, что обозначает заголовок, а не обычную ячейку.

Замените в своём HTML все элементы `<td>`, содержащие заголовки, на элементы `<th>`.

Сохраните HTML и загрузите его в браузер, и вы увидите, что заголовки теперь выглядят как заголовки.

По умолчанию к заголовкам таблицы применяется определённый стиль — они выделены жирным шрифтом и выровнены по центру, даже если вы не задавали для них стиль специально.

Заголовки дают дополнительное преимущество — вместе с атрибутом `scope` они помогают улучшить связь каждого заголовка со всеми данными строки или столбца одновременно, что довольно полезно

Слияние нескольких строк или столбцов

Иногда нам нужно, чтобы ячейки распространялись на несколько строк или столбцов. Возьмём простой пример, в котором приведены имена животных.

Иногда бывает нужно вывести имена людей рядом с именами животных.

А иногда это не требуется, и тогда мы хотим, чтобы имя животного занимало всю ширину.

Исходная разметка выглядит так:

```
<table> <tr> <th>Animals</th> </tr>
<tr> <th>Hippopotamus</th> </tr>
<tr> <th>Horse</th> <td>Mare</td> </tr>
<tr> <td>Stallion</td> </tr>
<tr> <th>Crocodile</th> </tr>
<tr> <th>Chicken</th> <td>Cock</td> </tr>
<tr> <td>Rooster</td> </tr> </table>
```

Атрибуты colspan и rowspan

Нужно, чтобы "Animals",
"Hippopotamus" и "Crocodile" распространялись
на два столбца, а "Horse" и "Chicken" - на две
строки.

Табличные заголовки и ячейки имеют
атрибуты colspan и rowspan, которые позволяют
это сделать.

Оба принимают безразмерное числовое
значение, которое равно количеству строк или
столбцов, на которые должны
распространяться ячейки.

Например, colspan="2" распространяет ячейку
на два столбца.

Стилизация столбцов

HTML позволяет указать, какой стиль нужно применять к целому столбцу данных сразу — для этого применяют элементы [<col>](#) и [<colgroup>](#).

Их ввели, поскольку задавать стиль для каждой ячейки в отдельности или использовать сложный селектор вроде [:nth-child\(\) \(en-US\)](#) было бы слишком утомительно.

Возьмём простой пример:

```
<table>
<tr> <th>Data 1</th>
<th style="background-color: yellow">Data 2</th> </tr>
<tr> <td>Calcutta</td> <td style="background-color: yellow">Orange</td>
</tr>
<tr> <td>Robots</td> <td style="background-color: yellow">Jazz</td> </tr>
</table>
```

Элемент CSS :nth-child()

The **:nth-child()** [CSS pseudo-class](#) сопоставляет элементы на основе их положения в группе братьев и сестер.

```
/* Selects the second <li> element in a list
```

```
*/ li:nth-child(2) { color: lime; } /*
```

Selects every fourth element among any group of siblings

```
*/ :nth-child(4n) { color: lime; }
```

Syntax

:nth-child() принимает один аргумент, который описывает шаблон для сопоставления индексов элементов в списке братьев и сестер. Индексы элементов основаны на 1.

Keyword values

odd Представляют элементы, числовое положение которых в ряду братьев и сестер нечетно: 1, 3, 5 и т.д.

even Представляют элементы, числовое положение которых в ряду братьев и сестер четно : 2, 4, 6, etc.

Альтернатива стилю столбцов

Нам пришлось повторить информацию о стиле для всех трёх ячеек в столбце (в реальном проекте, возможно, придётся вводить class для всех трёх и вводить правило в таблице стилей).

Вместо этого, мы можем задать информацию один раз, в элементе `<col>`. Элемент `<col>` задаётся в контейнере `<colgroup>` сразу же за открывающим тегом `<table>`.

Эффект, который мы видели выше, можно задать так:

<colgroup> <col> span

```
<table>
  <colgroup> <col> <col style="background-color: yellow"> </colgroup>
  <tr> <th>Data 1</th> <th>Data 2</th> </tr> <tr> <td>Calcutta</td> <td>Orange</td>
    </tr> <tr> <td>Robots</td> <td>Jazz</td> </tr>
</table>
```

Мы определяем два "стилизующих столбца". Мы не применяем стиль к первому столбцу, но пустой элемент <col> ввести необходимо — иначе ко 2 столбцу не будет применён стиль.

Если бы мы хотели применить информацию о стиле к обоим столбцам, мы могли бы просто ввести один элемент <col> с атрибутом span, таким образом:

```
<colgroup>
  <col style="background-color: yellow" span="2">
</colgroup>
```

Подобно colspan и rowspan, span принимает безразмерное числовое значение, указывающее, к какому количеству столбцов нужно применить данный стиль.

