

# Форматирование текста при помощи HTML. Списки

Модуль 2 (2 пары)

## Блочные и строчные теги

- **Блочными** (*block*) называются такие **элементы HTML**, которые по умолчанию занимают все доступное пространство внутри браузера или родительского элемента, даже если их содержимое совсем небольшое. **Строчными** (или *inline* — линейными) элементами называют такие элементы, которые занимают столько места, сколько в них находится текста, и размещаются они внутри блочных.
- Строчные элементы всегда будут расположены рядом друг с другом, если идут подряд в разметке документа, а блочные будут перенесены на следующую строку. Кроме того, блочные элементы предназначены для структурирования страницы, а строчные — для придания определенной функции тексту (ссылка, например) или для форматирования этого текста (`<b>` или `<i>`).
- Блочные всегда содержат как открывающий, так и закрывающий тег, а строчные могут состоять только из одного открывающего (например, `<br>` или `<img>`).
- **С точки зрения CSS**, различия между этими двумя группами задаются разными значениями одного и того же css-свойства: `display: block` (для блочных элементов) или `display: inline` (для строчных элементов). Еще давайте определимся, что такое *родительский элемент*, или *родительский контейнер*? Это тот элемент, в который будет вложен нужный нам тег. Чаще всего таким контейнером является тег `<body>`, т.к. все, что должно быть выведено на html-странице, помещается в этот тег.
- Но, если мы говорим о строчных тегах `<del>` и `<ins>`, то, скорей всего, они будут помещены в тег `<p>`, который и будет для них родительским. Теги `<del>` и `<ins>` в этом случае будут называться вложенными, или дочерними.

# Блочные и строчные теги

Таблица 1. Блочные элементы

Теги	Назначение тега
html	Корневой тег
body	Тело документа
h1-h6	Теги заголовков
p	Абзац
div	Блок контента
address	Адрес
article	Статья
aside	Боковая колонка
nav	Навигация на странице
main	Основное содержимое web-страницы
header	Верхняя часть страницы или раздела
footer	Нижняя часть страницы, раздела или блочной цитаты
section	Раздел страницы
audio	Вставка звука
video	Вставка видео
blockquote	Блочная цитата
caption	Заголовок таблицы
colgroup	Определяет группу столбцов таблицы
table	Таблица

# Блочные и строчные теги

Теги	Назначение тега
thead, tbody, tfoot	Для выделения групп строк таблицы
form	Для создания формы
datalist	Контент для заполнения поля формы
select	Выпадающий список
fieldset	Группа полей формы
legend	Заголовок для fieldset
dl, ul, ol	Теги списков
canvas	Для рисования графики с помощью JavaScript
details	Для создания интерактивного элемента с частично скрытым содержимым
summary	Видимая часть текста элемента details
figure	Обертка для элемента <img>
figcaption	Заголовок для тега <figure>
map	Интерактивная карта
iframe	Контейнер для загрузки содержимого другой html-страницы
pre	Преформатированный текст
progress	Индикатор выполнения

# Блочные и строчные теги

Таблица 2. Строчные элементы

Теги	Назначение тега
a	Ссылка
abbr	Аббревиатура
b, strong	Жирный текст
i, em	Курсивное начертание текста
dfn	Определение (англ. <i>definition</i> )
br	Перевод строки
button, input, textarea	Кнопка, текстовое поле и текстовая область — элементы формы
label	Метка для элемента формы
cite	Заголовок работы, ссылка на источник

## Блочные и строчные теги

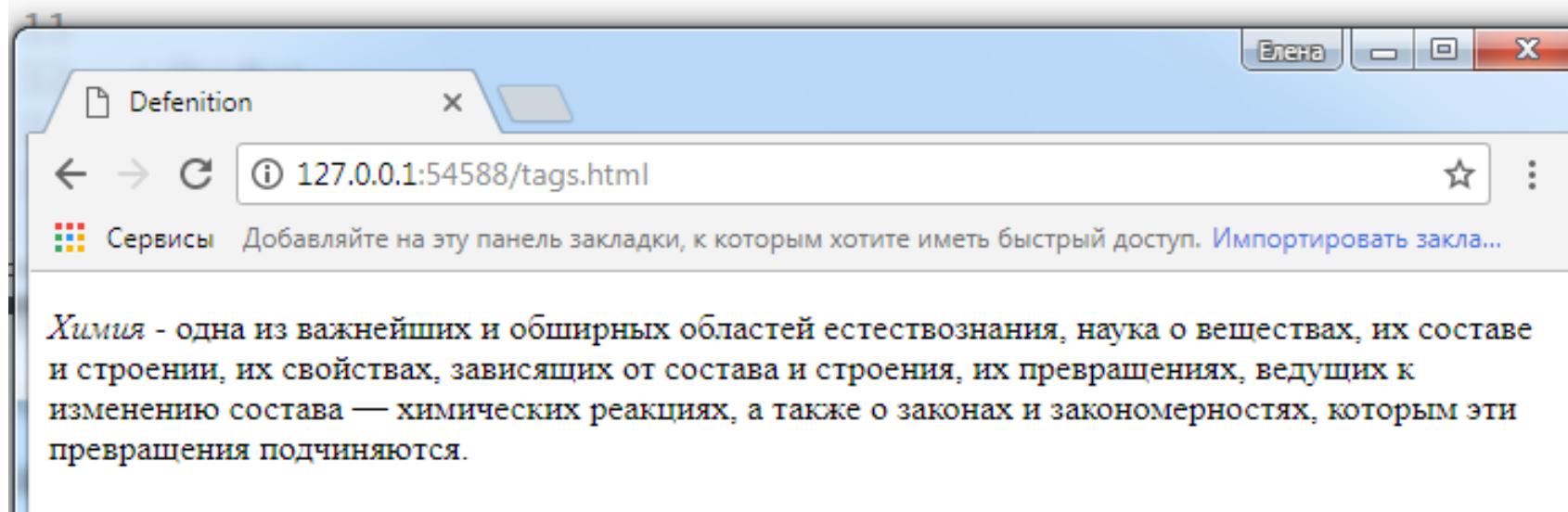
Теги	Назначение тега
code	Для выделения программного кода
kbd	Код, введенный с клавиатуры (англ. <i>keyboard</i> )
samp	Пример вывода чего-либо в компьютерной программе
var	Переменная в какой-либо программе (англ. <i>variable</i> )
col	Столбец в таблице
del	Текст, который был удален
ins	Текст, который был добавлен (часто следует за тегом <del>)
s	Зачеркнутый текст, т.е. текст, который не является уже правильным или достоверным. Лучше использовать тег <del>
span	Контейнер для задания css-форматирования определенного количества текста
sub	Нижний индекс (H <sub>2</sub> O)
sup	Верхний индекс (x <sup>2</sup> )
small	Более мелкий текст
img	Вставка изображений
mark	Выделенный текст
meter	горизонтальный счетчик
q	Короткая цитата

# Теги физического и логического форматирования

- В начале развития HTML добавлялись новые теги, которые позволяли, как правило, визуально изменять вид текста. Это используемые до сих пор и уже рассмотренные нами `<b>` и `<i>`, которые задают жирное и курсивное начертание текста. Существует также тег `<u>`, который подчеркивает текст, а тег `<s>` — зачеркивает его, `<small>` делает текст меньше, а теги `<sup>` и `<sub>`, соответственно, преобразуют текст в верхний и нижний индексы. Еще один тег — `<pre>` позволяет выводить текст в html-документе в том виде, в котором он был набран в текстовом редакторе, т.е. с сохранением всех переносов строк, табуляций и дополнительных пробелов.
- Можно сказать, что все эти теги физически изменяют внешний вид текста на html-странице. Поэтому их относят к группе тегов **физического форматирования**.
- В противовес этим тегам существуют теги **логического форматирования**, которые также физически изменяют внешний вид текста на странице, но их название содержит, как правило слово или его часть на английском языке.
- Например, тег `<del>` происходит от английского слова *delete* (удалять) и обозначает текст, удаленный в новой версии документа или по прошествии некоторого времени.
- Визуально он выводит текст перечеркнутым. Тег `<dfn>` происходит от английского слова *defenition* (определение), и предназначен именно для отображения каких-либо определений. В браузере он будет выделен курсивным начертанием текста

# Теги физического и логического форматирования

10 `<p><dfn>Химия</dfn>` – одна из важнейших и обширных областей естествознания, наука о веществах, их составе и строении, их свойствах, зависящих от состава и строения, их превращениях, ведущих к изменению состава — химических реакциях, а также о законах и закономерностях, которым эти превращения подчиняются. `</p>`





# Теги физического и логического форматирования

- В HTML5 есть много тегов логического форматирования, которые определяют *семантическую структуру документа*. Это такие теги, как `header` — шапка или заголовочная часть страницы или статьи, `main` — основная часть документа, `section` — раздел документа, `article` — статья, `footer` — нижняя часть документа или статьи. Эти теги были введены в качестве замены `div`-ам с соответствующими классами или `id` и определяют логику разметки страницы. В данном случае слово **семантика** указывает на соответствие внутренней разметки html-документа и его визуального форматирования. С точки зрения семантики рассматривают разметку страницы поисковики.
- Например, теги `<strong>` и `<em>` аналогично тегам `<b>` и `<i>` выводят текст жирным или курсивным начертанием, но при этом имеют еще и свой *семантический* смысл. Текст, заключенный в теги `<strong>` (усиление) имеет повышенную значимость своего содержимого, а текст в тегах `<em>` (эмфатическое ударение) содержит эмоциональный акцент на данной отрывке текста. В обычной речи мы выделяем такие слова голосом (интонацией, громкостью и др.).
- HTML5 принес семантическое содержание для тегов физического форматирования. Тот же тег `<b>` следует использовать для текста, на который необходимо обратить внимание читателя, но без усиления значимости текста или интонации, как для тега `<strong>`. Что касается тега `<i>`, то подразумевается, что он содержит текст, выбивающийся из общего окружения, но не имеющий эмоциональной окраски.

# Теги физического и логического форматирования

- Следует отметить, что довольно много тегов логического форматирования применяются не так уж и часто и на сайтах определенной тематики. Так, например, элементы `<code>`, `<var>`, `<samp>`, `<kbd>`, как правило, используются на сайтах, посвященных тому или иному языку программирования, а `<abbr>` и `<dfn>` чаще будут встречаться на сайтах-учебниках, предоставляющих информацию по какому-либо курсу (химии, физике, правоведению и т.п.).
- Очень удобно для таких элементов описать в CSS-свойствах единые для всего сайта правила форматирования.
- В HTML5 были также добавлены элементы, которые условно можно отнести к интерактивным. Это такие элементы, как `<details>` и `<summary>`, которые позволяют отображать скрытое содержимое по клику, а также `<progress>` `<meter>`, использование которых имеет смысл после изучения языка JavaScript, т.к. лучше всего ими управлять на этом языке.

## Устаревшие теги и атрибуты

- HTML, как язык гипертекстовой разметки, неоднократно изменялся и имел ряд стандартов, в котором вводились одни теги и отменялось использование других. Стандарт HTML5 внес множество новых тегов, но часть тех, которые были популярны в предыдущих версиях HTML, определил как **obsolete** — устаревший или **deprecated** — осуждаемый. Т.е. использование таких тегов не рекомендуется в вашем документе, поддерживающем стандарт HTML5, иначе документ не пройдет валидацию. Напомню, браузер определяет, к какому стандарту относится ваш документ по DOCTYPE, который указывается в самом начале разметки, и для HTML5 выглядит так:

`<!DOCTYPE html>`

- В Интернете существует масса уроков, созданных давно, в которых эти устаревшие теги или атрибуты используются и рекомендуются.
- Например, для увеличения размера шрифта ранее использовался тег `<big>`, который сейчас считается устаревшим, а вот тег `<small>`, который уменьшает размер шрифта, до сих пор используется. То же касается и тега `<strike>` — он перечеркивал текст в HTML4, но устарел в HTML5. Вместо него можно использовать тег `<s>` или лучше — тег `<del>`, указывающий на то, что перечеркнутая информация устарела или неверна. То же касается тега `<tt>`, который раньше использовался для вывода текста моноширинным шрифтом.

## Устаревшие теги и атрибуты

- Еще один пример — для обозначения аббревиатуры можно было использовать ранее два тега `<abbr>` или `<acronym>`. В стандарте HTML5 остался только `<abbr>`, а `<acronym>` получил статус устаревшего. Сделано это было в силу того, что `<acronym>` (устоявшееся сокращение, которое применяется как самостоятельное слово, например, ООН или DOM, или часто используемый в комментариях ИМНО, или *на англ. in my humble opinion*), хоть и отличается от аббревиатуры (тег `<abbr>`) по звучанию, но с точки зрения HTML имеет то же значение, поэтому необходимость в использовании акронимов отпала. Например, в документации сайта [developer.mozilla.org](http://developer.mozilla.org) можно увидеть такое сообщение

### `<acronym>`



#### Вышла из употребления

Эта возможность вышла из употребления. Хотя она может продолжать работать в некоторых браузерах, её использование не рекомендуется, поскольку она может быть удалена в любое время. Старайтесь избегать её использования.

