

Internet- технологии

ЛЕКЦИЯ №2
ЯЗЫК HTML

Содержание

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Содержание лекции №2</title>
  <meta charset="utf-8">
</head>
<body>
  <h2><i>Содержание</i></h2>
  <ul>
    <li>Списки</li>
    <li><table border="1"><tr><th>Таблицы</th></tr></table></li>
    <li><a href="#">Гиперссылки</a></li>
    <li>Основные понятия компьютерной графики.<br>Использование изображений в веб.</li>
    <li>Карты изображений</li>
    <li>Формы и управляющие элементы</li>
    <li>Снова <em>логическое</em> и <i>физическое</i> форматирование</li>
  </ul>
</body>
</html>
```

Содержание

- Списки
- | |
|---------|
| Таблицы |
|---------|
- [Гиперссылки](#)
- Основные понятия компьютерной графики.
Использование изображений в веб.
- Карты изображений
- Формы и управляющие элементы
- Снова *логическое* и *физическое* форматирование



Инструментарий

Редакторы:

- Brackets / Notepad++ / Sublime Text / Atom...
- VSC (Visual Studio Code);
- IDE: WebStorm, PhpStorm.



Работа с графикой:

- GIMP (GNU Image Manipulation Program).

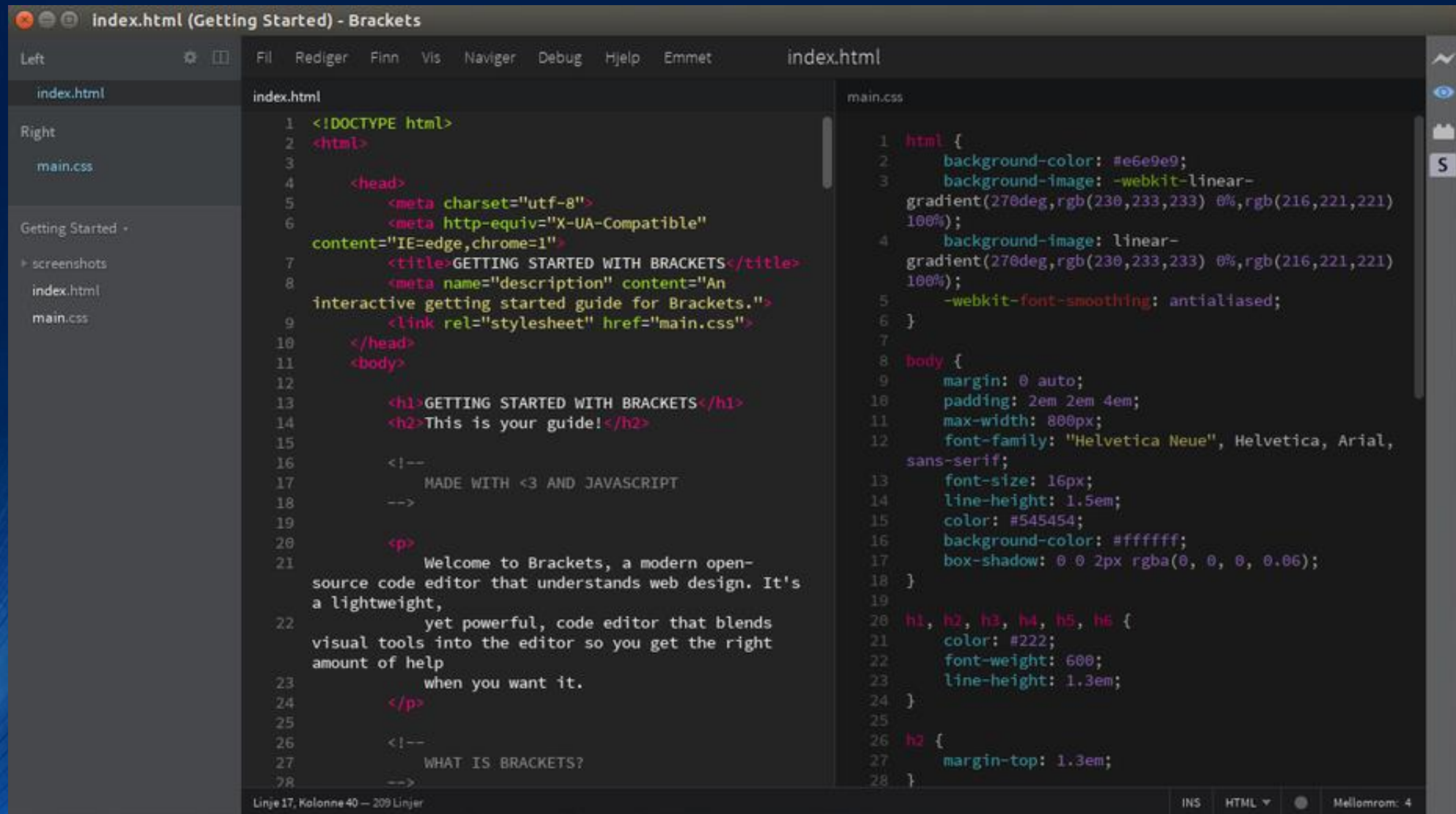


Браузер:

- Firefox;
- Google Chrome.



Инструментарий



The screenshot displays the Brackets code editor interface. The title bar reads "Index.html (Getting Started) - Brackets". The menu bar includes "File", "Editor", "View", "Navigator", "Debug", "Help", and "Emmet". The left sidebar shows a file tree with "index.html" and "main.css". The main workspace is split into two panes. The left pane shows the "index.html" file with the following content:

```
1 <!DOCTYPE html>
2 <html>
3
4   <head>
5     <meta charset="utf-8">
6     <meta http-equiv="X-UA-Compatible"
7       content="IE=edge,chrome=1">
8     <title>GETTING STARTED WITH BRACKETS</title>
9     <meta name="description" content="An
10      interactive getting started guide for Brackets.">
11     <link rel="stylesheet" href="main.css">
12   </head>
13   <body>
14
15     <h1>GETTING STARTED WITH BRACKETS</h1>
16     <h2>This is your guide!</h2>
17
18     <!--
19      MADE WITH <3 AND JAVASCRIPT
20    -->
21
22     <p>
23       Welcome to Brackets, a modern open-
24       source code editor that understands web design. It's
25       a lightweight,
26       yet powerful, code editor that blends
27       visual tools into the editor so you get the right
28       amount of help
29       when you want it.
30     </p>
31
32     <!--
33      WHAT IS BRACKETS?
34    -->
35  </body>
36</html>
```

The right pane shows the "main.css" file with the following content:

