

HTML+CSS (верстка)

Лекция 1

Короткое введение

- Последний стандарт **HTML** – версия **4.01**
- **XHTML** – переработанный HTML в соответствии со стандартом XML
- Стандарты неоднозначны, не все моменты регламентированы
- **Придерживайтесь стандартов**, осторожнее с нестандартными решениями

- Элементы и теги

<название_тега >

<название_тега >содержание тега < / название_тега >

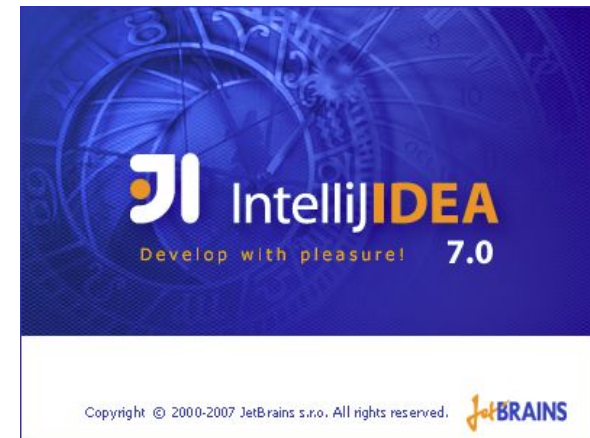
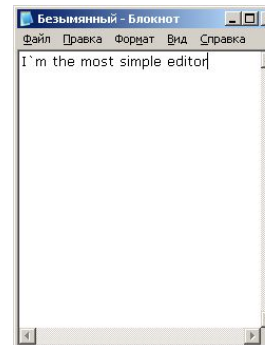
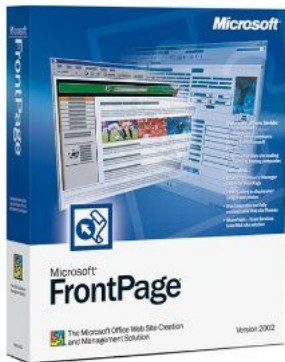
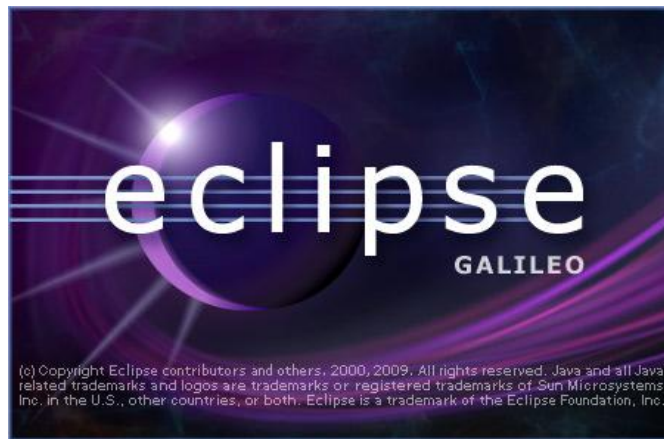
<p>Текст абзаца </p>

- HTML – это прежде всего **содержание**, данные.

HTML, как правило, только лишь советует как отображать документы.

Редакторы

- WYSIWYG или текстовые процессоры



Eclipse

- Свободный фреймворк для разработки модульных кроссплатформенных приложений
- **Плагины Eclipse** (Plug-in Development Environment)



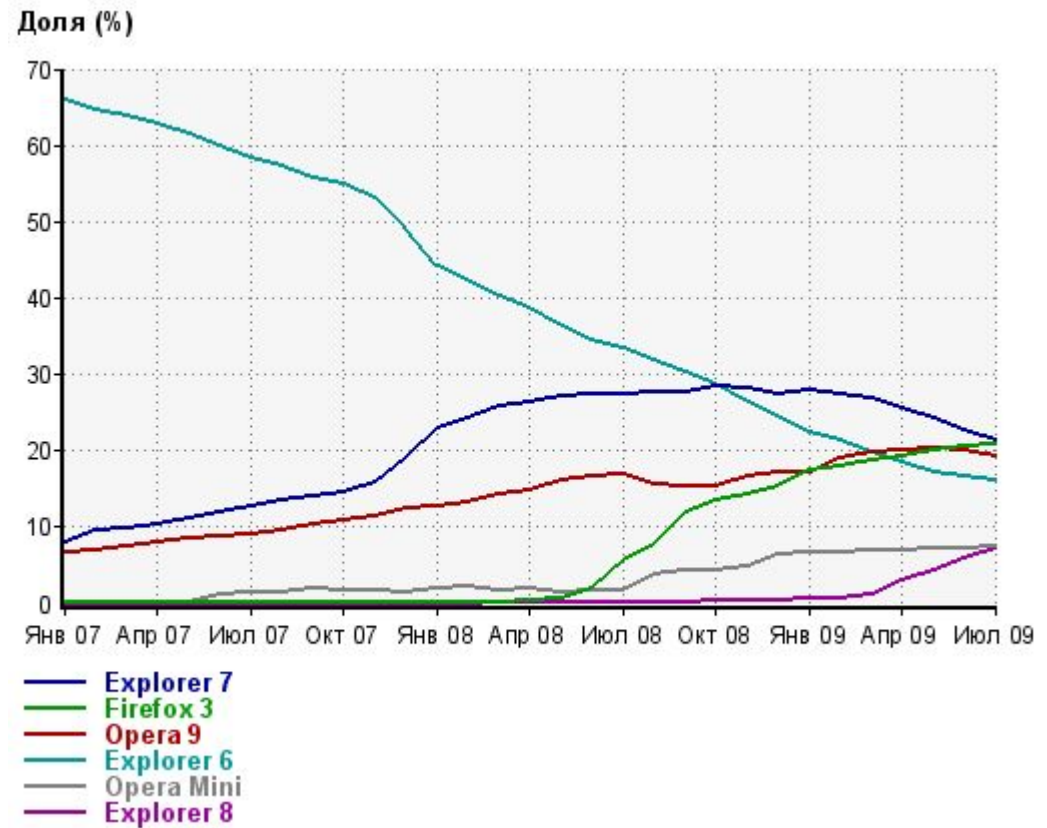
-  **Aptana** – отличные редакторы HTML и CSS

Преимущества Eclipse

- **Бесплатный** продукт с поддержкой
- Работа с проектами
- **Расширяемость и тонкая настройка**
- **Кроссплатформенность**
- **Отличные редакторы**
(автокомплит, шаблоны, навигация по коду...)
- Работа с разными языками программирования

