**Конспект по теме "Списки и таблицы"**

Списки

**Список в типографике** — способ оформления различного рода перечислений или перечней. Каждый элемент списка начинается с маркера списка или номера-буквы и весь текст списка не должен выступать влево за них.

Списки могут быть:

* упорядоченные или неупорядоченные. Если последовательность перечисленных элементов не зависит от какого-то определённого логического порядка, то такой список является неупорядоченным. Если же от перестановки перечисленных элементов изменяется смысл, то такой список называется упорядоченным;
* нумерованные и ненумерованные. Каждый элемент нумерованного списка начинается с порядкового номера или буквы в алфавитном порядке. Элемент ненумерованного списка начинается c маркера списка;
* одноуровневые (простые) или многоуровневые.

[Ссылка на Вики](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%28%D1%82%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0%29)

Виды списков

В вебе выделяют три основных вида списков:

1. Маркированные списки
2. Нумерованные списки
3. Списки описаний

Разница между ними в смысловой нагрузке. Правильно подбирайте список исходя из контента.

Если в списке важна последовательность действий, например кулинарный рецепт, то вам подойдет нумерованные списки.

Если нужно перечислить несколько пунктов без четкой их последовательности, то можно выбрать маркированный список.

В случае, когда требуется оформить несколько определений и дать им описание, то подойдет список описаний.

У списков есть стили по умолчанию. Визуально текст отбивается пустой строкой до начала и пустой строкой после окончания. Поэтому в тексте он выделяется.

Все пункты списка визуально немного смещены относительно левого края всего остального текста и перед каждым пунктом появляется либо маркер (по умолчанию черная точка), либо номер.

Это привлекает внимание читателя к тексту и повышает его читабельность.

Синтаксис

**Общее**

Во всех видах списков общим является то, что вложить внутрь родительского тега можно только теги пунктов списка. Все прочие теги должны находится внутри тега пункта списка. В том числе это относится к вложенным спискам.

**Маркированные списки**

ul – родительский тег. Все списки начинаются с него. Это парный тег.UL = *Unordered List* – неупорядоченный список.

li – тег пункта списка. Парный тег. LI = *List Item* – элемент списка.

Пример структуры простого маркированного списка. Представим ситуацию: нам на странице нужно перечислить вещи, которые понадобятся ребенку в походе.

Мы понимаем, что нам нужно использовать список – так будет нагляднее и понятнее. А так же родители ребенка смогут распечатать страницу и отмечать то, что положили в рюкзак.

Какой список выбрать? В этом случае порядок сбора вещей не важен. Главное – перечислить их все. Значит подойдет маркированный список.

Отлично. На странице в тексте, где должен быть список, мы добавляем родительский тег списка. Поскольку список маркированный, то это будет ul.

Каждая вещь должна идти отдельным пунктом списка. Это значит, что каждую вещь мы должны разместить между <li> и </li> – в тексте напротив поставится точка и после будет перенос строки.

Когда наш список закончен – закрываем его при помощи тега </ul>

В итоге разметка для нашего списка вещей в детский поход будет выглядеть так. (показать результат)

123456<ul>

<li>Компас</li>

<li>Палатка</li>

<li>Сменная одежда</li>

<li>Спальник</li>

</ul>

**Нумерованный список**

ol – родительский тег. Парный тег. OL = *Ordered List* – упорядоченный список. li – тег пункта списка.

Пример структуры простого списка.

Продолжим писать контент для сайта о детском походе. Теперь нам нужно разместить информацию о необходимых прививках и справках.

Очевидно, что тут нам тоже пригодится список. Прививки обязательно ставить в определенном порядке. Значит нам нужен нумерованный список.

Как и в предыдущем случае, каждую прививку и справку нужно разместить в отдельном пункте списка, чтобы напротив автоматически проставилась цифра.

В нужном месте текста обозначаем что здесь будет список – открываем тег <ol>.

Дальше размещаем все нужные нам пункты, оборачивая текст в теги <li>.

Значит наша разметка будет выглядеть следующим образом (показать результат):

123456<ol>

<li>Прививка от кори</li>

<li>Прививка от клещевого энцефалита</li>

<li>Прививка от оспы</li>

<li>Справка от педиатра</li>

</ol>

**Списки описаний**

В этом типе списков структура чуть сложнее, чем у предыдущих.

Здесь используются три тега: родительский, тег термина и тег определения.

dl – родительский парный тег. DL = *Description List* – список описаний.

dt – тег, в который заключается термин, парный. DT = *Definition Term* – термин.

dd – тег, в котором размещается определение для термина. DD = *Description* – описание.

Пример структуры простого списка. Теперь мы перешли к заполнению страницы с маршрутом похода.

Для игрового момента обычные пункты маршрута обозначены выдуманными словами.

Чтобы родителям было понятно, где же будут находится их дети – нам нужно дать расшифровку каждого выдуманного слова.

Для этого отлично подойдет список описаний. В нужном месте текста мы обозначаем, что здесь будет список.

Открываем тег <dl>.

Далее нам нужно написать выдуманное место на карте.