Таблица для редактирования

```
<!DOCTYPE html><html> <head> <meta charset="utf-8"> <title>School timetable</title>
  <style> html { font-family: sans-serif; } table { border-collapse: collapse; border: 2px
  solid rgb(200,200,200); letter-spacing: 1px; font-size: 0.8rem; } td, th { border: 1px
  solid rgb(190,190,190); padding: 10px 20px; } td { text-align: center; } caption {
  padding: 10px; } </style> </head>
<body> <h1>School timetable</h1> <table> <tr> <td>&nbsp;</td> <th>Mon</th>
  <th>Tues</th> <th>Wed</th> <th>Thurs</th> <th>Fri</th> <th>Sat</th>
  <th>Sun</th> </tr> <tr> <th>1st period</th> <td>English</td> <td>&nbsp;</td>
  <td>&nbsp;</td> <td>German</td> <td>Dutch</td> <td>&nbsp;</td>
  <td>&nbsp;</td> </tr> <tr> <th>2nd period</th> <td>English</td> <td>English</td>
  <td>&nbsp;</td> <td>German</td> <td>Dutch</td> <td>&nbsp;</td>
  <td>&nbsp;</td> </tr> <tr> <th>3rd period</th> <td>&nbsp;</td>
  <td>German</td> <td>&nbsp;</td> <td>German</td> <td>Dutch</td>
  <td>&nbsp;</td> <td>&nbsp;</td> </tr> <tr> <th>4th period</th> <td>&nbsp;</td>
  <td>English</td> <td>&nbsp;</td> <td>English</td> <td>Dutch</td>
  <td>&nbsp;</td> <td>&nbsp;</td> </tr> </table>
</body></html>
```

Тренировка

- Добавьте элемент `<colgroup>` вверху таблицы, сразу же под тегом `<table>`, куда вы сможете вставлять элементы `<col>`.
- Первые два столбца надо оставить без стиля..
- Добавьте цвет фона для третьего столбца. Значением атрибута `style` будет `background-color:#97DB9A;`
- Задайте ширину для четвёртого столбца. Значением атрибута `style` будет `width: 42px;`
- Добавьте цвет фона для пятого столбца. Значением атрибута `style` будет `background-color: #97DB9A;`
- Добавьте другой цвет фона и границу для шестого столбца, чтобы показать, что это особый день и она ведёт новый класс. Значениями атрибута `style` будут: `background-color:#DCC48E; border:4px solid #C1437A;`
- Последние два дня выходные; значением атрибута `style` будет `width: 42px;`

CSS: border-collapse

Свойство CSS "Свернуть границы" задает, имеют ли ячейки внутри <таблицы> общие или отдельные границы.

Когда ячейки свернуты, значение стиля границы вставки ведет себя как канавка, а начало ведет себя как гребень.

Когда ячейки разделены, расстояние между ячейками определяется свойством border-spacing.

Syntax

- `/* Keyword values */`
- `border-collapse: collapse;`
- `border-collapse: separate;`
- `/* Global values */`
- `border-collapse: inherit;`
- `border-collapse: initial;`
- `border-collapse: revert;`
- `border-collapse: unset;`

Продвинутые техники редактирования таблиц

Далее рассматриваем ещё несколько продвинутых возможностей в HTML таблицах — такие как заголовки/описание и группировка строк внутри head, body и footer секциях таблицы, а также доступность таблиц для пользователей с ограниченными возможностями.

Добавление заголовка к таблице с помощью <caption>

Вы можете добавить заголовок для таблицы установив его в элементе [<caption>](#) и этот элемент необходимо поместить внутрь элемента [<table>](#) (en-US). Причём вам нужно поместить его сразу после открытия тега <table>.

```
<table> <caption>Dinosaurs in the Jurassic period</caption> ... </table>
```

Заголовок отражает в себе описание контента таблицы.

Это полезно для всех читателей просматривающих страницу и желающих получить краткое представление от том полезна ли для них таблица, что особенно важно для слепых пользователей.

Вместо того чтобы читать содержимое множества ячеек чтобы понять о чем таблица, он или она могут полагаться на заголовок и принимать решение читать ли таблицу более подробно.

Заголовок помещают сразу после тега <table>.

Примечание: Атрибут [summary](#) (en-US) также может быть использован в <table> элементе предоставляя описание — это также читается скринридерами. Однако мы рекомендуем вместо этого использовать <caption> элемент, так как [summary deprecated](#) в HTML5 спецификации и не может быть прочитан зрячими пользователями (он не отображается на странице).