Браузеры

- Internet Explorer
- Firefox
- Opera
- Safari
- Chrome



Популярность браузеров по
Liveinternet в рунете

Другие инструменты



Firebug – не заменим для разработки и отладки HTML, CSS, Javascript



Pixel Perfect – плагин для подложки рисунка дизайна при верстке



IE Developer Toolbar,
IETester,
DebugBar for IETester

Теги, атрибуты. Блочные и строковые элементы

- `<html>`, `<head>`, `<title>`
- `<body bgcolor="#A0BEC4" onload="alert('Loaded');"></body>`
- `<!-- комментарий -->`
- Семантика, логичность кода
(``, ``, `<blockquote>`, `<code>`, ...)
- Спецсимволы: `<` `>` ...
- `<p>`, `<h1>`, `<h2>`, `<h3>`, ..., `
`, `<hr>`
- `текст`
- ``
- Вложенность тегов

Обработка ошибок, DTD

- Отсутствующие теги – плохая практика

`<p>`Первый абзац `<p>`Второй абзац `<h2>`Заголовок

- Игнорирование лишних тегов

`<p>`Первый абзац `<p><p>`Второй абзац `<uktag>`

- XHTML – намного строже

- `
` - не правильно, `
` - правильно
- Регистр символов в названии тегов
- `текст <i>текст текст</i>` - не правильно
- Обязательные теги, например `<html>`

- Инструменты проверки кода на ошибки:
HTML – валидатор

- ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

# Текст

- Текст, абзацы, перевод строки
- Лишние пробельные символы не учитываются

```
<p align="left" class="log" lang="ru" id="loginfo" style="color:blue" title="Некоторая статистическая информация">Скачано 10Кб</p>
```

- Заголовки, значение заголовков
- **<br>** и **<nobr>** — все браузеры поддерживают, но...
- **<pre>** — блок предварительно форматированного текста (все пробелы, переводы строк, и т.д., монош. шрифт)
- Физическая и логическая разметка

# Шрифт

- Немного терминов
- Название шрифта, семейство (serif, sans-serif, monospace...)
- Размер
- Цвет
- Межсимвольное расстояние
- Курсив, жирность
- `<font size=5 color="#FFFFFF" face="Courier, Arial">Б</font>`уква



# Теги логической разметки текста

Тег	Описание	Отображение
<code>&lt;abbr&gt;</code>	Выделение аббревиатур, сокращенная форма слова, фразы	подч. пункт.
<code>&lt;acronym&gt;</code>	Акроним, сокращение из первых букв слов	подч. пункт.
<code>&lt;cite&gt;</code>	Библиографическая ссылка, например, название журнала	курсив
<code>&lt;code&gt;</code>	Выделение кода программы	монош. уменьш.
<code>&lt;dfn&gt;</code>	Термины, впервые появившиеся в тексте	курсив
<code>&lt;em&gt;</code>	Визуальное выделение, логическое ударение	курсив
<code>&lt;kbd&gt;</code>	Текст, набранный на клавиатуре, название клавиш	монош.
<code>&lt;samp&gt;</code>	Результат программы; часть, вырванная из контекста	монош.
<code>&lt;strong&gt;</code>	Логическое подчеркивание текста, более сильное	полуж.
<code>&lt;var&gt;</code>	Имя переменной, данные, вводимые пользователем	курсив

# Основные теги физической разметки

Тег	Описание
<code>&lt;b&gt;</code>	Эквивалент <code>&lt;strong&gt;</code> , не несущий определенного смысла
<code>&lt;big&gt;</code>	Увеличение размера шрифта (на 1 больше, чем у окружающего текста)
<code>&lt;i&gt;</code>	Эквивалент <code>&lt;em&gt;</code> , не несущий определенного смысла
<code>&lt;small&gt;</code>	Тег, обратный тегу <code>&lt;big&gt;</code>
<code>&lt;s&gt;</code>	Устарел, делает текст <del>перечеркнутым</del>
<code>&lt;sub&gt;</code>	Чуть меньший размер шрифта, выше уровня строки на половину. Возводит текст в <sup>верхний индекс</sup>
<code>&lt;sup&gt;</code>	Чуть меньший размер шрифта, выше уровня строки на половину. Возводит текст в <sub>нижний индекс</sub>
<code>&lt;tt&gt;</code>	Моноширинный шрифт
<code>&lt;u&gt;</code>	Устарел, подчеркнутый текст

Не гарантируется именно такое отображение

Содержимое вышеупомянутых тегов – любые элементы допустимые в тексте.

Употребляться могут везде, где применяются элементы, относящиеся к

# Цитаты, адреса

- `<blockquote>` и `<q>` — цитаты (длинная и короткая)

```
<blockquote cite="http://lib.ru/" lang="en" style="color:green;">Длиинная цитата</blockquote>
```

- `<address>`Адрес`</address>` - адрес

# Линейки

- `<hr align="center" color="#000000" size="3" width="50%">`
- Использование атрибутов не рекомендуется
- Часто используется для логического отделения информационных блоков



# Изображения в документе HTML

- Формат? Все зависит от браузеров, в HTML нет спецификаций
- Формат GIF
  - сжатие "без потерь"
  - 256 цветов максимум
  - чересстрочный (всплывающий) и нормальный формат
  - прозрачность
  - анимация

- **Формат JPEG**
  - 24 битная палитра (16 млн цветов)
  - потеря качества рисунка
- **Формат PNG**
  - **ЦВЕТОВЫЕ СХЕМЫ:**
    - **truecolor**
    - **grayscale**
    - **индексированная палитра (GIF-подобная) – PNG-8**
  - альфа канал на 254 уровня
  - улучшенное сжатие без потерь
  - двухмерное чередование
  - гамма-коррекция
- **Формат MNG**

