

Введение в язык разметки HTML

Версии HTML,
Теги HTML: оформление
текста, рисунки, гиперссылки,
списки таблицы, формы

Общая информация

Виды занятий и контроля по МДК:

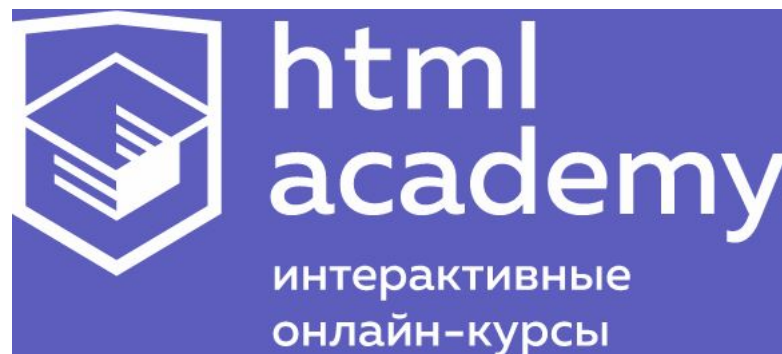
- Длительность МДК – 2 семестра

В текущем семестре:

- Лекции – 4,5 пар в неделю
- Лабораторные работы – 2 пары в неделю,
- Контрольные работы – 2(?)
- Диф. зачет

Лабораторные работы

- Интерактивные онлайн-курсы на HtmlAcademy.ru
- 18 бесплатных разделов с заданиями (9 обязательных)



Что планируется изучить

- HTML
- CSS
- JavaScript
- PHP
- Денвер (Denwer)
- CMS Joomla

Язык разметки HTML

HyperText Markup Language —

«язык разметки гипертекста»

Документ HTML

- Файл с расширением `htm` или `html`
- Имя – произвольное, но если это стартовая страница сайта (раздела сайта), то – `index.html` или `main.html`
(пример)

HTML

Основным форматом представления документов в сети Интернет является язык гипертекстовой разметки HTML (стандартный язык разметки документов во Всемирной паутине) .

HTML – это определенная совокупность правил (тегов), по которым оформляется документ. Теги показывают Интернет-браузеру, как следует отображать текст на Web-страничке.

Структура тега (пары тегов) всегда такова:

<название тега> ... </название тега>

Теги могут содержать атрибуты, характеризующие отображение информации внутри тега.

Тэги

Тэг – это команда языка HTML, которую выполняет браузер:

- непарные тэги

вставить
рисунок

```
<IMG SRC = "vasya.jpg">
```

- парные тэги (*контейнеры*)

открывающий

```
<TABLE>
```

```
...
```

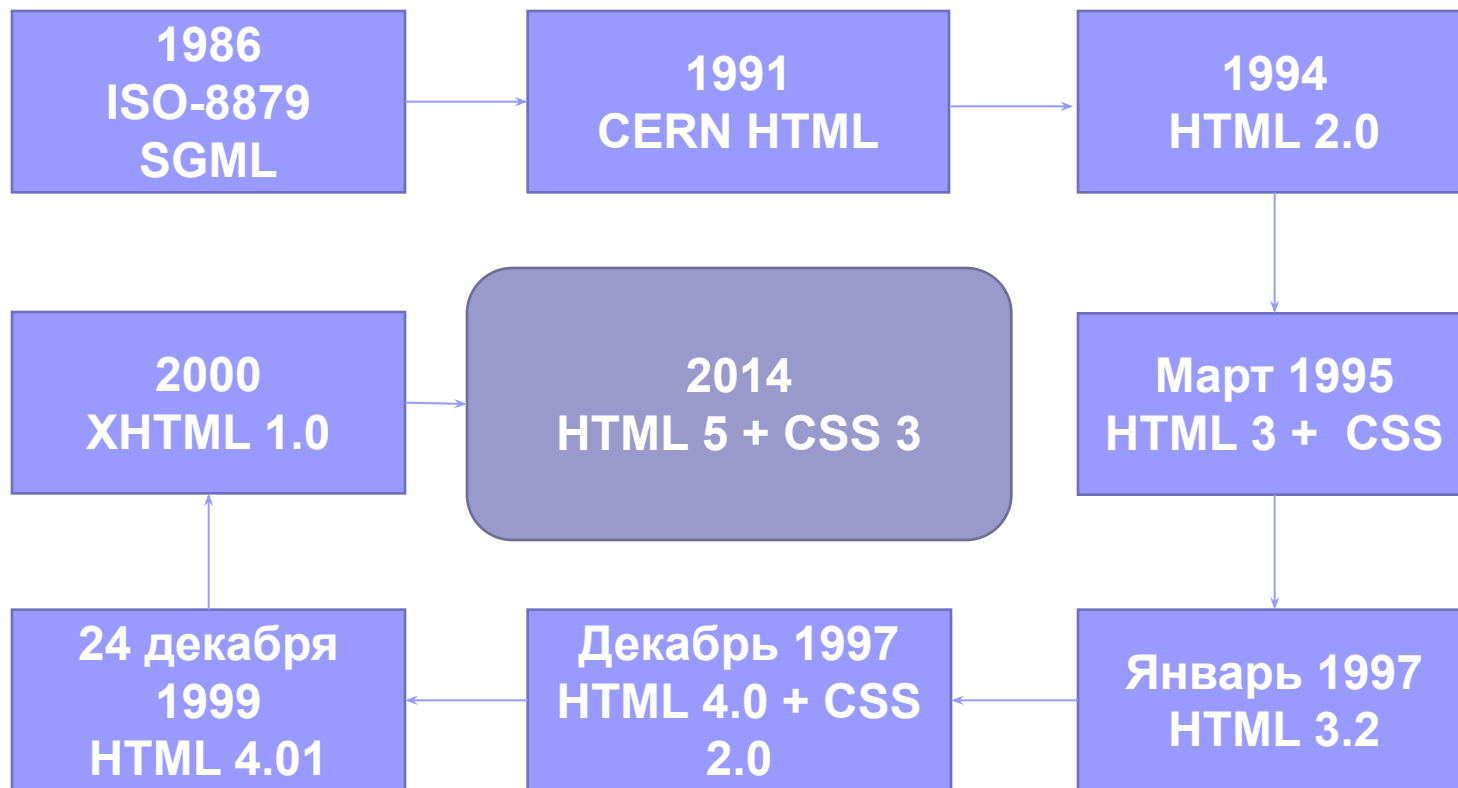
```
</TABLE>
```

область действия
тэга: описание
таблицы

закрывающий

Общее представление о языках разметки HTML, XML, XHTML

ЭВОЛЮЦИЯ ЯЗЫКОВ РАЗМЕТКИ.