#### Аннотация

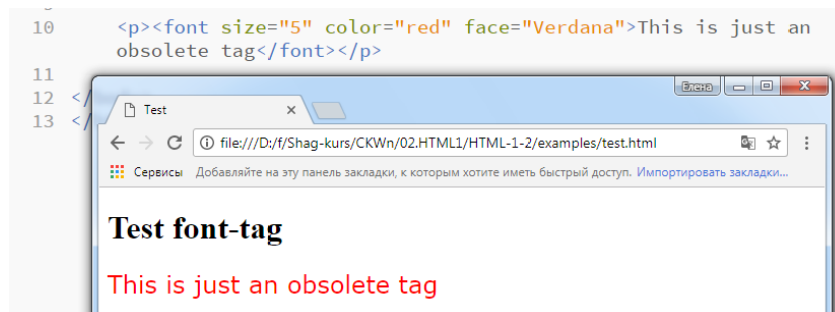
Элемент акронима (`<acronym>`) позволяет явно указать, что данная последовательность букв, являющаяся акронимом или аббревиатурой.



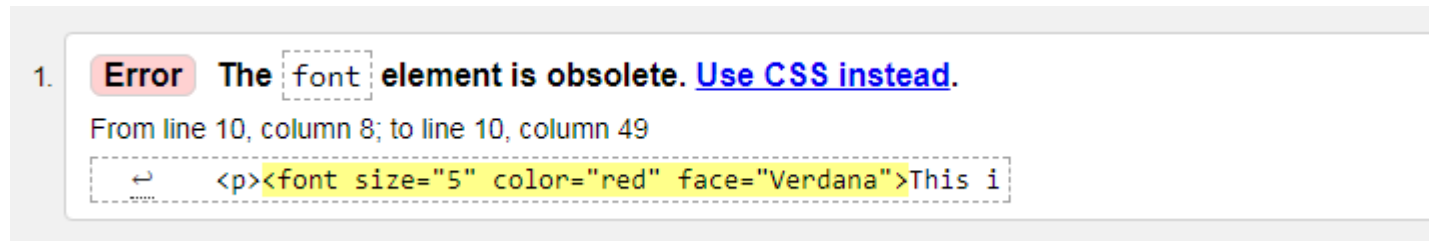
**Примечание:** Этот элемент был убран из спецификации HTML5, поэтому его не рекомендуется использовать. Вместо него используйте элемент `<abbr>`.

# Устаревшие теги и атрибуты

- Есть еще один устаревший тег, который почему-то невероятно популярен у студентов, начинающих изучать HTML, — это тег `<font>`, который позволял в предыдущих стандартах задать размер, цвет и семейство шрифта. Например, так



- HTML-валидатор для данного примера выдал следующую ошибку



- Фактически этот тег заменял ранее использование таких CSS-свойств, как `font-size`, `color` и `font-family`. Пожалуйста, не используйте в своих файлах этот тег.
- То же самое относится и к атрибутам, которые в более ранних стандартах применялись сплошь и рядом. Например, такой атрибут, как `align` имел следующие значения: `left`, `right`, `center`, `justify`, и очень часто применялся для форматирования текста в заголовках, абзацах, `div`-ах и даже для обтекания текстом картинок. В силу того, что все возможности этого атрибута можно заменить CSS-свойством `text-align` или для картинок — свойством `float`, данный атрибут является устаревшим.

# Устаревшие теги и атрибуты

## Deprecated Features in HTML5

- Invalid HTML
- Frames
- Applets
- `<xmp>` HTML Tag
- `<table width="">`
- `<table summary="">`
- `<table rules="">`
- `<table hspace="">`
- `<table height="">`
- `<sound>` HTML Tag
- `<select readonly>`
- `<plaintext>` HTML Tag
- `<ol start="">`
- `<noembed>` HTML Tag
- `<multicol>` HTML Tag
- `<multicol width="">`
- `<multicol gutter="">`
- `<multicol cols="">`
- `<marquee>` HTML Tag
- `<listing>` HTML Tag
- `<input language="">`
- `<img suppress="">`
- `<img start="">`
- `<img nosave>`
- `<img naturalsizeflag="">`
- `<img name="">`
- `<img lowsrc="">`
- `<img loop="">`
- `<img hspace="">`
- `<img dynsrc="">`
- `<img controls>`
- `<img border="">`
- `<img align="">`
- `<hype>` HTML Tag
- `<frameset>` HTML Tag
- `<frame>` HTML Tag
- `<form name="">`
- `<font>` HTML Tag
- `<font weight="">`
- `<font size="">`
- `<font point-size="">`
- `<font face="">`
- `<font color="">`
- `<center>` HTML Tag
- `<caption align="">`
- `<body topmargin="">`
- `<body text="">`
- `<body stylesrc="">`
- `<body scroll="">`
- `<big>` HTML Tag
- `<bgsound>` HTML Tag
- `<basefont>` HTML Tag
- `<basefont size="">` HTML Attribute
- `<basefont face="">` HTML Attribute
- `<basefont color="">` HTML Attribute
- `<applet>` HTML Tag
- `<applet width="">` HTML Attribute
- `<applet vspace="">` HTML Attribute
- `<applet name="">` HTML Attribute
- `<applet mayscript="">` HTML Attribute
- `<applet hspace="">` HTML Attribute
- `<applet height="">` HTML Attribute
- `<applet codebase="">` HTML Attribute
- `<applet code="">` HTML Attribute
- `<applet border="">` HTML Attribute
- `<applet archive="">` HTML Attribute
- `<applet align="">` HTML Attribute
- `<app>` HTML Tag
- `<a name="">` HTML Attribute

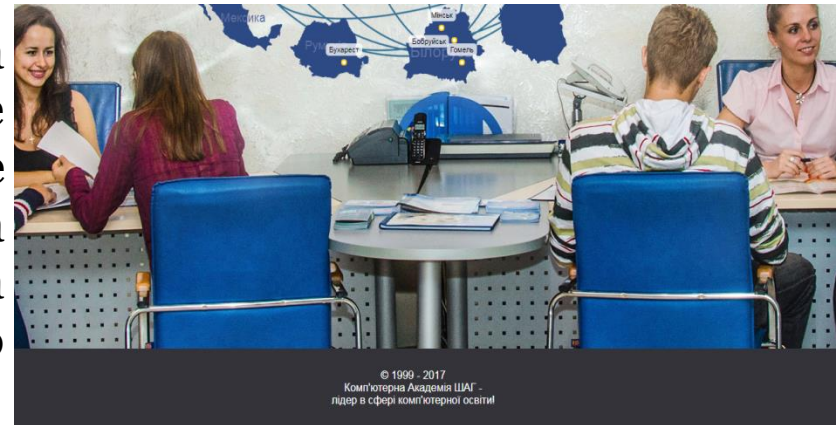


## Спецсимволы в HTML. Использование их на html-странице

- Возможно, вы сталкивались с символами в документах Word. Они обычно отсутствуют на клавиатуре, а вставляются с помощью специального пункта меню. Например, нельзя напечатать символ копирайта © или символы стрелок (↓ или ↑). Та же ситуация в HTML — символы можно либо скопировать из текста в том же Word, или добавить с помощью специальных кодов. Спецсимволы в HTML еще называют *мнемониками* (англ. — *entity*). Они имеют 2 варианта синтаксиса — в виде частей англоязычных слов или номеров, которые обязательно начинаются со знака «амперсанд» (&) и заканчиваются точкой с запятой (;). Использовать можно любой из вариантов, но следует учитывать, что спецсимволы без именного кода могут совсем не отображаться или отображаться некорректно в различных браузерах. Например, символ копирайта © можно представить, как мнемокод &copy; или &#169; В текстовом редакторе вы увидите набор символов, а на html-странице они будут представлять всего один символ.
- Есть ряд спецсимволов, знать которые следует на память, т.к. применять вы его будете в различных ситуациях.

# Знаки охраны авторского права или торговой марки

- Например, это знак того же копирайта (авторского права), без которого не обходится обычно не один сайт. Чаще всего, его размещают в нижней части сайта (подвале или футере). Например, на официальном сайте Академии ШАГ это выглядит так



- Также может быть использован рядом с названием фирмы знак торговой марки (спецсимвол &trade; или &#8482). Например, таким образом

© 2017 ООО "Аудиторские системы"™

- Используется, правда, этот знак значительно реже, чем знак копирайта.
- Что касается Brackets, то при вводе начальных символов мнемоника появляется подсказка, которая содержит и полный код, и его внешний вид на странице

```
<footer>  
  <p class="copyright">&copy; 2017 ООО "Аудиторские системы"&tr</p>  
</footer>  
<body>
```

&trade; ™



# Знаки охраны авторского права или торговой марки

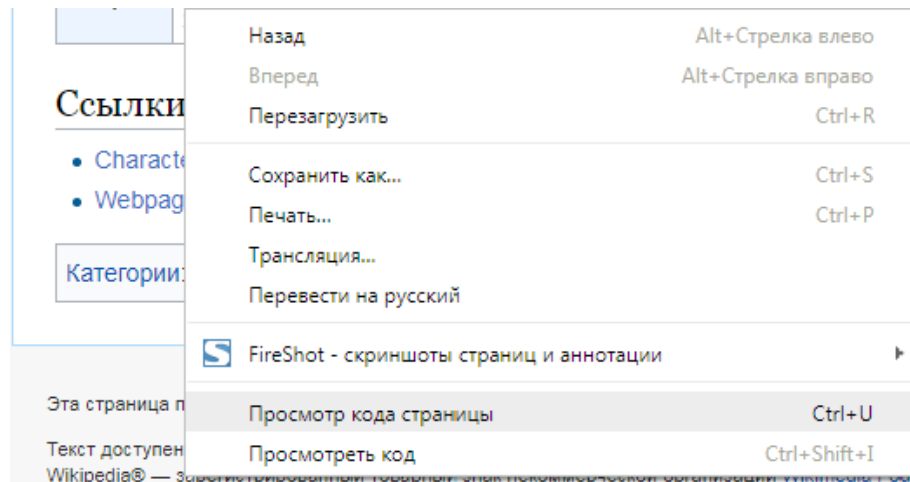
- Если у компании (предприятия), для которого сделан сайт, имеется зарегистрированная торговая марка, то рядом с названием этой компании (предприятия) можно указать знак ® (спецсимвол &reg; или &#174;). Например, это спецсимвол использован на сайте Википедии

Wikipedia® — зарегистрированный товарный знак некоммерческой организации [Wikimedia Foundation, Inc.](#)

- Выдержка из кода страницы

```
<br />
470 Wikipedia&reg; – зарегистрированный товарный знак некоммерческой организации <a
href="//wikimediafoundation.org/wiki/Приёмная">Wikimedia Foundation, Inc.</a><br /></li>
```

- Чтобы посмотреть код открытой в браузере страницы, можно нажать клавиши *Ctrl + U* или по правому клику в любом месте страницы выбрать в контекстном меню пункт «Просмотр кода страницы».