```
1 html {
2   background-color: #e6e9e9;
3   background-image: -webkit-linear-
4     gradient(270deg,rgb(230,233,233) 0%,rgb(216,221,221)
5     100%);
6   background-image: linear-
7     gradient(270deg,rgb(230,233,233) 0%,rgb(216,221,221)
8     100%);
9   -webkit-font-smoothing: antialiased;
10 }
11
12 body {
13   margin: 0 auto;
14   padding: 2em 2em 4em;
15   max-width: 800px;
16   font-family: "Helvetica Neue", Helvetica, Arial,
17   sans-serif;
18   font-size: 16px;
19   line-height: 1.5em;
20   color: #545454;
21   background-color: #ffffff;
22   box-shadow: 0 0 2px rgba(0, 0, 0, 0.06);
23 }
24
25 h1, h2, h3, h4, h5, h6 {
26   color: #222;
27   font-weight: 600;
28   line-height: 1.3em;
29 }
30
31 h2 {
32   margin-top: 1.3em;
33 }
```

The status bar at the bottom indicates "Line 17, Column 40 — 209 Lines".



Списки

Элементы списка представляют собой простые наборы **блочных** элементов, имеющих дополнительный блок, расположенный сбоку, который не участвует в компоновке.

Нумерованные – для перечисления элементов, следующих в строго определенном порядке. Если дополнить уже существующий список новыми значениями, то браузер автоматически пересчитает его.

Маркированные – для перечисления элементов, следующих в произвольном порядке.

Список определений – используется для форматирования словарей.

Многоуровневые списки используются для конкретизации информации определенных элементов. Для создания многоуровневых списков можно использовать как маркированные, так и нумерованные списки, а также их сочетание. Многоуровневый список получается путем вложения одного списка в тело другого.



Нумерованный список

Для создания нумерованных списков используются тэги `` и ``. Тэгом `` отмечается начало/окончание всего списка. Тэгом `` обозначают начало/конец отдельного элемента списка.

При помощи атрибута `type` можно изменить стиль нумерации:

- `type="A"` – в алфавитном порядке заглавными латинскими буквами (A, B, C).
- `type="a"` – нумерация прописными латинскими буквами (a, b, c).
- `type="I"` – нумерация большими римскими цифрами (I, II, III).
- `type="i"` – нумерация маленькими римскими цифрами (i, ii, iii).

По умолчанию элементы списка нумеруются по порядку 1, 2, 3... При помощи атрибутов `start` и `value` можно изменить порядок нумерации списка. `start` – служит для задания начального номера списка, отличного от 1. `value` – дает возможность назначить произвольный номер любому элементу списка. Атрибут `reversed` задает отображение списка в обратном порядке



Нумерованный список. Пример

``

`элемент 1`

`элемент 2`

`элемент 3`

``

`<ol type="A">`

`элемент 1`

`элемент 2`

`элемент 3`

``

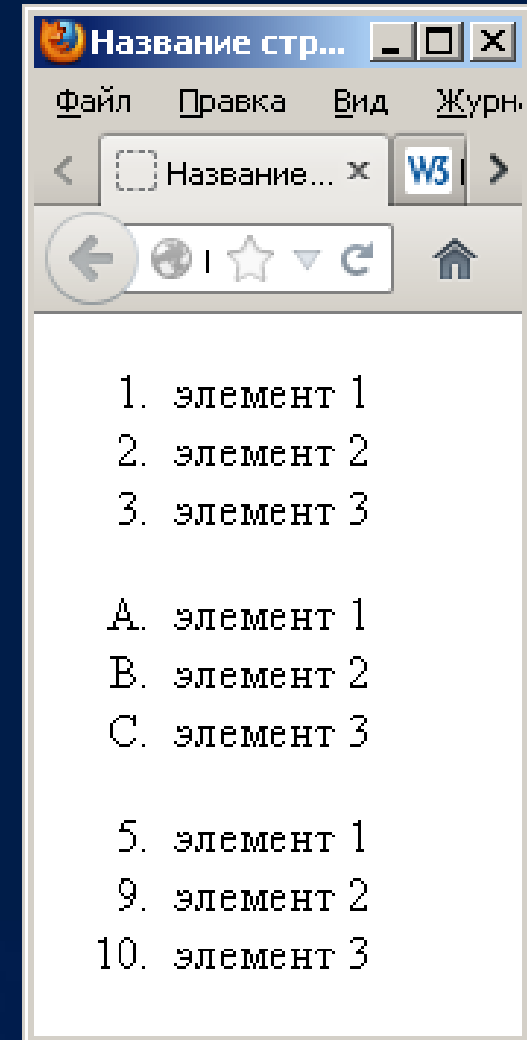
`<ol start="5">`

`элемент 1`

`<li value="9">элемент 2`

`элемент 3`

``



Маркированный список

Для создания маркированных списков используются тэги `` и ``. Тэгом `` (Unordered List) отмечается начало/окончание всего списка. Тэгом `` обозначают начало/конец отдельного элемента списка.

По умолчанию элементы списка маркируются черным кружком. При помощи атрибута `type` можно изменить стиль маркирования. В пределах одного списка можно использовать различную маркировку элементов списка. **Устарел!**