Отличия версий HTML

- Каждая новая версия включала дополнительные теги и атрибуты, некоторые признавались устаревшими
- Но основная тенденция развития языка HTML заключалась в стандартизации синтаксиса и верстки HTML-документа.
- Из-за многочисленных условностей при верстке допускалось большое количество ошибок.
- Было принято решение бороться с этим. В результате синтаксис языка HTML стал более строгим.
- HTML5 отличается от XHTML 1.0 только наличием новых тегов

Выбираем !DOCTYPE

- Каждый HTML-документ должен начинаться со строки !DOCTYPE. Она говорит браузеру, какую версию HTML использовали при создании HTML-страницы.
- Если не указать – разные браузеры будут вести себя по разному
- Для HTML 5: `<!DOCTYPE HTML>`

Виды !DOCTYPE

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">` — DOCTYPE для стандарта HTML 4.01 Strict
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">` — DOCTYPE для стандарта XHTML 1.0 Strict;
- Разновидность Strict (строгий) используется в том случае, если код документа полностью соответствует выбранной версии HTML. В Strict оформление и содержание полностью разделены между HTML и CSS.

Виды !DOCTYPE

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">` — DOCTYPE для стандарта HTML 4.01 Transitional
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">` — DOCTYPE для стандарта XHTML 1.0 Transitional
- Transitional - является переходный вариант от прошлой версии языка HTML к новой.
- Если в новой версии HTML какие-то теги были признаны устаревшими, то в стандарте Transitional они по-прежнему остаются рекомендуемыми для применения.

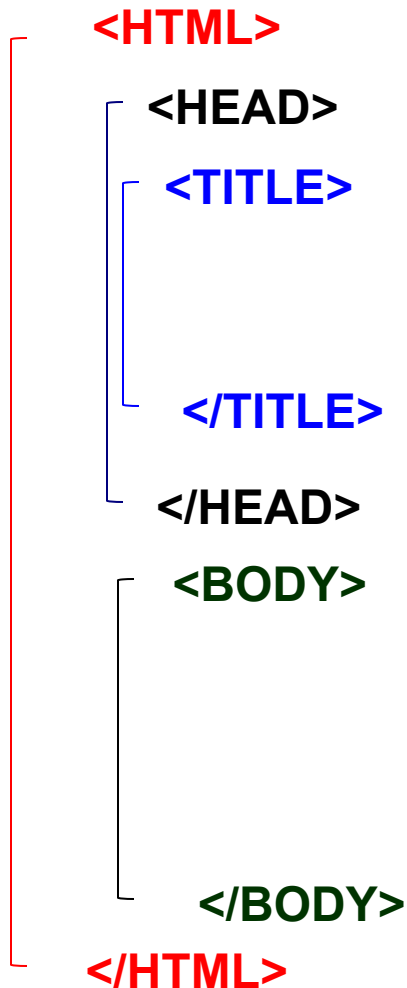
Виды !DOCTYPE

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">` — DOCTYPE для стандарта HTML 4.01 Frameset (с фреймами);
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">` — DOCTYPE для стандарта XHTML 1.0 Frameset (с фреймами);
- Разновидность Frameset аналогична Transitional, кроме того, в Frameset также разрешено применять теги для создания фреймов.

Виды !DOCTYPE

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">` — DOCTYPE для стандарта XHTML 1.1.

Структура Web-страницы



- HTML-код страницы помещается внутри контейнера **<HTML> ...</HTML>**
- Заголовок Web-страницы заключается в контейнер **<HEAD>...</HEAD>**
- Основное содержание страницы помещается в контейнер **<BODY>...</BODY>**
- Название Web-страницы содержится в контейнере **<TITLE>...</TITLE>** и выводится в строке заголовка браузера.

Простейшая Web-страница

first.html

шапка («голова»)

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Моя первая  
Web-страница</TITLE>
```

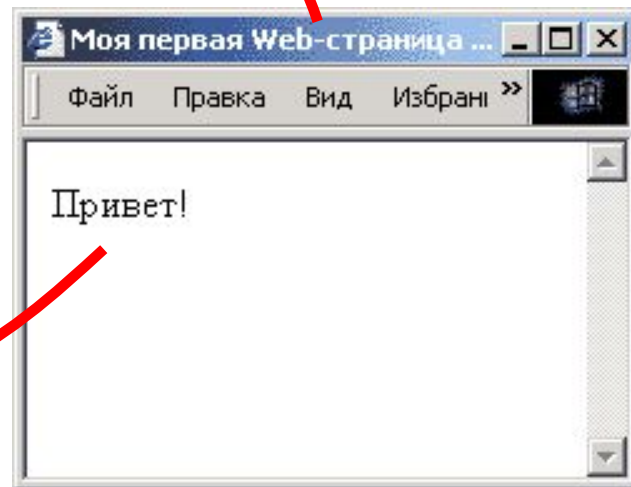
```
</HEAD>
```

```
<BODY>
```

```
Привет!
```

```
</BODY>
```

```
</HTML>
```



основная часть
(«тело»)

Создание файла веб-страницы

Открыть **Notepad++** и введите туда следующий текст:

```
<html>
<head>
<title> Моя первая веб-страничка </title>
<meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" />
</head>
<body>
Здравствуйте, это моя первая страница.<br>
Добро пожаловать! :)
</body>
</html>
```

Сохраним этот документ, присвоив ему имя *.html

ПО для верстки

Редакторы для верстки

- Конечно, сверстать HTML-документ можно и в Блокноте... Раньше это считалось верхом профессионализма. Сейчас же это считается верхом глупости
- Основное достоинство таких редакторов - подсветка HTML- кода, что позволяет сразу определить, где в коде находится ошибка
- Это позволяет минимизировать количество ошибок в HTML-документе и упростить его создание

Редакторы для верстки

- Notepad++
- phpDesignerPro
- Adobe Dreamweaver
- Hotdog
- HTML Pad
- Edit Plus
- ...

Валидаторы

- Валидаторы — сервисы Интернета, проверяющие HTML-документ на ошибки и следование правилам выбранного DOCTYPE
- Как правило, большинство ошибок в HTML-документе не сказываются на его отображении в браузерах.
- Валидатор предназначен прежде всего для разработчиков сайтов
- Наиболее популярным валидатором является сервис <http://validator.w3.org>
- Программа **Tidy** - ее можно скачать с сайта <http://tidy.sourceforge.net>

Теги HTML

Теги

В языке HTML все, что вводится в HTML-документе, будет выводиться на экран браузера сплошным текстом, то есть:

- переводы строк игнорируются и не влияют на отображение текста браузером
- несколько идущих подряд пробелов заменяются одним
- если браузер встретит какой-либо неизвестный ему тег, то проигнорирует его

Типы тегов

(на основе назначения тега либо его поведения)

- **Теги верхнего уровня** — теги, из которых состоит базовая структура документа — его каркас, то есть определяющие раздел заголовка и тела документа: **html, head, body**
- **Теги заголовка документа** — к данной группе относятся все теги, которые указываются внутри тега head: **title, meta, link**

Типы тегов

- **Блочные элементы** — в данный тип входят все теги, которые занимают всю доступную ширину экрана, независимо от того, сколько текста и других элементов находится внутри тега: **blockquote, div, h1, h2, h3, h4, h5, h6, hr, p, pre**
- Любой тег, идущий за блочным тегом, будет отображаться с новой строки

Типы тегов

- Встроенные (строчные) элементы — в отличие от блочных, теги данного типа занимают лишь ту ширину экрана, которая им реально нужна; при этом следующий за ними встроенный элемент на новую строку не переносится: **a, b, big, em, i, img, small, span, strong, sub, sup**
- Универсальные элементы — теги данного типа могут быть как блочными, так и строчными: **del, ins**

Типы тегов

- **Списки** — в данную группу тегов входят все теги, которые применяются для создания нумерованных либо ненумерованных списков: **ul, ol, li, dd, dt, dl**
- **Таблицы** — в данную группу входят теги, предназначенные для создания таблиц: **table, thead, tbody, td, th, tr**
- **Фреймы** — все теги, предназначенные для создания фреймов (области внутри HTML-документа, в которые загружается другой HTML-документ): **frame, frameset, iframe**

Блочные и строчные теги

Характеристики	Блочный тег	Строчный тег
Какие теги могут находиться внутри данных тегов?	Как блочные, так и строчные	Только строчные
Тег начинается с новой строки?	Да, всегда	Нет, переносятся на новую строку, только если на предыдущей нет места либо если предыдущий тег — блочный
Теги занимают всю доступную ширину окна браузера?	Да	Нет, их ширина равна ширине содержимого плюс размеры заданных отступов, полей и границ

Атрибуты и их значения

- Внутри одинарного или открывающего тега могут находиться атрибуты и их значения:

`<тег атрибут1="значение1"`

`атрибут2="значение2"> текст </тег>`

- Атрибуты уточняют функциональность тега
- Порядок расположения атрибутов в теге ни на что не влияет

Атрибуты и их значения

- В XHTML каждый атрибут обязан иметь значение
- Значения атрибута - либо определенные ключевые слова, либо любой пользовательский текст
- Значение атрибута указывается в кавычках ' (одинарных) или " (двойных). В других кавычках (` , ‘ , “ , « , ») - нельзя

`<body text = "white" bgcolor="black">`

Пример текста `</body>`

Типы значений атрибутов

Задающие цвет:

- специальное ключевое слово: black, blue и т. д.
- шестнадцатеричный код цвета: #ff0000 (красный), #000000 (черный) и т. Д

Задающие размер:

- просто целое число (размер в пикселах)
- целое число от 1 до 100 в процентах, определяющее размер элемента относительно размера родительского элемента либо окна браузера

Типы значений атрибутов

Задающие адрес в Интернете:

- абсолютный адрес — `http://microsoft.com`,
`http://mail.ru/content.html`
- относительный адрес — `/content.html`,
`../css/style.css`

Различные ключевые слова (selected, none и т.д.)

Основной синтаксис XHTML

- HTML-документ обязательно должен начинаться с DOCTYPE
- Все теги и атрибуты должны записываться строчными буквами (в нижнем регистре), так как язык XHTML регистрозависим
- Любые значения атрибутов необходимо заключать в кавычки
- Все парные теги должны иметь закрывающий тег
- Все одинарные теги должны завершаться слешем перед закрывающей скобкой:

, , <input />, <hr />, <meta />

Основной синтаксис XHTML

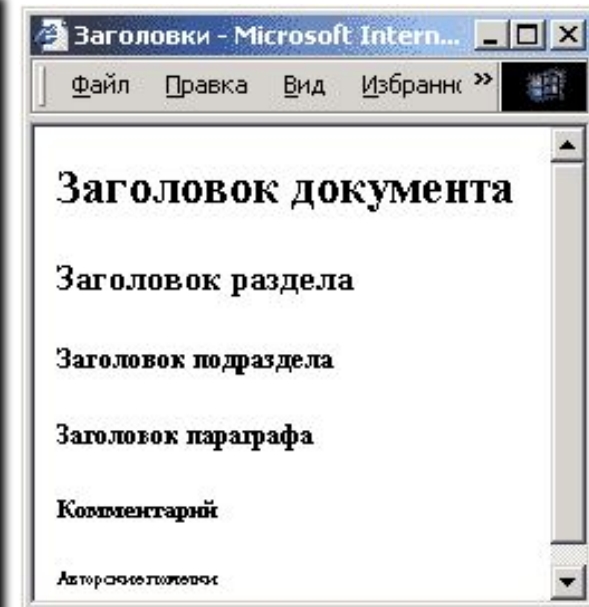
- Каждый тег должен быть корректно вложен в другой
- Сокращенные атрибуты (без значения) запрещены; для них в качестве значения обязательно нужно указывать название атрибута: *checked="checked"*, *compact="compact"*, *disabled="disabled"*
- Непосредственно внутри тега *body* не должно быть ни текста, ни изображений. Они должны находиться внутри парных тегов *span*, *div*, списков или таблиц

Web-страницы. Язык HTML

Оформление текста

Заголовки: h1 ... h6

```
<body>  
<h1>Заголовок документа</h1>  
<h2>Заголовок раздела</h2>  
<h3>Заголовок подраздела</h3>  
<h4>Заголовок параграфа</h4>  
<h5>Комментарий</h5>  
<h6>Авторские пометки</h6>  
</body>
```



Теги h1- h6 являются парными и блочными

Абзацы

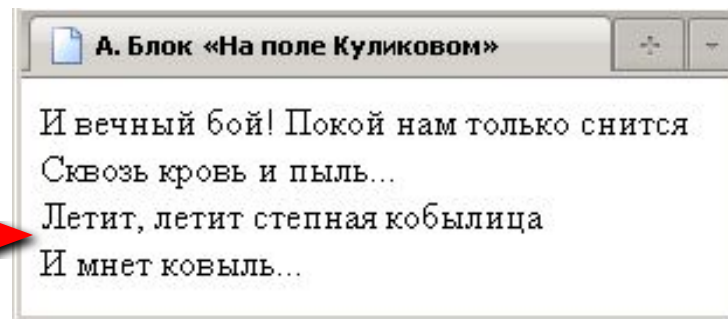
- переход на новую строку

И вечный бой! Покой
нам только снится

Сквозь кровь и пыль...

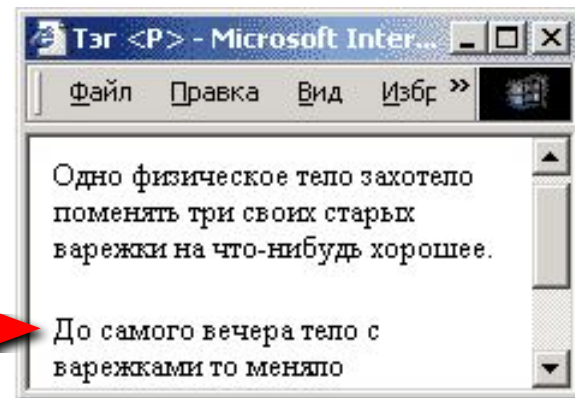
Летит, летит степная
кобылица

И мнет ковыль...



- абзац (с отступами)

<p>
Одно физическое тело захотело
поменять три своих старых варежки
на что-нибудь хорошее.
</p>
<p>
До самого вечера тело с
варежками ...
</p>



Физическая разметка

Курсив (<i>italic</i>)	<code><i>Вася</i></code> <code>Вася</code>	<i>Вася</i>
Жирный (<i>bold</i>)	<code>Вася</code> <code>Вася</code>	Вася
Верхний индекс (<i>superscript</i>)	<code>Вася<sup>2</sup></code>	Вася ²
Нижний индекс (<i>subscript</i>)	<code>Вася<sub>2</sub></code>	Вася ₂
Подчеркивание	<code><u>Вася</u></code>	<u>Вася</u>
Зачеркивание	<code><s>Вася</s></code>	Вася

Оформление текста

- Вывод «как есть»