Для этого используем тег термина и помещаем внутрь наше слово. Вот так <dt>Крюкландия</dt>. После этого мы даем расшифровку этого термина и пишем внутри тега dd.

В итоге у нас получится такая структура (показать результат):

12345678910<dl>

<dt>Стартур</dt>

<dd>Центральная площадь. Начальная точка похода, место сбора детей.</dd>

<dt>Теплостания</dt>

<dd>Ж/Д вокзал. Место посадки на электричку.</dd>

<dt>Полянд</dt>

<dd>Деревня Жабреи. Стоянка перед лесом, обед.</dd>

<dt>Лесвуд</dt>

<dd>Остановка в лесу. Место проведения веселых стартов.</dd>

</dl>

**Изменяем нумерацию упорядоченного списка**

Вернемся к нашему списку прививок и справок. Дежурный врач сказал, что после пункта «Прививка от энцефалита» обязательно должен идти текст о том, что срок давности прививки – не больше 3 месяцев.

Мы понимаем, что нам нужно разорвать список, вставить текст, а потом снова продолжить список. (показать пример)

Если мы просто разорвем список, а потом откроем новый, то второй список снова начнется с единицы. Что же делать?

Нужно начать нумерованный список не с единицы. На этот случай вы можете использовать HTML-атрибут start.

123456789<ol>

<li>Прививка от кори</li>

<li>Прививка от клещевого энцефалита</li>

</ol>

<p>Срок давности прививки – 3 месяца.</p>

<ol start="3">

<li>Прививка от оспы</li>

<li>Справка от педиатра</li>

</ol>

В качестве значения атрибута указываем число, с которого нужно начать этот конкретный список.

**Многоуровневые списки**

Отдельно стоит остановится на списках, состоящих из нескольких уровней.

В процессе создания многоуровнего списка стоит помнить о правиле, что в списке могут быть только пункты списка.

Это значит, что второй и все следующие уровни списка нужно поместить внутрь пункта предыдущего списка.

Вы можете комбинировать нумерованные и маркированные списки при вложении.

Инструктор, который пойдет с детьми в поход, боится, что под “Сменной одеждой” каждый родитель может понять разные предметы одежды. И просит нас уточнить этот пункт, добавив к нему пункты с конкретной обязательной одеждой.

Мы понимаем, что нам нужно поместить список в список. Есть более важные предметы одежды, которые обязательно должны быть в рюкзаке ребенка, а есть необязательные, но желательные. Мы можем отразить это при помощи нумерованного списка.

Чтобы создать второй уровень списка, в нужном пункте первого списка мы открываем новый список. В нашем случае это нумерованный список – используем <ol>.

Дальше размещаем информацию об одежде для ребенка – каждую вещь в новый пункт.

Когда все необходимое перечислено – закрываем второй уровень списка – </ol>.

И теперь закрываем пункт первого списка – </li>.

Результат:

123456789101112<ul>

<li>Компас</li>

<li>Палатка</li>

<li>Сменная одежда

<ol>

<li>Простые и шерстяные носки</li>

<li>Теплые вещи</li>

<li>Непромокаемая куртка</li>

</ol>

</li>

<li>Спальник</li>

</ul>

Таблицы

В походе проводились соревнования и каждый ребенок набрал определенное количество баллов. Организаторы решили разместить на сайте итоговые результаты.

На странице должны быть следующие данные: имя ребенка, название команды, сумма заработанных им баллов.

Мы понимаем, что списка в этом случае будет недостаточно. Родителям будет неудобно искать имя своего ребенка в списке.

Лучше всего для этих целей подойдет таблица. Все данные о ребенке находятся в одной строке – удобно искать и смотреть. Каждая колонка может иметь заголовок – просто будет определить, что значит текст/цифра, расположенные в ячейке.

Синтаксис

Таблица всегда состоит из строк и ячеек в строке. В нужном месте страницы мы создаем таблицу при помощи парного родительского тега table. Затем внутрь таблицы помещается необходимое количество строк при помощи парного тега tr. TR расшифровывается как *table row* – строка таблицы. В каждой строке может лежать любое (нужное вам) количество ячеек. Ячейки бывают двух типов. th – для заголовков колонки. TH – *table header cell* – ячейка “шапки” таблицы. Текст в такой ячейке отображается жирным шрифтом и выравнивается по центру. td – для всех остальных ячеек. TD – *table data cells* – ячейка данных таблицы.

Сколько ячеек вы разместите в ряду, столько колонок у вас получится. Важно помещать одинаковое количество ячеек в каждую строку для правильного внешнего вида таблицы и состыковки данных по вертикали и горизонтали.

**Стандартная структура.**

Организаторы похода прислали вам данные о каждом ребенке. Нам нужно разместить их в таблице. Всего в поход ходили 8 детей. Это значит, что нам нужно создать 8 строк.