```
<ul type="circle">  
  <li>элемент 1</li>  
  <li>элемент 2</li>  
  <li type="disc">элемент 3</li>  
  <li type="square">элемент 4</li>  
</ul>
```

- элемент 1
- элемент 2
- элемент 3
- элемент 4



Вложенный маркированный список

Пункт 1.

Пункт 2.

Подпункт 2.1.

Подпункт 2.2.

Подпункт 2.2.1.

Подпункт 2.2.2.

Подпункт 2.3.

Пункт 3.

- Пункт 1.
- Пункт 2.
 - Подпункт 2.1.
 - Подпункт 2.2.
 - Подпункт 2.2.1.
 - Подпункт 2.2.2.
 - Подпункт 2.3.
- Пункт 3.



Список определений

Каждый пункт состоит из двух частей: термин и его определение.

Для создания списков определений используют три тэга:

`<dl>` – начало/конец списка.

`<dt>` – начало/конец конкретного термина.

`<dd>` – начало/конец поясняющей статьи термина.

Тэги `<dt>` и `<dd>` не обязательно чередовать. Т.е. можно «привязать» к одному определению несколько терминов и наоборот.

`<dl>`

`<dt>Термин 1</dt>`

`<dd>Аннотация1 к термину 1</dd>`

`<dd>Аннотация2 к термину 1</dd>`

`<dt>Термин 2</dt>`

`<dd>Аннотация1 к термину 2</dd>`

`<dd>Аннотация2 к термину 2</dd>`

`</dl>`

Термин 1

Аннотация1 к термину 1

Аннотация2 к термину 1

Термин 2

Аннотация1 к термину 2

Аннотация2 к термину 2



Таблицы

Для добавления таблицы на веб-страницу используется тег `<table>`, который служит контейнером для строк и ячеек, которые задаются соответственно с помощью тегов `<tr>` и `<td>`. Для создания заголовочных ячеек вместо тега `<td>` используется тег `<th>`. Текст в такой ячейке отображается браузером с шрифтом жирного начертания и выравнивается по центру ячейки.

Тег `<caption>` предназначен для создания заголовка к таблице и может размещаться только внутри контейнера `<table>`, причем сразу после открывающего тега.

Атрибуты тегов `<td>` и `<th>`:

- `colspan` – определяет объединение ячеек по горизонтали.
- `rowspan` – определяет объединение ячеек по вертикали.

Основная функция таблиц в HTML – представление данных, также иногда используются для определения структуры тела документа. Не рекомендуется.



Таблицы. Пример

```
<table width="500" border="1">  
  <tr>    <td colspan="2">Начало таблицы</td>  </tr>  
  <caption>Заголовок таблицы,  
необязательный элемент.</caption>  
  <tr>  
    <th>Заголовок 1</th>  
    <th>Заголовок 2</th>  
  </tr>  
  <tr align="center">  
    <td rowspan="2" align="center">Ячейка 1-1, Ячейка 2-1</td>  
    <td>Ячейка 1-2</td>  
  </tr>  
  <tr>  
    <td>Ячейка 2-2</td>  
  </tr>  
</table>
```

Начало таблицы	
Заголовок 1	Заголовок 2
Ячейка 1-1, Ячейка 2-1	Ячейка 1-2
	Ячейка 2-2



Табличная верстка

Достоинства:

- простая организация многоколоночной страницы;
- «резиновый» макет.
- единообразие обработки браузерами.

Недостатки:

- долгая загрузка.
- лишний код.
- разделение выводимой информации, плохо для SEO.
- сложность оформления.
- несоответствие стандартам.



Гиперссылки

Ссылки позволяют переходить с одной веб-страницы на другую:

`текст ссылки (или изображение)`

Ссылка может вести не только на HTML-файлы, но и на файл любого типа.

`Почта`

- абсолютные гиперссылки должны начинаться с указания протокола (обычно http://)

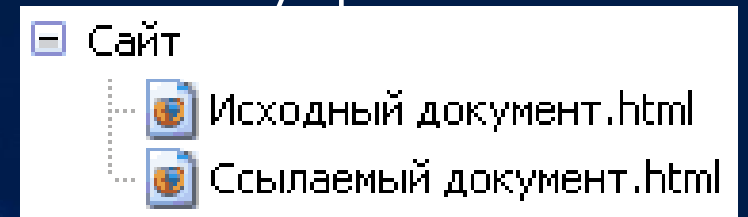
`https://en.wikipedia.org/wiki/Web_design`

`file:///D:/about.html`

`file:///D:/Education/%D0%9F%D1%80%D0%B5%D0%BF%D0%BE%D0%B4%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/Trening_2014/Lb1.html`

- относительная гиперссылка – адрес строится относительно текущего документа или корня сайта:

`Ссылка`



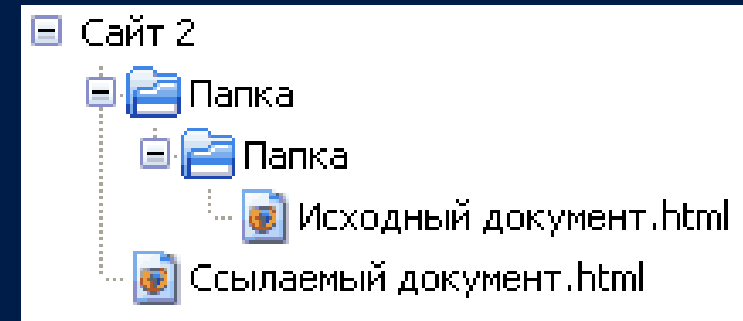
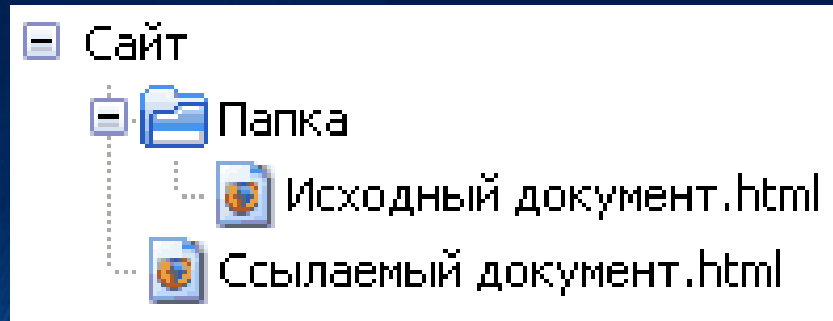
Гиперссылки. Примеры относительных ссылок

Файлы размещаются в разных папках:

`Ссылка`

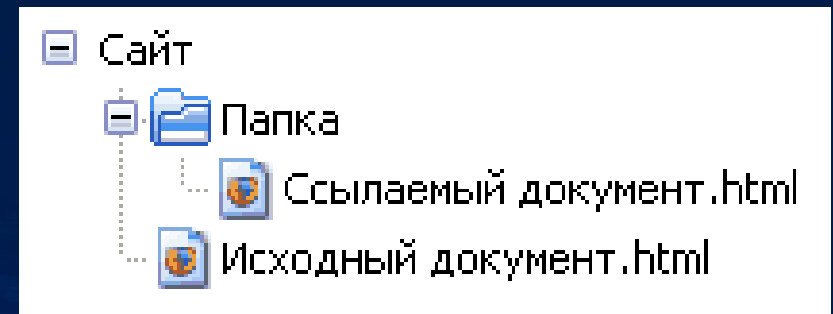
`Ссылка`

Две точки в данном случае означают выйти из текущей папки на уровень выше.



Исходный файл располагается в корне сайта, а файл, на который необходимо сделать ссылку - в папке:

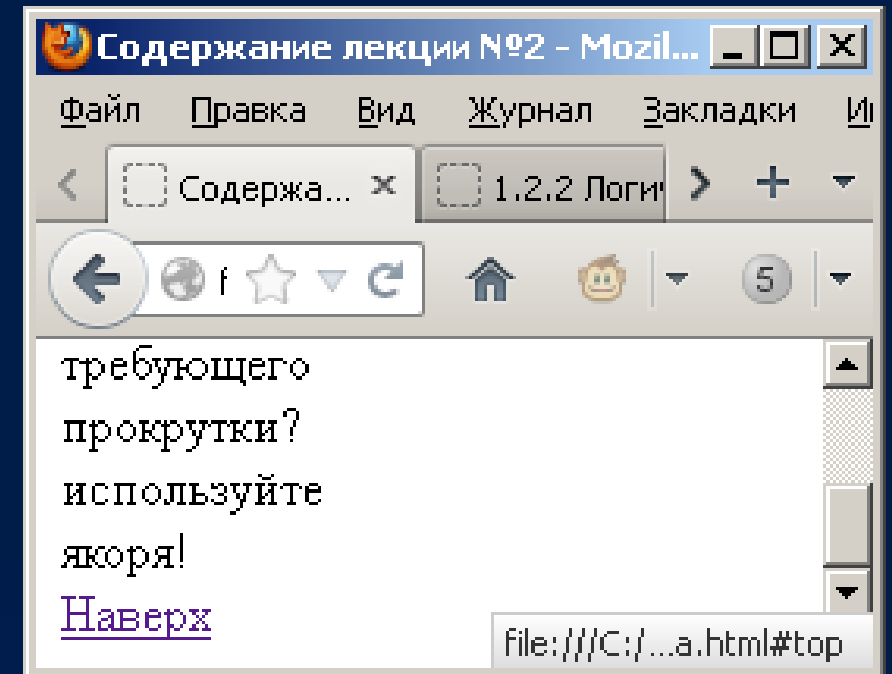
`Ссылка`



Гиперссылки. Якоря

Якорем называется закладка с уникальным именем на определенном месте веб-страницы, предназначенная для создания перехода к ней по ссылке. Якоря удобно применять в документах большого объема, чтобы можно было быстро переходить к нужному разделу. Метка – не обязательно `<a>`.