Добавление структуры с помощью `<thead>`, `<tfoot>` и `<tbody>`

Когда таблицы становятся более сложными по структуре полезно дать им более структурированное определение.

Отличный способ сделать это используя [<thead> \(en-US\)](#), [<tfoot>](#) и [<tbody> \(en-US\)](#), которые позволяют вам разметить header, footer и body секции таблицы.

Эти элементы не создают дополнительной доступности для пользователей со скринридерами и не приводят к какому-то визуальному улучшению при их использовании.

Зато они очень полезны при стилизации и разметке, как точки для добавления CSS к вашей таблице.

Далее несколько интересных примеров, в случае длинной таблицы вы можете сделать header и footer таблицы повторяемый на каждой печатной странице, или вы можете сделать body таблицы отображаемое на одной странице и иметь доступ ко всему содержимому контенту прокручивая вверх и вниз.

Использование `<thead>`, `<tfoot>` :

Элементом `<thead>` нужно обернуть часть таблицы которая относится к заголовку — обычно это первая строка содержащая заголовки колонок, но это не обязательно.

Если вы используете [<col>/<colgroup> \(en-US\)](#) элемент, тогда заголовок должен находиться ниже его.

Элементом `<tfoot>` нужно обернуть ту часть, которая относится к footer таблицы — например, это может быть последняя строка в которой отражаются суммы по столбцам таблицы.

Вы можете включить сюда footer таблицы, как и следовало ожидать, или чуть ниже заголовка таблицы (браузер всё равно отобразит его внизу таблицы).

<tbody>

Элементом <tbody> необходимо обернуть остальную часть содержимого таблицы которая не находится в header или footer таблицы. Этот блок располагают ниже заголовка таблицы или иногда footer таблицы, зависит от того какую структуру вы решите использовать

Примечание: <tbody> всегда включён в каждой таблице, неявно если не укажете его в коде.

Проверьте это, открыв один из примеров в котором не используется <tbody> и посмотрите HTML код в [browser developer tools](#) — вы увидите, что браузер добавил этот тег самостоятельно.

Это даёт больше контроля над структурой таблицы и стилем.

Пример

```
<!DOCTYPE html>
<html>
  <head> <meta charset="utf-8"> <title>My spending record</title> <link
    href="minimal-table.css" rel="stylesheet" type="text/css"> <style> </style>
  </head>
  <body> <h1>My spending record</h1> <table> <caption>How I chose to
    spend my money</caption> <tr> <th>Purchase</th> <th>Location</th>
    <th>Date</th> <th>Evaluation</th> <th>Cost (€)</th> </tr> <tr>
    <td>SUM</td> <td>118</td> </tr> <tr> <td>Haircut</td>
    <td>Hairdresser</td> <td>12/09</td> <td>Great idea</td> <td>30</td>
    </tr> <tr> <td>Lasagna</td> <td>Restaurant</td> <td>12/09</td>
    <td>Regrets</td> <td>18</td> </tr> <tr> <td>Shoes</td>
    <td>Shoeshop</td> <td>13/09</td> <td>Big regrets</td> <td>65</td>
    </tr> <tr> <td>Toothpaste</td> <td>Supermarket</td> <td>13/09</td>
    <td>Good</td> <td>5</td> </tr> </table>
  </body>
</html>
```

Упражнение: Добавление структуры таблицы

Попробуйте открыть это в браузере — вы увидите, что все выглядит классно, но могло бы быть лучше. Строка "SUM" которая содержит потраченные суммы кажется находится не в том месте и некоторые детали отсутствуют в коде.

Поместите очевидную строку заголовка внутрь `<thead>` элемента, строку "SUM" внутрь `<tfoot>` элемента и оставшийся контент внутрь `<tbody>` элемента.

Сохраните, перезагрузите и вы увидите, что добавление элемента `<tfoot>` привело к тому, что строка "SUM" опустилась к нижней части таблицы.

Далее, добавьте атрибут [colspan](#), чтобы ячейка "SUM" занимала первые четыре столбца, таким образом числовое значение "Cost" появится в последнем столбце.

Давайте добавим несколько простых дополнительных стилей к таблице, чтобы дать вам представление насколько эти элементы полезны при использовании CSS. Внутри в `<head>` вашего HTML документа вы увидите пустой элемент [<style>](#).

Внутри этого элемента добавьте следующие строки CSS кода:

```
tbody { font-size: 90%; font-style: italic; }  
tfoot { font-weight: bold; }
```


Вложенные таблицы

В одну таблицу вкладывать другую таблицу возможно, если вы используете полную структуру включая элемент `<table>`.

