# Занятие #9 Списки. Принцип модульности в верстке

# Списки:

**HTML списки** представляют собой набор сгруппированных абзацев текста, помеченных значками **(маркированный список)** или цифрами **(нумерованный список)**. Также существуют **списки определений**, которые состоят из пар термин - определение.

## 1. Маркированный список

Маркированный список представляет собой неупорядоченный список *(ненумерованный список)*. Создается с помощью тега <ul>.

Каждый элемент списка добавляется с помощью тега <li>

В качестве маркера элемента списка выступает метка, например, закрашенный кружок.

<ul>

<li> элемент маркированного списка </li>

<li> элемент маркированного списка </li>

<li> элемент маркированного списка </li>

</ul>

Тегам **<li>** и **<ul>** можно задавать атрибут **type** который устанавливает элементу списка или же всему списку целиком определённый вид маркера, но этот атрибут уже устарел и правильнее задавать стиль маркеров в CSS при помощи свойства list-style-type.

Для маркированных списков list-style-type может иметь одно из трёх значений:

**disk** - диск (закрашенный круг по умолчанию)

**circle** - полый круг

**square** – квадрат

В качестве маркера может использоваться картинка (например, картинка с названием ic-marker.gif) при помощи свойства list-style-image, в файле css это свойство записывается примерно так: **ul.list {list-style-image: url(images/ic-marker.gif); }**

### Недостатком использования правила list-style-type при верстке маркированных списков, является то, что бывает достаточно сложно спозиционировать маркер именно так как нарисовано в дизайне, кроме того позиционирование маркеров в разных браузерах может заметно отличаться. Именно поэтому наиболее часто маркеры в маркированных списках задаются следующим образом:

1) тегу <ul> задают list-style:none; - т.е. маркеры по умолчанию убирают;

2) тегам <li> задают отступ слева и

**2а)** бэкграундом к тегу <li> размещают картинку маркера;

**2б)** тегу <li> задают position:relative, создают для него псевдоэлемент ::before, задают этому элементу размеры (высоту и ширину) равные размерам картинки маркера, бэкграундом к этому псевдоэлементу задают картинку маркера и позиционируют псевдоэлементы в нужном месте при помощи position: absolute

## 2. Нумерованный список

Нумерованный список создается при помощи тегов <ol>, каждый пункт списка помещается в теги <li>

<ol>

<li>элемент нумерованного списка</li>

<li>элемент нумерованного списка</li>

</ol>

Для нумерованных списков можно задавать следующие значения

list-style-type:

**armenian** - традиционная армянская нумерация.

**decimal** - арабские числа (1, 2, 3, 4,...).

**decimal-leading-zero** - арабские числа с нулем впереди для цифр меньше десяти (01, 02, 03,...).

**georgian** - традиционная грузинская нумерация.

**lower-alpha** - строчные латинские буквы (a, b, c, d,...).

**lower-greek** - строчные греческие буквы (α, β, γ, δ,...).

**lower-latin** - это значение аналогично lower-alpha.

**lower-roman** - римские числа в нижнем регистре (i, ii, iii, iv, v,...).

**upper-alpha** - заглавные латинские буквы (A, B, C, D,...).

**upper-latin** - это значение аналогично upper-alpha.

**upper-roman** - римские числа в верхнем регистре (I, II, III, IV, V,...).

Изменение css-стилей номеров строк (цифр) в упорядоченных списках <ol> является более сложной задачей, по сравнению с изменением маркеров в не нумерованных списках. Но все же это можно сделать используя псевдоэлемент ::before и css-свойства content, сounter-increment, counter-reset.

Пример css-кода для стилизации нумерованного списка:

ol{  
 counter-reset: myCounter; /\*обнуляем счетчик внутри <ol>\*/

}  
li{  
 list-style: none; /\*убираем стиль счетчика по умолчанию\*/

}  
li:before { /\*создаем псевдоэлемент для элемента <li>\*/

counter-increment: myCounter; /\*инкрементируем счетчик для псевдоэлемента\*/

content:counter(myCounter); /\*выводим в content значение счетчика \*/

color: white;  
 background: #2980B9;  
 display: inline-block;  
 text-align: center;  
 margin: 5px 10px;  
 line-height: 40px;  
 width: 40px;  
 height: 40px;  
}

## 3. Список определений

Списки определений задаются при помощи тегов <dl>, <dt>, <dd>

Каждый такой список начинается с контейнера <dl>, куда входит тег <dt> создающий термин и тег <dd> задающий определение этого термина.