```
<body>
  <a name="top" id="top"></a>
  <br> Очень
  <br> много
  <br> контента,
  <br> требующего
  <br> прокрутки?
  <br> Используйте
  <br> якоря!<br>
  <a href="#top">Наверх</a>
</body>
```



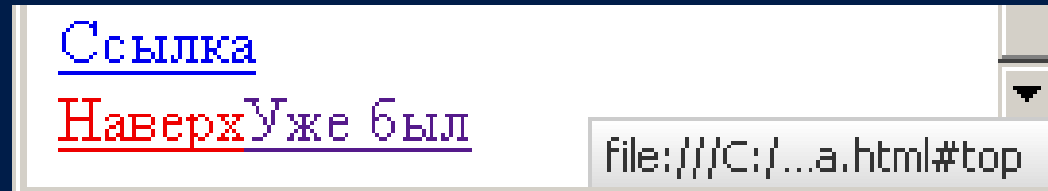
Гиперссылки. Состояния и атрибуты

Состояния гиперссылок на странице:

Непосещенная – изображаются синего цвета и с подчеркиванием.

Активная – ссылка помечается как активная в момент ее открытия. Цвет такой ссылки по умолчанию красный.

Посещенная – при переходе по ней по умолчанию меняет свой цвет на фиолетовый.



Атрибуты:

- **target** – в каком окне/**вкладке** открывать. Значения **_blank**, **_self**, **_parent**, **_top**.
- **title** – добавляет поясняющий текст к ссылке в виде всплывающей подсказки.

```
<a href="new.html" target="_blank"
  title="Комментарий">Ссылка</a>
```



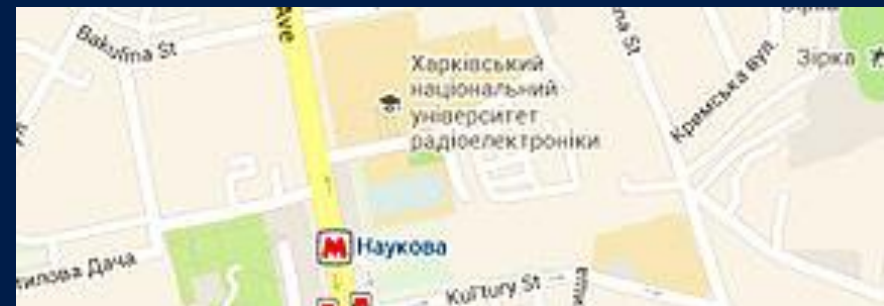
Использование графики в веб

Правильно подобранная и размещенная на Web-странице графика облегчает восприятие контента и делает страницу визуально привлекательней.

Направления использования графики:

- различные рисунки, диаграммы и графики;
- фон страницы и оформление ее элементов;
- рекламные объявления (банеры);
- логотипы и т.д.

Например, некоторые изображения с официального сайта университета:



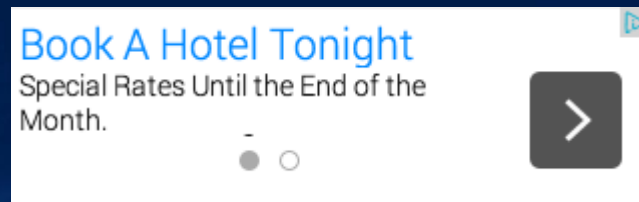
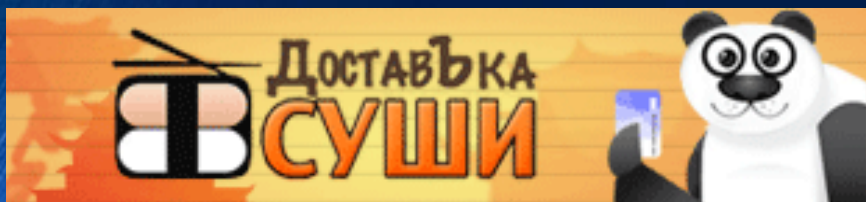
Думай смело, креативно!



Использование графики в веб. Баннеры

Баннер (англ. banner – транспарант) – графическое изображение рекламного характера, являющееся гиперссылкой на сайт рекламодателя или страницу с дополнительной информацией. Один из преобладающих форматов интернет-рекламы.

Частым явлением являются веб-страницы, направленные исключительно на получение прибыли от баннерной рекламы.



Использование графики в веб. Фон

Одним из наиболее важных аспектов графического оформления сайта является использование **фона**. Для создания фона, состоящего из одного сплошного цвета, используется атрибут **bgcolor** со hex-значением цвета. Для использования изображения в качестве фона используется атрибут **background** с адресом фонового рисунка. *Данные атрибуты устарели, используйте CSS!*