Это как правило не рекомендуется, так как делает разметку более запутанной и менее доступной для пользователей скринридеров, так в большинстве случаев вы можете просто вставить дополнительные ячейки/строки/столбцы в существующую таблицу.

Однако, иногда это необходимо, например, если вы хотите легко импортировать контент из других источников.

Разметка простой вложенной таблицы:

См. пример

Скринридеры будут определять все заголовки и использовать их создавая программные ассоциации между этими заголовками и ячейками к которым они относятся.

Сочетание заголовков столбцов и строк будет определять и интерпретировать данные в каждой ячейке так, что пользователи скринридеров могут интерпретировать таблицу также как это делают зрячие пользователи.

Атрибут scope

Атрибут [scope \(en-US\)](#), который может быть добавлен к элементу <th> он сообщает скринридеру какие ячейки точно являются заголовками — например, заголовков строки в которой он находится или столбца.

Можно однозначно определить заголовки столбцов :

```
<thead>
<tr>
<th scope="col">Purchase</th>
<th scope="col">Location</th>
<th scope="col">Date</th>
<th scope="col">Evaluation</th>
<th scope="col">Cost (€)</th>
</tr>
</thead>
```

И у каждой строки может быть определён заголовок, (если мы добавили заголовки строк и заголовки столбцов):

```
<tr>
<th scope="row">Haircut</th> <td>Hairdresser</td> <td>12/09</td> <td>Great idea</td>
<td>30</td>
</tr>
```

Использование заголовков столбцов и строк

Атрибут `scope` имеет ещё два возможных значения — `colgroup` и `rowgroup`. Они используются для заголовков, которые располагаются вверху ваших столбцов или строк.

Альтернатива атрибута `scope` это использование атрибутов [id](#) и [headers](#) задавая ассоциации между заголовками и ячейками. Этот способ выглядит следующим образом:

Вы устанавливаете уникальный `id` для каждого `<th>` элемента.

Вы устанавливаете атрибут `headers` для каждого `<td>` элемента. Каждый `headers` атрибут должен содержать список всех `id`, разделённый пробелами, ко всем `<th>` элементам которые действуют как заголовки для этой ячейки.

Это обеспечивает явное определение позиции для каждой ячейки вашей HTML таблицы, определяет заголовки столбцов и строк таблицы. Для того чтобы это работало

Атрибуты id и headers

Вернёмся к примеру с расчётом затрат, его можно переписать следующим образом:

```
<thead>
  <tr> <th id="purchase">Purchase</th>
  <th id="location">Location</th>
  <th id="date">Date</th>
  <th id="evaluation">Evaluation</th>
  <th id="cost">Cost (€)</th>
</tr>
</thead>
<tbody> <tr>
  <th id="haircut">Haircut</th>
  <td headers="location haircut">Hairdresser</td>
  <td headers="date haircut">12/09</td>
  <td headers="evaluation haircut">Great idea</td>
  <td headers="cost haircut">30</td>
</tr> ... </tbody>
```

Упражнение: играем со score и headers

Для заключительного упражнения мы, вначале создадим копию [items-sold.html](#) и [minimal-table.css](#) в новой папке.

Теперь попробуем добавить соответствующий атрибут `score`, который наиболее соответствует этой таблице.

И наконец попробуем сделать другую копию изначальных файлов, на этот раз делая таблицу более доступной используя атрибуты `id` и `headers`.

Мультимедиа и встраивание тег `img`

Вы можете встроить изображение используя абсолютный URL, например:

```
<img src=https://www.example.com/images/dinosaur.jpg>
```

Но это бессмысленно, так как он просто заставляет браузер делать больше работы, запрашивая каждый раз IP-адрес от DNS-сервера. Вы почти всегда будете держать свои изображения для сайта на том же сервере, что и ваш HTML.

Предупреждение: Большинство изображений защищены. Не отображайте изображения на вашем сайте пока:

- вы не будете владеть изображением
- у вас не будет письменного разрешения владельца изображения, или
- пока у вас не будет достаточно доказательств что изображение находится в открытом доступе.
- Нарушение авторских прав является незаконным. Кроме того, никогда не указывайте в своём атрибуте `src` ссылку на изображение, размещённое на чужом сайте. Это называется "хотлинкинг" (с англ. 'hotlinking' - 'горячая ссылка'). Запомните, кража пропускной способности чужого сайта незаконна. Это также замедляет вашу страницу и не позволяет вам контролировать, будет ли изображение удалено или заменено чем-то неприятным.