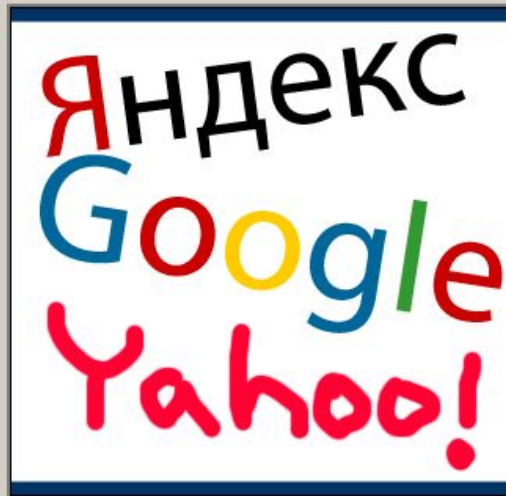


Исходное Оптимизация 2 варианта 4 варианта



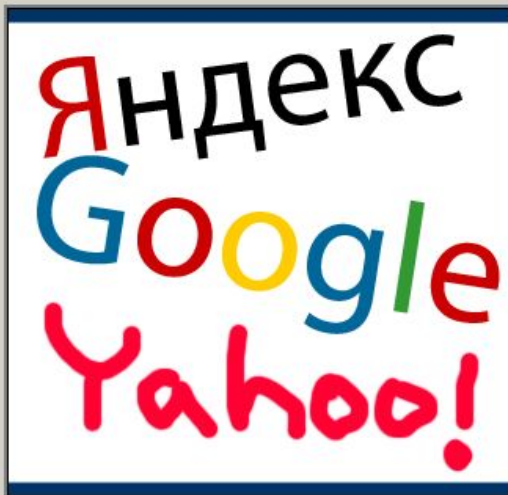
JPEG  
24,59K  
5 сек @ 56,6 кбит/с

качество: 60



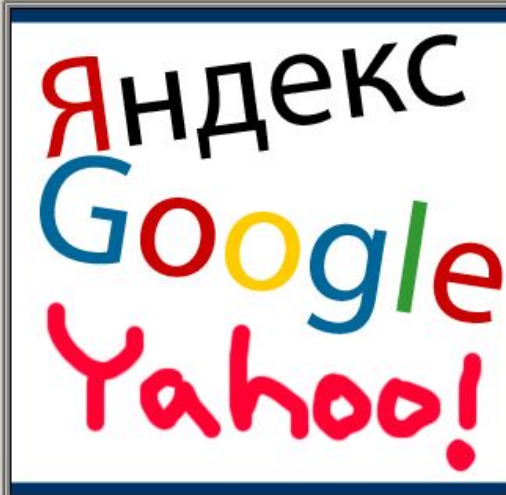
PNG-8  
10,41K  
3 сек @ 56,6 кбит/с

Дизеринг: 100%  
Перцепционная  
256 цвета



GIF  
12,36K  
3 сек @ 56,6 кбит/с

Дизеринг: 88%  
Селективная  
256 цвета



PNG-24  
19,2K  
4 сек @ 56,6 кбит/с

Набор: PNG-24

Прозрачность  Матовый:

Чересстрочно

Преобразовать в sRGB

Просмотр: Цвет монитора

Метаданные: Сведения об авт. правах и контакты

Таблица цветов

Размер изображения

Ш: 320 пикс. В: 310 пикс. Процент: 100 %

Качество: Бикубическая

Анимация

Параметры повторов: Навсегда

1 из 1

100% R: -- G: -- B: -- | Альфа: -- Шестнадц.: -- Индекс: --



Исходное    Оптимизация    2 варианта    4 варианта



JPEG  
57,69К  
11 сек @ 56,6 кбит/с

качество: 70



PNG-8  
73,54К  
14 сек @ 56,6 кбит/с

Дизеринг: 100%  
Перцепционная  
256 цвета



GIF  
80,91К  
16 сек @ 56,6 кбит/с

Дизеринг: 88%  
Селективная  
256 цвета



PNG-24  
209,5К  
39 сек @ 56,6 кбит/с

Набор: [Безимени]

PNG-24

Прозрачность    Матовый: [ ]

Чересстрочно

Преобразовать в sRGB

Просмотр: Цвет монитора

Метаданные: Сведения об авт. правах и контакты

Таблица цветов

Размер изображения

Ш: 320 пикс.    Проценты: 100 %

В: 310 пикс.    Качество: Бикубическая

Анимация

Параметры повторов: Навсегда

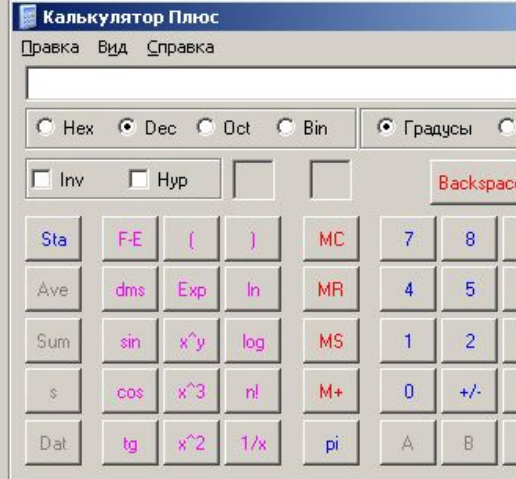
1 из 1

100%    R: --    G: --    B: --    |    Альфа: --    Шестнадц.: --    Индекс: --



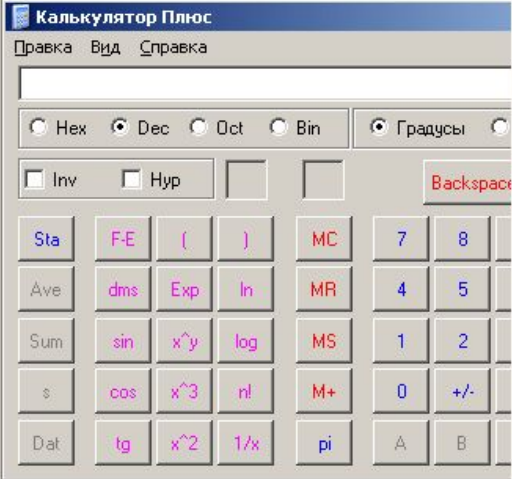


Исходное Оптимизация 2 варианта 4 варианта



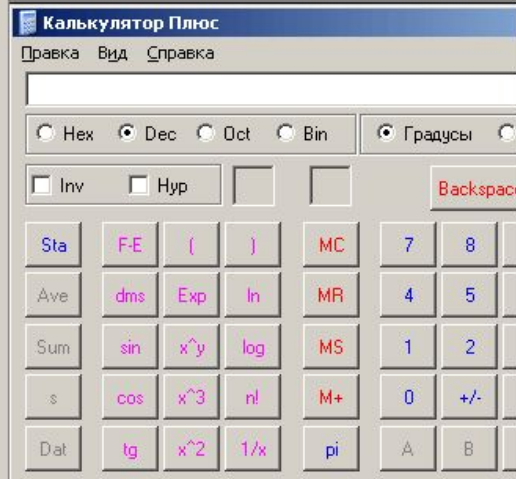
JPEG  
49,79K  
10 сек @ 56,6 кбит/с

качество: 70



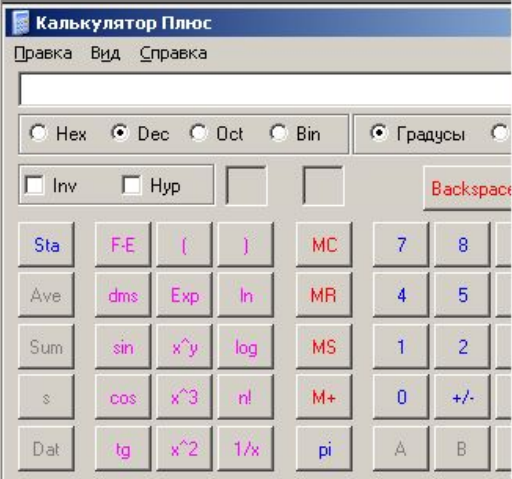
PNG-8  
5,005K  
2 сек @ 56,6 кбит/с

Дизайнинг: 100%  
Перцепционная  
256 цвета



GIF  
10,91K  
3 сек @ 56,6 кбит/с

Дизайнинг: 79%  
Селективная  
128 цвета



PNG-24  
7,087K  
2 сек @ 56,6 кбит/с

Набор: [Безимени]