```
body {  
    background-image: url(background.png);  
    background-position: left top;  
    background-repeat: repeat-x;  
}
```



Содержание

- Использование графики в веб. Баннеры, фон.
- Виды компьютерной графики. Форматы графических файлов.
- Изображения, карты изображений.
- Формы и управляющие элементы.
- Фреймы.
- Особенности HTML5.

Источником фона может служить рисунок небольшого размера, тиражированный так, чтобы заполнить все пространство окна.

Фон не должен снижать контрастность страницы и мешать чтению текста.



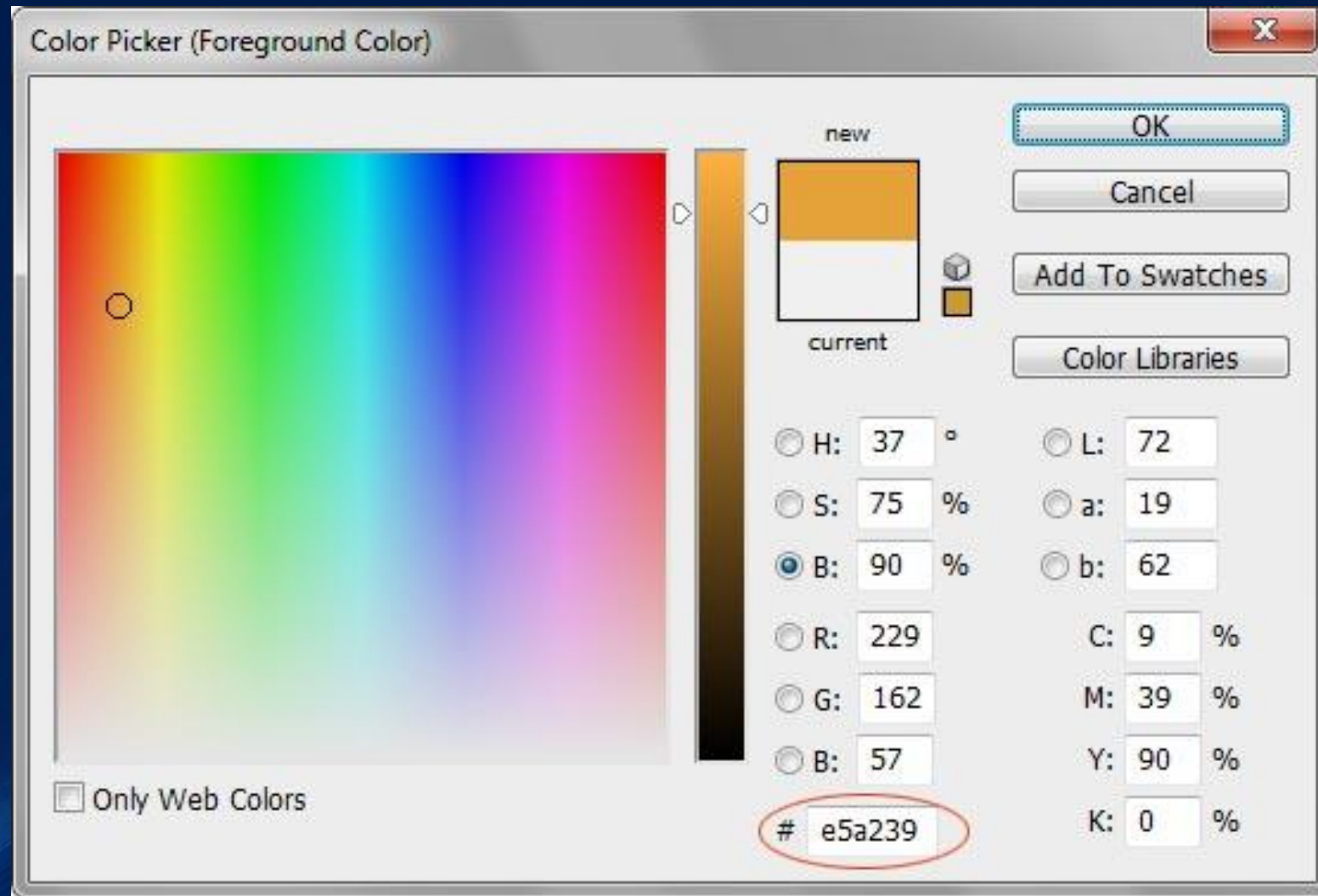
Указание цвета

Веб-цветами называются такие цвета, для каждой составляющей – красной, зелёной и синей, устанавливается одно из шести значений – 0 (00), 51 (33), 102 (66), 153 (99), 204 (CC), 255 (FF). Общее количество веб-цветов 216. Пример веб-цвета – #33ff66.

Имя цвета	Цвет	Описание	Шестнадцатеричное значение
black		Черный	#000000
gray		Серый	#808080
green		Зеленый	#008000
navy		Темно-синий	#000080
olive		Оливковый	#808000
purple		Фиолетовый	#800080
red		Красный	#ff0000
silver		Светло-серый	#c0c0c0
teal		Сине-зеленый	#008080
white		Белый	#ffffff



Указание цвета



Виды компьютерной графики

Критерий сравнения	Растровая графика	Векторная графика
Способ представления изображения	Растровое изображение строится из множества пикселей .	Описывается в виде последовательности команд (набора графических примитивов).
Области применения	Используется для представления реальных образов.	Схемы, карты, шрифты, логотипы.
Качество при редактировании изображения	При масштабировании и вращении возникают искажения	Векторные изображения могут быть легко преобразованы без потери качества.



Графические форматы

Растровые:

BMP, JPEG, GIF, PNG, APNG, ICO, TIFF, PCX, Raw, TGA, WBMP, XCF, WebP.



Векторные:

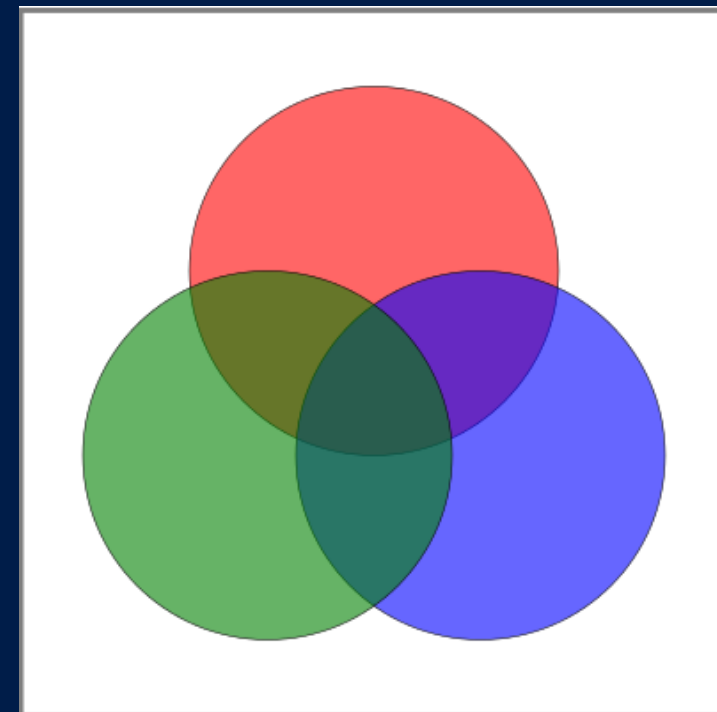
SVG, WMF, SWF, AI, CDR, EMF, PS, EPS, XPS.



Векторная графика. Пример

Пример содержимого svg-файла:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg version = "1.1"
  baseProfile="full"
  xmlns = "http://www.w3.org/2000/svg"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:ev = "http://www.w3.org/2001/xml-events"
  height = "400px" width = "400px">
  <rect x="0" y="0" width="400" height="400"
    fill="none" stroke="black" stroke-width="5px" stroke-opacity="0.5"/>
  <g fill-opacity="0.6" stroke="black" stroke-width="0.5px">
    <circle cx="200px" cy="200px" r="104px" fill="red" transform="translate( 0,-52)" />
    <circle cx="200px" cy="200px" r="104px" fill="blue" transform="translate( 60, 52)" />
    <circle cx="200px" cy="200px" r="104px" fill="green" transform="translate(-60, 52)" />
  </g>
</svg>
```