```
<pre>
```

```
И вечный бой! Покой  
    нам только снится  
Сквозь кровь и пыль...  
Летит, летит степная кобылица  
    И мнет ковыль...
```

```
</pre>
```

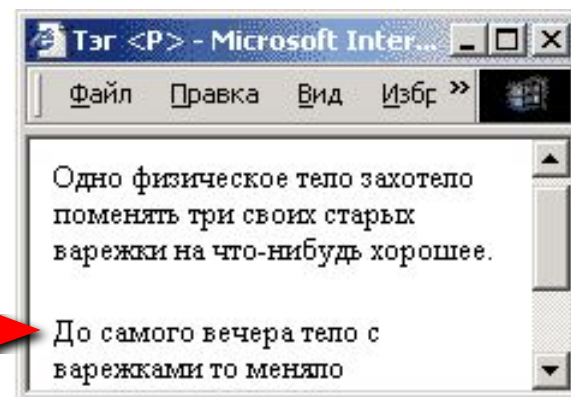
- Комментарии

```
<!-- это наш комментарий -->
```

```
<p>
```

```
Одно физическое тело захотело  
поменять три своих старых варежки  
на что-нибудь хорошее.
```

```
</p>
```



Контейнеры

Тег div

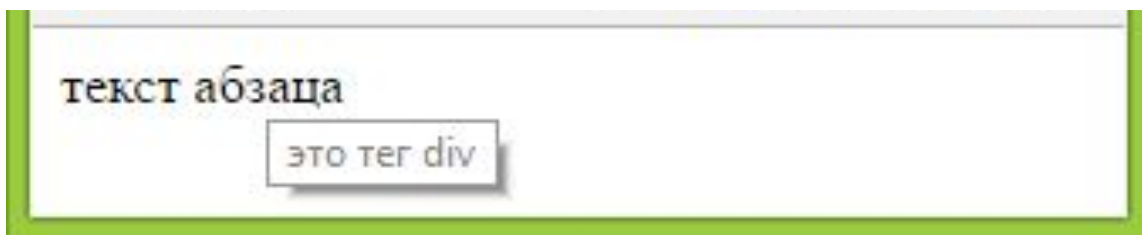
- Блочный тег, который служит для единственной цели — быть контейнером, в котором могут находиться другие теги, изображения или текст.
- *Вам нужно установить фон или границу для отдельного участка веб-страницы?*
- *Поместите нужный участок в тег div, после чего установите фон и/или границу для этого div.*

Тег span

- То же что и тег div, но является строчным, т. е. не требует для себя всей ширины окна браузера. Ему
- достаточно лишь той ширины, которую физически занимают все элементы, находящиеся в нем.

Атрибут `title`

- Позволяет добавить описание к любому элементу веб-страницы, в виде всплывающей подсказки
- Это необязательный атрибут
- Атрибут `title` можно использовать практически для любого тега, который может быть вложен внутрь тега `body`



Специальные символы

Символ	HTML-код	Название
—	—	(длинное) тире
	 	неразрывный пробел
©	©	символ авторского права
«	«	левая русская кавычка
»	»	правая русская кавычка
®	®	зарегистрированная торговая марка
°	°	градус
²	²	квадрат
³	³	куб
¼	¼	четверть
½	½	половина
×	×	знак умножения
÷	÷	знак деления

Web-страницы. Язык HTML

Рисунки, видео, аудио

Форматы рисунков

GIF (*Graphic Interchange Format*)



- сжатие без потерь
- прозрачные области
- анимация
- **только с палитрой** (2...256 цветов)

рисунки с четкими границами, мелкие рисунки

JPEG (*Joint Photographer Expert Group*)

- сжатие с потерями
- только *True Color* (16,7 млн. цветов)
- **нет анимации и прозрачности**

рисунки с размытыми границами, фото



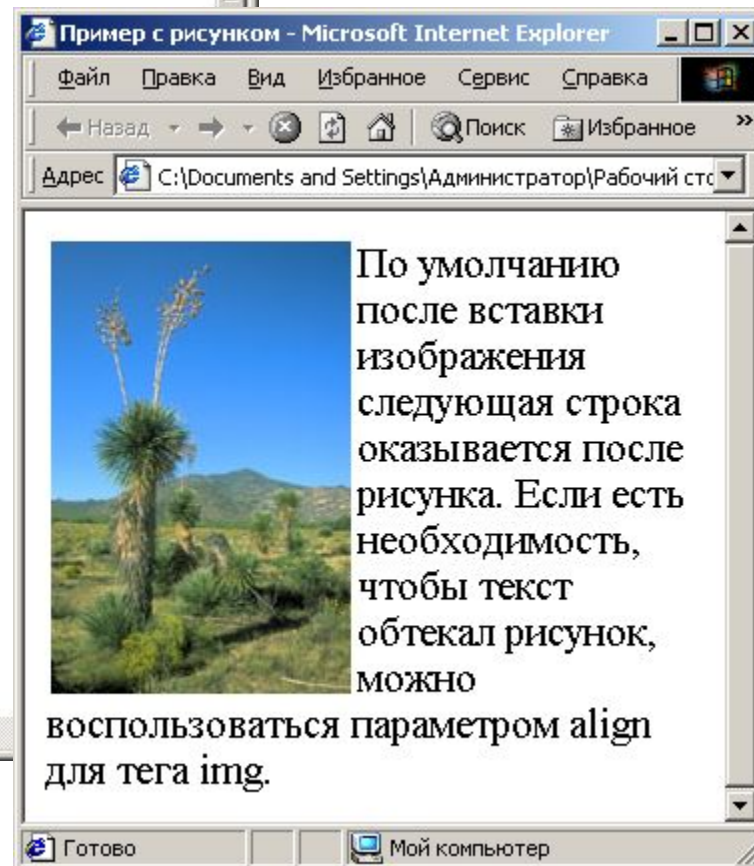
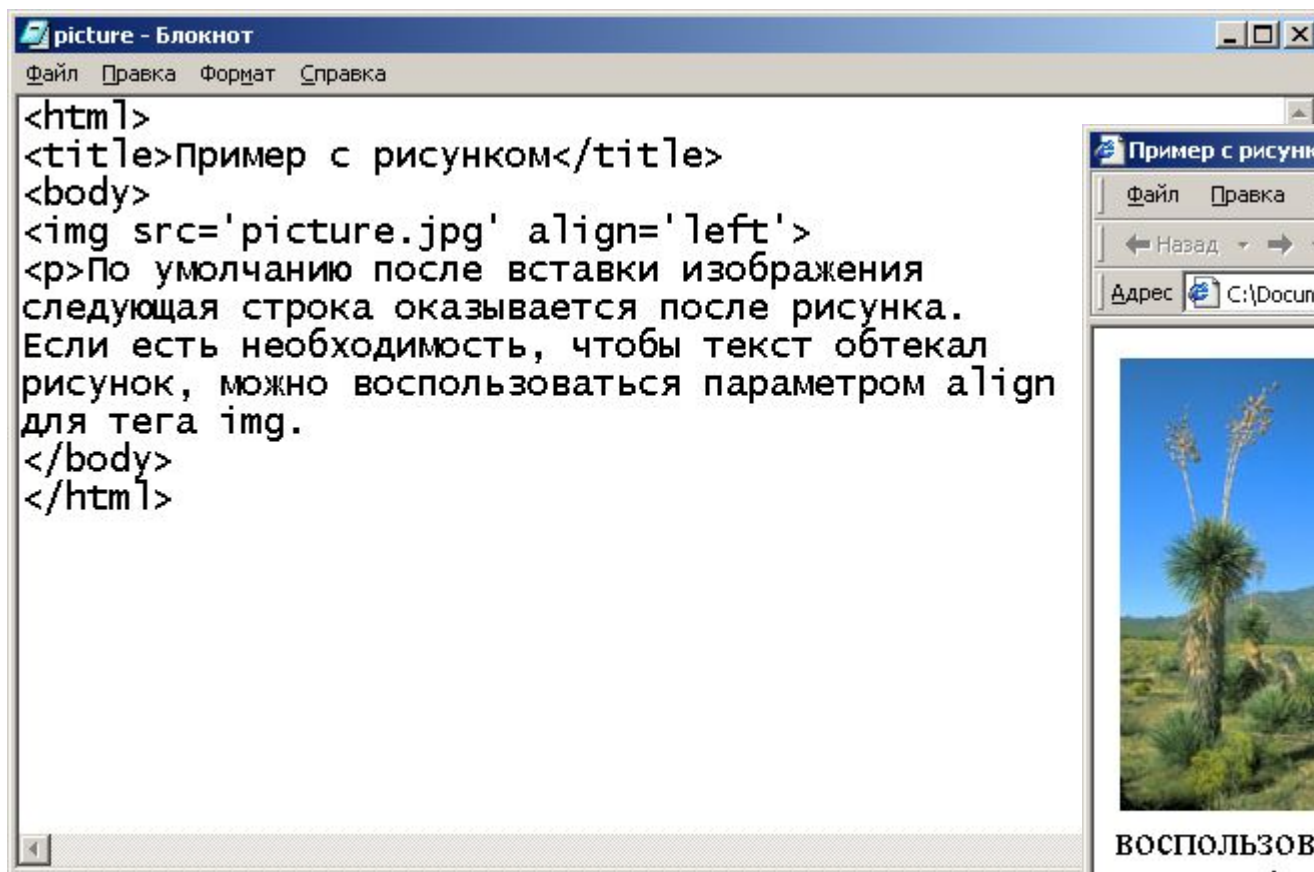
PNG (*Portable Network Graphic*)

- сжатие без потерь
- с палитрой (PNG-8) и *True Color* (PNG-24)
- прозрачность и полупрозрачность (альфа-канал)
- нет анимации
- **плохо сжимает мелкие рисунки**

Тег img