## Использование пробелов и переносов

- Существуют ситуации, когда некоторые словосочетания не могут быть перенесены на другую строку по словам. Самые распространенные варианты — это фамилия и инициалы, форма собственности и название компании, наименование города или поселка. Например, Иванов А. Ф., ООО «Стройинвест», г. Харьков. В этом случае для их «склеивания» вместо обычного пробела используют неразрывный пробел &nbsp; или &#160;.
- Сравните внешний вид текста на скриншоте:

### Без неразрывного пробела

Следует учитывать, что наша компания ООО  
"Стройинвесттрейд" в лице исполнительного директора Бойко В.  
Н. будет представлять на выставке "Экспо 2018" в январе 2018  
г. в г. Харькове все наши лучшие наработки

### С неразрывным пробелом

Следует учитывать, что наша компания  
ООО "Стройинвесттрейд" в лице исполнительного директора  
Бойко В. Н. будет представлять на выставке "Экспо 2018" в  
январе 2018 г. в г. Харькове все наши лучшие наработки

# Использование пробелов и переносов

## Мягкий перенос и неразрывный пробел

Согласно отчету о результатах социологического исследования, опубликованному Microsoft в 2015-м году, средний человек в возрасте от 18 до 24 лет способен сосредоточиться на чем-либо не более чем на 8 секунд. Это так называемый attention span, период непрерывной фокусировки внимания. Пятнадцать лет назад эта цифра составляла 12 секунд. У золотой рыбки этот показатель равен девяти секундам. То есть, если взять усредненно-го «вчерашнего тинейджера» и посадить в лабиринт, типа таких, на которых проверяют обучаемость мелких грызунов, то он будет беспомощно тыкаться в стены, так как паталогически не способен сфокусировать внимание на время, доста-

- Еще одна ситуация связана с необходимостью добавить переносы между частями слов, т.к. размер блока для текста может быть слишком мал, чтобы вместить длинные слова вроде «Зарегистрированный». Пока HTML не в полной мере поддерживает автоматический перенос слов по слогам, хотя это вопрос уже недалекого будущего. Сейчас при необходимости разорвать слишком длинное слово можно поставить в нужных местах знаки мягкого переноса (&shy; или &#173;). Ставить их следует в соответствии с нормами языка. Количество переносов зависит от длины слова и ваших потребностей. Знаки мягкого переноса можно использовать после каждого слога:

За&shy;ре&shy;ги&shy;стри&shy;ро&shy;ван&shy;ный

# Использование пробелов и переносов

- Что касается автоматического переноса слов, то для этого было введено css-свойство `hyphens`. Со значением `auto` при указании языка в атрибуте `lang` для тега `html`, будет выполнен автоматический перенос длинных слов в соответствии с нужным словарем. Например, тот текст, который ранее мы переносили с помощью спецсимвола `&shy`; теперь можно перенести, задав `css-стили`:

```
<html lang="ru">
```

```
...
```

```
<style>
```

```
.hyphens {
```

```
-moz-hyphens: auto;
```

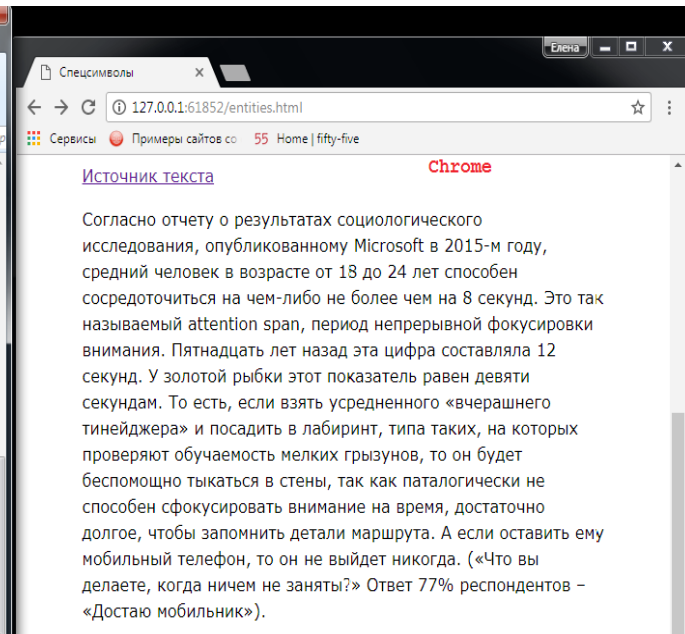
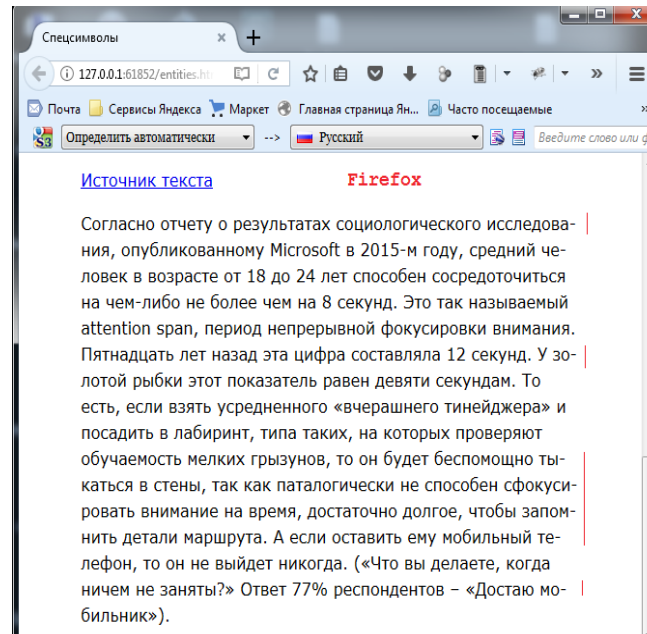
```
-ms-hyphens: auto;
```

```
-webkit-hyphens: auto;
```

```
hyphens: auto;
```

```
}
```

```
</style>
```



- Минус этого способа то, что такого рода переносы поддерживаются не всеми браузерами. Например, если в Firefox текст отобразился с переносами, то в Chrome мы видим «рваные края» с правой стороны

# Использование пробелов и переносов

- Для того чтобы быть в курсе, насколько поддерживается то или иное CSS-свойство в браузерах, можно сделать запрос на сайте [caniuse.com](http://caniuse.com).

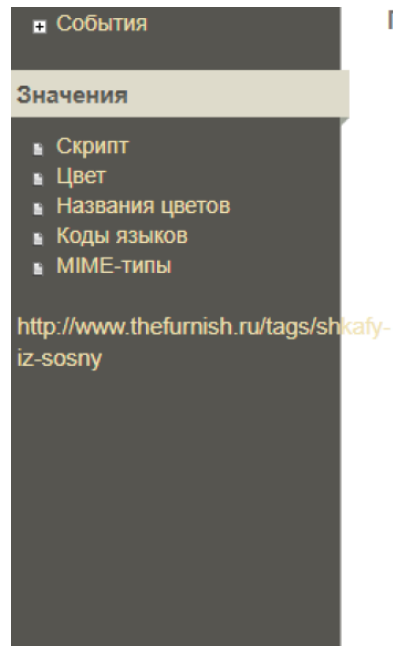
На скриншоте видно, что зеленым цветом отмечены браузеры и их версии, которые поддерживают это свойство, красным цветом — которые не поддерживают, оливковым цветом — поддерживают, но с каким-то условием (например, Chrome с версии 55 поддерживает только на платформах Android или Mac).

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	BlackBerry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
		40	46		32												
		41	47		33												
		42	48	3.1	34												
		43	49	3.2	35												
		44	50	4	36	3.2											
		45	51	5	37	4.1											
		46	52	5.1	38	4.3											
		47	53	6	39	5.1		2.1									
		48	54	6.1	40	6.1		2.2									
		49	55	7	41	7.1		2.3									
		50	56	7.1	42	8		3									
6		51	57	8	43	8.4		4									
7	12	52	58	9	44	9.2		4.1									
8	13	53	59	9.1	45	9.3		4.3									
9	14	54	60	10	46	10.2		4.4		12							
10	15	55	61	10.1	47	10.3		4.4.4	7	12.1			10		4		
11	16	56	62	11	48	11	all	56	10	37	61	56	11	11.4	5	1.2	7.12
		57	63	TP	49												
		58	64		50												
		59	65														

- Знак «минус» показывает, что свойство поддерживается с так называемым «вендорным префиксом»:
  - -moz- — для Mozilla FireFox,
  - -ms- — для Internet Explorer,
  - -webkit- — для Safari, Chrome и других браузеров на основе Chromium — движка от Chrome.

## Использование знаков

- Спецсимволов для знаков существует огромное количество, но хотелось бы выделить 2, которые относятся к html-коду.
- Например, на таких сайтах-справочниках, как <http://htmlbook.ru/> или <https://webref.ru/> есть масса примеров кода, который вы можете скопировать в свой текстовый редактор и просмотреть «вживую». Как же возможно вывести на страницу теги, которые браузер должен интерпретировать и показать как текст? Для этого все угловые скобки «кодируют» с помощью спецсимволов: левая угловая скобка (знак меньше <) — как &lt;, правая угловая скобка (знак больше >) — как &gt;.



Пример

```

<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Тег KBD</title>
  <style>
    kbd {
      font-weight: bold; /* Жирное начертание */
    }
  </style>
</head>
<body>

  <p>Для быстрого закрашивания выделенной области цветом переднего плана
  используйте комбинацию клавиш <kbd>&lt;Alt&gt;+&lt;Backspace&gt;</kbd>
  или <kbd>&lt;Alt&gt;+&lt;Delete&gt;</kbd>.
  Для закрашивания выделенной области фоновым цветом
  воспользуйтесь комбинацией
  <kbd>&lt;Ctrl&gt;+&lt;Backspace&gt;</kbd> или
  <kbd>&lt;Ctrl&gt;+&lt;Delete&gt;</kbd>.</p>

</body>
</html>

```

# Использование знаков

HTML

Результат

JSFiddle

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>mark</title>
  </head>
  <body>
    <header>
      <h1>Следы невиданных зверей</h1>
    </header>
    <article>
      История о том, как возле столовой появились загадочные розовые
      следы с <mark>шестью пальцами</mark>, и почему это случилось.
    </article>
  </body>
</html>
```

Атрибуты

Универсальные

События

Значения

МIME-типы

Адрес

Коды языков

Названия цветов

Проценты

Скрипт

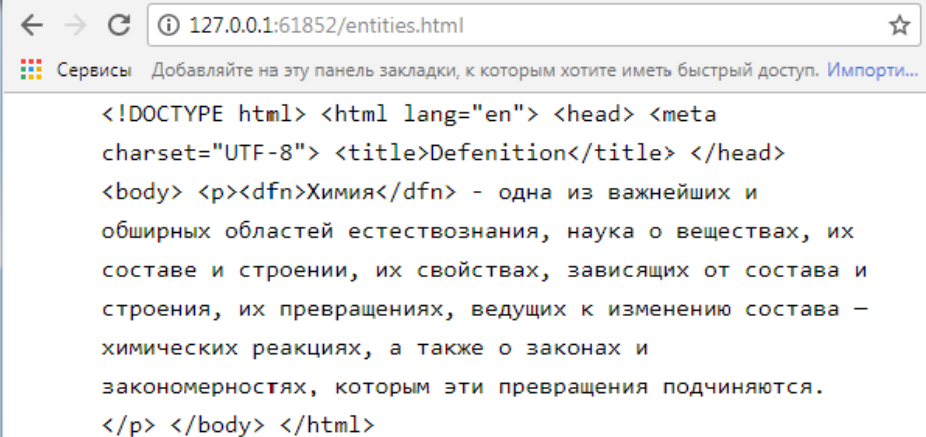
- После этого код будет выводиться как обычный текст. В примере в файле entities.html видно, что html-код перестал быть таковым в текстовом редакторе за счет замены скобок на спецсимволы, но выглядит как html-код на странице

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Defenition</title>
</head>

<body>
  <p><dfn>Химия</dfn> -
  одна из важнейших и обширных областей
  естествознания, наука о веществах, их
  составе и строении, их свойствах,
  зависящих от состава и строения, их
  превращениях, ведущих к изменению состава
  – химических реакциях, а также о законах
  и закономерностях, которым эти
  превращения подчиняются.</p>
</body>

</html>
```





## Использование знаков

- Для того чтобы сохранить внешний вид строк текста в редакторе и на странице, необходимо задать следующие CSS-свойства для тега `<code>`:

```
code {
    white-space: pre;
}
```

```
42 -->
43 <code>
44     &lt;!DOCTYPE html&gt;
45     &lt;html lang="en"&gt;
46
47     &lt;head&gt;
48         &lt;meta charset="UTF-8"&gt;
49         &lt;title&gt;Defenition&lt;/title&gt;
50     &lt;/head&gt;
51
52     &lt;body&gt;
53         &lt;p&gt;&lt;dfn&gt;Химия&lt;/dfn&gt; -
54         одна из важнейших и обширных
55         областей естествознания, наука о
56         веществах,
57         их составе и строении, их свойствах,
58         зависящих от состава и строения, их
59         превращениях,
60         ведущих к изменению состава – химических
61         реакциях,
62         а также о законах и закономерностях,
63         которым эти превращения
64         подчиняются.&lt;/p&gt;
65     &lt;/body&gt;
66
67     &lt;/html&gt;
68 </code>
```

Сервисы Добавляйте на эту панель закладки, к которым хотите иметь быстрый доступ. [Иморти...](#)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Defenition</title>
</head>

<body>
    <p><dfn>Химия</dfn> - одна из важнейших и обширных
    областей естествознания, наука о веществах,
    их составе и строении, их свойствах,
    зависящих от состава и строения, их превращениях,
    ведущих к изменению состава – химических реакциях,
    а также о законах и закономерностях,
    которым эти превращения подчиняются.</p>
</body>