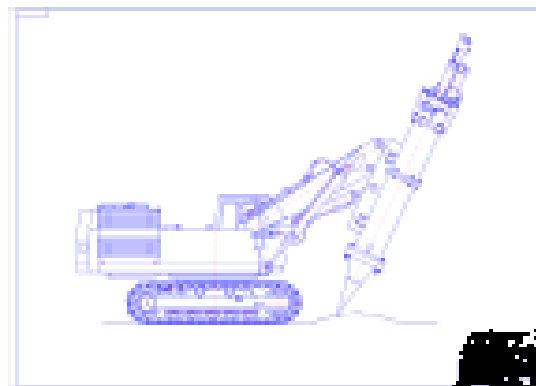


Использование изображений в веб

Для вставки изображения используются одинарный тег ``. Адрес файла с картинкой задаётся через атрибут `src`. Другим обязательным атрибутом является атрибут `alt`, который устанавливает альтернативный текст для изображений, позволяющий получить текстовую информацию о рисунке при отключенной в браузере загрузке изображений, а также поисковыми системами.

Если необходимо, то рисунок можно сделать ссылкой, поместив `` в контейнер `<a>`.

```
<a href="index.php"></a>
```



file:///C:/Documents and Settings/IG5/Мои документы/index.php



Карты изображений

Рисунки также могут применяться в качестве карт-изображений. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые области произвольной формы, где каждая из зон служит отдельной ссылкой.

Элемент `<map>` служит контейнером для элементов `<area>`, которые определяют активные области для карт-изображений. Связь элемента `` с клиентской картой-изображением выполняется при помощи единого идентификатора как в ``, задаваемого атрибутом `usemap`, так и в `<map>`, устанавливаемого атрибутом `name`.

Синтаксис:

```

```

```
<map name="mapid">
```

```
  <area href="path1.html" shape="rect" coords="12,34,56,78">
```