PNG-24

Прозрачность Матовый:

Чересстрочно

Преобразовать в sRGB

Просмотр: Цвет монитора

Метаданные: Сведения об авт. правах и контакты

Таблица цветов

Размер изображения

Ш: 480 пикс. Проценты: 100 %

В: 309 пикс. Качество: Бикубическая

Анимация

Параметры повторов: Навсегда

1 из 1

100% R: -- G: -- B: -- | Альфа: -- Шестнадц.: -- Индекс: --



Исходное   Оптимизация   2 варианта   4 варианта

<p>JPEG 607 байт 1 сек @ 56,6 кбит/с</p>	<p>качество: 60 PNG-8 541 байт 1 сек @ 56,6 кбит/с</p>
<p>GIF 365 байт 1 сек @ 56,6 кбит/с</p>	<p>Дизеринг: 88% Селективная 89 цвета PNG-24 539 байт 1 сек @ 56,6 кбит/с</p>

Набор: [Безимени]

PNG-24

Прозрачность   Матовый: [ ]

Чересстрочно

Преобразовать в sRGB

Просмотр: [Цвет монитора]

Метаданные: [Сведения об авт. правах и контакты]

Таблица цветов

Размер изображения

Ш: 18 пикс.   Проценты: 100 %

В: 17 пикс.   Качество: Бикубическая

Анимация

Параметры повторов: [Навсегда]

1 из 1

100%   R: --   G: --   B: --   |   Альфа: --   Шестнадц.: --   Индекс: --

# Итог

- JPEG – для фотографий, изображений с большим количеством градиентов и цветов
- GIF – для маленьких рисунков, иконок, пиктограмм, полезен, когда края – четкие
- PNG8 – когда не много цветов, четкие края, например скриншоты windows - окон
- PNG24 – полупрозрачности, изображение без потери качества, большое количество цветов

# Применение изображений в HTML

- ``
- `align="bottom | left | middle | right | top"`
- Заливка с помощью `width` или `height`
- Бывает, что изображения отключены...
- `ismap`, `usemap`
- `onAbort` – только ie, `onError`, `onLoad`
- Фоновые изображения



# Карты на изображениях

- Серверные карты

```



```

- Клиентские карты

```

<map name="menumap">
 <area shape="rect" coords="0, 0, 50, 50"
href="index.html#p1">
 <area shape="circle" coords="100, 100, 20"
href="index.html#p2">
</map>
```

# Применим карту?



# Гиперссылки

- Основа гипертекста

- URL

scheme:scheme\_specific\_part

<схема>://<логин>:<пароль>@<хост>:<порт>/<URL - путь>

http://ru.wikipedia.org:80/wiki/HTTP?get#GET

`<a id="GET" name="GET"></a>`

- Только US-ASCII – СИМВОЛЫ

- `http://yandex.ru/yandsearch?text=%2F+%D0%BA%D0%BE%D0%B4%20%2F` (/ код /)

- `<a href="foto.jpg" target="_blank">foto</a>` -  
ссылка с относительным адресом

- `mailto:mail@mail.ru?subject=Subject`

# Отношения. Мета-теги.

- Мы имеем дело не с документом, а с проектом, набором документов

- `<head>`  
`<base href="http://rezonans.ru/" />`  
`<link rel="stylesheet" type="text/css" href="style.css" />`  
`<meta http-equiv="Content-Type" content="text/html;`  
`charset=utf-8" />`  
`<meta name="generator" content="MediaWiki 1.16alpha-wmf" />`  
`<meta name="keywords" content="ASCII,Ref-en,Английский`  
`язык,Соединённые Штаты Америки,Стандарт,Код,Информация,`  
`Диакритический знак,Акут,Man (Unix),Язык" />`  
`</head>`

# СПИСКИ

```
<ul type="square">
 Страны
 <ul type="circle">
 Англия
 Швейцария

 Города

<ol type="A" start="3">
<!-- A, a, I, i, 1 -->
 Крокодил Гена
 <li value="6">Чебурашка

```

# Списки определений

```
<dl>
```

```
 <dt>Термин 1</dt>
```

```
 <dd>Определение первого термина</dd>
```

```
 <dt>Термин 2</dt>
```

```
 <dd>Определение второго термина</dd>
```

```
</dl>
```

# Формы

- Процесс заполнения, отправки
- `<form action="form.php" enctype="multipart/form-data" method="POST"></form>`
- События
- `<input name="my_name">` (controls)
- `type=text, password, file`
- `size="30" maxlength="20" value="my name"`
- `accept="image/*"`

# Поля форм

- `<input type="checkbox" name="border" value="top" checked="checked" />`
- `<input type="radio" name="border" value="top" checked>`
- `<input type="submit" value="отправить" name="submit">`
- `<input type="reset" value="Сбросить форму">`
- `<input type="image" src="кнопка.jpg" name="btn" />`
- `<input type="button" value="надпись">`
- `<input type="hidden" name="user" value="34tr3sdf" />`



# Поля форм 2

- `<button type="submit" > текст</button>`
- `<textarea name="longtext" rows="10" cols="40"></textarea>`
- `<select name="xxx" multiple="multiple" size="5"><option selected="selected">texxxt</option></select>`
- `accesskey, disabled, readonly`
- `<label for="name-field">Введите имя:</label><input name="name" id="name-field" />`
- `<fieldset><legend align="top"></legend></fieldset>`

# Таблицы

- `<table cellspacing="0" cellpadding="0" border="0" frame="below" rules="cols" width="100%">`  
`<caption>Название таблицы</caption>`  
`<thead>`  
`<tr align="left" valign="middle">`  
`<th>111</th>`  
`<td></td>`  
`</tr>`  
`</thead>`  
`</table>`
- background, bgcolor, bordercolor, cols, height, title, nowrap
- colspan, rowspan
- `<tfoot>`, `<tbody>`
- `<colgroup>`, `<col>`

# Фреймы

- `<html><head><title>Фреймы</title></head>  
<frameset rows="30%, *">  
<frame src="fr1.htm" ></frame>  
<frame src="fr2.htm"></frame>  
</frameset>  
</html>`
- Рамки для frameset
- noresize, scrolling, frameborder для frame
- `<noframes>`
- `<iframe height="100" width="400" src="fr1.htm">Не работают фреймы?</iframe>`
- target у ссылки, `<base>`

# Объекты и апплеты

- `<object>`  
`<param name="name" value="1" />`  
`</object>`
- Вложенные object
- `<embed src="x.swf"`  
`type="application/x-shockwave-f`  
`lash" />`