```

```



Рисунки в документе

из той же папки:

image
(изображение)

source
(источник)

```

```

из другой папки:

```

```

```

```

с другого сервера:

```

```

Атрибуты тега `img`

```

```

- `width`, `height` — ширина или высота задается в пикселах
- `src` — url картинки
- `alt` — альтернативный текст, который выводится, если изображение не загрузилось

Аудио

`<audio><source src="music.mp3"></audio>`

- `autoplay` — проигрывание при загрузке страницы
- `controls` — панель управления плеером
- `loop` — автоповтор
- `preload` — загрузка файла вместе с загрузкой страницы
- `src` — путь до трека

Видео

```
<video height="300" poster="pictures/poster.jpg">  
  <source src="pictures/llm.mp4">  
</video>
```

- poster – стартовая картинка-заместитель

Web-страницы. Язык HTML

Гиперссылки

Ссылки на другие страницы сайта

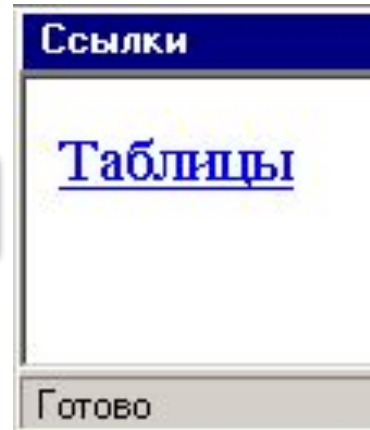
- страница в той же папке

anchor (якорь)

```
<a
```

```
href="table.htm">Таблицы</a>
```

*hyper reference
(гиперссылка)*



- страница во вложенной папке

```
<a href="example/ex1.htm">Пример</a>
```

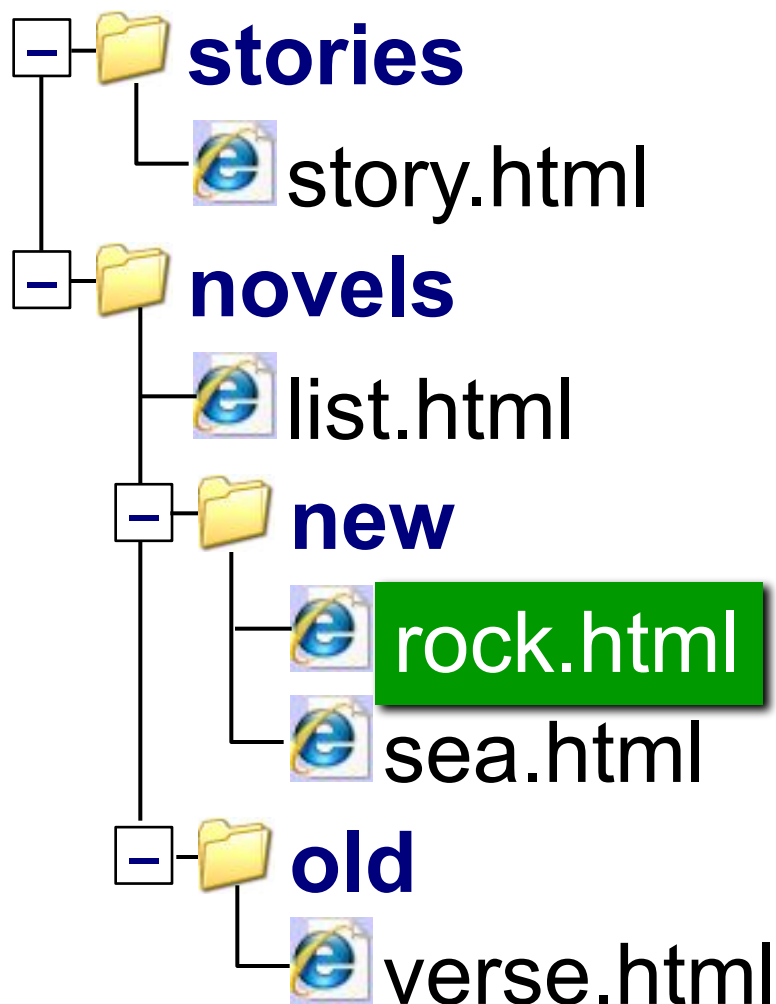
- страница в соседней папке

```
<a
```

```
href=" ../text/text1.htm">Текст</a>
```

выйти из текущей папки

Примеры (ссылки из файла *rock.html*)



Ссылки на другие сайты

- на главную страницу сайта

```
<a href="http://www.mail.ru">Почта</a>
```

index.htm, index.html, default.asp, ...

- на конкретную страницу сайта (URL)

```
<a href="http://www.vasya.ru/text/a.htm">  
Васин текст</a>
```

- на файл для скачивания

```
<a href="http://www.vasya.ru/prog.zip">  
Скачать</a>
```

Ссылки внутри страницы

```
<a name="up"></a>
```

переход на метку

```
<a href="#chap1">Глава 1</a><br>
```

```
<a href="#chap2">Глава 2</a><br>
```

```
<a id="chap1"></a>
```

метка (якорь)

```
<h1>Глава 1</h1>
```

Это текст главы 1. Это текст главы 1.

Это текст главы 1. Это текст главы 1.
1.