</html>
```

# CSS-свойства для форматирования текста

- **1. Свойство** `white-space` отвечает за отображение пробелов между словами. По умолчанию значение `normal` устанавливает автоматические переносы слов по достижению правого края браузера или родительского блока.

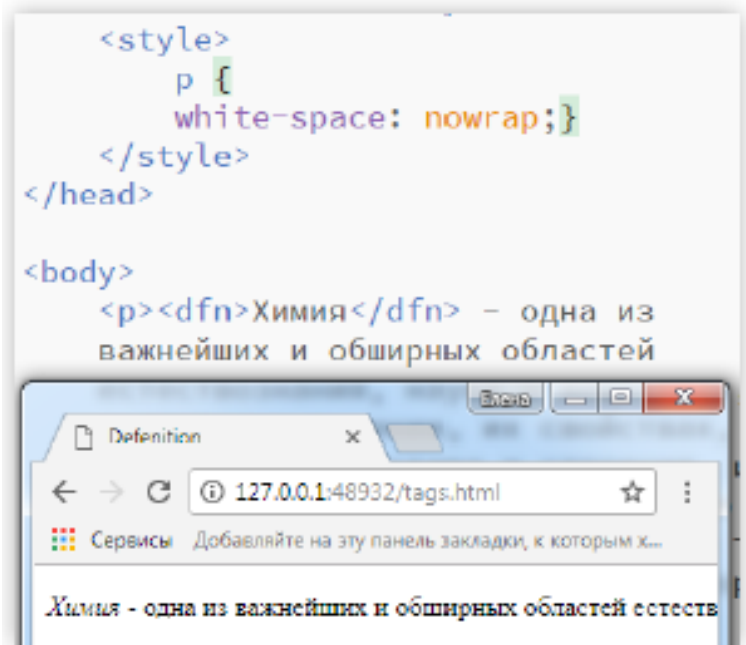
```
white-space: normal | nowrap | pre | pre-line | pre-wrap
```

- **Значение** `nowrap` не учитывает пробелы и переносы строк в коде HTML, и весь текст отображается как одна строка. При этом может быть добавлена горизонтальная полоса прокрутки, если текст не помещается по ширине родительского контейнера. Только добавление тега `<br>` переносит текст на новую строку.

- **Значение** `pre` отображает текст с учётом всех пробелов, табуляций и переносов, какие присутствуют в коде HTML. Для слишком длинной строки в браузере будет добавлена горизонтальная полоса прокрутки.

- **Значение** `pre-line` не учитывает пробелы, а сам текст будет перенесён на следующую строку, если он не помещается в родительский контейнер.

- **Значение** `pre-wrap` сохраняет все пробелы и переносы, которые были в текстовом редакторе, но если строка по ширине не помещается в родительский контейнер, то текст автоматически будет перенесён на следующую строку.



# CSS-свойства

## для форматирования текста

- **2. Свойство** `word-break` отвечает за перенос строк внутри слов, которые не помещаются внутриродительского контейнера по ширине. Имеет следующие значения:

```
word-break: normal | break-all | keep-all
```

- **Значение** `normal` (по умолчанию) переносит слова целиком по достижении ими правого края родительского блока

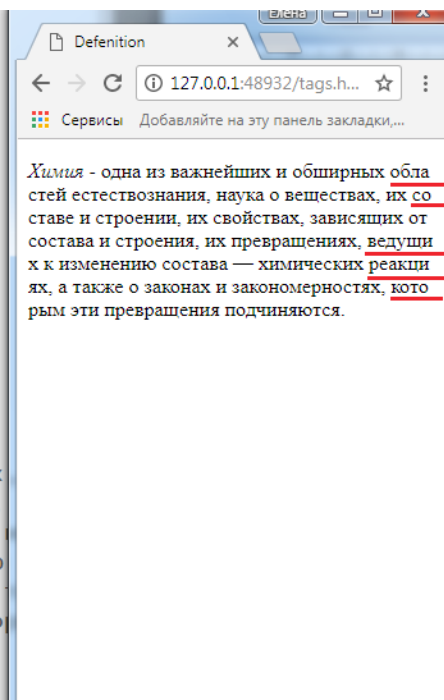
- **Значение** `break-all` добавляет перенос строк так, чтобы слово поместилось в заданную ширину родительского блока (не может быть использовано для текста на китайском, корейском или японском языках).

- **Значение** `keep-all` запрещает перенос строк для текста на китайском, корейском или японском языках. Для остальных языков соответствует значению `normal`.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Defenition</title>
  <style>
    p {
      word-break: break-all;
    }
  </style>
</head>

<body>
  <p><dfn>Химия</dfn> - одна из
  важнейших и обширных областей
  естествознания, наука о веществах
  составе и строении, их свойствах,
  зависящих от состава и строения,
  превращениях, ведущих к изменению
  состава — химических реакциях, а
  о законах и закономерностях, кото
  эти превращения подчиняются.</p>
</body>
```



- ```
word-wrap: normal | break-word
```

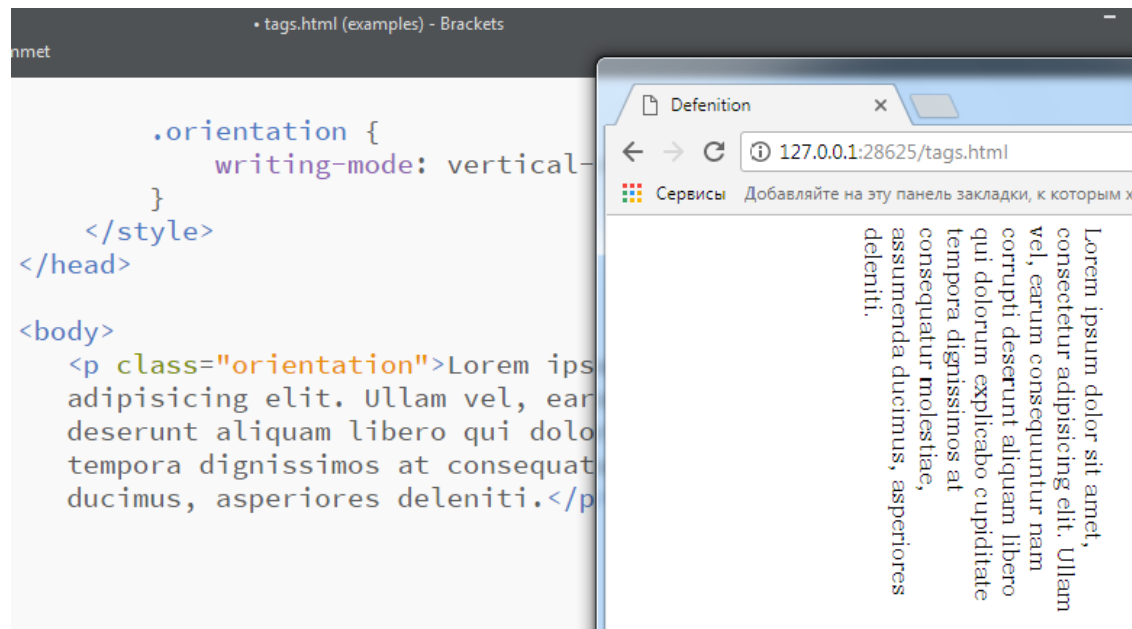
- 
- The screenshot shows a web browser window with a tab titled "tags.html (examples) - Brackets". The address bar displays "127.0.0.1:48932/tags.html". A "Definition" popup is visible, showing the definition for the CSS property "break-word". The definition text reads: "This paragraph contains a very long word: soherewehaveaveryveryveryveryverylongword. There are we use word-wrap: break-word property long word will break and wrap to the next line." The browser's status bar at the bottom shows "met".

# CSS-свойства для форматирования текста

- 4. Одно из новых свойств, добавленных в спецификации CSS3, — `writing-mode` — определяет направление текста на странице — горизонтальное или вертикальное. По умолчанию имеет значение `horizontal-tb`:

```
writing-mode: horizontal-tb | vertical-rl | vertical-lr
```

- Значение** `horizontal-tb` определяет направление текста по горизонтали сверху вниз и слева направо.
- Значение** `vertical-rl` определяет направление текста по вертикали сверху вниз и справа налево.
- Значение** `vertical-lr` определяет направление текста по вертикали сверху вниз и слева направо. Применяется ко всем элементам, кроме ячеек и строк таблицы

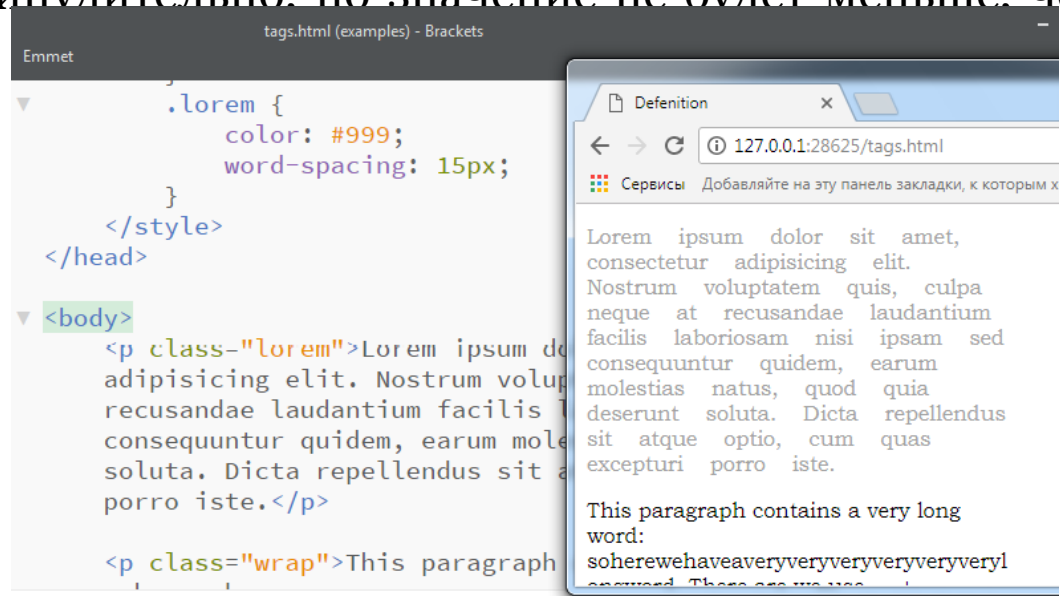


## CSS-свойства для форматирования текста

- **Свойство** `word-spacing` определяет интервал между словами. По умолчанию значение `normal` устанавливает обычный интервал, соответствующий параметрам шрифта. Можно также указать размер в любых единицах измерения, кроме %:

```
word-spacing: число в px, pt, em, но не в % | normal
Emmet: WOS
```

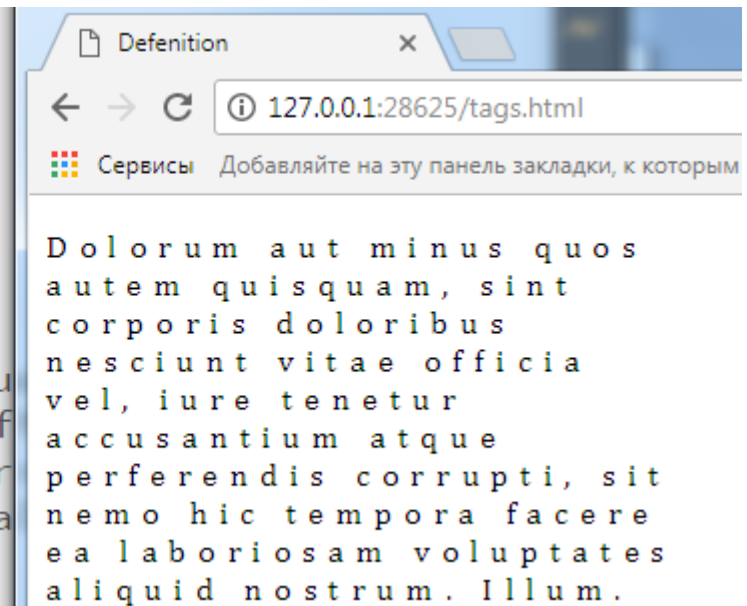
- Поддерживаются также отрицательные значения, уменьшающие расстояние между словами. В случае, когда для текста задано свойство, обеспечивающее выравнивание по ширине (`text-align: justify`), интервал между словами будет установлен принудительно, но значение не будет меньше, чем указанное в `word-spacing`.