# div и span

- Div – блочный
- Span – строчный
- title, dir, lang, style, class

# CSS 2

- Стиль – это правило отображения тега
- Встроенные стили, стили документа, внешние таблицы
- `<link>` или `@import`
- `@media` – правило
- `/* комментарий к стилям */`

# Синтаксис

- `selector {property:value; property1:value1;}`
- `p {color: red; text-decoration: underline;}`
- `p {font-family: Georgia, 'Times New Roman', Times, serif;}`
- `p {border: 1px solid red;}`
- **Стиль элемента явно указан, либо унаследован, либо взят по умолчанию**

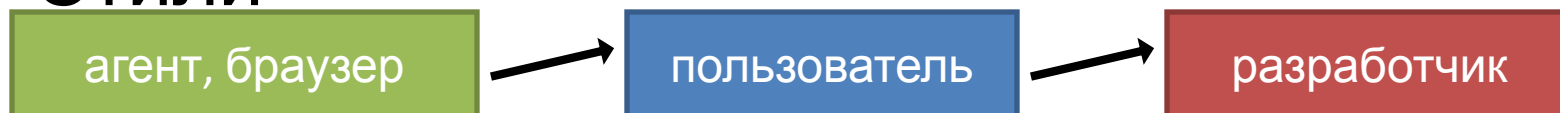
# Селекторы

- `p {}` /\* типовой, по тегу \*/
- `#id1 {}` /\* идентификатор \*/
- `.class1 {}` /\* по имени класса \*/
- `* {}` /\* универсальный \*/
- `*[align="right"] {}` /\* по атрибутам \*/
- `p#id1.class1.class2 {}`
- `p:first-line {}` /\* псевдоэлементы \*/
- `div, table, .class1 {}` /\* групповой \*/
- `ul ul {}` /\* контекст, наследующие \*/
- `ol > li {}` /\* дочерний \*/
- `li + li {}` /\* соседние \*/



# Приоритеты стилей

- **Стили**



- **!important**



- **Порядок каскадирования:**

- По источнику ("ближе" к тегу)
- Специфичность, более узкое, точное определение
- Порядок следования

- **Как правило стили приоритетнее атрибутов**

# Специфичность

a – id; b – класс, псевдокласс, атрибут; c – имя тега.

```
* {} /* a=0 b=0 c=0 -> специфичность = 0 */
li {} /* a=0 b=0 c=1 -> специфичность = 1 */
ul li {} /* a=0 b=0 c=2 -> специфичность = 2 */
ul ol+li {} /* a=0 b=0 c=3 -> специфичность = 3 */
ul ol li.red {} /* a=0 b=1 c=3 -> специфичность = 13 */
li.red.level {} /* a=0 b=2 c=1 -> специфичность = 21 */
#x34y {} /* a=1 b=0 c=0 -> специфичность = 100 */
/* style="" -> специфичность = 1000 */
```

# Размеры, цвета, URL в CSS

- Ключевые слова, `inherit`
- `url(http://localhost/1.jpg)`
- `red`, `#7788AA`, `rgb(12,11,34)`
- Размеры:
  - **em** — ширина буквы `m` в настоящем шрифте. Например, `p {text-indent: 3em}` задаст красную строку абзаца шириной в три буквы `m`.
  - **px** — пикселы
  - **pt** — пункты. Один пункт = 1/72 дюйма.
  - **%** — проценты
  - **ex** — ширина буквы `x`
  - **in** — дюймы
  - **cm** — сантиметры
  - **mm** — миллиметры
  - **pc** — размер в пиках. (12 пунктов).

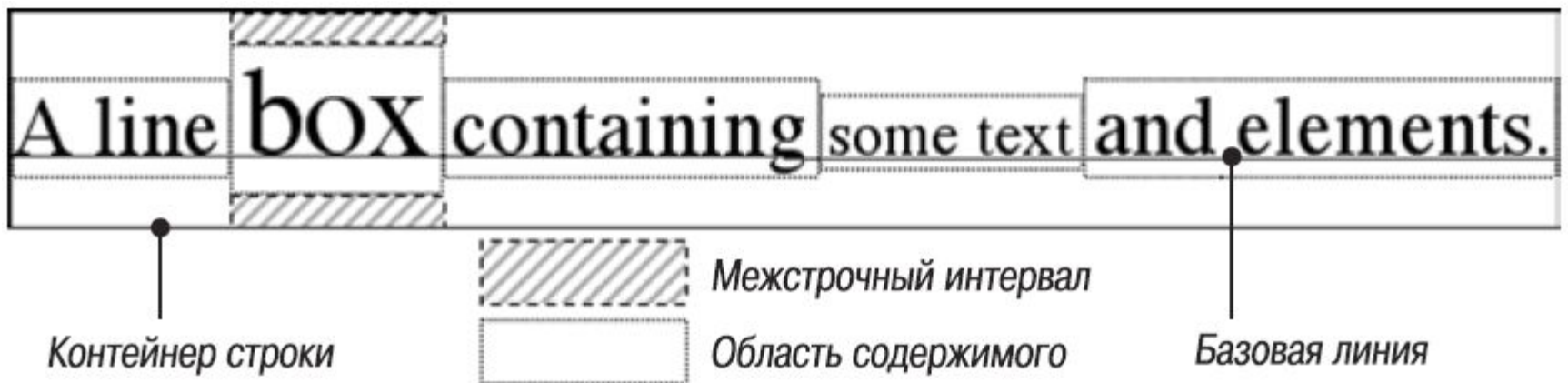
# Шрифты

- font-family: Georgia, 'Times New Roman', Times, serif; (с засечками, без, рукописные..)
- font-size: larger;
- font-style: italic;
- font-variant: small-caps;
- font-weight: bold; (400)
- font: [font-style | | font-variant | | font-weight]  
**font-size** [/line-height] **font-family**
- Загрузка шрифтов, @font-face