```
<a href="#up">Наверх</a>
```

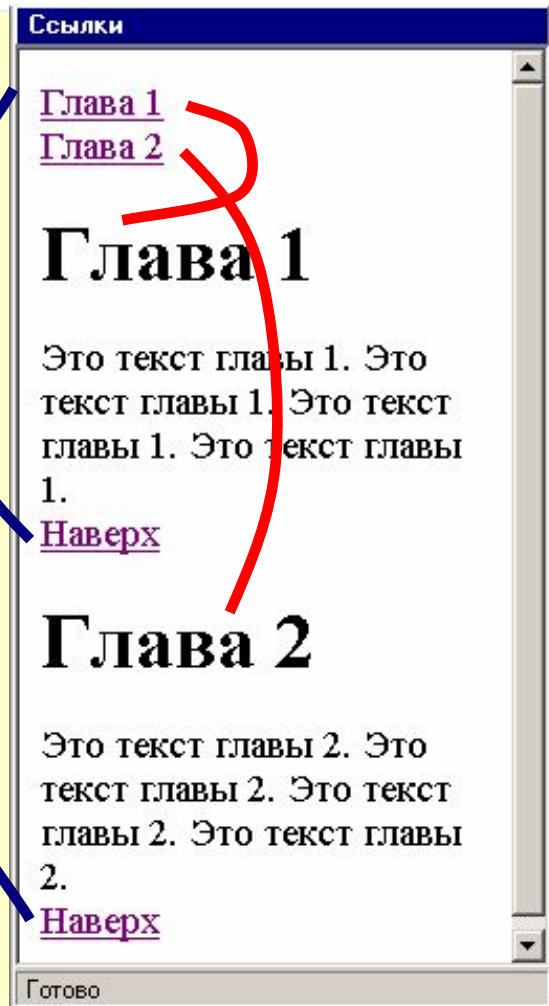
```
<a id="chap2"></a>
```

```
<h1>Глава 2</h1>
```

Это текст главы 2. Это текст главы 2.

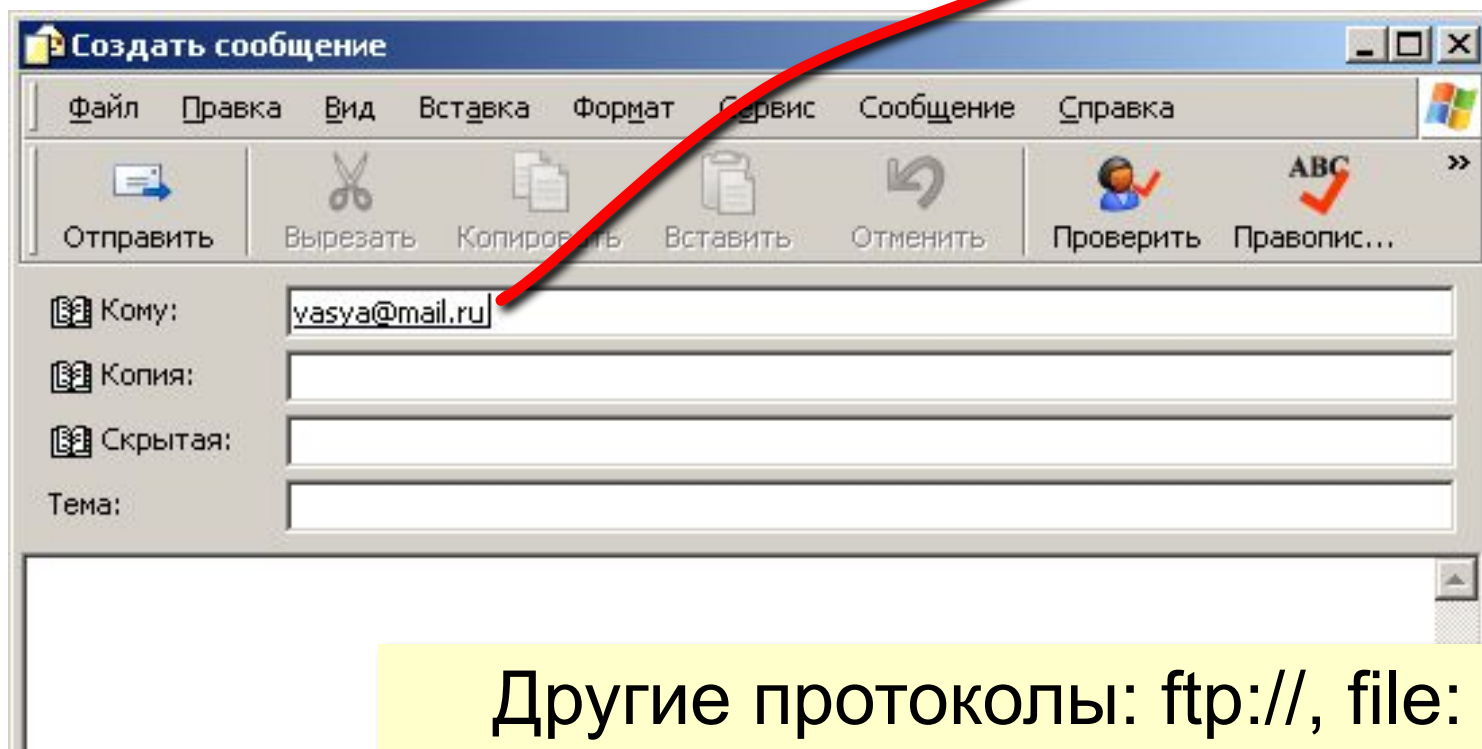
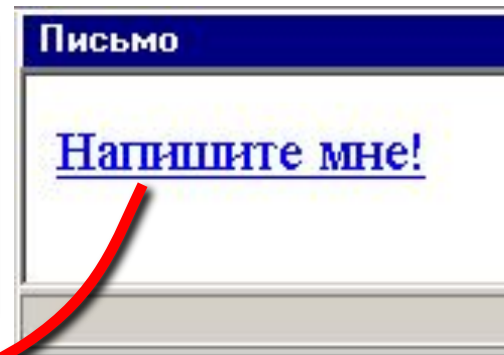
Это текст главы 2. Это текст главы 2.
2.