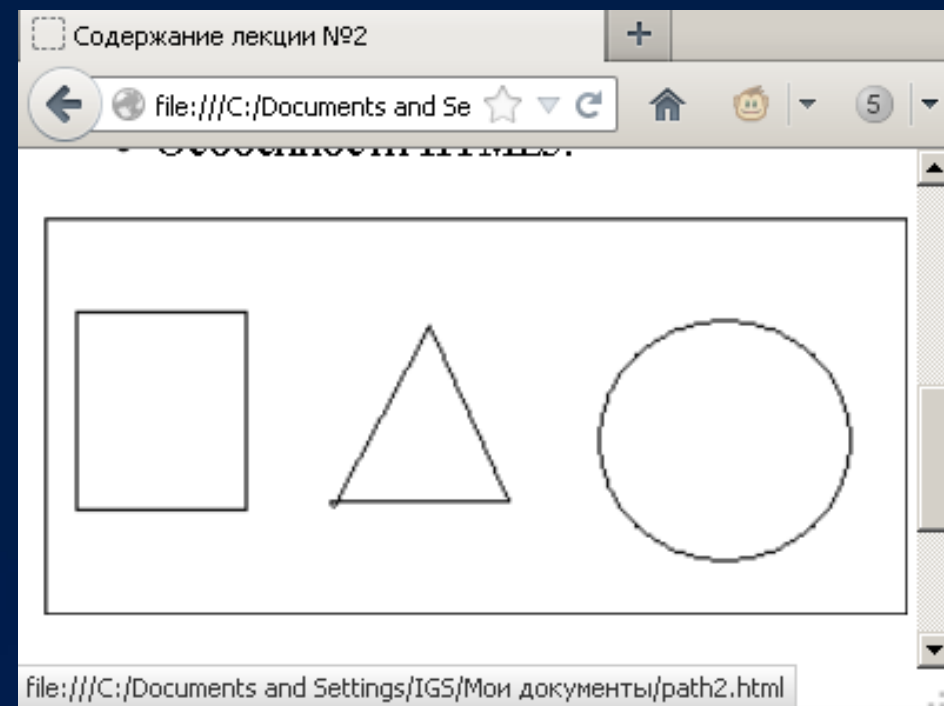
```
</map>
```



Карты изображений. Пример

```

<map name="hmap">
  <area href="path1.html" shape="rect"
coords="12,34,73,103">
  <area href="path2.html" shape="poly"
coords="100,100,160,100,140,40">
  <area href="path3.html" shape="circle"
coords="240,80,45">
</map>
```



Формы и управляющие элементы

Форма – это инструмент, с помощью которого HTML-документ может послать обработчику некоторую информацию (в том числе и введенную пользователем) в виде пар «имя=значение».

Форма характеризуется **адресом обработчика** (адрес программы на веб-сервере, которая будет обрабатывать содержимое данных формы) и включает в себя набор **элементов формы**, которые представляют собой стандартные поля для ввода информации. В частности, кнопка отправки данных формы на сервер (кнопка SUBMIT).

```
<form action="handler.php">  
  <b>Получу ли я зачет?</b><br>  
  <input type="radio" name="answer" value="a1">Да<br>  
  <input type="radio" name="answer" value="a2">Нет<br>  
  <input type="submit">  
</form>
```

Получу ли я зачет?

☐ Да

☐ Нет

Отправить запрос



Формы. Атрибуты тега <form>

Атрибут	Описание
accept-charset	Устанавливает кодировку, в которой сервер может принимать и обрабатывать данные.
action	Адрес программы или документа, который обрабатывает данные формы. Различают два метода – get и post.
autocomplete	Включает автозаполнение полей формы.
enctype	Способ кодирования данных формы.
method	Метод протокола HTTP.
name	Имя формы.
novalidate	Отменяет встроенную проверку данных формы на корректность ввода.
target	Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.



Формы. Методы отправки

Значения атрибута method:

get – предназначен для передачи данных в **адресной строке**. Пары «имя=значение» присоединяются в этом случае к адресу после вопросительного знака и разделяются между собой амперсандом (символ &).
`handler.php?answer=a1&name=SomeName`

post – посылает на сервер данные в HTTP-запросе браузера. Это позволяет отправлять большее количество данных, чем доступно методу get, поскольку у get установлено ограничение в 4 Кб. Большие объёмы данных используются в почтовых службах, при заполнении базы данных, пересылке файлов и др.



Формы и управляющие элементы

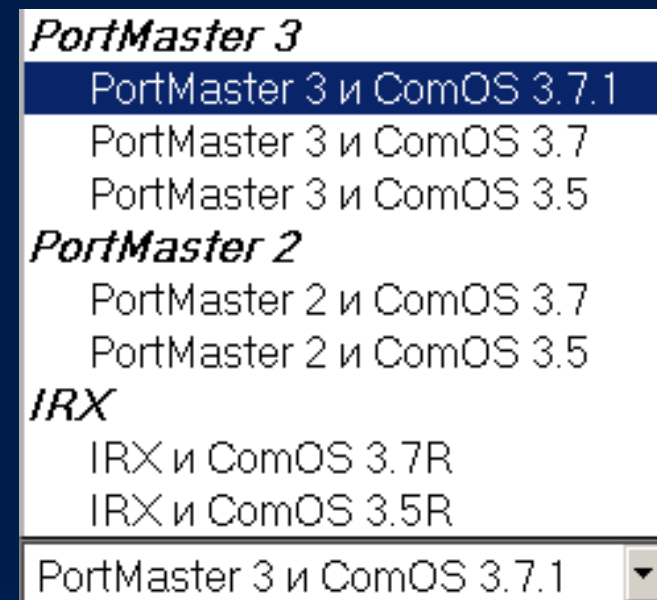
Тег	Описание
<form>	Устанавливает форму на веб-странице.
<button>	Создаёт на веб-странице кнопку.
<input>	Позволяет создавать разные элементы интерфейса.
<textarea>	Создаёт поле для многострочного текста.
<keygen>	Используется для генерации ключей.
<fieldset>	Предназначен для группирования элементов формы.
<label>	Устанавливает метку для элемента формы.
<legend>	Создаёт заголовок группы элементов формы.
<select>	Создаёт раскрывающийся список.
<option>	Определяет отдельные пункты списка, создаваемого с помощью контейнера select.
<optgroup>	Группирует элементы списка.



Управляющие элементы. <select>

<select> позволяет создать элемент интерфейса в виде **раскрывающегося списка**, а также список с одним или множественным выбором. Каждый пункт создаётся с помощью элемента <option>, который должен быть вложен в контейнер <select>. Если планируется отправлять данные списка на сервер, то требуется поместить <select> внутри формы.

```
<select name="ComOS">
  <optgroup label="PortMaster 2">
    <option label="3.7" value="pm2_3.7">PortMaster 2 и ComOS 3.7
  </option><option label="3.5" value="pm2_3.5">PortMaster 2 и ComOS 3.5
  </option>
</optgroup>
<optgroup label="IRX">
  <option label="3.7R" value="IRX_3.7R">IRX и ComOS 3.7R
  </option><option label="3.5R" value="IRX_3.5R">IRX и ComOS 3.5R
  </option>
</optgroup>
</select>
```