# Свойства текста

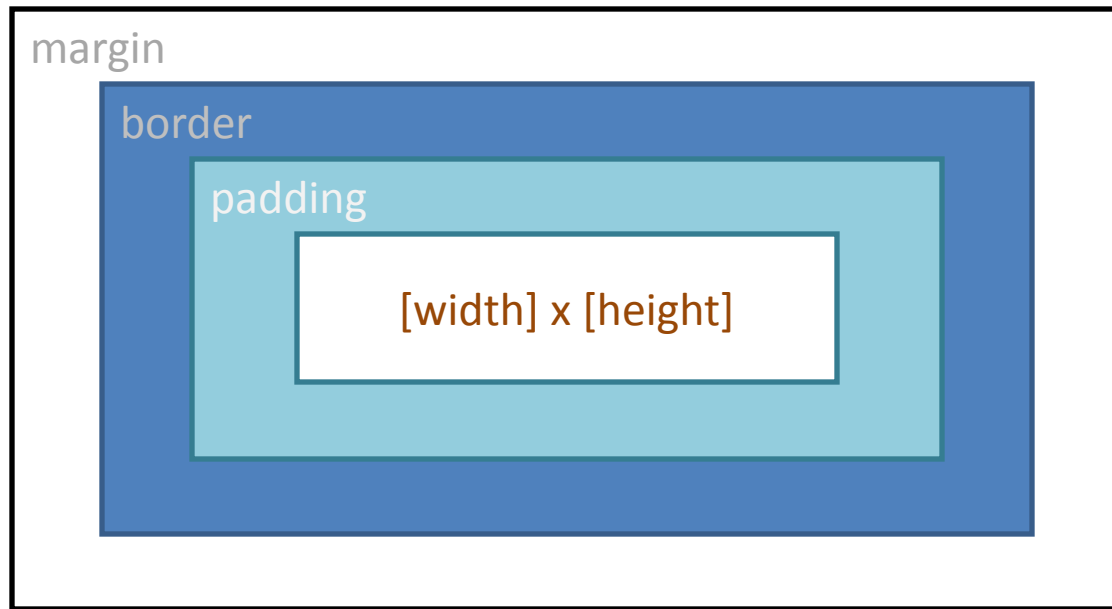
- letter-spacing: 2px;
- line-height: 120%; (наслед. вычисл. от родителя)
- text-align: right;
- text-decoration: blink | line-through | overline | underline | none
- text-indent: -5em;
- text-transform: capitalize | lowercase | uppercase | none
- vertical-align: baseline | bottom | middle | sub | super | text-bottom | text-top | top | значение | проценты (только строк. и замещ. элементы)
- word-spacing: 10em;
- white-space: normal | nowrap | pre (pre в ie6 работает 😞)
- Другие языки...

# Контейнер строки



# Свойства контейнеров

- Блочная модель CSS, горизонтальное и вертикальное форматирование (по 7 свойств)



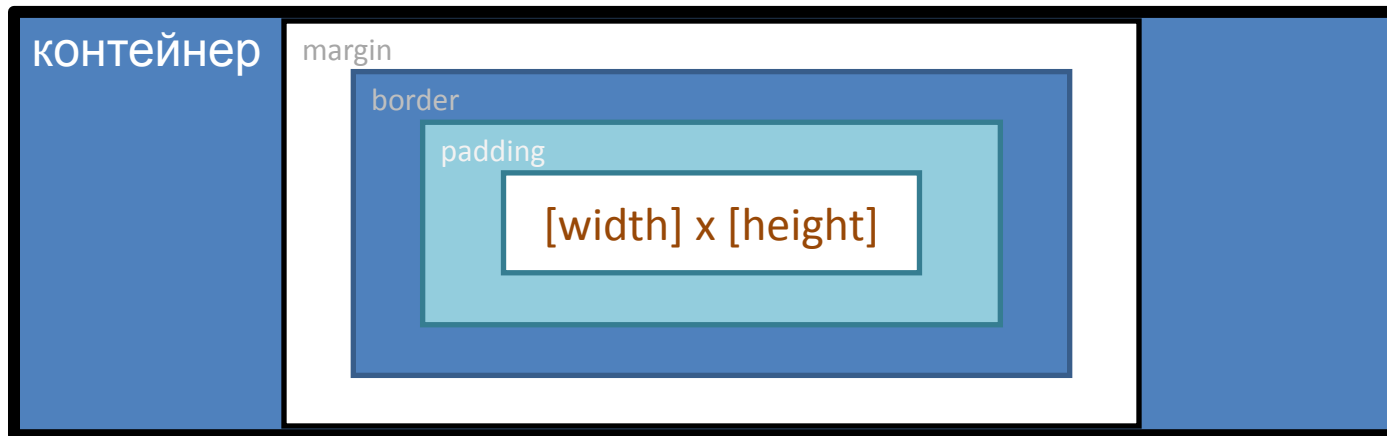
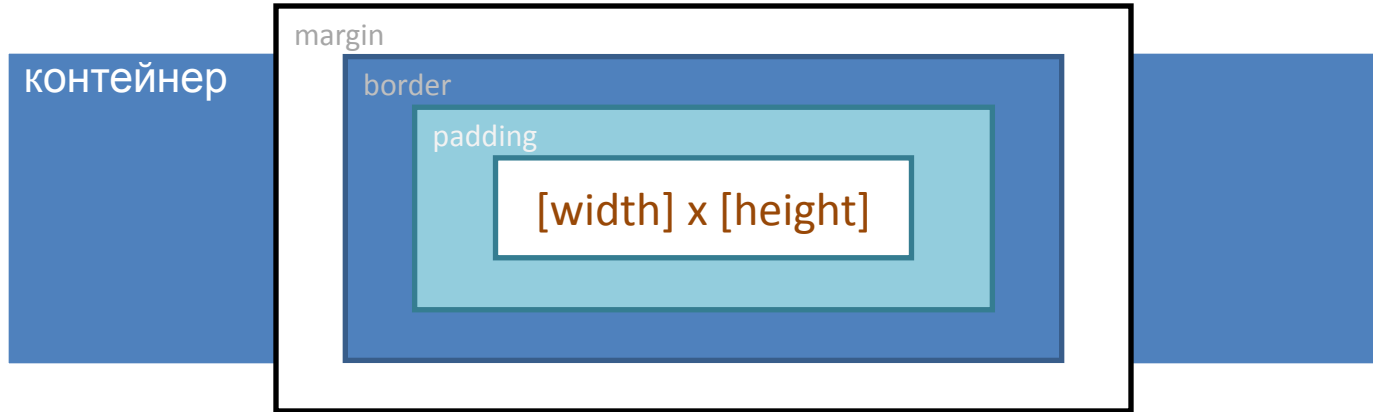
- top, right, bottom, left

# Горизонтальное форматирование

- Значение auto – заполняет всю оставшуюся ширину контейнера (margin, width)
- Отрицательные поля
- Сумма 7 размеров дочернего элемента = width родительского
- Для замещающих элементов размеры auto равны реальным размерам
- Размеры замещающих элементов изменяются пропорционально, если задавать одно из них