```
<a href="#up">Наверх</a>
```



Запуск почтовой программы

```
<a href="mailto:vasya@mail.ru">  
Напишите мне!  
</a>
```



Другие протоколы: ftp://, file: , ...

Варианты открытия ссылки

`я ссылка`

- атрибут `target` — имя окна или фрейма, куда браузер будет загружать документ
- `_blank` — в новой вкладке/окне
- `_self` — в том же окне, что и ссылка
- `_parent` — (с фреймами) в родительском фрейме
- `_top` - (с фреймами) в текущем окне, заменяя собой все фреймы

Web-страницы. Язык HTML

Списки

Списочные элементы

- `` - пронумерованный перечень элементов (ordered lists)
- `` - маркированный перечень элементов (bulleted lists)
- `` - отдельный пункт списка

Списочные элементы

Нумерованный:

```
<ol>
```

```
<li>text1</li>
```

```
<li>text2</li>
```

```
<li>text3</li>
```

```
<li>text4</li>
```

```
</ol>
```

Ненумерованный:

```
<ul>
```

```
<li>text1</li>
```

```
<li>text2</li>
```

```
<li>text3</li>
```

```
<li>text4</li>
```

```
</ul>
```

Web-страницы. Язык HTML

Таблицы

Теги для таблицы

- В тег `<table>` включается множество элементов для описания структуры таблицы
- Элемент `<tbody>` группирует строки таблицы
- Тег `<tr>` для описания конкретной строки
- Каждый тег `<tr>` содержит элементы `<th>` и `<td>` для описания заголовков ячеек данных и самих данных ячейки, соответственно
- Другие не обязательные элементы позволяют описывать название таблицы `<caption>`, а также начальные (header) и конечные (footer) строки (`<thead>` и `<tfoot>`)

Таблицы

<table>

<thead><th></th>	<th></th>	<th></th></thead>
<tbody> <td></td>	<td></td>	<td></td>
<td></td>	<td></td>	<td></td> </tbody>
<tfoot><td></td>	<td></td>	<td></td></tfoot>

</table>

Таблицы

```
<table>  
  <thead><th>...</th></thead>  
  <tfoot><td>..</td></tfoot>  
  <tbody>  
    <tr><td>..</td></tr>  
    <tr><td>..</td></tr>  
  </tbody>  
</table>
```

*Подвал таблицы определяется
выше содержимого таблицы!*

Задание HTML таблицы

```
<table border cellspacing="5" cellpadding="10">
```

```
  <caption>Состав команды</caption>
```

```
  <tr>
```

```
    <th>ФИО</th>
```

```
    <th>Должность</th>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>Иванов А.П.</td>
```

```
    <td>программист</td>
```

```
  </tr>
```

```
  <tr>
```

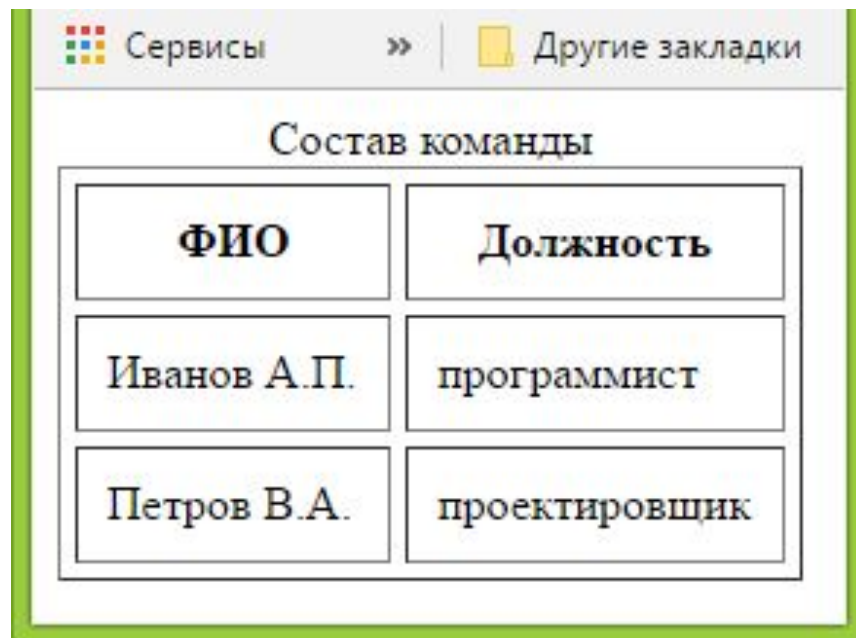
```
    <td>Петров В.А.</td>
```

```
    <td>проектировщик</td>
```

```
  </tr>
```

```
</table>
```

Заголовок таблицы



ФИО	Должность
Иванов А.П.	программист
Петров В.А.	проектировщик

Таблицы. Объединение ячеек

`<table>`

<td rowspan=2></td>	<td></td>	<td></td>
	<td colspan=2></td>	
<td></td>	<td></td>	<td></td>

`</table>`

Описание структуры страницы с помощью таблиц

- С помощью таблиц можно создать хорошую схему (планировку таблицы) *(4 слайд 44)*
- Не обязательно рисовать саму таблицу (толщина линии может быть = 0). Но таблицы позволяет задавать взаимное расположение элементов страницы относительно друг друга (что находится слева, справа и снизу, сверху).
- Однако, таблицы имеют следующие недостатки:
- Таблицы требуют добавления большого количества кода.
- Таблицы трудно удалить или изменить.

Разметка на основе таблиц

Заголовок сайта	
Набор гиперссылок для перехода между страницами сайта	Полезное содержимое сайта
Сведения о правах разработчика	

Сделать самим

Web-страницы. Язык HTML

Формы

Формы HTML

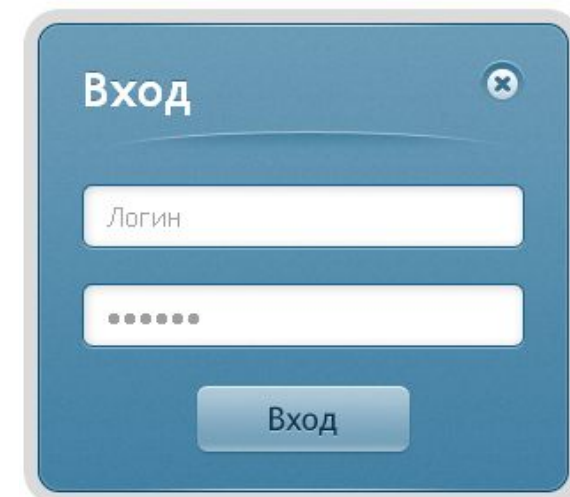
- Используются для описания частей страницы в которой пользователь может просматривать имеющиеся данные, корректировать и вводить новые данные
- С помощью форм выполняется сбор данных, вводимых пользователем для взаимодействия с Web-приложением

Типичные примеры форм:

- *поиск по сайту*
- *отправки сообщения админу*
- *регистрации на сайте*
- *опросы*



A search form with a white input field containing the placeholder text "Поиск..." and an orange button labeled "Найти".



A login form titled "Вход" (Login) with a close button (X) in the top right corner. It contains two input fields: the first is labeled "Логин" (Login) and the second contains masked characters ".....". Below the fields is a blue button labeled "Вход" (Login).

Формы HTML

- Функциональность форм обеспечивается через языки JavaScript и/или PHP
- Формы в HTML могут состоять из:
 - Текстовых полей
 - Текстовых областей
 - Раскрывающихся списков
 - Кнопок
 - Флажков
 - Переключателей

Формы HTML

- Форма задается парным блочным тегом `<form>`
- Тег `<form>` обычно включает набор элементов управления:
- Элемент `<input>` для ввода пользователем данных
- Элемент `<select>` для задания возможности выбора из списка альтернатив
- Элемент `<textarea>` для ввода многострочного текста

Описание формы в HTML

- Форма задается парным блочным тегом `<form>`
`<form method="get" action="URL">`
Элементы_формы_и_другие_элементы_HTML
`</form>`
- **method** - метод послыки запроса с данными из формы:
- *get*: данные из формы добавляются в конец URL
- *post*: данные из формы добавляется в заголовок запроса
- **action** - описывает URL, который будет вызываться для обработки данных формы

Элементы формы

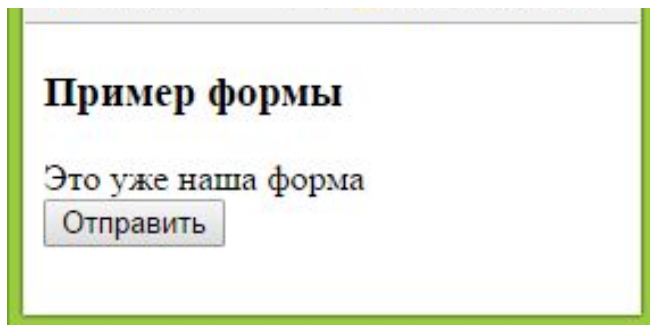
- Многие элементы формы создаются тегом `<input>`
`<input type="тип" name="имя" value="значение"/>`

Типы управляющих элементов `<input>`

- Кнопки «Передать» (`type="submit"`)
- Кнопки «Очистить» (`type="reset"`)
- Кнопки (`type="button"`)
- Кнопки-картинки (`type="image"`)
- Поля ввода (`type="text"`)
- Пароли (`type="password"`)
- Переключатели (`type="radio"`)
- Флажки (`type="checkbox"`)
- Скрытый (`type="hidden"`)

Пример формы HTML