## CSS-свойства для форматирования текста

- 6. **Свойство** `letter-spacing` позволяет установить интервал между символами для селектора. По умолчанию это расстояние определяется в зависимости от шрифта. Можно указывать отрицательные значения, если это необходимо и не портит вид текста.

```
.lettering {
  font-family: Cambria;
  letter-spacing: 6px;
}
</style>
</head>
<body>
  <p class="lettering">Dolorum aut minus qu
  sint corporis doloribus nesciunt vitae of
  tenetur accusantium atque perferendis cor
  tempora facere ea laboriosam voluptates a
  Illum.</p>
```



`letter-spacing`: число в px, pt, em, но не в % | normal

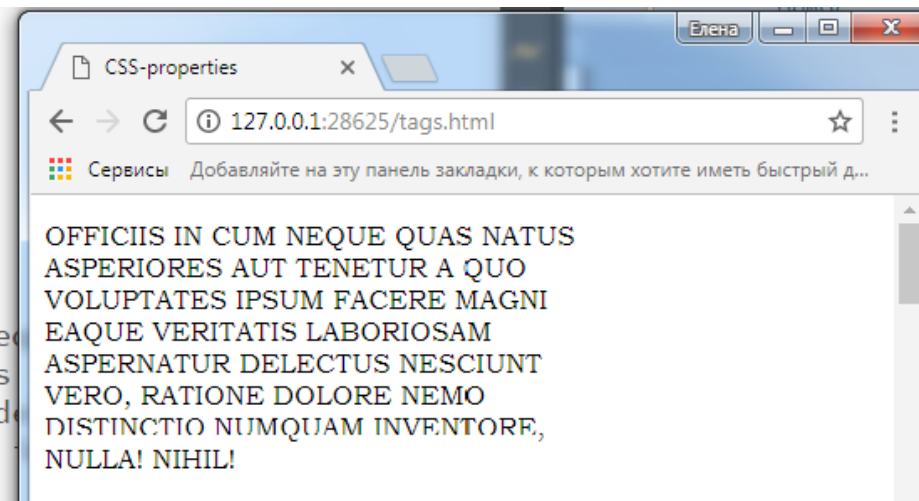
Emmet: `ltsn`, `lts:число`  
`.lettering {letter-spacing: 6px;}`

## CSS-свойства для форматирования текста

- 7. **Свойство** `text-transform` позволяет изменить регистр текста, преобразуя символы в заглавные (значение `uppercase`) или строчные (значение `lowercase`). Значение `none` оставляет текст неизменным, а `capitalize` преобразует первые буквы каждого слова в заглавные (больше характерно для английского языка).

```
text-transform: capitalize | lowercase | uppercase | none
```

```
.transform {  
    text-transform: uppercase;  
}  
</style>  
</head>  
  
<body>  
    <p class="transform">Officiis in cum ne  
asperiores aut tenetur a quo voluptates  
eaque veritatis laboriosam aspernatur de  
ratione dolore nemo distinctio numquam  
Nihil!</p>
```





## CSS-свойства для форматирования текста

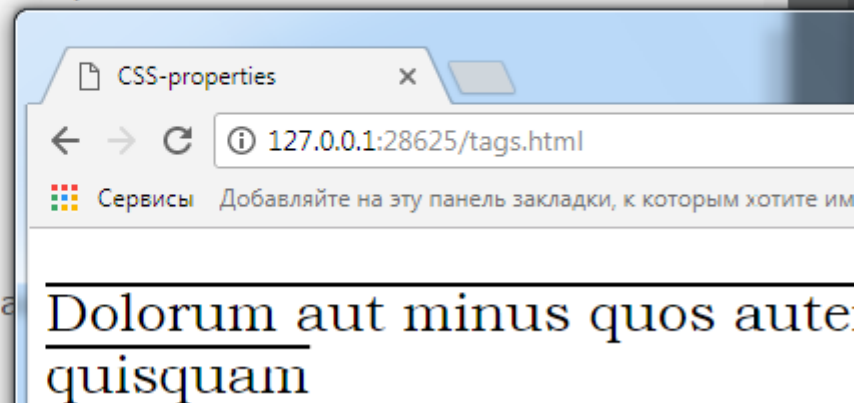
- **8. Свойство** `text-decoration` добавляет оформление
- текста в виде подчеркивания (значение `underline`), пере-
- черкивания (значение `line-through`) или надчеркивания
- (значение `overline`). Значение `none` оставляет текст неиз-
- менным.

```
text-decoration: underline | line-through | overline | none
```

```
.decoration {
  font-size: 1.7em;
  text-decoration: overline;
}
</style>
</head>
```

```
<body>
```

```
<p class="decoration">Dolorum aut  
quisquam</p>
```



## CSS-свойства для форматирования текста

- Свойство `text-decoration` состоит из 3-х свойств, которые можно задавать как по отдельности, так и вместе в объединенном свойстве `text-decoration`:

```
text-decoration: text-decoration-line |  
text-decoration-style | text-decoration-color
```

- Свойство `text-decoration-line` определяет вид линии и имеет значения, как в основном свойстве:

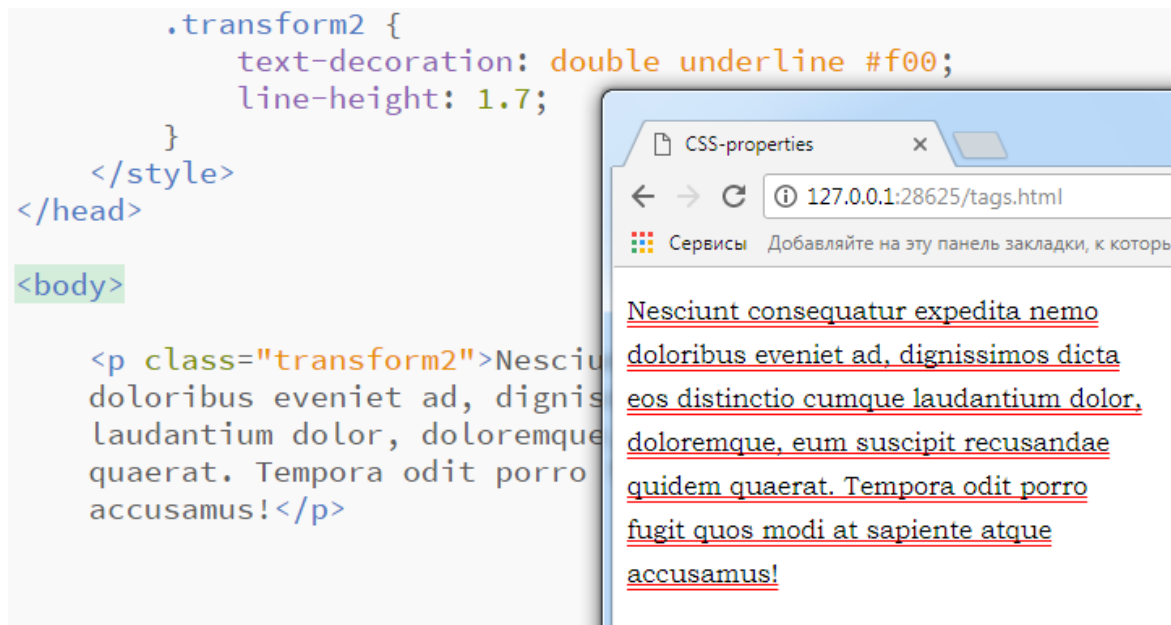
```
text-decoration-line: line-through | overline |  
underline | none
```

- Можно добавлять несколько линий, перечисляя значения через запятую. Вид линии определяется свойством `text-decoration-style`:

```
text-decoration-style: solid | double | dotted |  
dashed | wavy
```

# CSS-свойства для форматирования текста

- Значения этого свойства таковы:
  - solid — сплошная одинарная линия
  - double — двойная линия.
  - dotted — точечная линия.
  - dashed — пунктирная линия.
  - wavy — волнистая линия.
- Свойство text-decoration-color определяет цвет линии. Его значения можно задавать так же, как и для свойства color.

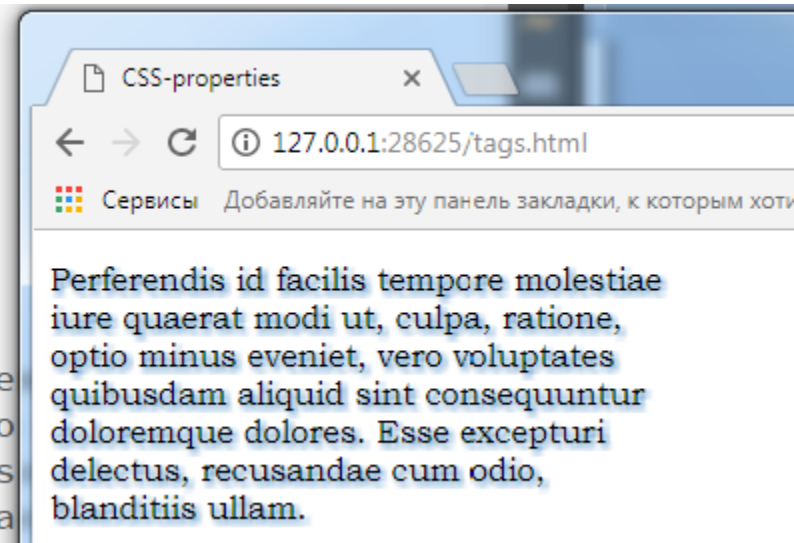


## CSS-свойства для форматирования текста

- **Свойство** `text-shadow` позволяет добавить тень к тексту. В качестве значений указывают смещение по горизонтали (`offX`), смещение по вертикали (`offY`), радиус размытия (`blur`), и цвет (`color`).
- По умолчанию текст не имеет тени (значение `none`).

```
text-shadow: none | offX offY blur color
```

```
.shadow {  
    text-shadow: 2px 2px 3px #176ab2;  
}  
</style>  
</head>  
  
<body>  
    <p class="shadow">Perferendis id facilis te  
iure quaerat modi ut, culpa, ratione, optio  
vero voluptates quibusdam aliquid sint cons  
dolores. Esse excepturi delectus, recusanda
```



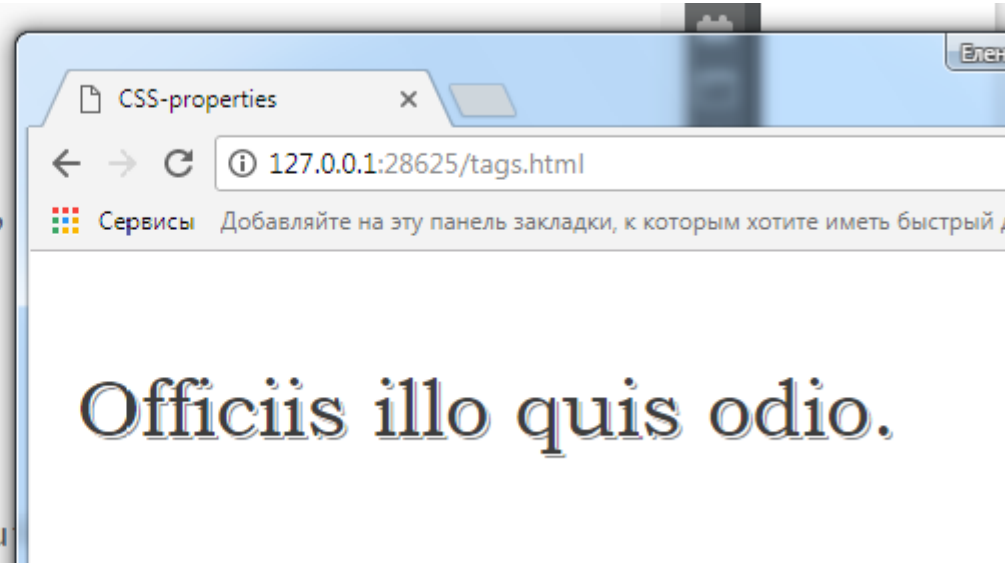
## CSS-свойства для форматирования текста

- Теней можно добавлять несколько, перечисляя их параметры через запятую.

```
text-shadow: offX1 offY1 blur1 color1, offX2 offY2
            blur2 color2, offX3 offY3 blur3 color3;
```

```
.shadow2 {
  padding: 15px;
  color: #333;
  font-size: 2.5em;
  text-shadow: 1px 1px white,
              2px 2px #777;
}
</style>
</head>

<body>
  <p class="shadow2">Officiis illo qu
```



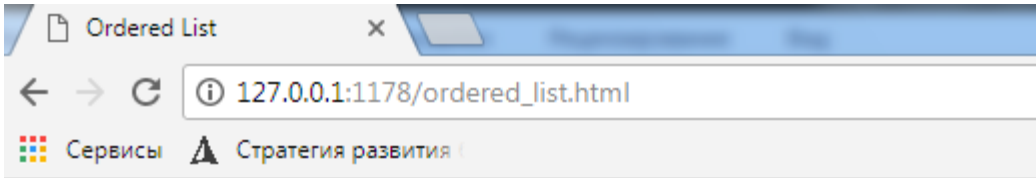
# Списки

• На множестве страниц в интернете используют перечисление каких-либо вариантов услуг или товаров. Для такого перечисления отлично подходят списки. В HTML их существует 3 вида:

- списки определений,
- Упорядоченные (нумерованные)
- неупорядоченные (маркированные).
- С помощью списков можно упорядочить и систематизировать различные виды данных и представить их в удобном для пользователя виде.