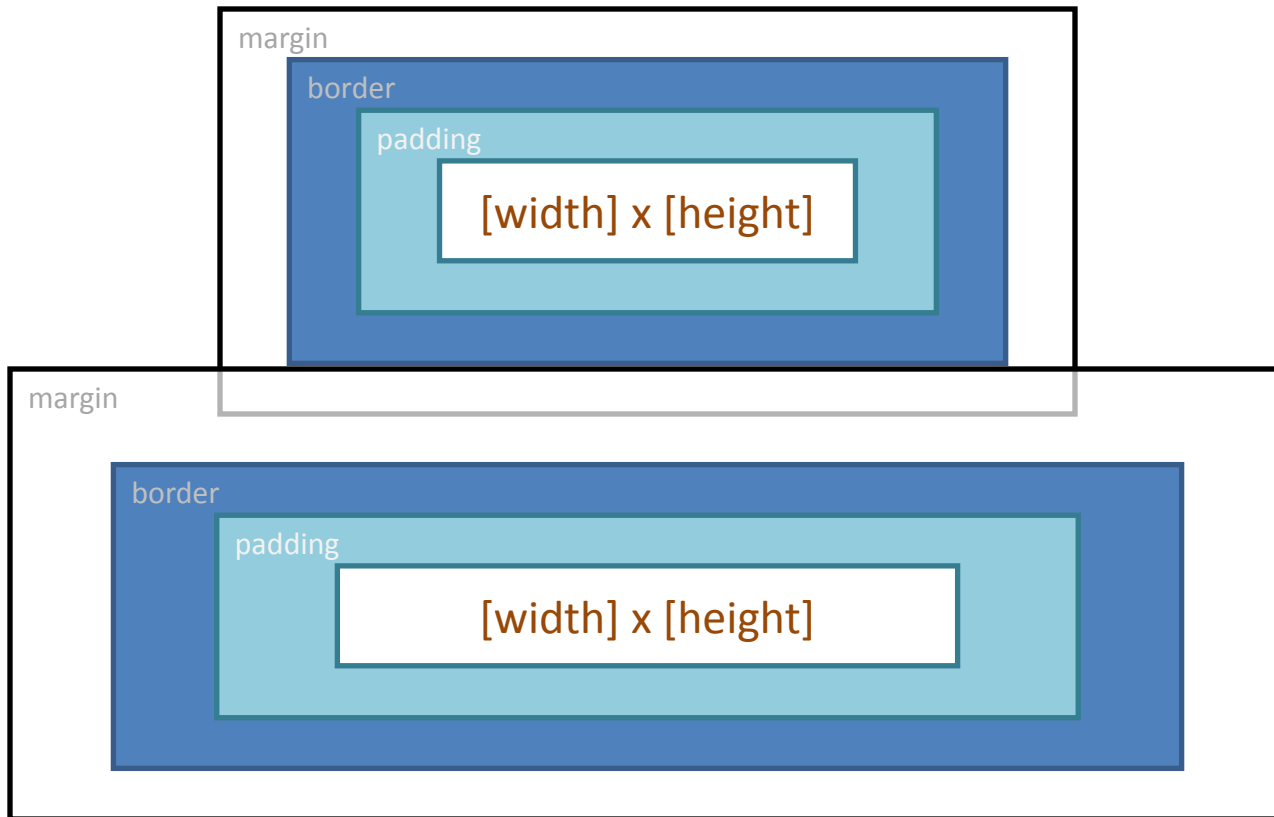
# height=auto



# Вертикальное форматирование

- Высота по содержимому (auto)
- Или через height. Если содержимого больше чем height – прокрутка
- margin-top или bottom равное auto = 0
- Высота в процентах – от блока контейнера, но...
- Центрирование по вертикали через процентные margin и высоту блока-контейнера
- Сворачивание полей
- Отрицательные margin

# Сворачивание вертикальных полей



# Форматирование строчных элементов

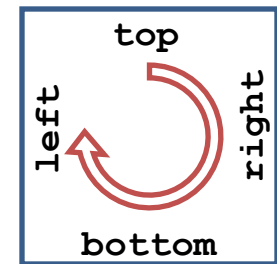
- Многострочный строковый элемент, рамки, фон
- Строковый блок и контейнер строки
- Отступы, рамки и поля незамещаемых элементов не оказывают влияния на высоту строкового блока в отличие от замещаемых
- Вспомним line-height
- Если в строке есть замещаемый элемент, то его реальная высота, padding и margin, border влияют на высоту строкового блока

# Изменение представления элемента

- `display: block | inline | inline-table | list-item | none | run-in | table | table-caption | table-cell | table-column-group | table-footer-group | table-header-group | table-row | table-row-group | inline-block`
- Пример употребления `display:block`
- Пример употребления `display:inline`
- `display` определяет только лишь представление элемента, но не его тип, не его суть
- `display: inline-block`
- Иногда значение `display` вычисляется в зависимости от значения свойств `float` или `position` (абс. – или `table` или `block`)

# Поля













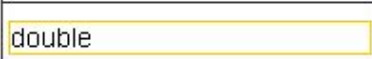
















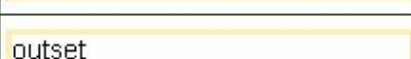


- По умолчанию `margin=0`
- У некоторых элементов есть поля по умолч.
- `img {margin: 1em;}`
- `img {margin: 1em 2em 3em 4em;}`
- `img {margin: 1em 2em;}`
- `img {margin: 1em 2em 3em;}`
- Процентные значения от ширины родительского элемента
- `margin-left`, `margin-right`, `margin-top`, `margin-bottom`
- Поля строчковых элементов (левое и правое)



# Рамки

- Рамки внутри полей, они ограничивают фон
- Ширина, стиль, цвет
- Ширина по умолч.=medium или none
- Цвет по умолч.=цвет элемента
- border-style: стиль (TRBL) или отдельно
- border-width: thin | medium | thick | значение (TRBL) или отдельно, % - запрещены
- Если border-style=none, то border-width=0
- border-color: color (TRBL) или отдельно
- Цвет рамки может = transparent

# Рамки, стили рамок

1 пиксел	3 пиксела	5 пикселей	7 пикселей
			
			
			
			
			
			
			
			

- border-top, border-right, border-bottom, border-left: border-width || border-style || color
- border: border-width || border-style || color
- У строковых элементов тоже могут быть



# Отступы

- padding: значение (любые меры, проценты) (TRBL)
- Залиты фоном
- Отступы не сворачиваются
- % относительно ширины родителя (причем и верхние и нижние поля)
- padding-top, padding-right, padding-bottom, padding-left
- Можно применять к строковым элементам

# Цвета, фон

- `color: <цвет> | inherit`
- Свойство `color` для элементов форм
- `background-color: цвет | transparent`  
(умолч.)
- `background-image: url(путь к файлу) | none`
- `background-color + background-image`
- `background-repeat: no-repeat | repeat | repeat-x | repeat-y`

# ФОН

- `background-position: [проценты | значение] | [left | center | right] || [top | center | bottom]`  
(если одно задано, второе – center)
- `background-attachment: fixed | scroll`  
(<http://www.meyerweb.com/eric/css/edge/complexspiral/glassy.html>)
- `background: background-attachment || background-color || background-image || background-position || background-repeat`

# Перемещение, плавающая МОДЕЛЬ

- Рисунки, параграф с float
- Задание ширины обязательно
- Поля не сворачиваются
- Перемещаемый элемент генерирует блочный элемент
- Правила перемещаемых элементов (стр.327)
- Перемещаемый элемент увеличивается, чтобы вместить всех перемещаемых потомков
- Отрицательные поля
- Если ширина больше – перемещаемый элемент окажется за боковым краем родителя
- clear: both | left | none | right

# Позиционирование

- position: absolute | fixed | relative | static
- static – нормальный поток
- relative – смещение элемента с теми же размерами и начальными координатами
- absolute – удаление из нормального потока.