<dl>

<dt>Термин 1</dt>

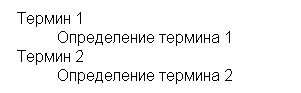
<dd>Определение термина 1</dd>

<dt>Термин 2</dt>

<dd>Определение термина 2</dd>

</dl>

В браузере такой список (без стилей) будет выглядеть так:

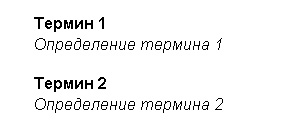


Если добавить стили, например такие:

dl dt{font-weight: bold;}

dl dd{margin:0; padding:0 0 20px; font-style:italic;}

В браузере этот список примет вид:

****

## 4. Принцип модульности в верстке

Сайты и экранные интерфейсы состоят из разнокалиберных элементов: навигационных шапок, меню, кнопок, иконок, панелей инструментов, лент новостей, тизеров, форм ввода, виджетов и плашек. Тексты выводятся из базы данных и имеют непредсказуемую длину.

Принцип модульности в верстке помогает структурно упорядочить компоновку всего материала на веб-странице. Основа принципа модульности в верстке -

макет должен быть «нарезан» на аккуратные непересекающиеся прямоугольники-модули.

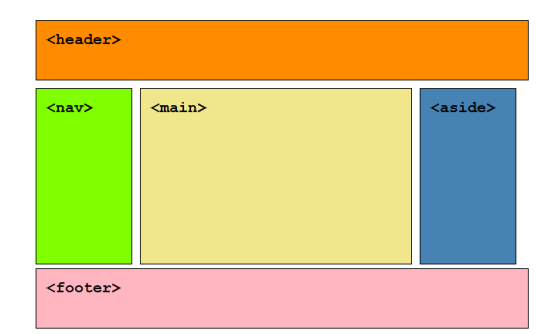
Модули не должны пересекаться и налезать друг на друга. Они должны иметь прямоугольную форму, согласовываться друг с другом по ширине и высоте и составлять вместе общий прямоугольник страницы или экрана.

Таким образом, с точки зрения модульной верстки, веб-страница фактически рассматривается как набор прямоугольных блоков, которые выкладываются в определенном порядке.

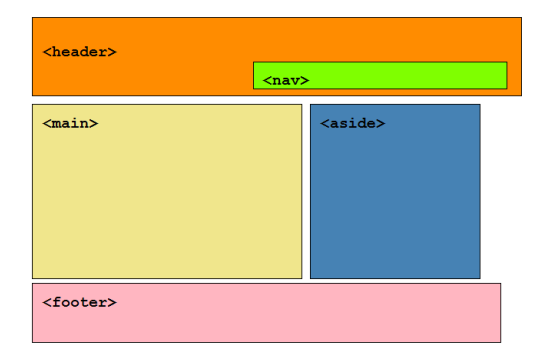
Типичный веб-сайт состоит из шапки, главного меню, блока с основным содержанием, боковых колонок, подвала. Расположение этих главных прямоугольников- модулей формирует структуру страницы, поэтому назовем их **структурными блоками**. Эти блоки могут быть расположены друг под другом, в несколько колонок или ещё сложнее.

Расположение структурных блоков на странице представляет собой своего рода каркас. Этот каркас должен надежно держать содержимое страницы и ни при каких обстоятельствах не должен ломаться ни в одном из действующих браузеров.

Ниже показан каркас страницы с шапкой вверху, навигацией слева, основным контентом в центре, дополнительной информацией справа и подвалом внизу. Так как контентная часть содержит 3 колонки, такой макет часто называют 3-х колоночным



Пример 2-х колоночный макет, в котором блок навигации вложен в шапку



*Каркас страницы должен быть гибким, иными словами должна быть предусмотрена возможность при необходимости легко перестраивать структурные блоки а также менять их ширину, что особенно важно в случае адаптивной верстки.*

Внутрь структурных блоков станицы вкладываются более мелкие независимые блоки - назовем их просто **модули**.

Модули должны быть независимыми, т.е. расположение элементов в одном модуле никаким образом не должно влиять на расположение элементов в других модулях.

Модули должны быть сверстаны так, чтобы их легко было переместить из одного структурного блока в другой или поменять местами в пределах одного структурного блока.