# Упорядоченный (нумерованный) список

- Упорядоченный (нумерованный) список представляет собой блок с элементами, каждый из которых по умолчанию имеет номер в виде арабской цифры



## Результат проекта

1. Работающая новая стратегия, новые идеи, новые продукты
2. Трансформация бизнеса и команды
3. Амбициозные и достижимые цели бизнеса и команды
4. Фокусировка и скоординированность действий
5. Освоение методологии разработки стратегии
6. Рост, развитие и капитализация бизнеса

```
<ol>
  <li> Работающая новая стратегия, новые идеи,
    новые продукты </li>
  <li> Трансформация бизнеса и команды </li>
  <li> Амбициозные и достижимые цели бизнеса
    и команды </li>
  <li> Фокусировка и скоординированность
    действий </li>
  <li> Освоение методологии разработки
    стратегии </li>
  <li> Рост, развитие и капитализация бизнеса </li>
</ol>
```

- В html-разметке нумерованный список формируется с помощью тега `<ol>` (от англ. *ordered list*) и вложенных в него тегов `<li>` (от англ. *list item*).

- Вместо арабских цифр мы можем указать в качестве нумерации римские цифры или буквы латинского алфавита, причем как строчные, так и прописные, а также указать нумерацию на основе греческого, армянского или грузинского языка.

- Для этого в HTML4 существовал атрибут `type`, который на данный момент является устаревшим. В HTML5 вместо него принято указывать тип списка с помощью стилей, а именно с помощью css-свойства `list-style-type`:

```
list-style-type: decimal (по умолчанию, можно не
указывать) | decimal-leading-zero | armenian |
georgian | lower-alpha | lower-greek | lower-latin |
lower-roman | upper-alpha | upper-latin |
upper-roman;
```



# Упорядоченный (нумерованный) список

```
list-style-type: decimal-leading-zero;
```

01. Работающая новая стратегия, новые идеи, новые продукты
02. Трансформация бизнеса и команды
03. Амбициозные и достижимые цели бизнеса и команды
04. Фокусировка и скоординированность действий
05. Освоение методологии разработки стратегии
06. Рост, развитие и капитализация бизнеса

## Наша компания предоставляет такие услуги

```
list-style-type: lower-alpha; (lower-latin)
```

- а. Зеленый офис
- б. Зоны отдыха
- в. Медицинское страхование
- г. Мультиспортивный абонемент
- д. Спортзал
- е. Дисконтная программа
- ж. Корпоративные праздники
- з. Корпоративные тренинги
- и. Кухня

## Формат разработки

```
list-style-type: upper-alpha;
```

- А. Стратегическая сессия
- Б. Стратегический аудит
- С. Межсессионная разработка
- Д. Серия интервью с командой
- Е. Рыночные исследования

```
list-style-type: upper-roman;
```

- I. Куда стремиться?
- II. На каком поле играть?
- III. Каким образом добиться победы?
- IV. В чем нужно быть сильнее?
- V. Какие управленческие системы нужно внедрить?

## Почему стоит путешествовать с нами по Грузии

```
list-style-type: georgian;
```

- ა. Готовые туры, разработанные знатоками Грузии
- ბ. Уже более 10 лет мы работаем в сфере туризма
- გ. Грузинско-русская команда
- დ. Свой автопарк и свои водители
- ე. Своя команда обученных гидов
- ვ. Дружелюбный и гибкий персонал
- ზ. Мы на связи всегда (24/7)

## Города и популярные курорты Армении

```
list-style-type: armenian;
```

- Ա. Ереван, столица Армении
- Բ. Гюмри, культурная столица Армении
- Գ. Цахкадзор, горнолыжный курорт Армении
- Դ. Шуши, культурный центр Арцаха
- Ե. Джермук, высокогорный курорт

## Почему стоит купить недвижимость в компании D

```
list-style-type: lower-greek;
```

- α. Более 20 лет на рынке недвижимости. Эти годы позволили нам приобрести
- β. Наша компания является застройщиком с огромным опытом работы в сфере чужих объектов и их законность.
- γ. Большой выбор недвижимости в Греции в сегменте эконом-класса.
- δ. Весь персонал компании говорит на английском языке.
- ε. Сопровождение клиента на всех этапах приобретения недвижимости, как и
- ζ. Возможность размещения в собственных апартаментах компании в Диони

## Расшифровка значений css-свойства list-style-type

- арабские цифры (1, 2, 3, ...) — decimal;
- арабские цифры с ведущим нулём для цифр первой десятки (01, 02, 03, ..., 09) — decimal-leading-zero;
- армянская нумерация — armenian;
- грузинская нумерация — georgian;
- прописные латинские буквы (A, B, C, ...) — upper-alpha или upper-latin;
- строчные латинские буквы (a, b, c, ...) — lower-alpha или lower-latin;
- римские цифры в верхнем регистре (I, II, III, ...) — upper-roman;
- римские цифры в нижнем регистре (i, ii, iii, ...) — lower-roman;
- строчные греческие буквы — lower-greek.

# Разрыв списка

- Нумерованные списки имеют также дополнительные атрибуты: `start` для тега `<ol>` и `value` для тега `<li>`. В этом случае список начинается не с цифры 1, а с той, которая указана в атрибуте `start`. Как правило, этот атрибут бывает необходим, если список нужно разорвать каким-либо пояснением или картинкой. Следует отметить, что используется он крайне редко.

## Требования к кандидату на должность в компании IT-Softwear

На собеседовании с кандидатом на вакантную должность в первую очередь оцениваются такие показатели:

- Знания и умения кандидата
- Образование
- Опыт работы в IT
- Навыки работы в команде

После успешного прохождения собеседования, кандидату будет предложены следующие тесты:

- Тест на определение уровня владения английским языком
- Внешняя сертификация
- Оценка знаний
- Оценка эффективности работы в условиях сжатых сроков

```
<p>На собеседовании с кандидатом на вакантную должность
в первую очередь оцениваются такие показатели:</p>
```

```
<ol>
  <li>Знания и умения кандидата</li>
  <li>Образование</li>
  <li>Опыт работы в IT</li>
  <li>Навыки работы в команде</li>
</ol>
```

```
<p>После успешного прохождения собеседования,
кандидату будет предложены следующие тесты:
</p>
```

```
<ol start="5">
  <li>Тест на определение уровня владения английским
языком</li>
  <li>Внешняя сертификация</li>
  <li>Оценка знаний</li>
  <li>Оценка эффективности работы в условиях
установки сжатых сроков</li>
</ol>
```

## Смена порядка нумерации списка

- Так же редко используется атрибут value для элемента li, который меняет порядок нумерации, начиная с той цифры, которая указана в качестве значения атрибута.
- Этот атрибут имеет приоритет перед атрибутом start, если они указаны в одном и том же списке. Кроме того, эти 2 атрибута применяются к любой нумерации (латинскими буквами, римскими цифрами и т.п.). В примере выделены красным цветом те элементы li, у которых есть атрибут value

### Акции, бонусные программы, мероприятия

1. система раннего бронирования (РБ);
2. «горящие туры» по спецпредложениям (СП1-СП4);
3. специальные предложения от партнеров (СПП);
10. система скидок постоянным и корпоративным клиентам;
11. помощь в оформлении банковского кредита на путешествие;
12. регулярно проводятся акции и бонусные программы;
30. «наши друзья – ваши друзья» - мы предлагаем вам лояльность и различные льготные условия при приобретении услуг у наших партнеров;
31. регулярные мероприятия и дни открытых дверей, посвященные туристическим новинкам и открытию туристического сезона!
32. поощрение клиентов подарочными сертификатами.

```
<ol>
  <li>система раннего бронирования (РБ);</li>
  <li>"горящие туры" по спецпредложениям
    (СП1-СП4);</li>
  <li>специальные предложения от партнеров (СПП);</li>
  <li value="10">система скидок постоянным
    и корпоративным клиентам;</li>
  <li>помощь в оформлении банковского кредита
    на путешествие;</li>
  <li>регулярно проводятся акции и бонусные
    программы;</li>
  <li value="30">"наши друзья – ваши друзья" -
    мы предлагаем вам лояльность ...;</li>
  <li>регулярные мероприятия и дни открытых дверей,
    посвященные туристическим новинкам и открытию
    туристического сезона! </li>
  <li>поощрение клиентов подарочными сертификатами.</li>
</ol>
```

# Изменение порядка нумерации на обратный

- Еще один малоиспользуемый атрибут для тега `<ol>` — `reversed` — изменяет порядок нумерации списка на обратный. Т.е. вместо 1, 2, 3, 4 будет 4, 3, 2, 1. Например, так:

## Обратный счет для детей

10. Десять
9. Девять
8. Восемь
7. Семь
6. Шесть
5. Пять
4. Четыре
3. Три
2. Два
1. Один

```
<ol reversed>
  <li>Десять</li>
  <li>Девять</li>
  <li>Восемь</li>
  <li>Семь</li>
  <li>Шесть</li>
  <li>Пять</li>
  <li>Четыре</li>
  <li>Три</li>
  <li>Два</li>
  <li>Один</li>
</ol>
```

# Маркированный список

- Маркированный список отличается от нумерованного тем, что вместо цифр имеет маркеры (по умолчанию в виде черных кружочков). Также маркированный список называется неупорядоченным (unordered list), т.к. нумерация в нем не нужна.
- Формируется маркированный список аналогично нумерованному двумя тегами: `<ul>` (от англ. *Unordered list*) и уже известным вам `<li>`

## Категории

- Биографии (255)
- Боевики (1581)
- Вестерны (76)
- Военные (237)
- Детективы (627)
- Документ (64)
- Драмы (2814)
- Исторические (237)
- Комедии (1983)

```
<ul>
  <li>Биографии (255)</li>
  <li>Боевики (1581)</li>
  <li>Вестерны (76)</li>
  <li>Военные (237)</li>
  <li>Детективы (627)</li>
  <li>Документ (64)</li>
  <li>Драмы (2814)</li>
  <li>Исторические (237)</li>
  <li>Комедии (1983)</li>
</ul>
```

- С помощью устаревшего атрибута `type` ранее определялся вид маркера, который на данный момент заменяется css-свойством `list-style-type`:

```
list-style-type: disc | circle | square | none
```

- disc — темный кружок (значение по умолчанию, можно не указывать);
- circle — пустой кружок;
- square — темный квадрат;
- none — маркеры (нумерация) отсутствует.

# Примеры маркированного списка

## Responsibilities:

```
list-style-type: disc; (default)
```

- Lead in the development of hardening the web application in protecting PII (Personally Identifying Information)
- Work side-by-side with product owners to create products and features to help protect users information while remaining seamless to the experience
- Find and build creative solutions to satisfy general audiences and weather enthusiasts while building in compliance for GDPR
- Something about integrating with external data sources/CMS to generate consent flow screens
- Write fluid/liquid/responsive HTML and CSS that works across all three screens (phone, tablet, and desktop)
- Designing and implementing solutions collaboratively to add value to software development efforts in the most cost efficient way. Must be able to effectively communicate JavaScript/HTML/CSS designs to other team members for they could implement the design.
- Ability to define, document and write Unit Test for front-end web technologies
- Create custom JavaScript libraries and utilize APIs
- Knowledge and experience in any JavaScript frameworks, including Angular
- Strong aptitude for both giving and taking direction
- Strong documentation skills to work properly with the general audience (not necessarily technical)

## Requirements:

```
list-style-type: circle;
```

- Core aspects of software development using HTML5, CSS and JavaScript
- 3+ years experience in web applications development
- Strong OOP and Functional software design ability. Able to develop systems that meet architectural objectives including reusable, scalable code
- Experience with Angular + NodeJS
- Exposure to build tools like Webpack and Jenkins
- Working knowledge of client- and server-side Javascript
- Ability to care about his/her products and the people who use them
- Ability to follow the timeline and meet deadlines
- Analytical and detail oriented, with the ability to prioritize, execute and deliver projects on time

## As a Plus:

```
list-style-type: none;
```

Github profile and contributed projects  
Development projects and code samples you can demo



# Список определений

- Списки определений нужны в тех сайтах, которые посвящены, как правило, обучению. Например, отличный сайт [webref.ru](http://webref.ru) использует списки определений для указания атрибутов тегов или вариантов свойств.
- Создается список определений с помощью 3-х тегов:
  - **dl** — definition list — основной, или родительский тег;
  - **dt** — definition termin — вложенный тег;
  - **dd** — definition description — вложенный тег.