Генерация структурного блока.

- fixed – абсолютно позиционированный элемент, но блок-контейнер – окно просмотра.

# Позиционирование 2

- Блок-контейнер для элементов с position = relative | static – родитель
- Блок-контейнер для элементов с position = absolute – ближайший предок с position != static. Если ближайший предок – строковый, то контейнер – предок. Если таких элементов нет – начальный блок-контейнер

# Свойства смещения

- top, right, bottom, left: <длина> | <процентное значение> | auto | inherit
- Ширина и высота
- min-width, min-height: <длина> | <процентное значение> | inherit
- max-width, max-height: <длина> | <процентное значение> | none | inherit

# Переполнение и отсечение содержимого

- `overflow: visible | hidden | scroll | auto | inherit`
- `overflow-x` и `overflow-y`
- `clip: rect(top, right, bottom, left) | rect(Y1, X1, Y2, X2) | auto | inherit`



# Видимость элементов

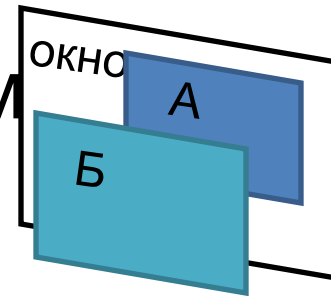
- `visibility: visible | hidden | collapse | inherit`

# Абсолютное позиционирование

- Абсолютное позиционирование относительно...
- Перекрытие элементов
- Размеры и размещение
- $\text{left} + \text{margin-left} + \text{border-left-width} + \text{padding-left} + \text{width} + \text{padding-right} + \text{border-right-width} + \text{margin-right} + \text{right} = \text{ширина блока-контейнера}$
- Значение `auto`
- `right` или `bottom` компенсируют, если все свойства заданы

# Размещение по оси z

- z-index: число | auto
- Может быть отрицательным
- Локальный контекст занесения в стек



# Фиксированное позиционирование

- Относительно окна браузера
- Удобно, например, для баннеров или для меню

# Относительное позиционирование

- Смещение относительно текущего положения

# Верстка таблиц

- Объединять ячейки через CSS нельзя
- У ячеек нет полей

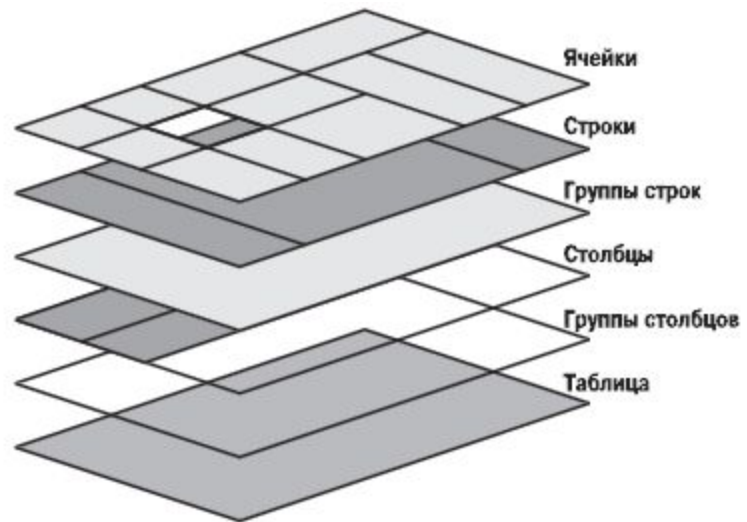
	Свойство CSS	Аналог HTML
СВОЙСТВО display	display: table	table
	display: table-row	tr
	display: table-head	thead
	display: table-row-group	tbody
	display: table-footer-group	tfoot
	display: table-column	col
	display: table-column-group	colgroup
	display: table-cell	td, th
	display: table-caption	caption

# Таблицы. Продолжение

- `display: inline-table` – таблица строкового уровня (типа `inline-block`)
- Значения можно присвоить любым элементам и сделать на основе них таблицу
- Анонимные объекты таблицы
- ```
<table>  
<td>Name:</td>  
<td><input type="text"></td>  
</table>
```

Таблицы. Продолжение

- Слои таблицы
- `caption-side: top | bottom`
- `border-collapse: collapse | separate | inherit`
- `border-spacing: <длина гор.> <длина верт.>? | inherit`
- `empty-cells: show | hide | inherit`



Сливающиеся рамки ячеек

- Если `display: table | inline-table` у элемента не может быть отступов, только поля
- Рамки могут применяться к таблице, ячейкам, строкам, группам строк, столбцам и группам столбцов
- Между рамками ячеек не может быть никаких промежутков; рамки сливаются и центрируются

Размеры таблиц

- Ширина
- table-layout: auto | fixed | inherit
- Скорость рендера с фиксированной шириной больше
- Если сумма ширин столбцов больше ширины таблицы, то берется первое
- Высота: или заданная или как сумма высот строк
- Вертикальное выравнивание – не то же самое что и для строковых элементов

Элементы списка

- Управлять размещением сложно
- `list-style-position: inside | outside` – вне содержимого или как строковый маркер в начале содержимого
- `list-style-type: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none`
- `list-style-image: none | url('путь к файлу')`
- `list-style: list-style-type | list-style-position | list-style-image`

















Генерируемое содержимое

- Например маркеры списка...
- `a[href]:before {content: "(link)";}`
- Ограничения на `display`
- `content`: строка | `attr(атрибут)` | `open-quote` | `close-quote` | `no-open-quote` | `no-close-quote` | `url` }
- Теги недопустимы
- Генерируемые кавычки
- Счетчики
- `counter-reset` и `counter-increment`

Курсоры

- `cursor: [[<uri>,* [auto | default | pointer | crosshair | move | e-resize | ne-resize | nw-resize | n-resize | s-resize | sw-resize | w-resize | text | wait | help | progress]] | inherit`
- `a[href] {cursor: pointer;}`

Курсоры

	default		n-resize
	crosshair		ne-resize
	help		e-resize
	move		se-resize
	pointer		s-resize
	progress		sw-resize
	text		w-resize
	wait		nw-resize

Контур

- Не участвуют в потоке документа
- Могут употребляться вместе с рамками
- `outline-style: none | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit`
(TRBL – нет)
- `outline-width: thin | medium | thick | <длина> | inherit`
- `outline-color: <цвет> | invert | inherit`
- `outline: [<outline-color> || <outline-style> || <outline-width>] | inherit`