Открываем таблицу при помощи родительского тега table и размещаем внутри 8 строк – tr. Внутри строк ничего писать не нужно. Все данные будут лежать в ячейках, находящихся в строке. <https://codepen.io/solarrust/pen/NAkQZZ?editors=1000>

Далее в каждой строке нам нужно создать 3 ячейки – с именем ребенка, с названием команды и с количеством заработанных баллов. <http://codepen.io/solarrust/pen/ZONOJV?editors=1000>

И, наконец, заполним таблицу данными детей. <http://codepen.io/solarrust/pen/oLKvNZ?editors=1000>

**Заголовки колонок**

Отлично, мы внесли данные в таблицу. Родители заходят на сайт и не понимают, что за цифра в последней колонке. Мы поняли, что нам нужно подписать колонки и обозначить, что за данные в каждой из них находятся. Чтобы создать строку с заголовками колонок и отделить эту строку от основных данных таблицы используется парный тег thead – *table head*. thead следует использовать в самом начале таблицы. Что логично, поскольку заголовки колонок должны располагаться сверху. Допускается использование не более одного thead в пределах одной таблицы. Должен идти в разметке сразу после тега table. Внутрь вкладывается тег ряда – tr, а в него теги ячеек – td.

Давайте зададим заголовки нашим колонкам. <http://codepen.io/solarrust/pen/qNzZgR?editors=1000>

**Тело таблицы**

После thead должен идти основной контент таблицы. Его так же группируют по смыслу, для этого используется парный тег tbody – table body. Тел таблицы может быть много. Например, мы возьмем данные только одной команды и разобьем их по дням похода.

Пример с использованием tbody <http://codepen.io/solarrust/pen/RRXbNy?editors=1000>

Теперь нам нужно подвести итог и посчитать, сколько всего баллов набрала команда Грифиндор. В наверняка не раз видели таблицы с расчетами, где в нижней строке выводится сумма или какой-то вывод из приведенных выше цифр. Именно для таких строк ИТОГО используется парный тег tfoot – *table footer*. Синтаксис его использования ничем не отличается от предыдущих логических элементов thead и tbody. Давайте подсчитаем и выведем общее количество баллов команды

Пример с использованием tfoot <http://codepen.io/solarrust/pen/AXZApL?editors=1000>

**Атрибуты**

Мы видим, что в строках с днями недели у нас есть пустые ячейки. Та же ситуация в последней строке ИТОГО. Почему бы не объединить пустые ячейки? Так таблица станет более читаемой.

Что делать, если нам нужно объединить несколько ячеек по горизонтали?

Без паники! Все придумано до нас. Для этих целей используются специальные html-атрибуты colspan. span дословно переводится как охват. Это хорошо объясняет, что именно делают атрибуты.

Если мы укажем у тега td атрибут colspan и в значении напишем 3, то ячейка растянется, охватит собою место под 3 ячейки.

Так как одна ячейка может занимать несколько позиций в строке или в столбце, или и там и там одновременно, то важно следить за суммарным количеством ячеек в строках и столбцах с учетом охвата.

Это схоже с тем, как работает кнопка [button](https://github.com/netology-code/wm-materials/blob/master/lessons/i/button.png) в Excel.

Значением этих атрибутов может быть только целое положительное число.

Пример с использованием colspan. Давайте объединим пустые ячейки в строках с днями недели и в последней строке с итоговым количеством баллов. <http://codepen.io/solarrust/pen/RRXbNy?editors=1000>

Поскольку у всех детей в этой таблице одинаковое название команды, то мы можем объединить все три ячейки напротив имен детей в одну по высоте. Для этого используется атрибут rowspan.

<http://codepen.io/solarrust/pen/qNZWRP?editors=1000>

Как вы возможно заметили, в строках не хватает ячеек. Причина в том, что для охвата нужно будет освободить место. По факту атрибуты не склеивают ячейки по ширине или высоте. Они растягивают одну ячейку на указанное количество строк или колонок. Соответственно, для того, чтобы таблица не развалилась нужно освободить место под это растягивание.

figure

Было принято решение наградить детей, занявших первое и второе место. И создать на сайте почетную доску, где будет размещена фотография победителя и его имя. Мы можем отдельно разместить картинку и отдельно написать текст. Но такой код семантически не будет связан между собой. Но в html есть специальная пара тегов, которые пригодятся нам при создании доски почета. Это тег figure – родительский парный тег и располагающийся внутри него тег figcaption. figcaption так же парный тег. При этом он является не обязательным – фигура может сущестовать и без описания.

Представьте газеты. Под фотографиями в газете есть подпись. <http://screencast.com/t/AzoFqfLx9K30>

Фотография победителя вместе с именем – это фигура – используем тег figure. Имя ребенка – это пояснение для фотографии – используем figcaption. При этом figcaption располагается внутри тега figure. Прямо как ячейка в строке таблице, пункт списка – внутри тега списка.

Синтаксис

Давайте создадим фигуру с фотографией Гермионы. <http://codepen.io/solarrust/pen/grVYGq?editors=1000>

Теперь под фотографией нам нужно написать, кто на ней изображен. Для этого внутри figure вставляем тег figcaption.

<http://codepen.io/solarrust/pen/RRXbpY?editors=1000>

Ура! Мы отлично справились с контентом на сайте детского похода. А так же выучили три вида списков, таблицы и фигуры.