Альтернативный текст

Следующий атрибут — alt.

Его значением должно быть текстовое описание изображения для использования в ситуациях, когда изображение не может быть просмотрено / отображено или отрисовка занимает много времени из-за медленного интернет-соединения.

```

```

Итак, в каких случаях текст из атрибута alt может быть нам полезен?

- Пользователь с нарушением зрения использует [устройство чтения с экрана](#), которое может читать вслух описание элементов веб-страницы. На самом деле, наличие текста в атрибуте alt для описания изображения может быть полезно для большинства пользователей.
- В случае, если была допущена ошибка в имени файла или пути к нему (как было описано выше).
- Браузер не поддерживает формат данного изображения. Некоторые люди до сих пор используют текстовые браузеры, такие как [Lynx](#), которые вместо изображений отображают текст из атрибута alt.
- Если вы хотите добавить возможность найти ваше изображение с помощью поисковых систем. Например, поисковые системы могут искать совпадения поисковых запросов с текстом атрибута alt.
- Если пользователи отключили отображение изображений на странице для уменьшения объёма передаваемых данных и для сокращения элементов, отвлекающих внимание.

Ширина и высота

Вы можете использовать атрибуты width и height, чтобы указать ширину и высоту вашего изображения.

```

```

Но если изображение не будет показано, например, когда пользователь только что перешёл на страницу, а оно ещё не успело загрузиться, вы укажете браузеру оставить место для отрисовки изображения:

Это хорошая практика, в результате страница загрузится быстрее и более гладко.

Однако, вы не должны изменять размеры ваших изображений используя HTML атрибуты.

Если вы установите размер изображения слишком большим, то в конечном итоге вы столкнётесь с изображениями, которые выглядят зернистыми, размытыми или слишком маленькими, и потратите трафик для загрузки изображения, которое не будет соответствовать нуждам пользователя

Заголовок изображения

Как и для [ССЫЛОК](#), вы также можете добавить атрибут title для изображений, чтобы при необходимости предоставить дополнительную информацию.

В нашем примере это так:

```

```

Это даёт нам всплывающую подсказку при наведении курсора мыши, также как и в ссылках.

Однако это не рекомендуется - title имеет ряд проблем с доступностью, в основном из-за того, что поддержка программ чтения с экрана очень непредсказуема, и большинство браузеров не будут отображать её, если вы не наведёте курсор мыши (например, нет доступа для пользователей клавиатуры).

Зачастую лучше включить такого рода вспомогательную информацию в основной текст статьи, чем прикреплять её к изображению.

Однако, она полезна в некоторых обстоятельствах; например, в галереях изображений, когда у вас нет места для их заголовков.

Придание изображению структуры и установка заголовка

Начиная разговор о заголовках, есть множество путей как вы можете добавить заголовок к своему изображению. Для примера, нет ничего, что может вас остановить сделать это таким образом:

```
<div class="figure">
```

```

```

```
<p>A T-Rex on display in the Manchester University Museum.</p> </div>
```

Это нормально. Это содержит всё что вам нужно, и красиво стилизуется с помощью CSS.

Но, есть проблема: здесь нет ничего, что семантически связывает изображение с его заголовком, и это может вызвать сложности для читателей.

Например, когда у вас есть 50 изображений и заголовков, какой заголовок идёт вместе с каким изображением?

Придание изображению структуры

Лучшим решением будет использование элементов HTML5 [<figure>](#) и [<figcaption>](#).

Они были созданы исключительно для этой цели: предоставить семантический контейнер для рисунков и чётко связать рисунок с заголовком.

Наш пример выше мог бы быть переписан так:

```
<figure> 
```

```
<figcaption>A T-Rex on display in the Manchester University Museum.</figcaption>
```

```
</figure>
```

Элемент [<figcaption>](#) говорит браузерам и вспомогательной технологии, что заголовок описывает содержимое элемента [<figure>](#).

<figure>

Тег <figure> не является изображением. Он представляет собой независимый структурный элемент, который:

- Передаёт смысл компактным, интуитивно понятным способом.
- Может использоваться в различных местах страницы.
- Предоставляет ценную информацию, поддерживающую основной текст.

Тег <figure> может быть несколькими изображениями, куском кода, аудио, видео, уравнением, таблицей, либо чем-то другим.

Литература

- [HTML таблицы основы - Изучение веб-разработки | MDN \(mozilla.org\)](#)
- [learning-area/timetable.html at master · mdn/learning-area · GitHub](#)
- [Изображения в HTML - Изучение веб-разработки | MDN \(mozilla.org\)](#)