## Атрибуты

<b>type</b>	Устанавливает вид маркера списка.
<b>reversed</b>	Нумерация в списке становится по убыванию (3,2,1). <a href="#">HTML5</a>
<b>start</b>	Задаёт число, с которого будет начинаться нумерованный список. <a href="#">HTML5</a>

Также для этого элемента доступны [универсальные атрибуты](#) и [события](#).

```
its Console Sources Network Performance Memory Application Security Audits
▼ <dl class="param">
  ▼ <dt>
    <a href="/html/ol/type">type</a> == $0
  </dt>
  <dd>Устанавливает вид маркера списка.</dd>
  ▼ <dt>
    <a href="/html/ol/reversed">reversed</a>
  </dt>
  ▼ <dd>
    "Нумерация в списке становится по убыванию (3,2,1). "
    ▶ <span class="attr-HTML5">...</span>
  </dd>
  ▼ <dt>
    <a href="/html/ol/start">start</a>
```

Код списка выглядит так:

```
<dl>
  <dt>Термин 1</dt>
  <dd>Описание термина 1</dd>
  <dt>Термин 2</dt>
  <dd>Описание термина 2</dd>
  <dt>Термин 3</dt>
  <dd>Описание термина 3</dd>
</dl>
```



# Список определений

## Интернет

совокупность сетей, применяющих единый протокол обмена (точнее, обширное семейство из сотен протоколов) для передачи информации

## Локальная сеть

соединение нескольких компьютеров при помощи технических и программных средств, которые позволяют объединить файловые системы входящих в нее компьютеров

## Глобальная сеть

такое соединение компьютеров, при котором допускается использование информации, физически находящейся на других компьютерах.

## Провайдер сетевых услуг

владелец конкретной сети, с которой клиент находится в юридических отношениях

- На скриншоте видно, что по сравнению с термином в элементе dt элемент dd имеет смещение от левого края окна браузера на 40px вправо. Этот отступ определяется свойством `margin-left` или в браузере Хром — `webkit-marginstart`.
- За счет этого определение визуально отличается от термина и привлекает к себе внимание

```
dd {                                user agent stylesheet
  display: block;
  -webkit-margin-start: 40px;
}
```

# Список определений

- Список определений хорошо форматировать с помощью css. Для сайта, который использует много вариантов таких списков, имеет смысл назначить css-свойства для селектора тега, не прибегая ни к использованию селекторов классов, ни селекторов id. Например, список на скриншоте выше можно отформатировать так

## Интернет

совокупность сетей, применяющих единый протокол обмена (точнее, обширное семейство из сотен протоколов) для передачи информации

## Локальная сеть

соединение нескольких компьютеров при помощи технических и программных средств, которые позволяют объединить файловые системы входящих в нее компьютеров

## Глобальная сеть

такое соединение компьютеров, при котором допускается использование информации, физически находящейся на других компьютерах.

## Провайдер сетевых услуг

владелец конкретной сети, с которой клиент находится в юридических отношениях

## Код css-стилей:

```
<style>
  dt {
    font-weight: bold;
    color: #036011;
  }
  dd {
    color: #333;
    margin-top: 10px;
    margin-bottom: 10px;
  }
</style>
```

## Многоуровневые, или вложенные списки

- Многоуровневым списком является список, внутри элементов `<li>` которого размещается список второго уровня. Т.е. второй список вкладывается внутрь первого, поэтому такие списки называются еще вложенными (файл *inner\_list.html*).
- В многоуровневом списке можно использовать как упорядоченные, так и неупорядоченные списки, а также списки определений, чередуя их в любой последовательности. Важно, чтобы вложенный список находился именно в элементе `<li>` родительского списка. Количество уровней в списках может быть любым, хотя больше 3-х уровней вложенности бывает крайне редко.
- Пример правильного кода:

```
<ol>
  <li>Площади
    <ul>
      <li>Дворцовая площадь</li>
      <li>Рыночная площадь</li>
    </ul>
  </li>

  <li>Замки
    <ul>
      <li>Королевский замок</li>
      <li>Виланувский дворец</li>
    </ul>
  </li>

  <li>Старе Място</li>
```

```
    <li>Музеи
      <ul>
        <li>Исторический музей</li>
        <li>Музей Войска Польского</li>
        <li>Музей карикатуры</li>
        <li>Музей Марии Кюри</li>
        <li>Музей Шопена</li>
        <li>Национальный музей</li>
      </ul>
    </li>

    <li>Краковское предместье</li>
  </ol>
```

- Площади
  - Дворцовая площадь
  - Рыночная площадь
- Замки
  - Королевский замок
  - Виланувский дворец
- Старе Място
- Музеи
  - Исторический музей
  - Музей Войска Польского
  - Музей карикатуры
  - Музей Марии Кюри
  - Музей Шопена
  - Национальный музей
- Краковское предместье

# Многоуровневые, или вложенные списки

- Вариант ошибочного кода

```
<ol>
  <li>Площади</li>
  <ul>
    <li>Дворцовая площадь</li>
    <li>Рыночная площадь</li>
  </ul>

  <li>Замки</li>
  <ul>
    <li>Королевский замок</li>
    <li>Вилянувский дворец</li>
  </ul>

  <li>Старе Место</li>
```

```
<li>Музеи</li>
<ul>
  <li>Исторический музей</li>
  <li>Музей Войска Польского</li>
  <li>Музей карикатуры</li>
  <li>Музей Марии Кюри</li>
  <li>Музей Шопена</li>
  <li>Национальный музей</li>
</ul>
<li>Краковское предместье</li>
</ol>
```

- Ошибка заключается в том, что вложенный список вставлен не внутрь элемента `<li>`, а между двумя соседними элементами, а это недопустимо с точки зрения валидатора:

- Error** Element `ul` not allowed as child of element `ol` in this context. (Suppressing further errors from this subtree.)

From line 10, column 9; to line 10, column 12

```
> <ul>
```

Contexts in which element `ul` may be used:

Where [flow content](#) is expected.

Content model for element `ol`:

Zero or more `li` and [script-supporting elements](#).

- Error** Element `ul` not allowed as child of element `ol` in this context. (Suppressing further errors from this subtree.)

From line 16, column 9; to line 16, column 12

```
> <ul>
```

Contexts in which element `ul` may be used:

Where [flow content](#) is expected.

Content model for element `ol`:

Zero or more `li` and [script-supporting elements](#).

## CSS-стили для списков

- CSS-свойства, о которых пойдет речь ниже, назначают обычно для маркированных или нумерованных списков, причем указывать их можно как для родительских элементов (ul, ol), так и для вложенных (li). Одно из стилевых свойств списков мы уже рассмотрели. Это свойство list-style-type, определяющее вид маркера или цифры:

```
list-style-type: decimal | decimal-leading-zero |  
armenian | georgian | lower-alpha | lower-greek |  
lower-latin | lower-roman | upper-alpha | upper-latin |  
upper-roman | disc | circle | square | none;
```

- В предыдущих примерах мы разделяли его по значениям, которые были разными для нумерованных и маркированных списков. Вы можете назначить любое из значений для любого вида списка, однако логика html-разметки тогда потеряется. Довольно часто маркеры или цифры в списках «прячут», используя значение list-style-type: none. Например, так делают при форматировании меню.

```
Аббревиатуры Emmet: lst, lstdlz, lstlr, lstur, lstd,  
lstc, lsts, lstn
```

- Второе свойство — это позиция цифр или маркеров. Определяется свойством list-style-position с такими значениями:

```
list-style-position: outside | inside | inherit
```

## CSS-стили для списков

- По умолчанию у всех списков `list-style-position: outside`, поэтому использовать это значение вы будете редко. Оно подразумевает, что маркеры (цифры) размещаются слева от текста элемента списка. Значение `list-style-position: inside` размещает маркеры (цифры) внутри текста элемента списка.

### 1. `list-style-position: inside`

- Консультативная помощь пациенту на каждом этапе реабилитационного процесса,
- Жесткие стандарты проверки качества медицинских и дополнительных услуг.
- Выделение бюджета на инвестиции в образование и повышение квалификации сотрудников центра,
- Надежность результатов работы специалистов — основной принцип оказания медицинской услуги.

### • `list-style-position: inside;`

- стойкое и долгосрочное улучшение качества медицинских услуг
- разработка терапевтических протоколов с учетом расширяющихся видов патологий пациентов центра
- гибкость и способность быстро адаптироваться к любым клинически сложным случаям
- научно-исследовательская работа и внедрение инноваций
- разработка и внедрение систем реабилитационных манипуляций и новых технологий

### 1. `list-style-position: outside; (default)`

- Консультативная помощь пациенту на каждом этапе реабилитационного процесса,
- Жесткие стандарты проверки качества медицинских и дополнительных услуг.
- Выделение бюджета на инвестиции в образование и повышение квалификации сотрудников центра,
- Надежность результатов работы специалистов — основной принцип оказания медицинской услуги.

### • `list-style-position: outside; (default)`

- стойкое и долгосрочное улучшение качества медицинских услуг
- разработка терапевтических протоколов с учетом расширяющихся видов патологий пациентов центра
- гибкость и способность быстро адаптироваться к любым клинически сложным случаям
- научно-исследовательская работа и внедрение инноваций
- разработка и внедрение систем реабилитационных манипуляций и новых технологий

- Значение `inherit` подразумевает, что вложенный список наследует это значение от родительского элемента. Поскольку вложенные элементы списков обычно имеют такие же свойства, как и родительские, поэтому для них используют это значение не так уж часто. Аббревиатуры Emmet: `lsp`, `lspi`, `lspo`

## Использование CSS-стили для списков

- Третьим, достаточно популярным свойством, является `list-style-image`, которое позволяет установить изображение в качестве маркера списка:

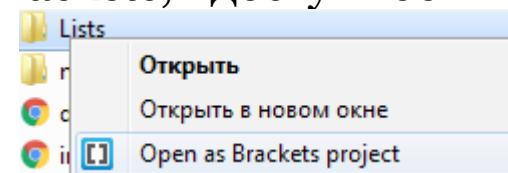
```
list-style-image: none |  
                    url(путь к файлу изображения.расш);  
list-style-image: url(markers/arrow.gif);
```

- Или

```
list-style-image: url(http://mysite.com/images/ arrow.gif);
```

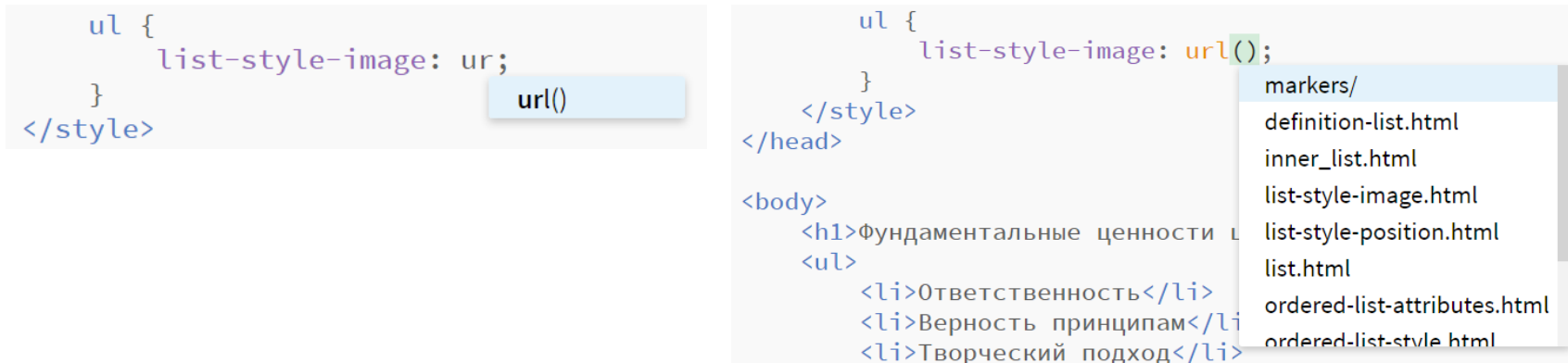
- В первом примере путь к файлу указывается относительно местоположения html-файла и папки с изображениями для маркеров (они должны находиться в общей папке). Во втором — путь указывается с учетом доменного имени вашего сайта. Поскольку на данном этапе сайт у вас вряд ли существует, то единственным вариантом для аудиторных и домашних заданий будет первый пример.

- Что касается текстового редактора Brackets, то в нем существуют очень удобные подсказки, позволяющие легко указать путь к файлу изображения-маркера, а также посмотреть при наведении на строку пути к изображению, как выглядит файл, и его размер. Важным условием для этих подсказок является открытие папки в качестве проекта в Brackets, доступное из контекстного меню проводника.



## CSS-стили для списков

- Как только вы написали свойство `list-style-image: url()` и оставили курсор в скобках, Brackets начинает подсказывать вам, показывая папки, которые есть в вашем проекте, и файлы в этих папках:



- В списке будут также и все файлы, которые есть в текущей папке, поэтому будьте внимательны. Свой выбор вы можете подтвердить, нажав клавишу `Enter` или сделав клик левой кнопкой мыши.