Управляющие элементы. <input>. Типы

Тип	Описание	Вид
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.	<input type="text"/>
button	Кнопка.	<input type="button" value="Кнопка"/>
checkbox	Флажки. Позволяют выбрать более одного варианта из предложенных.	<input type="checkbox"/> Чай <input type="checkbox"/> Кофе
file	Поле для ввода имени файла, который пересылается на сервер.	<input type="text"/> <input type="button" value="Обзор..."/>
hidden	Скрытое поле. Оно никак не отображается на веб-странице.	
password	Скрытое текстовое поле, все символы показываются звездочками.	<input type="password"/>
radio	Переключатели. Используются, когда следует выбрать один вариант из нескольких предложенных.	<input type="radio"/> Чай <input type="radio"/> Кофе
reset	Кнопка для возвращения данных формы в первоначальное значение.	<input type="button" value="Сброс"/>
submit	Кнопка для отправки данных формы на сервер.	<input type="button" value="Отправить запрос"/>



Управляющие элементы. <input>. Типы

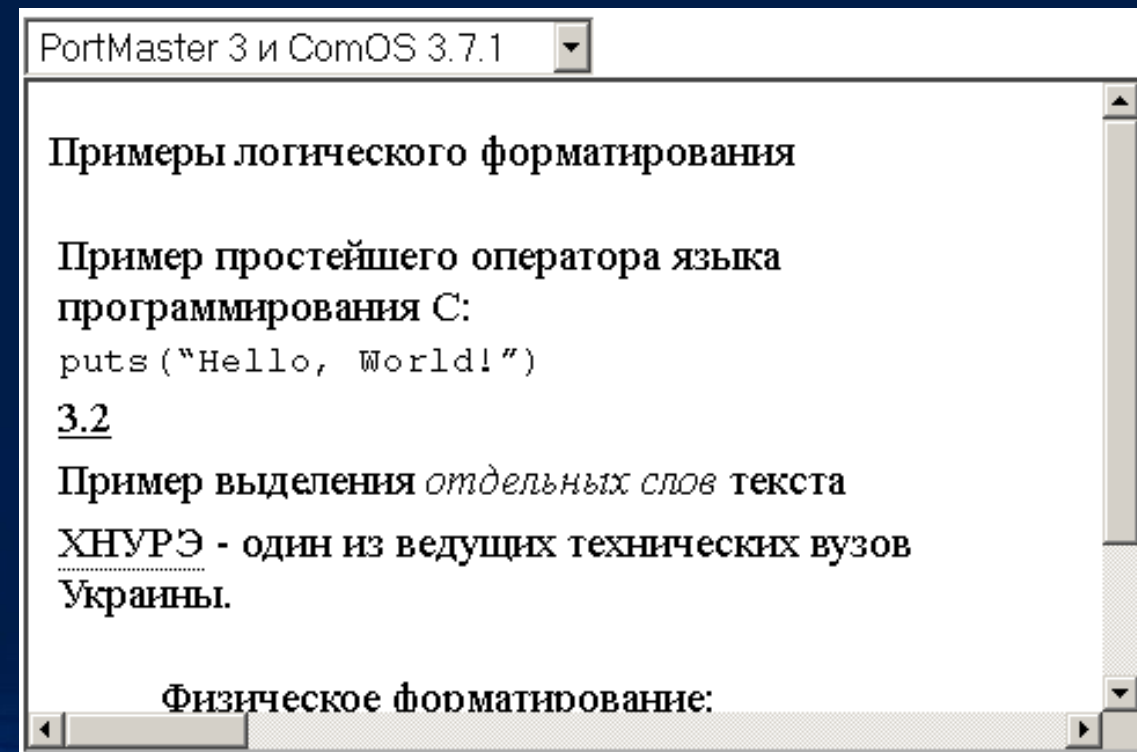
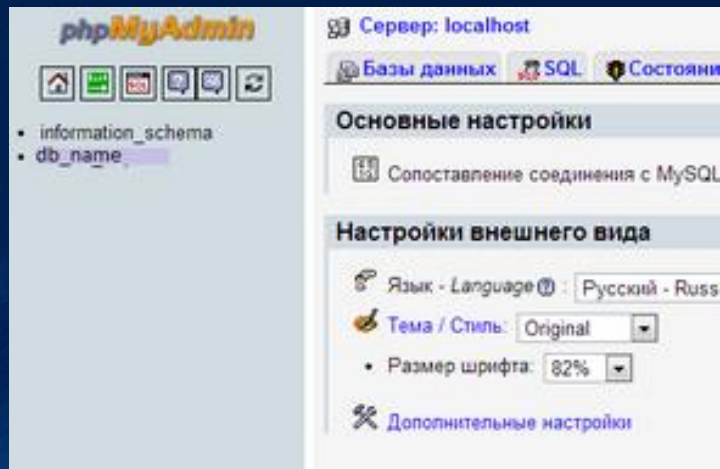
Тип	Описание
color	Виджет для выбора цвета.
date	Поле для выбора календарной даты.
datetime	Указание даты и времени.
datetime-local	Указание местной даты и времени.
email	Для адресов электронной почты.
number	Ввод чисел.
range	Ползунок для выбора чисел в указанном диапазоне.
search	Поле для поиска.
tel	Для телефонных номеров.
time	Для времени.
url	Для веб-адресов.
week	Выбор недели.



Фреймы

Фреймы разделяют окно браузера на отдельные области, расположенные рядом друг с другом. В каждую из таких областей загружается отдельная веб-страница. Практически вышли из употребления, за исключением плавающего (встроенного) фрейма, который можно добавлять в любое место веб-страницы с помощью тега `<iframe>`.

```
<iframe src="Lb1_1.html" width="400"
height="240">Ваш браузер не
поддерживает встроенные фреймы!
</iframe>
```



Вопросы

- Виды списков. Как изменить порядок нумерации в списке?
- Теги для создания таблиц. Как объединить три ячейки по вертикали?
- Для чего используются гиперссылки? В чем разница между абсолютными и относительными гиперссылками?
- Как обеспечить переход к другому разделу текущего документа?
- Возможные состояния гиперссылок.
- Способы указания фона для веб-документа.
- Недостатки использования растровой графики.
- Форматы графических файлов для векторной графики.
- Что такое карта изображений?
- Для чего используются формы? Как устроены формы?
- Назовите основные управляющие элементы.
- Отличия использования элементов `<input type="button">` и `<button>`.
- Недостатки использования фреймов. Плавающие фреймы.