```
<form method="get" action="">  
  <h3>Пример формы</h3>  
  <div>Это уже наша форма</div>  
  <input type="submit" value="Отправить" name="" />  
</form>
```

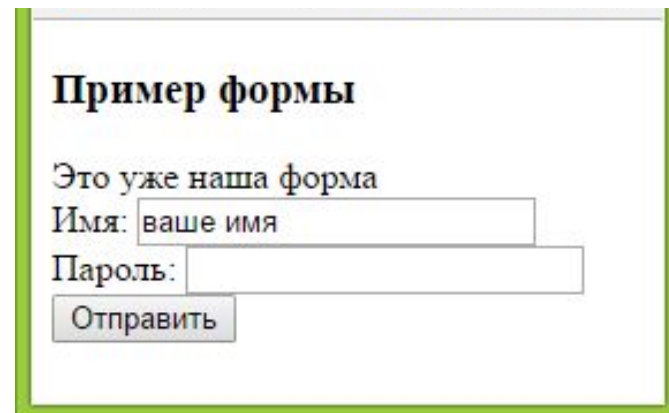


Пример формы

Это уже наша форма



расширить



Пример формы

Это уже наша форма

Имя:

Пароль:

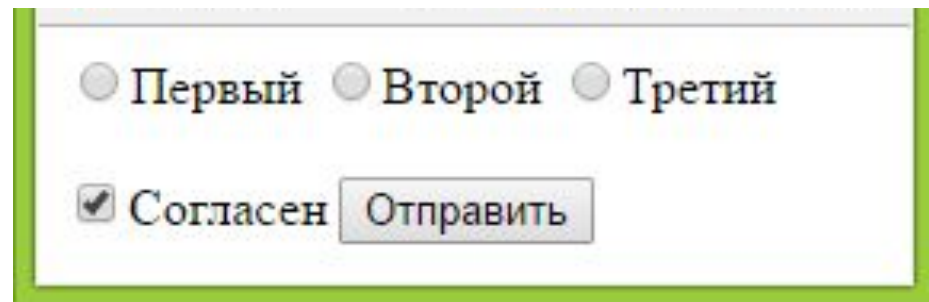
Флажки и переключатели

- **Переключатель** создается тегом `<input>` с атрибутом `type = "checkbox"`

```
<input type="checkbox" name="ch" value="1"/>
```

- **Флажок** создается тегом `<input>` с атрибутом `type = "radio"`

```
<input type="radio" name="radio" value="1" />
```



○ Первый ○ Второй ○ Третий

☒ Согласен

The image shows a screenshot of a web form. It contains three radio buttons labeled 'Первый', 'Второй', and 'Третий'. Below them is a checkbox labeled 'Согласен' which is checked, and a button labeled 'Отправить'.

Скрытый элемент

- Иногда при отправке формы на сервер нужно передать некое значение, но посетителю его видеть не надо
- Так могут передаваться значения, которые посетитель указал на предыдущих шагах мастера. Или какой-то секретный ключ, который применяется формой для проверки легитимности полученных данных
- Для этих целей в HTML используется тег `<input>` с атрибутом `type="hidden"`
`<input type="hidden" value="1234" name="userid" />`

Другие варианты тега input

Менее популярные варианты атрибута type тега <input>:

- **button** — позволяет создать кнопку
- **file** — используется для создания кнопки выбора файла на локальном компьютере, который будет загружен на сервер
- **image** — аналог значения submit (создает кнопку отправки формы), но также позволяет указать картинку, которая будет отображаться вместо стандартной кнопки
- **reset** — создает кнопку для сброса всех значений, которые пользователь внес в форму, к значениям по умолчанию



Выпадающие списки

- Выпадающий список – тег `<select>`
- Строка списка задается тегом `<option>`

```
<select name="form_sel" size="2">  
  <option value="1">Первый</option>  
  <option value="2">Второй</option>  
  <option value="3" selected>Третий</option>  
</select>
```

Текстовая область

- Для создания многострочного текстового поля следует использовать парный блочный тег `<textarea>`
- ```

<form method="get" action="">
 <h3>Пример формы</h3>
 <div>Это уже наша форма</div>
 Имя: <input type="text" name="txt" value="ваше имя" />
 Пароль: <input type="password" name="psw" value="" />
 <textarea name="bigtext" rows="3">просто
какой-то
текст</textarea>
 <input type="submit" value="Отправить" name="" />
</form>

```



# Замечание по обработке HTML форм

- При нажатии кнопки «submit» данные формы передаются на сервер – по адресу указанному в атрибуте `action="URL"`; если атрибута нет или он пустой, то передаются той же странице, в которой записана форма
- На серверной стороне сайта необходимо создать способ сохранения и обработки данных передаваемых на сервер

# Тег `meta` и его виды

- `<meta name="Keywords" content="Екатеринбург, портал, новости, форумы"/>`
- `<meta name="Description" content="Хабрахабр — самое крупное в Рунете.."/>`
- атрибут `name` — устанавливает модификатор мета-тега

# Тег meta и его виды

- `<meta name="viewport" content="width=device-width, maximum-scale=1, initial-scale=1, user-scalable=no" />`
- width — ширина
- maximum-scale — максимальный масштаб
  - initial-scale — начальный масштаб
  - user-scalable — изменение масштаба (yes/no)