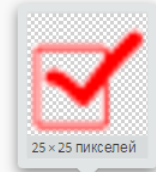




## CSS-стили для списков

- При наведении на текст в скобках вы увидите, как выглядит изображение и каков его размер (на скриншоте 25×25px):
- Для того чтобы поменять изображение на любое другое, нужно просто поставить курсор после слэша и нажать Ctrl+пробел:
- Для замены одного изображения другим нужно просто выбрать его и нажать Enter — и название файла будет автоматически заменено, даже если не было выделено. При выборе различных файлов для перемещения по списку можно использовать не только мышь, но и клавиши со стрелками вверх и вниз. В результате получим следующий вид списка

```
<style>
  body {
    font-family: sans-serif;
    width: 80%;
    margin: auto;
  }
  ul {list-style-image: url(markers/check.png);}
</style>
```



```
ul {
  list-style-image: url(markers/check.png);
}
</style>
```

markers/flags/  
 markers/arr.gif  
 markers/arrb.gif  
 markers/arrow.gif

## Фундаментальные ценности центра

- ✓ Ответственность
- ✓ Верность принципам
- ✓ Творческий подход
- ✓ Эффективность
- ✓ Качество
- ✓ Профессионализм
- ✓ Новаторство
- ✓ Командная работа

## CSS-стили для списков

- **Примечание:** свойство *list-style-image* имеет приоритет перед *list-style-type*, т.е. изображение всегда будет перекрывать маркер или номер элемента списка.
- В случае если нужно убрать в одном из элементов списка назначенный маркер-изображение, используют значение *none*:

```
list-style-image: none;
Аббревиатура Emmet: lsi, lisi, lisin
```

- Универсальное свойство *list-style* позволяет задать через пробел все свойства из вышеперечисленных, т.е. указать и стиль маркера, и его позицию, и изображение. Перечисление идет в такой последовательности:

```
list-style: list-style-type list-style-position
list-style-image;
```

- Например:

```
list-style: upper-latin inside url(markers/somepic.png);
```














### Фундаментальные ценности центра

- ... Ответственность
- ... Верность принципам
- ... Творческий подход
- ... Эффективность
- ... Качество
- ... Профессионализм
- ... Новаторство
- ... Командная работа

## CSS-стили для СПИСКОВ

```
list-style: url(markers/actor-icon.png);
```

### Список актеров в фильме "Лучшее предложение"

-  Джеффри Раш — Вирджил Олдман — главная роль
-  Джим Стёрджесс — Роберт — главная роль
-  *Сильвия Хукс — Клара — главная роль*
-  Дональд Сазерленд — Билли
-  Филип Джексон — Фрэд
-  Дермот Кроули — Ламберт
-  *Лия Кебеде — Сара*
-  Кируна Стамелл — девушка в баре
-  Джон Бенфилд — бармен
-  Клаус Таубер — ассистент Вирджила
-  Максимилиан Дирр — ассистент Вирджила
-  Ген Сето — домоправитель
-  *Сильвия Де Фанти — менеджер галереи*

## Использование псевдоэлемента `::before` для стилей списков

- Зачастую хочется изменить нумерацию в упорядоченном списке, задав для него скобки после номера, а для неупорядоченного списка — тире вместо маркера. Для этого необходимо отключить стандартную нумерацию или маркеры с помощью свойства `list-style-type: none`. А затем использовать псевдоэлемент `::before` для элемента `<li>`.
- Псевдоэлементы имеют приставку псевдо- перед знакомым вам уже словом «элемент», потому что изначально в html-коде их нет. Кстати, их не будет в коде вообще, потому как появляются они исключительно с помощью CSS.
- Псевдоэлементов существует несколько, а именно `::first-line`, `::first-letter`, `::selection`, `::before` и `::after`. Причем для них доступен синтаксис и с двумя двоеточиями перед наименованием (CSS3), и с одним (CSS2), т.к. в версии CSS3 было принято решение отделить псевдоэлементы от псевдоклассов, добавив к псевдоэлементам второе двоеточие. Поэтому не удивляйтесь, если вы увидите разное их написание.
- По поводу `:before` и `:after` можно сказать, что эти псевдоэлементы позволяют добавить некий контент (текстовое или визуальное содержимое) перед текстом элемента или в конце элемента, т.е. после содержимого элемента. Причем текст псевдоэлемента выделяться не будет, а CSS-свойства для него можно задать такие же, как и для обычных элементов.
- По умолчанию `:before` и `:after` являются строчными элементами, и имеют такой же шрифт, цвет текста и начертание, что и у основного элемента.

# Использование псевдоэлемента ::before для стилей списков

- Например, псевдоэлементы :before и :after для абзаца, у которого в стилях указано курсивное начертание, тоже будут выведены курсивом:

```
p { font-style: italic; }
p:before, p:after { background-color: #ccc; }
p:before { content: 'Start'; }
p:after { content: "End"; }
```

*Start* Lorem ipsum dolor sit amet, consectetur adipiscing elit. Illo qui quod eius, vitae accusamus eveniet! Ipsum doloribus dolore sed numquam! *End*

*Start* Quas quia veniam autem delectus, non repellat consequatur, amet corporis ullam odit deserunt, ab itaque ipsam assumenda aliquid. Nulla, qui. *End*

- коде в свойстве content указан в кавычках (двойных или одинарных) текст, который будет выведен до и после текста абзаца.
- Примечание:** свойство content является обязательным для псевдоэлементов ::before ::after. Если это свойство не будет указано, даже с пустыми кавычками (" " или ' '), то на странице эти псевдоэлементы не отобразятся, какие бы еще свойства вы не писали для них.

# Использование псевдоэлемента ::before для стилей списков

- Что касается упорядоченных списков, то в них обычно используют только псевдоэлемент :before для изменения нумерации, указывая некое имя счетчика (например, number) в нескольких свойствах:

```
ol { list-style-type: none;
    counter-reset: number; }

ol li::before {
    content: counter(number) ') ';
    counter-increment: number; }
```

## Rotational motion

- 1) Angular momentum (Introduction)
- 2) Angular velocity
- 3) Centrifugal force
- 4) Centripetal force
- 5) Circular motion
- 6) Tangential velocity
- 7) Torque

- Для неупорядоченных списков в свойстве content указывают, например, текст с пробелом вместо маркера

```
ul { list-style-type: none;
    font-style: italic; }

ul li::before { content: '- ';
                color: #04b548; }
```

## Conservation of energy and momentum

- Conservation of energy
- Elastic collision
- Energy
- Inelastic collision
- Inertia
- Kinetic energy
- Moment of inertia
- Momentum
- Potential energy
- Rotational energy

# Оформление многоуровневых списков.

## Вложенные селекторы

- Для оформления многоуровневого списка недостаточно написать css-правила только для селектора элемента (ol, ul, li). Чаще всего для них (и не только) придется применять контекстные (или вложенные) селекторы. Они подразумевают, что css-свойства назначаются для тега, вложенного в другой тег, например, селектор

```
ol ul {color: red}
```

подразумевает, что красного цвета будет только тот маркированный список (ul), который вложен в нумерованный список (ol).

- Количество селекторов может быть любым, но следует понимать, что css-правила назначаются для самого последнего селектора, но с учетом его вложенности во все предыдущие. Т.е. читается такой селектор справа налево.
- Например, код

```
.season ul li { font-style: italic; }
```

говорит о том, что курсивное начертание будет только у элементов li маркированного списка ul, который находится в элементе с классом season.

# Оформление многоуровневых списков.

## Вложенные селекторы

- Например, необходимо оформить список с перечислением времен года так, как показано на изображении. Если проанализировать его внешний вид, то станет понятно, что это нумерованный список с 2-мя уровнями вложенности, причем вложенный список может быть как нумерованным, так и маркированным, т.к. цифры или маркеры все равно будут перекрыты изображениями. Поскольку нумерация не будет видна, имеет смысл сделать вложенный список маркированным.

## Времена года

### 1. Зима

- ❄ Декабрь
- ❄ Январь
- ❄ Февраль

### 2. Весна

- 🌱 Март
- 🌱 Апрель
- 🌱 Май

### 3. Лето

- ☀ Июнь
- ☀ Июль
- ☀ Август

### 4. Осень

- 🍂 Сентябрь
- 🍂 Октябрь
- 🍂 Ноябрь



# Оформление многоуровневых списков.

## Вложенные селекторы

- Поэтому начальная html-разметка будет такой:

```
<h2>Времена года</h2>

<ol>

  <li>Зима
    <ul>
      <li>Декабрь</li>
      <li>Январь</li>
      <li>Февраль</li>
    </ul>
  </li>

  <li>Весна
    <ul>
      <li>Март</li>
      <li>Апрель</li>
      <li>Май</li>
    </ul>
  </li>

  <li>Лето
    <ul>
      <li>Июнь</li>
      <li>Июль</li>
      <li>Август</li>
    </ul>
  </li>

  <li>Осень
    <ul>
      <li>Сентябрь</li>
      <li>Октябрь</li>
      <li>Ноябрь</li>
    </ul>
  </li>

</ol>
```

### Времена года

1. Зима
  - Декабрь
  - Январь
  - Февраль
2. Весна
  - Март
  - Апрель
  - Май
3. Лето
  - Июнь
  - Июль
  - Август
4. Осень
  - Сентябрь
  - Октябрь
  - Ноябрь

### Времена года

1. Зима
  - Декабрь
  - Январь
  - Февраль
2. Весна
  - Март
  - Апрель
  - Май
3. Лето
  - Июнь
  - Июль
  - Август
4. Осень
  - Сентябрь
  - Октябрь
  - Ноябрь

# Оформление многоуровневых списков.

## Вложенные селекторы

- Теперь нужно сделать так, чтобы шрифт родительского списка `<ol>` был жирным, а дочернего списка `<ul>` — обычным, или нормальным. Поскольку элементы `<li>` наследуют свойства родительского элемента, назначим для селектора `ol` следующее свойство:

```
ol { font-weight: 600; }
```

- И получим в результате список, в котором все элементы получили жирное начертание. Чтобы вложенный список «вернул» нормальное начертание, необходимо указать следующее css-правило:

```
ul { font-weight: 200; }
```

- Теперь список имеет следующий вид: элементы родительского списка — с жирным начертанием, элементы вложенного списка — с обычным, или нормальным начертанием.

## Времена года

### 1. Зима

- Декабрь
- Январь
- Февраль

### 2. Весна

- Март
- Апрель
- Май

### 3. Лето

- Июнь
- Июль
- Август

### 4. Осень

- Сентябрь
- Октябрь
- Ноябрь

# Оформление многоуровневых списков.

## Вложенные селекторы

- Следующим шагом будет назначение цвета для различных блоков текста: синего — для месяцев зимы, зеленого — для весенних месяцев и т.д. Проще всего это сделать, задав класс или id каждому из элементов li родительского списка. Поскольку значения цветов будем использовать индивидуально для каждого блока месяцев, используем атрибут id:

```
<ol>
  <li id="winter">Зима
    <ul>
      <li>Декабрь</li>
      <li>Январь</li>
      <li>Февраль</li>
    </ul>
  </li>

  <li id="spring">Весна
    <ul>
      <li>Март</li>
      <li>Апрель</li>
      <li>Май</li>
    </ul>
  </li>

  <li id="summer">Лето
    <ul>
      <li>Июнь</li>
      <li>Июль</li>
      <li>Август</li>
    </ul>
  </li>
```

```
    <li id="autumn">Осень
      <ul>
        <li>Сентябрь</li>
        <li>Октябрь</li>
        <li>Ноябрь</li>
      </ul>
    </li>
  </ol>
```

```
#winter { color: rgb(0, 0, 200); }
#spring { color: rgb(0, 190, 0); }
#summer { color: #ffd24c; }
#autumn { color: #d96c00; }
```

### 1. Зима

- Декабрь
- Январь
- Февраль

### 2. Весна

- Март
- Апрель
- Май

### 3. Лето

- Июнь
- Июль
- Август

### 4. Осень

- Сентябрь
- Октябрь
- Ноябрь

# Оформление многоуровневых списков.

## Вложенные селекторы

- Внешний вид блоков изменился в соответствии с назначенными цветами. Последней является задача по определению маркеров в виде различных изображений. Можно было бы назначить каждому из 3-х элементов `<li>` атрибут класс и задать для него свойство `list-style-image`, но проще использовать здесь контекстный селектор такого вида:

```
#winter li { list-style-image: url(markers/snow.png); }
#spring li { list-style-image: url(markers/leaf.png); }
#summer ul { list-style-image: url(markers/sun.png); }
#autumn ul { list-style-image: url(markers/rain.png); }
```

- Обратите внимание, что в первых двух селекторах мы назначали маркер-изображение для элемента `li`, вложенного в определенный селектор `id`, а в третьем и четвертом — для элемента `ul`. При этом визуально разница незаметна. Произошло это по той причине, что свойства, назначенные для всего списка (`ul` в примере), наследуются его элементами `li`, а свойства для элементов `li` применяются к ним непосредственно.

### Времена года

#### 1. Зима

- ❄ Декабрь
- ❄ Январь
- ❄ Февраль

#### 2. Весна

- 🌱 Март
- 🌱 Апрель
- 🌱 Май

#### 3. Лето

- ☀ Июнь
- ☀ Июль
- ☀ Август

#### 4. Осень

- 🍂 Сентябрь
- 🍂 Октябрь
- 🍂 Ноябрь

**Спасибо за внимание.**